User's Manual

16

RL78/G24

User's Manual: Hardware

16-Bit Single-Chip Microcontrollers

Renesas Electronics
www.renesas.com

Rev.1.10 Nov 2023

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Contact Information

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

Trademarks

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

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

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

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

How to Use This Manual

Readers

This manual is intended for user engineers who wish to understand the functions of the RL78/G24 and design and develop application systems and programs for these devices.
The target products are as follows.

- 20-pin: R7F101G6x (x = E, G)
- 24-pin: R7F101G7x (x = E, G)
- 25-pin: R7F101G8x (x = E, G)
- 30-pin: R7F101GAx (x = E, G)
- 32-pin: R7F101GBx (x = E, G)
- 40-pin: R7F101GEx (x = E, G)
- 44-pin: R7F101GFx (x = E, G)
- 48-pin: R7F101GGx (x = E, G)
- 52-pin: R7F101GJx (x = E, G)
- 64-pin: R7F101GLx (x = E, G)

Purpose

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

Organization

The RL78/G24 manual is separated into two parts: this manual and User's Manual: Software (common to the RL78 family).

RL78/G24 User's Manual: Hardware (This Manual)	**RL78 Family User's Manual: Software**
• Pin functions • Internal block functions • Interrupts • Other on-chip peripheral functions • Electrical characteristics	• CPU functions • Instruction set • Explanation of each instruction

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/G24 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 numbers are represented as xxxx or xxxxB, where each x is 0 or 1.

Decimal numbers are represented as xxxx, where each x is a numeral from 0 to 9.

Hexadecimal numbers are represented as xxxxH, where each x is a number from 0 to 9 or a letter from A to F.

Related Documents

Documents Related to Devices

Document Name	Document No.
RL78/G24 User's Manual: Hardware	This manual
RL78 Family User's Manual: Software	R01US0015E

Documents Related to Flash Memory Programming and On-chip Debugging

Document Name	Document No.
PG-FP6 Flash Memory Programmer User's Manual	**Note 1**
E2 Emulator User's Manual	R20UT3538E
E2 Emulator Lite RTE0T0002LKCE00000R User's Manual	R20UT3240E
Renesas Flash Programmer Flash Memory Programming Software User's Manual	**Note 2**

Note 1. For a list of the documents relevant to the PG-FP6, visit the Web page below.
https://www.renesas.com/us/en/software-tool/pg-fp6

Note 2. For a list of the documents relevant to the Renesas Flash Programmer, visit the Web page below.
https://www.renesas.com/us/en/software-tool/renesas-flash-programmer-programming-gui

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

Section 1 Outline

1.1 Features

Ultra-low power consumption technology

- V_{DD} = single power supply voltage of 1.6 to 5.5 V
- HALT mode
- STOP mode
  High-speed wakeup from the STOP mode is possible.
- SNOOZE mode

RL78 CPU core

- CISC architecture with 3-stage pipeline
- The minimum instruction execution time can be changed from high to ultra-low speed.
  - High speed: 0.02083 µs at 48 MHz operation with the high-speed on-chip oscillator clock or the PLL clock
  - Ultra-low speed: 30.5 µs at 32.768 kHz operation with the subsystem clock
- Multiply/divide/multiply & accumulate instructions are supported.
- Address space: 1 Mbyte
- General-purpose registers: (8-bit register × 8) × 4 banks
- On-chip RAM: 12 Kbytes

FAA core

- Multiplication: 32-bit signed × 32-bit signed → 32-bit signed
- Results of 64-bit multiplication can be right-shifted by a desired number of bits.
- Addition: 32-bit signed + 32-bit signed → 32-bit signed (internally calculated with 33-bit precision)
- Subtraction: 32-bit signed - 32-bit signed → 32-bit signed (internally calculated with 33-bit precision)
- Limit operation: Operation parameter registers (33 bits × 4) in which upper and lower limits can be set.
- Operation parameter registers (32 bits × 6)
- Address pointer registers (12 bits × 6)
- On-chip code RAM: 4 Kbytes
- On-chip data RAM: 2 Kbytes
- Multiple interrupts available
- A 32-byte shared memory is included for sharing of data by the RL78 CPU and FAA core.

Divider

- 32-bit ÷ 32-bit = 32-bit unsigned

Code flash memory

- 64 or 128 Kbytes
- Block size: 2 Kbytes
- Security function: Prohibition of block erase and rewriting
- On-chip debugging
- Self-programming with boot swapping and flash shield window

Data flash memory

- 4 Kbytes
- Background operation (BGO): Instructions can be executed from the program memory while rewriting the data flash memory.
- Number of rewrites: 1,000,000 times (typ.)

High-speed on-chip oscillator

- Selectable from among 64 MHz, 48 MHz, 32 MHz, 24 MHz, 16 MHz, 12 MHz, 8 MHz, 6 MHz, 4 MHz, 3 MHz, 2 MHz, and 1 MHz
- High accuracy: ±1.0% (V_{DD} = 1.8 to 5.5 V, T_A = –20 to +85°C)

Middle-speed on-chip oscillator

- Selectable from among 4 MHz, 2 MHz, and 1 MHz with adjustability

Low-speed on-chip oscillator

- 32.768 kHz (typ.) with adjustability

Operating ambient temperature

- T_A = –40 to +85°C (2D: Consumer applications)
- T_A = –40 to +105°C (3C: Industrial applications)
- T_A = –40 to +125°C (4C: Industrial applications)

Power management and resetting

- On-chip power-on-reset (POR) circuit
- On-chip voltage detectors (LVD0 and LVD1)

Data transfer controller (DTC)

- Transfer modes: Normal transfer mode, repeat transfer mode, and block transfer mode
- Activated by interrupt sources
- Chain transfer

Event link controller (ELC)

- 34 event signals can be set up between specified peripheral functions.

Serial interfaces

- Two to six simplified SPIs (CSIs[Note 1])
- Two to three UART/UART (LIN-bus supported) interfaces
- Two to six I^2C/Simplified I^2C interfaces
- Single digital addressable lighting interface (DALI)
- Single I^2C (SM/PM bus) interface

Timers

- 16-bit timers: 4-channel timer array unit (TAU)
  Single-channel timer RJ
  2-channel timer RD2 with PWMOPA
  Single-channel timer RG2
  Single-channel timer RX
- 32-bit interval timer: Single channel in 32-bit counter mode
  Two channels in 16-bit counter mode
  Four channels in 8-bit counter mode
- Single-channel realtime clock: Counting of one second to 99 years, alarm interrupt, and clock correction
- Single-channel watchdog timer: Operates with the low-speed on-chip oscillator clock
- Three 16-bit timers KB30, KB31, and KB32: Two outputs each (up to six outputs), complementary output timers for power control, timer restart, smooth start, PWM output gating, dithering, fast and asynchronous forced output stop triggered by a comparator or external interrupt, single or interleaved power factor correction (PFC) control, maximum frequency limit, fixed off control, pulse characteristics measurement, multi-phase operation, output with 651-ps average resolution (operation at 96 MHz and with the dithering being applied)

A/D converter

- 8-/10-/12-bit resolution
- 12 to 23 analog input channels: Two channels include a sample and hold circuit each.
- Internal reference voltage (1.48 V) and temperature sensor

D/A converter

- 8-/10-bit resolution (V_{DD} = 2.7 to 5.5 V)
- Two to three analog output channels
- Output voltage: 0 V to V_{DD}
- Realtime output

Comparator module

- Four channels
- The external reference voltage and the D/A converter output are selectable as the reference voltage.
- Time window output functioning with the timer array unit

Programmable gain amplifier

- Single amplifier

Input/output port pins

- Number of port pins: 26 to 120
  N-ch open drain I/O pins (withstand voltage of 6 V): 2 to 4
  N-ch open drain I/O pins (withstand voltage of V_{DD}**Note 2**/withstand voltage of EV_{DD}**Note 3**): 7 to 19
  Controlled current drive port pins: 2 to 8
- Can be set to N-ch open drain or TTL input buffer, and use of an on-chip pull-up resistor can be specified.
- Connectable to a device with different voltage (1.8, 2.5, or 3 V)

Others

- Key interrupt
- Clock output/buzzer output controller
- Binary-coded decimal (BCD) 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. This applies to the 20- to 52-pin products.

Note 3. This applies to the 64-pin products.

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

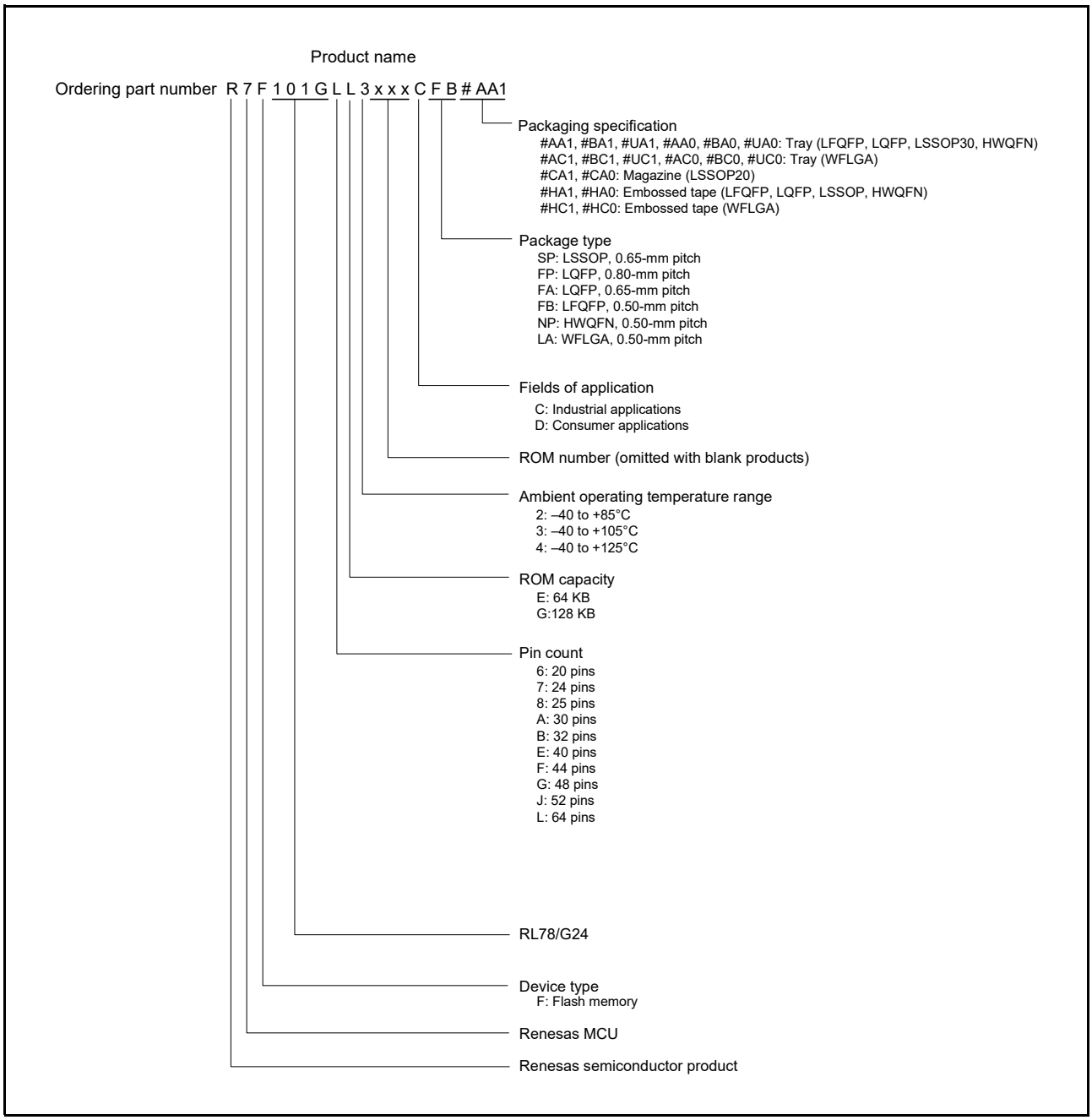
● ROM and RAM capacities

Flash ROM	Data flash memory	RAM	RL78/G24				
			20 pins	24 pins	25 pins	30 pins	32 pins
128 KB	4 KB	12 KB	R7F101G6G	R7F101G7G	R7F101G8G	R7F101GAG	R7F101GBG
64 KB			R7F101G6E	R7F101G7E	R7F101G8E	R7F101GAE	R7F101GBE

Flash ROM	Data flash memory	RAM	RL78/G24				
			40 pins	44 pins	48 pins	52 pins	64 pins
128 KB	4 KB	12 KB	R7F101GEG	R7F101GFG	R7F101GGG	R7F101GJG	R7F101GLG
64 KB			R7F101GEE	R7F101GFE	R7F101GGE	R7F101GJE	R7F101GLE

1.2 List of Part Numbers

<R> Figure 1 - 1 Part Number, Memory Size, and Package of RL78/G24



<R> Table 1 - 1 List of Ordering Part Numbers

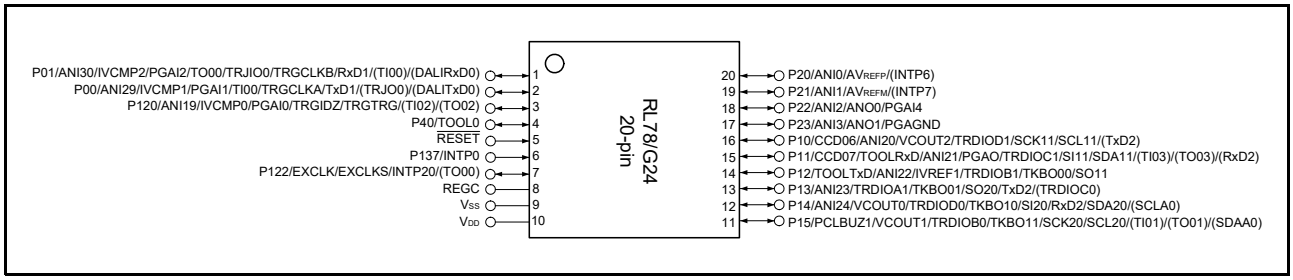
Pin Count	Package	Fields of Application Note	Ordering Part Number		Renesas Code
			Product Name	Packaging Specification	
20	20-pin plastic LSSOP (4.4 × 6.5 mm, 0.65-mm pitch)	C	R7F101G6G4CSP, R7F101G6E4CSP, R7F101G6G3CSP, R7F101G6E3CSP	#CA1, #CA0, #HA1, #HA0	PLSP0020JB-A
		D	R7F101G6G2DSP, R7F101G6E2DSP		
24	24-pin plastic HWQFN (4 × 4 mm, 0.50-mm pitch)	C	R7F101G7G4CNP, R7F101G7E4CNP, R7F101G7G3CNP, R7F101G7E3CNP	#AA1, #AA0, #BA1, #BA0, #UA1, #UA0, #HA1, #HA0	PWQN0024KG-A
		D	R7F101G7G2DNP, R7F101G7E2DNP		
25	25-pin plastic WFLGA (3 × 3 mm, 0.50-mm pitch)	C	R7F101G8G3CLA, R7F101G8E3CLA	#AC1, #AC0, #BC1, #BC0, #UC1, #UC0, #HC1, #HC0	PWLG0025KB-A
		D	R7F101G8G2DLA, R7F101G8E2DLA		
30	30-pin plastic LSSOP (7.62 mm (300), 0.65-mm pitch)	C	R7F101GAG4CSP, R7F101GAE4CSP, R7F101GAG3CSP, R7F101GAE3CSP	#AA1, #AA0, #BA1, #BA0, #UA1, #UA0, #HA1, #HA0	PLSP0030JB-B
		D	R7F101GAG2DSP, R7F101GAE2DSP		
32	32-pin plastic HWQFN (5 × 5 mm, 0.50-mm pitch)	C	R7F101GBG4CNP, R7F101GBE4CNP, R7F101GBG3CNP, R7F101GBE3CNP	#AA1, #AA0, #BA1, #BA0, #UA1, #UA0, #HA1, #HA0	PWQN0032KE-A
		D	R7F101GBG2DNP, R7F101GBE2DNP		
	32-pin plastic LQFP (7 × 7 mm, 0.80-mm pitch)	C	R7F101GBG3CFP, R7F101GBE3CFP	#AA1, #AA0, #BA1, #BA0, #UA1, #UA0, #HA1, #HA0	PLQP0032GB-A
		D	R7F101GBG2DFP, R7F101GBE2DFP		
40	40-pin plastic HWQFN (6 × 6 mm, 0.50-mm pitch)	C	R7F101GEG4CNP, R7F101GEE4CNP, R7F101GEG3CNP, R7F101GEE3CNP	#AA1, #AA0, #BA1, #BA0, #UA1, #UA0, #HA1, #HA0	PWQN0040KD-A
		D	R7F101GEG2DNP, R7F101GEE2DNP		
44	44-pin plastic LQFP (10 × 10 mm, 0.80-mm pitch)	C	R7F101GFG3CFP, R7F101GFE3CFP	#AA1, #AA0, #BA1, #BA0, #UA1, #UA0, #HA1, #HA0	PLQP0044GC-A
		D	R7F101GFG2DFP, R7F101GFE2DFP		
48	48-pin plastic LFQFP (7 × 7 mm, 0.50-mm pitch)	C	R7F101GGG4CFB, R7F101GGE4CFB, R7F101GGG3CFB, R7F101GGE3CFB	#AA1, #AA0, #BA1, #BA0, #UA1, #UA0, #HA1, #HA0	PLQP0048KB-B
		D	R7F101GGG2DFB, R7F101GGE2DFB		
	48-pin plastic HWQFN (7 × 7 mm, 0.50-mm pitch)	C	R7F101GGG3CNP, R7F101GGE3CNP	#AA1, #AA0, #BA1, #BA0, #UA1, #UA0, #HA1, #HA0	PWQN0048KC-A
		D	R7F101GGG2DNP, R7F101GGE2DNP		
52	52-pin plastic LQFP (10 × 10 mm, 0.65-mm pitch)	C	R7F101GJG4CFA, R7F101GJE4CFA, R7F101GJG3CFA, R7F101GJE3CFA	#AA1, #AA0, #BA1, #BA0, #UA1, #UA0, #HA1, #HA0	PLQP0052JA-A
		D	R7F101GJG2DFA, R7F101GJE2DFA		
64	64-pin plastic LQFP (12 × 12 mm, 0.65-mm pitch)	C	R7F101GLG3CFA, R7F101GLE3CFA	#AA1, #AA0, #BA1, #BA0, #UA1, #UA0, #HA1, #HA0	PLQP0064JA-A
		D	R7F101GLG2DFA, R7F101GLE2DFA		
	64-pin plastic LFQFP (10 × 10 mm, 0.50-mm pitch)	C	R7F101GLG3CFB, R7F101GLE3CFB	#AA1, #AA0, #BA1, #BA0, #UA1, #UA0, #HA1, #HA0	PLQP0064KB-C
		D	R7F101GLG2DFB, R7F101GLE2DFB		

Note For the fields of application, see **Figure 1 - 1 Part Number, Memory Size, and Package of RL78/G24**.

1.3 Pin Configuration (Top View)

1.3.1 20-pin products

- 20-pin plastic LSSOP (4.4 × 6.5 mm, 0.65-mm pitch)



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

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

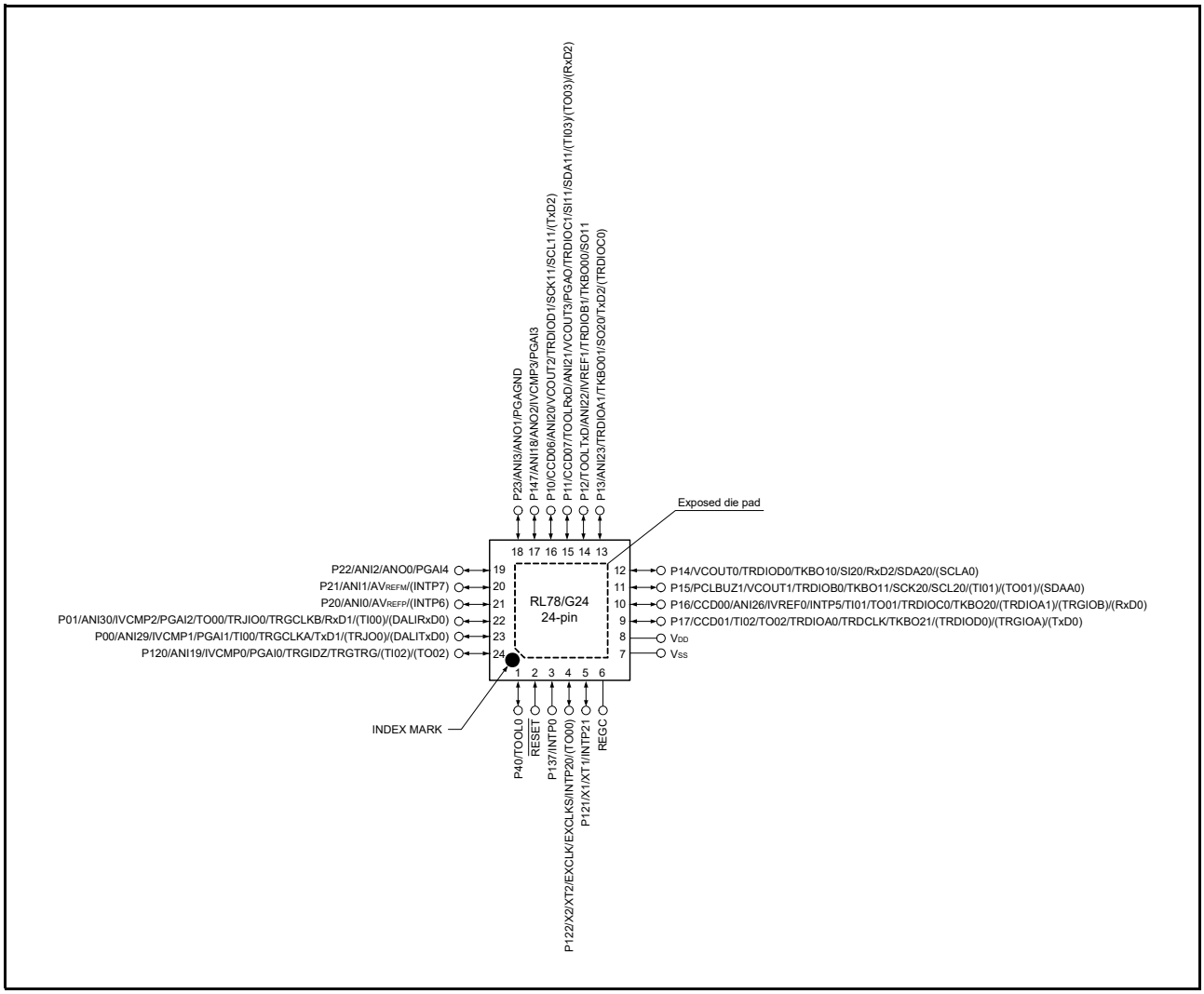
Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register x (PIORx). Refer to **Figure 7 - 8 Format of Peripheral I/O Redirection Register 0 (PIOR0)** to **Figure 7 - 11 Format of Peripheral I/O Redirection Register 3 (PIOR3)**.

Table 1 - 2 Multiplexed Pin Functions of the 20-pin Products

Pin Number	I/O		Power supply, system clock, and debugging	Analog Circuits				HMIs		Timers						Communications Interfaces		
20LSSOP	Digital port	Controlled current drive port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Programmable gain amplifier (PGA)	Interrupt (INTP)	Key interrupt (KR)	Timer array unit (TAU)	Timer RJ (TRJ)	Timer RD2 (TRD2)	Timer RG2 (TRG2)	16-bit timers KB30, KB31, and KB32 (TMKB3)	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Digital addressable lighting interface (DALI)
1	P01	—	—	ANI30	—	IVCMP2	PGAI2	—	—	TO00/ (TI00)	TRJIO0	—	TRGCLKB	—	—	RxD1	—	(DALIRxD0)
2	P00	—	—	ANI29	—	IVCMP1	PGAI1	—	—	TI00	(TRJO0)	—	TRGCLKA	—	—	TxD1	—	(DALITxD0)
3	P120	—	—	ANI19	—	IVCMP0	PGAI0	—	—	(TI02)/ (TO02)	—	—	TRGIDZ/ TRGTRG	—	—	—	—	—
4	P40	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
5	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
6	P137	—	—	—	—	—	—	INTP0	—	—	—	—	—	—	—	—	—	—
7	P122	—	EXCLK/ EXCLKS	—	—	—	—	INTP20	—	(TO00)	—	—	—	—	—	—	—	—
8	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
9	—	—	V_{SS}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
10	—	—	V_{DD}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
11	P15	—	PCLBUZ1	—	—	VCOUT1	—	—	—	(TI01)/ (TO01)	—	TRDIOB0	—	TKBO11	—	SCK20/ SCL20	(SDAA0)	—
12	P14	—	—	ANI24	—	VCOUT0	—	—	—	—	—	TRDIOD0	—	TKBO10	—	SI20/ RxD2/ SDA20	(SCLA0)	—
13	P13	—	—	ANI23	—	—	—	—	—	—	—	TRDIOA1/ (TRDIOC0)	—	TKBO01	—	SO20/ TxD2	—	—
14	P12	—	TOOLTxD	ANI22	—	IVREF1	—	—	—	—	—	TRDIOB1	—	TKBO00	—	SO11	—	—
15	P11	CCD07	TOOLRxD	ANI21	—	—	PGAO	—	—	(TI03)/ (TO03)	—	TRDIOC1	—	—	—	SI11/ SDA11/ (RxD2)	—	—
16	P10	CCD06	—	ANI20	—	VCOUT2	—	—	—	—	—	TRDIOD1	—	—	—	SCK11/ SCL11/ (TxD2)	—	—
17	P23	—	—	ANI3	ANO1	—	PGAGND	—	—	—	—	—	—	—	—	—	—	—
18	P22	—	—	ANI2	ANO0	—	PGAI4	—	—	—	—	—	—	—	—	—	—	—
19	P21	—	—	ANI1/ AV_{REFM}	—	—	—	(INTP7)	—	—	—	—	—	—	—	—	—	—
20	P20	—	—	ANI0/ AV_{REFP}	—	—	—	(INTP6)	—	—	—	—	—	—	—	—	—	—

1.3.2 24-pin products

- 24-pin plastic HWQFN (4 × 4 mm, 0.50-mm pitch)



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

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

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register x (PIORx). Refer to **Figure 7 - 8 Format of Peripheral I/O Redirection Register 0 (PIOR0)** to **Figure 7 - 11 Format of Peripheral I/O Redirection Register 3 (PIOR3)**.

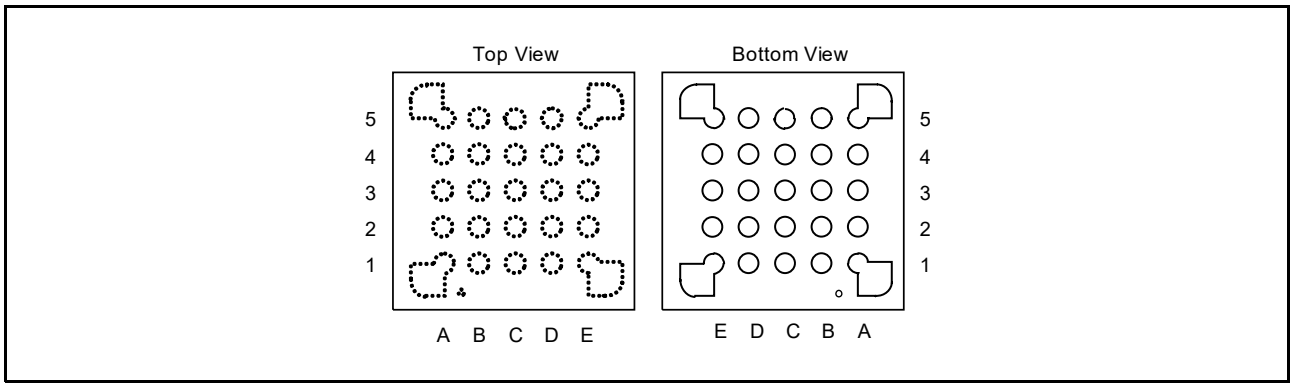
Remark 3. For the product in a QFN package, we recommend soldering the exposed die pad onto a plated area of the printed circuit board that has no electrical connections.

Table 1 - 3 Multiplexed Pin Functions of the 24-pin Products

Pin Number	I/O			Analog Circuits				HMIs		Timers						Communications Interfaces		
24HWQFN	Digital port	Controlled current drive port	Power supply, system clock, and debugging	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Programmable gain amplifier (PGA)	Interrupt (INTP)	Key interrupt (KR)	Timer array unit (TAU)	Timer RJ (TRJ)	Timer RD2 (TRD2)	Timer RG2 (TRG2)	16-bit timers KB30, KB31, and KB32 (TMKB3)	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Digital addressable lighting interface (DALI)
1	P40	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
2	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
3	P137	—	—	—	—	—	—	INTP0	—	—	—	—	—	—	—	—	—	—
4	P122	—	X2/XT2/ EXCLK/ EXCLKS	—	—	—	—	INTP20	—	(TO00)	—	—	—	—	—	—	—	—
5	P121	—	X1/XT1	—	—	—	—	INTP21	—	—	—	—	—	—	—	—	—	—
6	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
7	—	—	V_{SS}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
8	—	—	V_{DD}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
9	P17	CCD01	—	—	—	—	—	—	—	TI02/ TO02	—	TRDIOA0/ (TRDIOD0)/ TRDCLK	(TRGIOA)	TKBO21	—	(TxD0)	—	—
10	P16	CCD00	—	ANI26	—	IVREF0	—	INTP5	—	TI01/ TO01	—	TRDIOC0/ (TRDIOA1)	(TRGIOB)	TKBO20	—	(RxD0)	—	—
11	P15	—	PCLBUZ1	—	—	VCOUT1	—	—	—	(TI01)/ (TO01)	—	TRDIOB0	—	TKBO11	—	SCK20/ SCL20	(SDAA0)	—
12	P14	—	—	—	—	VCOUT0	—	—	—	—	—	TRDIOD0	—	TKBO10	—	SI20/ RxD2/ SDA20	(SCLA0)	—
13	P13	—	—	ANI23	—	—	—	—	—	—	—	TRDIOA1/ (TRDIOC0)	—	TKBO01	—	SO20/ TxD2	—	—
14	P12	—	TOOLTxD	ANI22	—	IVREF1	—	—	—	—	—	TRDIOB1	—	TKBO00	—	SO11	—	—
15	P11	CCD07	TOOLRxD	ANI21	—	VCOUT3	PGAO	—	—	(TI03)/ (TO03)	—	TRDIOC1	—	—	—	SI11/ SDA11/ (RxD2)	—	—
16	P10	CCD06	—	ANI20	—	VCOUT2	—	—	—	—	—	TRDIOD1	—	—	—	SCK11/ SCL11/ (TxD2)	—	—
17	P147	—	—	ANI18	ANO2	IVCMP3	PGAI3	—	—	—	—	—	—	—	—	—	—	—
18	P23	—	—	ANI3	ANO1	—	PGAGND	—	—	—	—	—	—	—	—	—	—	—
19	P22	—	—	ANI2	ANO0	—	PGAI4	—	—	—	—	—	—	—	—	—	—	—
20	P21	—	—	ANI1/ AV_{REFM}	—	—	—	(INTP7)	—	—	—	—	—	—	—	—	—	—
21	P20	—	—	ANI0/ AV_{REFP}	—	—	—	(INTP6)	—	—	—	—	—	—	—	—	—	—
22	P01	—	—	ANI30	—	IVCMP2	PGAI2	—	—	TO00/ (TI00)	TRJIO0	—	TRGCLKB	—	—	RxD1	—	(DALIRxD0)
23	P00	—	—	ANI29	—	IVCMP1	PGAI1	—	—	TI00	(TRJO0)	—	TRGCLKA	—	—	TxD1	—	(DALITxD0)
24	P120	—	—	ANI19	—	IVCMP0	PGAI0	—	—	(TI02)/ (TO02)	—	—	TRGIDZ/ TRGTRG	—	—	—	—	—

1.3.3 25-pin products

- 25-pin plastic WFLGA (3 × 3 mm, 0.50-mm pitch)



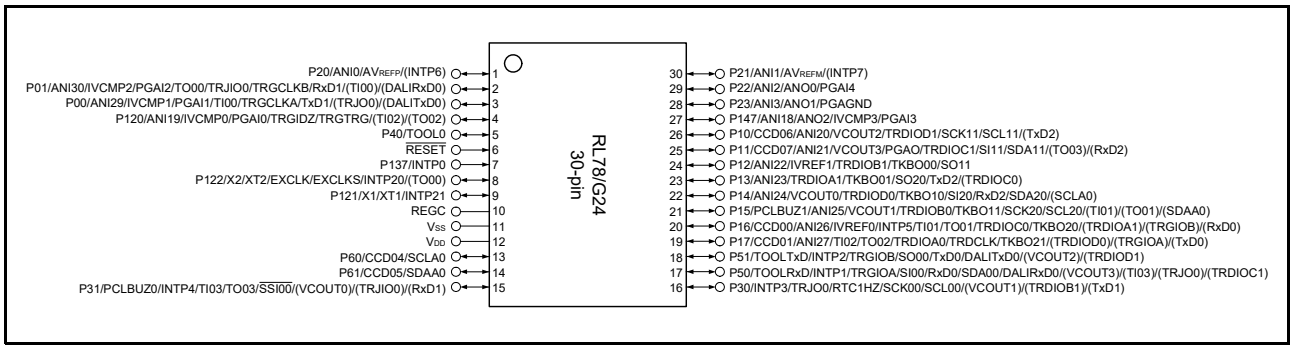
	A	B	C	D	E
5	P40/TOOL0	$\overline{\text{RESET}}$	P01/ANI30/IVCMP2/PGAI2/ TO00/TRJIO0/TRGCLKB/ RxD1/(TI00)/(DALIRxD0)	P22/ANI2/ANO0/PGAI4	P147/ANI18/ANO2/ IVCMP3/PGAI3
4	P122/X2/XT2/EXCLK/ EXCLKS/INTP20/(TO00)	P137/INTP0	P00/ANI29/IVCMP1/PGAI1/ TI00/TRGCLKA/TxD1/ (TRJO0)/(DALITxD0)	P21/ANI1/AVREFM/(INTP7)	P10/CCD06/ANI20/ VCOUT2/TRDIOD1/SCK11/ SCL11/(TxD2)
3	P121/X1/XT1/INTP21	VDD	P20/ANI0/AVREFP/(INTP6)	P12/TOOLTxD/ANI22/ IVREF1/TRDIOB1/ TKBO00/SO11	P11/CCD07/TOOLRxD/ ANI21/VCOUT3/PGAO/ TRDIOC1/SI11/SDA11/ (TI03)/(TO03)/(RxD2)
2	REGC	VSS	P23/ANI3/ANO1/PGAGND	P14/ANI24/VCOUT0/ TRDIOD0/TKBO10/SI20/ RxD2/SDA20/(SCLA0)	P13/ANI23/TRDIOA1/ TKBO01/SO20/TxD2/ (TRDIOC0)
1	P17/CCD01/ANI27/TI02/ TO02/TRDIOA0/TRDCLK/ TKBO21/(TRDIOD0)/ (TRGIOA)/(TxD0)	P16/CCD00/ANI26/ IVREF0/INTP5/TI01/TO01/ TRDIOC0/TKBO20/ (TRDIOA1)/(TRGIOB)/ (RxD0)	P120/ANI19/IVCMP0/ PGAI0/TRGIDZ/TRGTRG/ (TI02)/(TO02)	P15/PCLBUZ1/ANI25/ VCOUT1/TRDIOB0/ TKBO11/SCK20/SCL20/ (TI01)/(TO01)/(SDAA0)	P130

Table 1 - 4 Multiplexed Pin Functions of the 25-pin Products

Pin Number	I/O			Analog Circuits				HMIs		Timers						Communications Interfaces		
25WFLGA	Digital port	Controlled current drive port	Power supply, system clock, and debugging	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Programmable gain amplifier (PGA)	Interrupt (INTP)	Key interrupt (KR)	Timer array unit (TAU)	Timer RJ (TRJ)	Timer RD2 (TRD2)	Timer RG2 (TRG2)	16-bit timers KB30, KB31, and KB32 (TMKB3)	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Digital addressable lighting interface (DALI)
A1	P17	CCD01	—	ANI27	—	—	—	—	—	TI02/ TO02	—	TRDIOA0/ (TRDIOD0)/ TRDCLK	(TRGIOA)	TKBO21	—	(TxD0)	—	—
A2	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
A3	P121	—	X1/XT1	—	—	—	—	INTP21	—	—	—	—	—	—	—	—	—	—
A4	P122	—	X2/XT2/ EXCLK/ EXCLKS	—	—	—	—	INTP20	—	(TO00)	—	—	—	—	—	—	—	—
A5	P40	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
B1	P16	CCD00	—	ANI26	—	IVREF0	—	INTP5	—	TI01/ TO01	—	TRDIOC0/ (TRDIOA1)	(TRGIOB)	TKBO20	—	(RxD0)	—	—
B2	—	—	V_{SS}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
B3	—	—	V_{DD}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
B4	P137	—	—	—	—	—	—	INTP0	—	—	—	—	—	—	—	—	—	—
B5	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
C1	P120	—	—	ANI19	—	IVCMP0	PGAI0	—	—	(TI02)/ (TO02)	—	—	TRGIDZ/ TRGTRG	—	—	—	—	—
C2	P23	—	—	ANI3	ANO1	—	PGAGND	—	—	—	—	—	—	—	—	—	—	—
C3	P20	—	—	ANI0/ AV_{REFP}	—	—	—	(INTP6)	—	—	—	—	—	—	—	—	—	—
C4	P00	—	—	ANI29	—	IVCMP1	PGAI1	—	—	TI00	(TRJO0)	—	TRGCLKA	—	—	TxD1	—	(DALITxD0)
C5	P01	—	—	ANI30	—	IVCMP2	PGAI2	—	—	TO00/ (TI00)	TRJIO0	—	TRGCLKB	—	—	RxD1	—	(DALIRxD0)
D1	P15	—	PCLBUZ1	ANI25	—	VCOUT1	—	—	—	(TI01)/ (TO01)	—	TRDIOB0	—	TKBO11	—	SCK20/ SCL20	(SDAA0)	—
D2	P14	—	—	ANI24	—	VCOUT0	—	—	—	—	—	TRDIOD0	—	TKBO10	—	SI20/ RxD2/ SDA20	(SCLA0)	—
D3	P12	—	TOOLTxD	ANI22	—	IVREF1	—	—	—	—	—	TRDIOB1	—	TKBO00	—	SO11	—	—
D4	P21	—	—	ANI1/ AV_{REFM}	—	—	—	(INTP7)	—	—	—	—	—	—	—	—	—	—
D5	P22	—	—	ANI2	ANO0	—	PGAI4	—	—	—	—	—	—	—	—	—	—	—
E1	P130	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
E2	P13	—	—	ANI23	—	—	—	—	—	—	—	TRDIOA1/ (TRDIOC0)	—	TKBO01	—	SO20/ TxD2	—	—
E3	P11	CCD07	TOOLRxD	ANI21	—	VCOUT3	PGAO	—	—	(TI03)/ (TO03)	—	TRDIOC1	—	—	—	SI11/ SDA11/ (RxD2)	—	—
E4	P10	CCD06	—	ANI20	—	VCOUT2	—	—	—	—	—	TRDIOD1	—	—	—	SCK11/ SCL11/ (TxD2)	—	—
E5	P147	—	—	ANI18	ANO2	IVCMP3	PGAI3	—	—	—	—	—	—	—	—	—	—	—

1.3.4 30-pin products

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



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

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

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register x (PIORx). Refer to **Figure 7 - 8 Format of Peripheral I/O Redirection Register 0 (PIOR0)** to **Figure 7 - 11 Format of Peripheral I/O Redirection Register 3 (PIOR3)**.

Table 1 - 5 Multiplexed Pin Functions of the 30-pin Products (1/2)

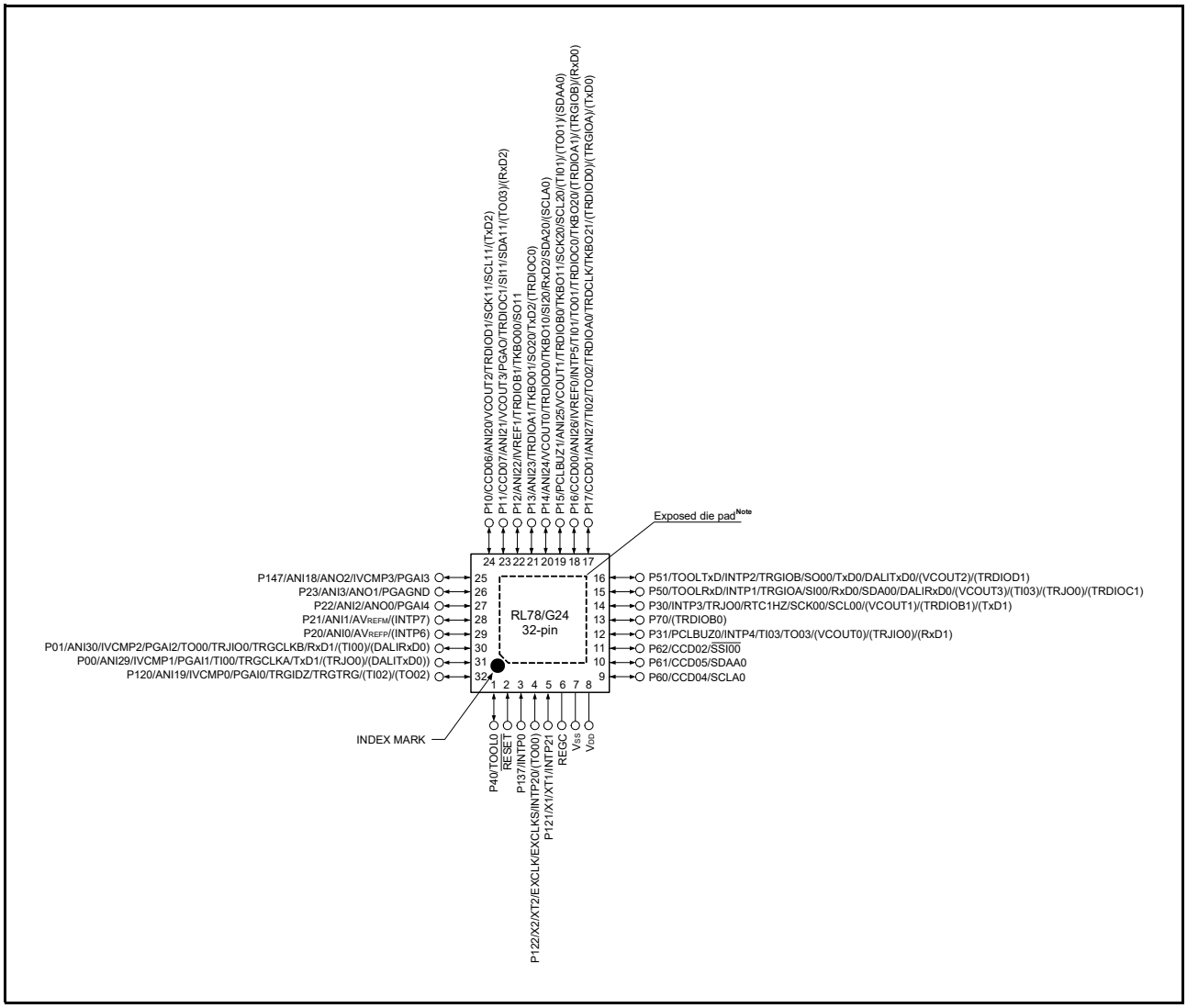
Pin Number	I/O			Analog Circuits				HMIs		Timers						Communications Interfaces		
30LSSOP	Digital port	Controlled current drive port	Power supply, system clock, and debugging	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Programmable gain amplifier (PGA)	Interrupt (INTP)	Key interrupt (KR)	Timer array unit (TAU)	Timer RJ (TRJ)	Timer RD2 (TRD2)	Timer RG2 (TRG2)	16-bit timers KB30, KB31, and KB32 (TMKB3)	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Digital addressable lighting interface (DALI)
1	P20	—	—	ANI0/ AVREFP	—	—	—	(INTP6)	—	—	—	—	—	—	—	—	—	—
2	P01	—	—	ANI30	—	IVCMP2	PGAI2	—	—	TO00/ (TI00)	TRJIO0	—	TRGCLKB	—	—	RxD1	—	(DALIRxD0)
3	P00	—	—	ANI29	—	IVCMP1	PGAI1	—	—	TI00	(TRJO0)	—	TRGCLKA	—	—	TxD1	—	(DALITxD0)
4	P120	—	—	ANI19	—	IVCMP0	PGAI0	—	—	(TI02)/ (TO02)	—	—	TRGIDZ/ TRGTRG	—	—	—	—	—
5	P40	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
6	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
7	P137	—	—	—	—	—	—	INTP0	—	—	—	—	—	—	—	—	—	—
8	P122	—	X2/XT2/ EXCLK/ EXCLKS	—	—	—	—	INTP20	—	(TO00)	—	—	—	—	—	—	—	—
9	P121	—	X1/XT1	—	—	—	—	INTP21	—	—	—	—	—	—	—	—	—	—
10	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
11	—	—	VSS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
12	—	—	VDD	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
13	P60	CCD04	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SCLA0	—
14	P61	CCD05	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDAA0	—
15	P31	—	PCLBUZ0	—	—	(VCOUT0)	—	INTP4	—	TI03/ TO03	(TRJIO0)	—	—	—	—	$\overline{\text{SSI00}}$/ (RxD1)	—	—
16	P30	—	—	—	—	(VCOUT1)	—	INTP3	—	—	TRJO0	(TRDIOB1)	—	—	RTC1HZ	SCK00/ SCL00/ (TxD1)	—	—
17	P50	—	TOOLRxD	—	—	(VCOUT3)	—	INTP1	—	(TI03)	(TRJO0)	(TRDIOC1)	TRGIOA	—	—	SI00/ RxD0/ SDA00	—	DALIRxD0
18	P51	—	TOOLTxD	—	—	(VCOUT2)	—	INTP2	—	—	—	(TRDIOD1)	TRGIOB	—	—	SO00/ TxD0	—	DALITxD0
19	P17	CCD01	—	ANI27	—	—	—	—	—	TI02/ TO02	—	TRDIOA0/ (TRDIOD0)/ TRDCLK	(TRGIOA)	TKBO21	—	(TxD0)	—	—
20	P16	CCD00	—	ANI26	—	IVREF0	—	INTP5	—	TI01/ TO01	—	TRDIOC0/ (TRDIOA1)	(TRGIOB)	TKBO20	—	(RxD0)	—	—
21	P15	—	PCLBUZ1	ANI25	—	VCOUT1	—	—	—	(TI01)/ (TO01)	—	TRDIOB0	—	TKBO11	—	SCK20/ SCL20	(SDAA0)	—
22	P14	—	—	ANI24	—	VCOUT0	—	—	—	—	—	TRDIOD0	—	TKBO10	—	SI20/ RxD2/ SDA20	(SCLA0)	—
23	P13	—	—	ANI23	—	—	—	—	—	—	—	TRDIOA1/ (TRDIOC0)	—	TKBO01	—	SO20/ TxD2	—	—
24	P12	—	—	ANI22	—	IVREF1	—	—	—	—	—	TRDIOB1	—	TKBO00	—	SO11	—	—
25	P11	CCD07	—	ANI21	—	VCOUT3	PGAO	—	—	(TO03)	—	TRDIOC1	—	—	—	SI11/ SDA11/ (RxD2)	—	—
26	P10	CCD06	—	ANI20	—	VCOUT2	—	—	—	—	—	TRDIOD1	—	—	—	SCK11/ SCL11/ (TxD2)	—	—

Table 1 - 5 Multiplexed Pin Functions of the 30-pin Products (2/2)

Pin Number	I/O		Power supply, system clock, and debugging	Analog Circuits				HMIs		Timers						Communications Interfaces		
30LSSOP	Digital port	Controlled current drive port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Programmable gain amplifier (PGA)	Interrupt (INTP)	Key interrupt (KR)	Timer array unit (TAU)	Timer RJ (TRJ)	Timer RD2 (TRD2)	Timer RG2 (TRG2)	16-bit timers KB30, KB31, and KB32 (TMKB3)	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Digital addressable lighting interface (DALI)
27	P147	—	—	ANI18	ANO2	IVCMP3	PGAI3	—	—	—	—	—	—	—	—	—	—	—
28	P23	—	—	ANI3	ANO1	—	PGAGND	—	—	—	—	—	—	—	—	—	—	—
29	P22	—	—	ANI2	ANO0	—	PGAI4	—	—	—	—	—	—	—	—	—	—	—
30	P21	—	—	ANI1/ AVREFM	—	—	—	(INTP7)	—	—	—	—	—	—	—	—	—	—

1.3.5 32-pin products

- 32-pin plastic HWQFN (5 × 5 mm, 0.50-mm pitch)
- 32-pin plastic LQFP (7 × 7 mm, 0.80-mm pitch)



Note The 32-pin plastic LQFP (7 × 7 mm, 0.80-mm pitch) products do not have an exposed die pad.

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

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

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register x (PIORx). Refer to **Figure 7 - 8 Format of Peripheral I/O Redirection Register 0 (PIOR0)** to **Figure 7 - 11 Format of Peripheral I/O Redirection Register 3 (PIOR3)**.

Remark 3. For the product in a QFN package, we recommend soldering the exposed die pad onto a plated area of the printed circuit board that has no electrical connections.

Table 1 - 6 Multiplexed Pin Functions of the 32-pin Products (1/2)

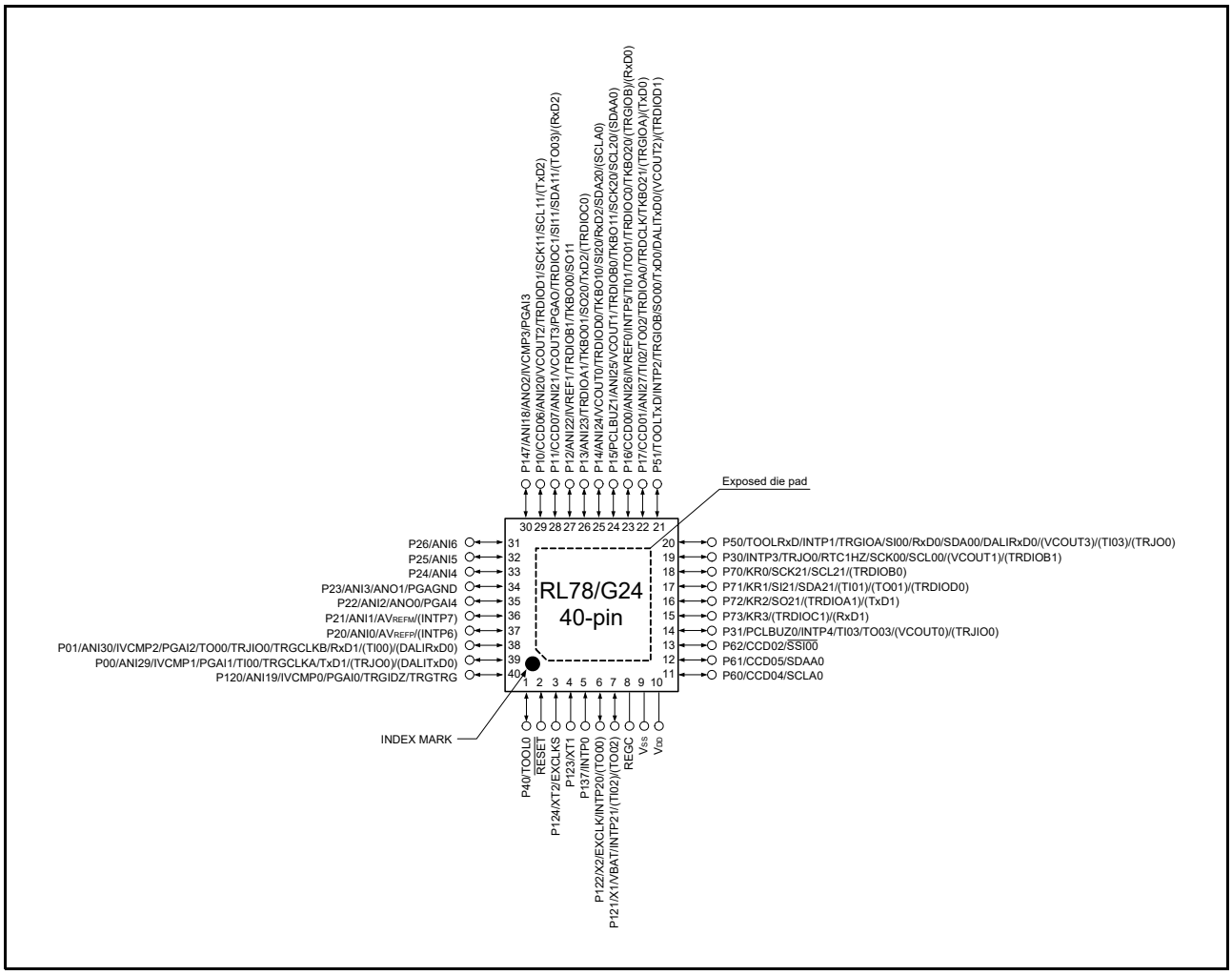
Pin Number	I/O			Analog Circuits				HMIs		Timers						Communications Interfaces		
32HWQFN, 32LQFP	Digital port	Controlled current drive port	Power supply, system clock, and debugging	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Programmable gain amplifier (PGA)	Interrupt (INTP)	Key interrupt (KR)	Timer array unit (TAU)	Timer RJ (TRJ)	Timer RD2 (TRD2)	Timer RG2 (TRG2)	16-bit timers KB30, KB31, and KB32 (TMKB3)	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Digital addressable lighting interface (DALI)
1	P40	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
2	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
3	P137	—	—	—	—	—	—	INTP0	—	—	—	—	—	—	—	—	—	—
4	P122	—	X2/XT2/ EXCLK/ EXCLKS	—	—	—	—	INTP20	—	(TO00)	—	—	—	—	—	—	—	—
5	P121	—	X1/XT1	—	—	—	—	INTP21	—	—	—	—	—	—	—	—	—	—
6	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
7	—	—	V_{SS}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
8	—	—	V_{DD}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
9	P60	CCD04	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SCLA0	—
10	P61	CCD05	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDAA0	—
11	P62	CCD02	—	—	—	—	—	—	—	—	—	—	—	—	—	$\overline{\text{SSI00}}$	—	—
12	P31	—	PCLBUZ0	—	—	(VCOUT0)	—	INTP4	—	TI03/ TO03	(TRJIO0)	—	—	—	—	(RxD1)	—	—
13	P70	—	—	—	—	—	—	—	—	—	—	(TRDIOB0)	—	—	—	—	—	—
14	P30	—	—	—	—	(VCOUT1)	—	INTP3	—	—	TRJO0	(TRDIOB1)	—	—	RTC1HZ	SCK00/ SCL00/ (TxD1)	—	—
15	P50	—	TOOLRxD	—	—	(VCOUT3)	—	INTP1	—	(TI03)	(TRJO0)	(TRDIOC1)	TRGIOA	—	—	SI00/ RxD0/ SDA00	—	DALIRxD0
16	P51	—	TOOLTxD	—	—	(VCOUT2)	—	INTP2	—	—	—	(TRDIOD1)	TRGIOB	—	—	SO00/ TxD0	—	DALITxD0
17	P17	CCD01	—	ANI27	—	—	—	—	—	TI02/ TO02	—	TRDIOA0/ (TRDIOD0)/ TRDCLK	(TRGIOA)	TKBO21	—	(TxD0)	—	—
18	P16	CCD00	—	ANI26	—	IVREF0	—	INTP5	—	TI01/ TO01	—	TRDIOC0/ (TRDIOA1)	(TRGIOB)	TKBO20	—	(RxD0)	—	—
19	P15	—	PCLBUZ1	ANI25	—	VCOUT1	—	—	—	(TI01)/ (TO01)	—	TRDIOB0	—	TKBO11	—	SCK20/ SCL20	(SDAA0)	—
20	P14	—	—	ANI24	—	VCOUT0	—	—	—	—	—	TRDIOD0	—	TKBO10	—	SI20/ RxD2/ SDA20	(SCLA0)	—
21	P13	—	—	ANI23	—	—	—	—	—	—	—	TRDIOA1/ (TRDIOC0)	—	TKBO01	—	SO20/ TxD2	—	—
22	P12	—	—	ANI22	—	IVREF1	—	—	—	—	—	TRDIOB1	—	TKBO00	—	SO11	—	—
23	P11	CCD07	—	ANI21	—	VCOUT3	PGAO	—	—	(TO03)	—	TRDIOC1	—	—	—	SI11/ SDA11/ (RxD2)	—	—
24	P10	CCD06	—	ANI20	—	VCOUT2	—	—	—	—	—	TRDIOD1	—	—	—	SCK11/ SCL11/ (TxD2)	—	—
25	P147	—	—	ANI18	ANO2	IVCMP3	PGAI3	—	—	—	—	—	—	—	—	—	—	—
26	P23	—	—	ANI3	ANO1	—	PGAGND	—	—	—	—	—	—	—	—	—	—	—
27	P22	—	—	ANI2	ANO0	—	PGAI4	—	—	—	—	—	—	—	—	—	—	—

Table 1 - 6 Multiplexed Pin Functions of the 32-pin Products (2/2)

Pin Number	I/O		Power supply, system clock, and debugging	Analog Circuits				HMIs		Timers						Communications Interfaces		
32HWQFN, 32LQFP	Digital port	Controlled current drive port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Programmable gain amplifier (PGA)	Interrupt (INTP)	Key interrupt (KR)	Timer array unit (TAU)	Timer RJ (TRJ)	Timer RD2 (TRD2)	Timer RG2 (TRG2)	16-bit timers KB30, KB31, and KB32 (TMKB3)	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Digital addressable lighting interface (DALI)
28	P21	—	—	ANI1/ AVREFM	—	—	—	(INTP7)	—	—	—	—	—	—	—	—	—	—
29	P20	—	—	ANI0/ AVREFP	—	—	—	(INTP6)	—	—	—	—	—	—	—	—	—	—
30	P01	—	—	ANI30	—	IVCMP2	PGAI2	—	—	TO00/ (TI00)	TRJIO0	—	TRGCLKB	—	—	RxD1	—	(DALIRxD0)
31	P00	—	—	ANI29	—	IVCMP1	PGAI1	—	—	TI00	(TRJO0)	—	TRGCLKA	—	—	TxD1	—	(DALITxD0)
32	P120	—	—	ANI19	—	IVCMP0	PGAI0	—	—	(TI02)/ (TO02)	—	—	TRGIDZ/ TRGTRG	—	—	—	—	—

1.3.6 40-pin products

- 40 -pin plastic HWQFN (6 × 6 mm, 0.50-mm pitch)



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

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

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register x (PIORx). Refer to **Figure 7 - 8 Format of Peripheral I/O Redirection Register 0 (PIOR0)** to **Figure 7 - 11 Format of Peripheral I/O Redirection Register 3 (PIOR3)**.

Remark 3. For the product in a QFN package, we recommend soldering the exposed die pad onto a plated area of the printed circuit board that has no electrical connections.

Table 1 - 7 Multiplexed Pin Functions of the 40-pin Products (1/2)

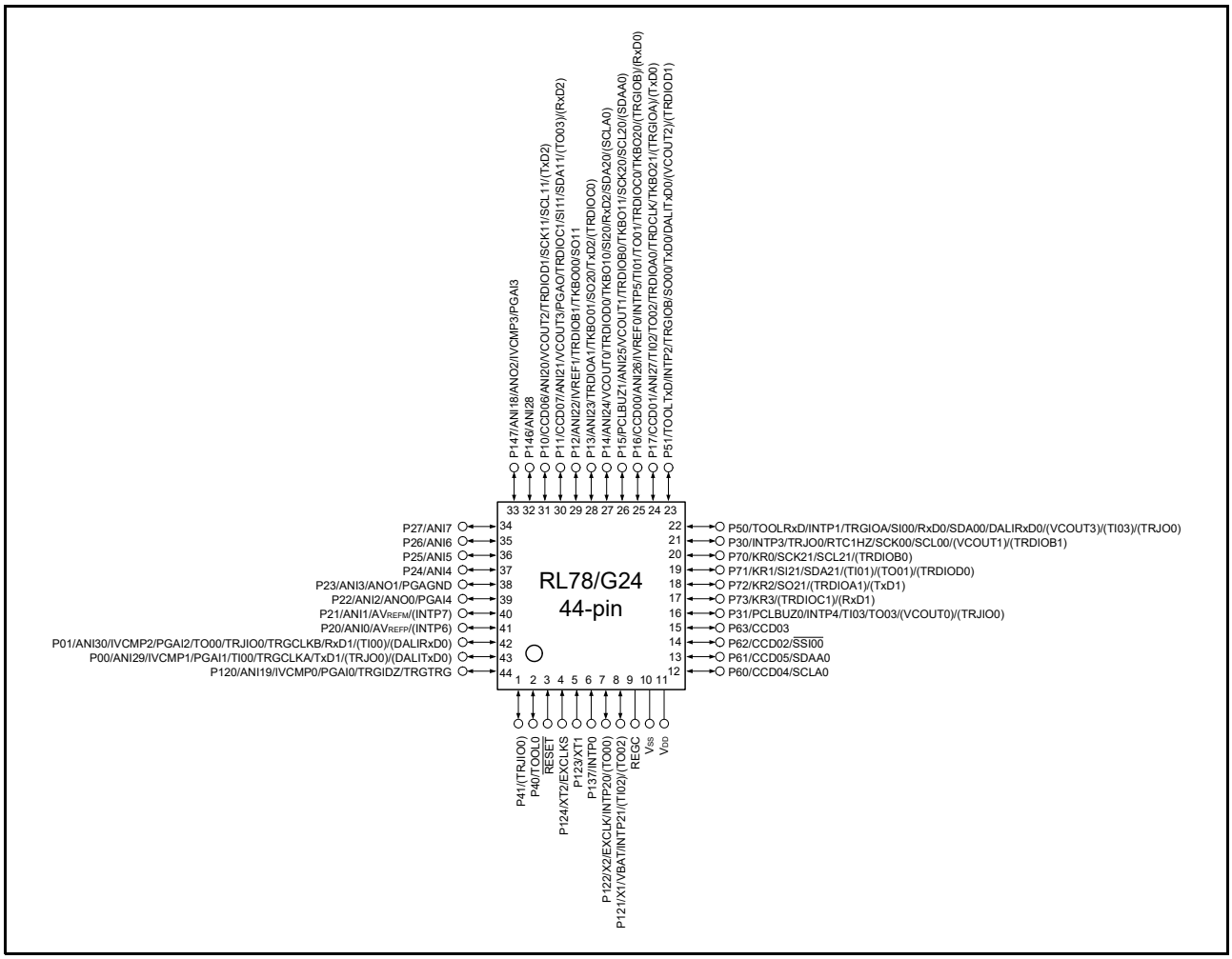
Pin Number	I/O		Power supply, system clock, and debugging	Analog Circuits				HMIs		Timers						Communications Interfaces		
40HWQFN	Digital port	Controlled current drive port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Programmable gain amplifier (PGA)	Interrupt (INTP)	Key interrupt (KR)	Timer array unit (TAU)	Timer RJ (TRJ)	Timer RD2 (TRD2)	Timer RG2 (TRG2)	16-bit timers KB30, KB31, and KB32 (TMKB3)	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Digital addressable lighting interface (DALI)
1	P40	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
2	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
3	P124	—	XT2/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
4	P123	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
5	P137	—	—	—	—	—	—	INTP0	—	—	—	—	—	—	—	—	—	—
6	P122	—	X2/EXCLK	—	—	—	—	INTP20	—	(TO00)	—	—	—	—	—	—	—	—
7	P121	—	X1/VBAT	—	—	—	—	INTP21	—	(TI02)/ (TO02)	—	—	—	—	—	—	—	—
8	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
9	—	—	V_{SS}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
10	—	—	V_{DD}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
11	P60	CCD04	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SCLA0	—
12	P61	CCD05	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDAA0	—
13	P62	CCD02	—	—	—	—	—	—	—	—	—	—	—	—	—	$\overline{\text{SSI00}}$	—	—
14	P31	—	PCLBUZ0	—	—	(VCOUT0)	—	INTP4	—	TI03/ TO03	(TRJIO0)	—	—	—	—	—	—	—
15	P73	—	—	—	—	—	—	—	KR3	—	—	(TRDIOC1)	—	—	—	(RxD1)	—	—
16	P72	—	—	—	—	—	—	—	KR2	—	—	(TRDIOA1)	—	—	—	SO21/ (TxD1)	—	—
17	P71	—	—	—	—	—	—	—	KR1	(TI01)/ (TO01)	—	(TRDIOD0)	—	—	—	SI21/ SDA21	—	—
18	P70	—	—	—	—	—	—	—	KR0	—	—	(TRDIOB0)	—	—	—	SCK21/ SCL21	—	—
19	P30	—	—	—	—	(VCOUT1)	—	INTP3	—	—	TRJO0	(TRDIOB1)	—	—	RTC1HZ	SCK00/ SCL00	—	—
20	P50	—	TOOLRxD	—	—	(VCOUT3)	—	INTP1	—	(TI03)	(TRJO0)	—	TRGIOA	—	—	SI00/ RxD0/ SDA00	—	DALIRxD0
21	P51	—	TOOLTxD	—	—	(VCOUT2)	—	INTP2	—	—	—	(TRDIOD1)	TRGIOB	—	—	SO00/ TxD0	—	DALITxD0
22	P17	CCD01	—	ANI27	—	—	—	—	—	TI02/ TO02	—	TRDIOA0/ TRDCLK	(TRGIOA)	TKBO21	—	(TxD0)	—	—
23	P16	CCD00	—	ANI26	—	IVREF0	—	INTP5	—	TI01/ TO01	—	TRDIOC0	(TRGIOB)	TKBO20	—	(RxD0)	—	—
24	P15	—	PCLBUZ1	ANI25	—	VCOUT1	—	—	—	—	—	TRDIOB0	—	TKBO11	—	SCK20/ SCL20	(SDAA0)	—
25	P14	—	—	ANI24	—	VCOUT0	—	—	—	—	—	TRDIOD0	—	TKBO10	—	SI20/ RxD2/ SDA20	(SCLA0)	—
26	P13	—	—	ANI23	—	—	—	—	—	—	—	TRDIOA1/ (TRDIOC0)	—	TKBO01	—	SO20/ TxD2	—	—
27	P12	—	—	ANI22	—	IVREF1	—	—	—	—	—	TRDIOB1	—	TKBO00	—	SO11	—	—
28	P11	CCD07	—	ANI21	—	VCOUT3	PGAO	—	—	(TO03)	—	TRDIOC1	—	—	—	SI11/ SDA11/ (RxD2)	—	—

Table 1 - 7 Multiplexed Pin Functions of the 40-pin Products (2/2)

Pin Number	I/O		Power supply, system clock, and debugging	Analog Circuits				HMIs		Timers						Communications Interfaces		
40HWQFN	Digital port	Controlled current drive port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Programmable gain amplifier (PGA)	Interrupt (INTP)	Key interrupt (KR)	Timer array unit (TAU)	Timer RJ (TRJ)	Timer RD2 (TRD2)	Timer RG2 (TRG2)	16-bit timers KB30, KB31, and KB32 (TMKB3)	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Digital addressable lighting interface (DALI)
29	P10	CCD06	—	ANI20	—	VCOUT2	—	—	—	—	—	TRDIOD1	—	—	—	SCK11/ SCL11/ (TxD2)	—	—
30	P147	—	—	ANI18	ANO2	IVCMP3	PGAI3	—	—	—	—	—	—	—	—	—	—	—
31	P26	—	—	ANI6	—	—	—	—	—	—	—	—	—	—	—	—	—	—
32	P25	—	—	ANI5	—	—	—	—	—	—	—	—	—	—	—	—	—	—
33	P24	—	—	ANI4	—	—	—	—	—	—	—	—	—	—	—	—	—	—
34	P23	—	—	ANI3	ANO1	—	PGAGND	—	—	—	—	—	—	—	—	—	—	—
35	P22	—	—	ANI2	ANO0	—	PGAI4	—	—	—	—	—	—	—	—	—	—	—
36	P21	—	—	ANI1/ AV_{REFM}	—	—	—	(INTP7)	—	—	—	—	—	—	—	—	—	—
37	P20	—	—	ANI0/ AV_{REFP}	—	—	—	(INTP6)	—	—	—	—	—	—	—	—	—	—
38	P01	—	—	ANI30	—	IVCMP2	PGAI2	—	—	TO00/ (TI00)	TRJIO0	—	TRGCLKB	—	—	RxD1	—	(DALIRxD0)
39	P00	—	—	ANI29	—	IVCMP1	PGAI1	—	—	TI00	(TRJO0)	—	TRGCLKA	—	—	TxD1	—	(DALITxD0)
40	P120	—	—	ANI19	—	IVCMP0	PGAI0	—	—	—	—	—	TRGIDZ/ TRGTRG	—	—	—	—	—

1.3.7 44-pin products

- 44-pin plastic LQFP (10 × 10 mm, 0.80-mm pitch)



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

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

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register x (PIORx). Refer to **Figure 7 - 8 Format of Peripheral I/O Redirection Register 0 (PIOR0)** to **Figure 7 - 11 Format of Peripheral I/O Redirection Register 3 (PIOR3)**.

Table 1 - 8 Multiplexed Pin Functions of the 44-pin Products (1/2)

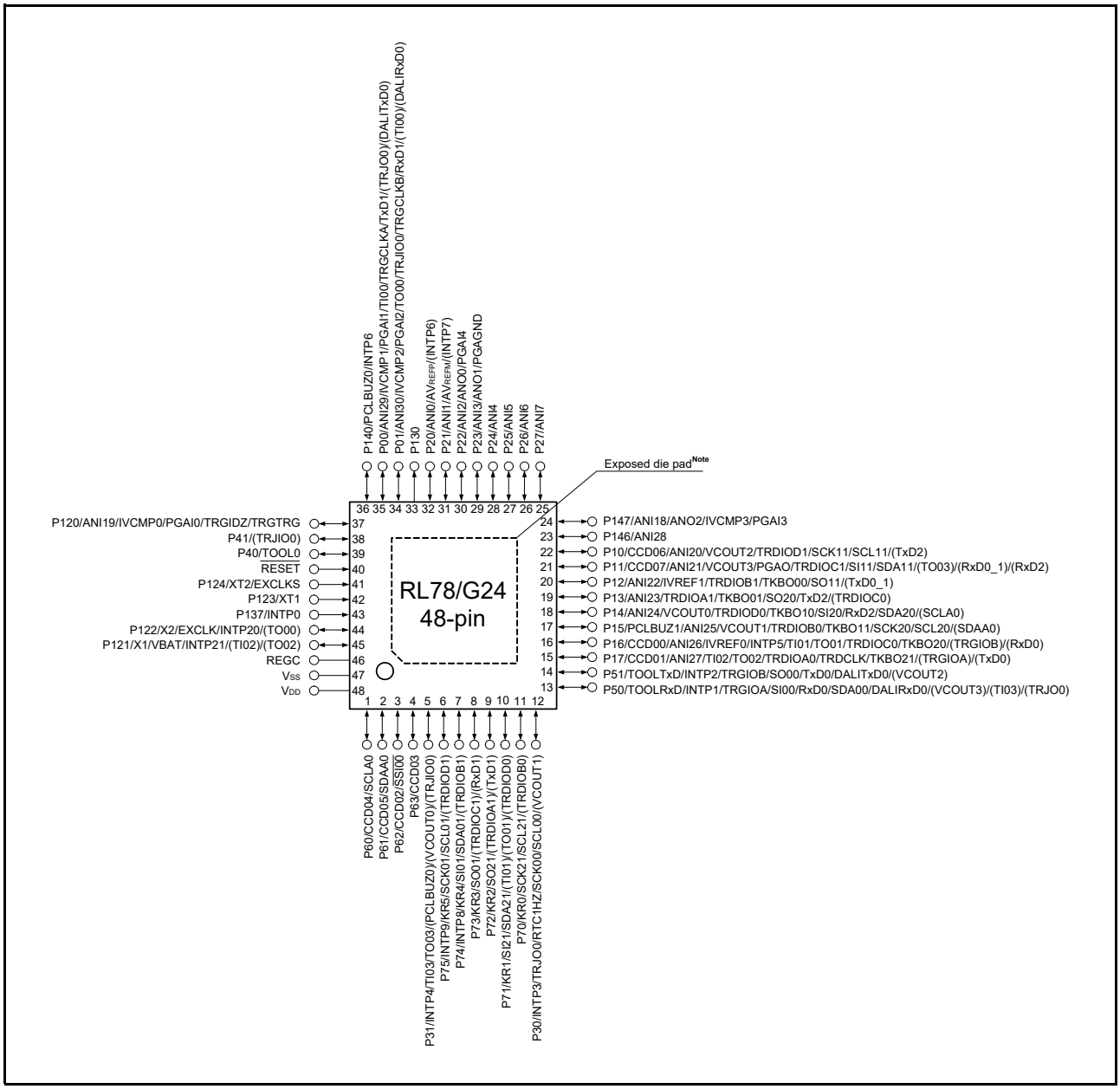
Pin Number	I/O			Analog Circuits				HMIs		Timers						Communications Interfaces		
44LQFP	Digital port	Controlled current drive port	Power supply, system clock, and debugging	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Programmable gain amplifier (PGA)	Interrupt (INTP)	Key interrupt (KR)	Timer array unit (TAU)	Timer RJ (TRJ)	Timer RD2 (TRD2)	Timer RG2 (TRG2)	16-bit timers KB30, KB31, and KB32 (TMKB3)	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Digital addressable lighting interface (DALI)
1	P41	—	—	—	—	—	—	—	—	—	(TRJIO0)	—	—	—	—	—	—	—
2	P40	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
3	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
4	P124	—	XT2/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
5	P123	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
6	P137	—	—	—	—	—	—	INTP0	—	—	—	—	—	—	—	—	—	—
7	P122	—	X2/EXCLK	—	—	—	—	INTP20	—	(TO00)	—	—	—	—	—	—	—	—
8	P121	—	X1/VBAT	—	—	—	—	INTP21	—	(TI02)/ (TO02)	—	—	—	—	—	—	—	—
9	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
10	—	—	V_{SS}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
11	—	—	V_{DD}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
12	P60	CCD04	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SCLA0	—
13	P61	CCD05	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDAA0	—
14	P62	CCD02	—	—	—	—	—	—	—	—	—	—	—	—	—	$\overline{\text{SSI00}}$	—	—
15	P63	CCD03	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
16	P31	—	PCLBUZ0	—	—	(VCOUT0)	—	INTP4	—	TI03/ TO03	(TRJIO0)	—	—	—	—	—	—	—
17	P73	—	—	—	—	—	—	—	KR3	—	—	(TRDIOC1)	—	—	—	(RxD1)	—	—
18	P72	—	—	—	—	—	—	—	KR2	—	—	(TRDIOA1)	—	—	—	SO21/ (TxD1)	—	—
19	P71	—	—	—	—	—	—	—	KR1	(TI01)/ (TO01)	—	(TRDIOD0)	—	—	—	SI21/S DA21	—	—
20	P70	—	—	—	—	—	—	—	KR0	—	—	(TRDIOB0)	—	—	—	SCK21/ SCL21	—	—
21	P30	—	—	—	—	(VCOUT1)	—	INTP3	—	—	TRJO0	(TRDIOB1)	—	—	RTC1HZ	SCK00/ SCL00	—	—
22	P50	—	TOOLRxD	—	—	(VCOUT3)	—	INTP1	—	(TI03)	(TRJO0)	—	TRGIOA	—	—	SI00/ RxD0/ SDA00	—	DALIRxD0
23	P51	—	TOOLTxD	—	—	(VCOUT2)	—	INTP2	—	—	—	(TRDIOD1)	TRGIOB	—	—	SO00/ TxD0	—	DALITxD0
24	P17	CCD01	—	ANI27	—	—	—	—	—	TI02/ TO02	—	TRDIOA0/ TRDCLK	(TRGIOA)	TKBO21	—	(TxD0)	—	—
25	P16	CCD00	—	ANI26	—	IVREF0	—	INTP5	—	TI01/ TO01	—	TRDIOC0	(TRGIOB)	TKBO20	—	(RxD0)	—	—
26	P15	—	PCLBUZ1	ANI25	—	VCOUT1	—	—	—	—	—	TRDIOB0	—	TKBO11	—	SCK20/ SCL20	—	(SDAA0)
27	P14	—	—	ANI24	—	VCOUT0	—	—	—	—	—	TRDIOD0	—	TKBO10	—	SI20/ RxD2/ SDA20	(SCLA0)	—
28	P13	—	—	ANI23	—	—	—	—	—	—	—	TRDIOA1/ (TRDIOC0)	—	TKBO01	—	SO20/ TxD2	—	—
29	P12	—	—	ANI22	—	IVREF1	—	—	—	—	—	TRDIOB1	—	TKBO00	—	SO11	—	—

Table 1 - 8 Multiplexed Pin Functions of the 44-pin Products (2/2)

Pin Number	I/O		Power supply, system clock, and debugging	Analog Circuits				HMIs		Timers						Communications Interfaces		
44LQFP	Digital port	Controlled current drive port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Programmable gain amplifier (PGA)	Interrupt (INTP)	Key interrupt (KR)	Timer array unit (TAU)	Timer RJ (TRJ)	Timer RD2 (TRD2)	Timer RG2 (TRG2)	16-bit timers KB30, KB31, and KB32 (TMKB3)	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Digital addressable lighting interface (DALI)
30	P11	CCD07	—	ANI21	—	VCOUT3	PGAO	—	—	(TO03)	—	TRDIOC1	—	—	—	SI11/ SDA11/ (RxD2)	—	—
31	P10	CCD06	—	ANI20	—	VCOUT2	—	—	—	—	—	TRDIOD1	—	—	—	SCK11/ SCL11/ (TxD2)	—	—
32	P146	—	—	ANI28	—	—	—	—	—	—	—	—	—	—	—	—	—	—
33	P147	—	—	ANI18	ANO2	IVCMP3	PGAI3	—	—	—	—	—	—	—	—	—	—	—
34	P27	—	—	ANI7	—	—	—	—	—	—	—	—	—	—	—	—	—	—
35	P26	—	—	ANI6	—	—	—	—	—	—	—	—	—	—	—	—	—	—
36	P25	—	—	ANI5	—	—	—	—	—	—	—	—	—	—	—	—	—	—
37	P24	—	—	ANI4	—	—	—	—	—	—	—	—	—	—	—	—	—	—
38	P23	—	—	ANI3	ANO1	—	PGAGND	—	—	—	—	—	—	—	—	—	—	—
39	P22	—	—	ANI2	ANO0	—	PGAI4	—	—	—	—	—	—	—	—	—	—	—
40	P21	—	—	ANI1/ AV_{REFM}	—	—	—	(INTP7)	—	—	—	—	—	—	—	—	—	—
41	P20	—	—	ANI0/ AV_{REFP}	—	—	—	(INTP6)	—	—	—	—	—	—	—	—	—	—
42	P01	—	—	ANI30	—	IVCMP2	PGAI2	—	—	TO00/ (TI00)	TRJIO0	—	TRGCLKB	—	—	RxD1	—	(DALIRxD0)
43	P00	—	—	ANI29	—	IVCMP1	PGAI1	—	—	TI00	(TRJO0)	—	TRGCLKA	—	—	TxD1	—	(DALITxD0)
44	P120	—	—	ANI19	—	IVCMP0	PGAI0	—	—	—	—	—	TRGIDZ/ TRGTRG	—	—	—	—	—

1.3.8 48-pin products

- 48-pin plastic LFQFP (7 × 7 mm, 0.50-mm pitch)
- 48-pin plastic HWQFN (7 × 7 mm, 0.50-mm pitch)



Note The 48-pin plastic LQFP (7 × 7 mm, 0.50-mm pitch) products do not have an exposed die pad.

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

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

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register x (PIORx). Refer to **Figure 7 - 8 Format of Peripheral I/O Redirection Register 0 (PIOR0)** to **Figure 7 - 11 Format of Peripheral I/O Redirection Register 3 (PIOR3)**.

Remark 3. For the product in a QFN package, we recommend soldering the exposed die pad onto a plated area of the printed circuit board that has no electrical connections.

Table 1 - 9 Multiplexed Pin Functions of the 48-pin Products (1/2)

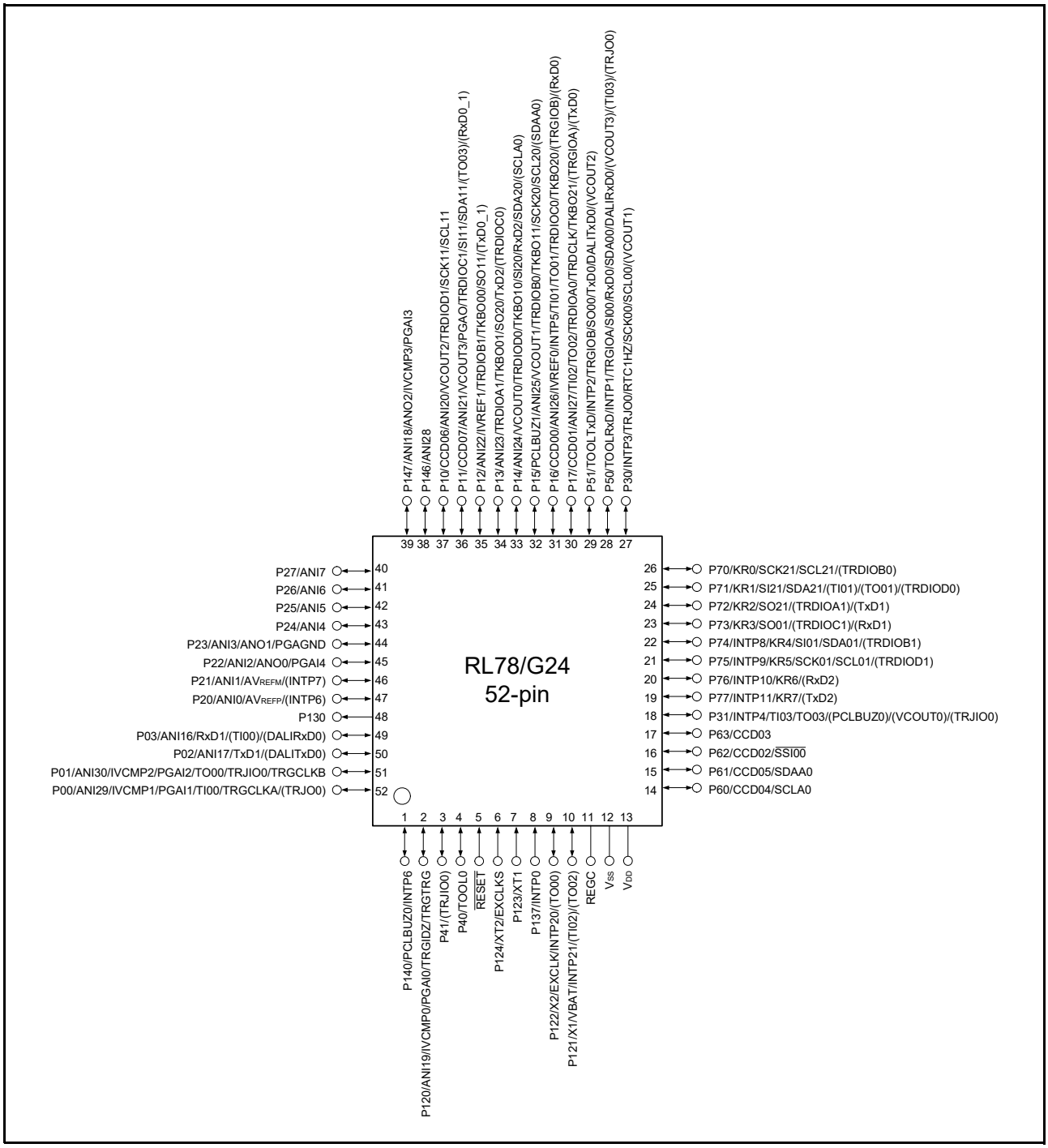
Pin Number	I/O			Analog Circuits				HMIs		Timers						Communications Interfaces		
48LFQFP, 48HWQFN	Digital port	Controlled current drive port	Power supply, system clock, and debugging	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Programmable gain amplifier (PGA)	Interrupt (INTP)	Key interrupt (KR)	Timer array unit (TAU)	Timer RJ (TRJ)	Timer RD2 (TRD2)	Timer RG2 (TRG2)	16-bit timers KB30, KB31, and KB32 (TMKB3)	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Digital addressable lighting interface (DALI)
1	P60	CCD04	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SCLA0	—
2	P61	CCD05	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDAA0	—
3	P62	CCD02	—	—	—	—	—	—	—	—	—	—	—	—	—	$\overline{SSI00}$	—	—
4	P63	CCD03	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
5	P31	—	(PCLBUZ0)	—	—	(VCOUT0)	—	INTP4	—	TI03/ TO03	(TRJIO0)	—	—	—	—	—	—	—
6	P75	—	—	—	—	—	—	INTP9	KR5	—	—	(TRDIOD1)	—	—	—	SCK01/ SCL01	—	—
7	P74	—	—	—	—	—	—	INTP8	KR4	—	—	(TRDIOB1)	—	—	—	SI01/ SDA01	—	—
8	P73	—	—	—	—	—	—	—	KR3	—	—	(TRDIOC1)	—	—	—	SO01/ (RxD1)	—	—
9	P72	—	—	—	—	—	—	—	KR2	—	—	(TRDIOA1)	—	—	—	SO21/ (TxD1)	—	—
10	P71	—	—	—	—	—	—	—	KR1	(TI01)/ (TO01)	—	(TRDIOD0)	—	—	—	SI21/ SDA21	—	—
11	P70	—	—	—	—	—	—	—	KR0	—	—	(TRDIOB0)	—	—	—	SCK21/ SCL21	—	—
12	P30	—	—	—	—	(VCOUT1)	—	INTP3	—	—	TRJO0	—	—	—	RTC1HZ	SCK00/ SCL00	—	—
13	P50	—	TOOLRxD	—	—	(VCOUT3)	—	INTP1	—	(TI03)	(TRJO0)	—	TRGIOA	—	—	SI00/ RxD0/ SDA00	—	DALIRxD0
14	P51	—	TOOLTxD	—	—	(VCOUT2)	—	INTP2	—	—	—	—	TRGIOB	—	—	SO00/ TxD0	—	DALITxD0
15	P17	CCD01	—	ANI27	—	—	—	—	—	TI02/ TO02	—	TRDIOA0/ TRDCLK	(TRGIOA)	TKBO21	—	(TxD0)	—	—
16	P16	CCD00	—	ANI26	—	IVREF0	—	INTP5	—	TI01/ TO01	—	TRDIOC0	(TRGIOB)	TKBO20	—	(RxD0)	—	—
17	P15	—	PCLBUZ1	ANI25	—	VCOUT1	—	—	—	—	—	TRDIOB0	—	TKBO11	—	SCK20/ SCL20	—	(SDAA0)
18	P14	—	—	ANI24	—	VCOUT0	—	—	—	—	—	TRDIOD0	—	TKBO10	—	SI20/ RxD2/ SDA20	—	(SCLA0)
19	P13	—	—	ANI23	—	—	—	—	—	—	—	TRDIOA1/ (TRDIOC0)	—	TKBO01	—	SO20/ TxD2	—	—
20	P12	—	—	ANI22	—	IVREF1	—	—	—	—	—	TRDIOB1	—	TKBO00	—	SO11/ (TxD0_ 1)	—	—
21	P11	CCD07	—	ANI21	—	VCOUT3	PGAO	—	—	(TO03)	—	TRDIOC1	—	—	—	SI11/ SDA11/ (RxD0_ 1)/ (RxD2)	—	—
22	P10	CCD06	—	ANI20	—	VCOUT2	—	—	—	—	—	TRDIOD1	—	—	—	SCK11/ SCL11/ (TxD2)	—	—
23	P146	—	—	ANI28	—	—	—	—	—	—	—	—	—	—	—	—	—	—
24	P147	—	—	ANI18	ANO2	IVCMP3	PGAI3	—	—	—	—	—	—	—	—	—	—	—

Table 1 - 9 Multiplexed Pin Functions of the 48-pin Products (2/2)

Pin Number	I/O		Power supply, system clock, and debugging	Analog Circuits				HMIs		Timers						Communications Interfaces		
48LFQFP, 48HWQFN	Digital port	Controlled current drive port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Programmable gain amplifier (PGA)	Interrupt (INTP)	Key interrupt (KR)	Timer array unit (TAU)	Timer RJ (TRJ)	Timer RD2 (TRD2)	Timer RG2 (TRG2)	16-bit timers KB30, KB31, and KB32 (TMKB3)	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Digital addressable lighting interface (DALI)
25	P27	—	—	ANI7	—	—	—	—	—	—	—	—	—	—	—	—	—	—
26	P26	—	—	ANI6	—	—	—	—	—	—	—	—	—	—	—	—	—	—
27	P25	—	—	ANI5	—	—	—	—	—	—	—	—	—	—	—	—	—	—
28	P24	—	—	ANI4	—	—	—	—	—	—	—	—	—	—	—	—	—	—
29	P23	—	—	ANI3	ANO1	—	PGAGND	—	—	—	—	—	—	—	—	—	—	—
30	P22	—	—	ANI2	ANO0	—	PGAI4	—	—	—	—	—	—	—	—	—	—	—
31	P21	—	—	ANI1/ AVREFM	—	—	—	(INTP7)	—	—	—	—	—	—	—	—	—	—
32	P20	—	—	ANI0/ AVREFP	—	—	—	(INTP6)	—	—	—	—	—	—	—	—	—	—
33	P130	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
34	P01	—	—	ANI30	—	IVCMP2	PGAI2	—	—	TO00/ (TI00)	TRJIO0	—	TRGCLKB	—	—	RxD1	—	(DALIRxD0)
35	P00	—	—	ANI29	—	IVCMP1	PGAI1	—	—	TI00	(TRJO0)	—	TRGCLKA	—	—	TxD1	—	(DALITxD0)
36	P140		PCLBUZ0	—	—	—	—	INTP6	—	—	—	—	—	—	—	—	—	—
37	P120	—	—	ANI19	—	IVCMP0	PGAI0	—	—	—	—	—	TRGIDZ/ TRGTRG	—	—	—	—	—
38	P41	—	—	—	—	—	—	—	—	—	(TRJIO0)	—	—	—	—	—	—	—
39	P40	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
40	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
41	P124	—	XT2/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
42	P123	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
43	P137	—	—	—	—	—	—	INTP0	—	—	—	—	—	—	—	—	—	—
44	P122	—	X2/EXCLK	—	—	—	—	INTP20	—	(TO00)	—	—	—	—	—	—	—	—
45	P121	—	X1/VBAT	—	—	—	—	INTP21	—	(TI02)/ (TO02)	—	—	—	—	—	—	—	—
46	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
47	—	—	VSS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
48	—	—	VDD	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

1.3.9 52-pin products

- 52-pin plastic LQFP (10 × 10 mm, 0.65-mm pitch)



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

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

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register x (PIORx). Refer to **Figure 7 - 8 Format of Peripheral I/O Redirection Register 0 (PIOR0)** to **Figure 7 - 11 Format of Peripheral I/O Redirection Register 3 (PIOR3)**.

Table 1 - 10 Multiplexed Pin Functions of the 52-pin Products (1/2)

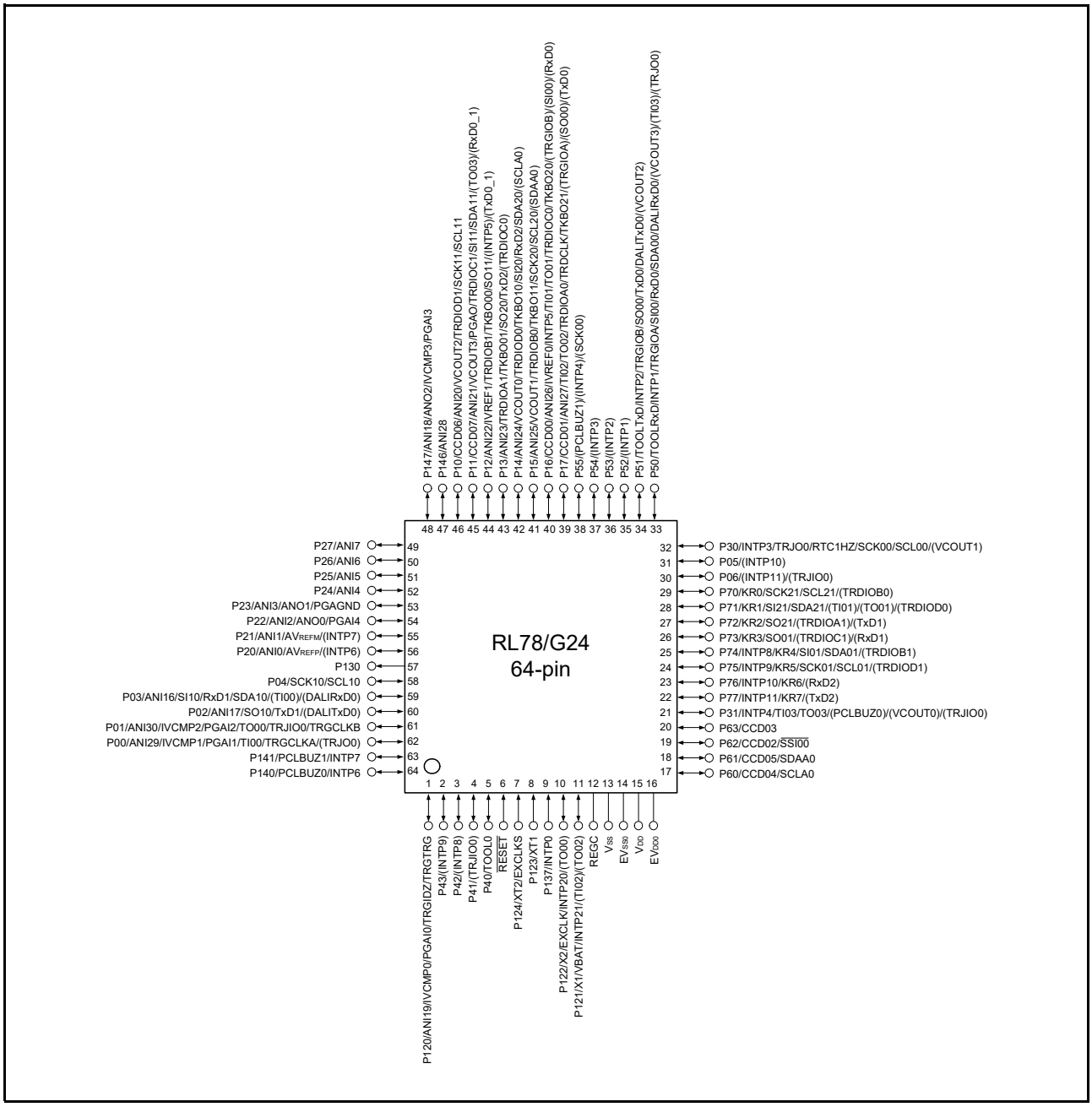
Pin Number	I/O			Analog Circuits				HMIs		Timers						Communications Interfaces		
52LQFP	Digital port	Controlled current drive port	Power supply, system clock, and debugging	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Programmable gain amplifier (PGA)	Interrupt (INTP)	Key interrupt (KR)	Timer array unit (TAU)	Timer RJ (TRJ)	Timer RD2 (TRD2)	Timer RG2 (TRG2)	16-bit timers KB30, KB31, and KB32 (TMKB3)	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Digital addressable lighting interface (DALI)
1	P140	—	PCLBUZ0	—	—	—	—	INTP6	—	—	—	—	—	—	—	—	—	—
2	P120	—	—	ANI19	—	IVCMP0	PGAI0	—	—	—	—	—	TRGIDZ/ TRGTRG	—	—	—	—	—
3	P41	—	—	—	—	—	—	—	—	—	(TRJIO0)	—	—	—	—	—	—	—
4	P40	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
5	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
6	P124	—	XT2/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
7	P123	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
8	P137	—	—	—	—	—	—	INTP0	—	—	—	—	—	—	—	—	—	—
9	P122	—	X2/EXCLK	—	—	—	—	INTP20	—	(TO00)	—	—	—	—	—	—	—	—
10	P121	—	X1/VBAT	—	—	—	—	INTP21	—	(TI02)/ (TO02)	—	—	—	—	—	—	—	—
11	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
12	—	—	V_{SS}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
13	—	—	V_{DD}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
14	P60	CCD04	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SCLA0	—
15	P61	CCD05	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDAA0	—
16	P62	CCD02	—	—	—	—	—	—	—	—	—	—	—	—	—	$\overline{\text{SSI00}}$	—	—
17	P63	CCD03	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
18	P31	—	(PCLBUZ0)	—	—	(VCOUT0)	—	INTP4	—	TI03/ TO03	(TRJIO0)	—	—	—	—	—	—	—
19	P77	—	—	—	—	—	—	INTP11	KR7	—	—	—	—	—	—	(TxD2)	—	—
20	P76	—	—	—	—	—	—	INTP10	KR6	—	—	—	—	—	—	(RxD2)	—	—
21	P75	—	—	—	—	—	—	INTP9	KR5	—	—	(TRDIOD1)	—	—	—	SCK01/ SCL01	—	—
22	P74	—	—	—	—	—	—	INTP8	KR4	—	—	(TRDIOB1)	—	—	—	SI01/ SDA01	—	—
23	P73	—	—	—	—	—	—	—	KR3	—	—	(TRDIOC1)	—	—	—	SO01/ (RxD1)	—	—
24	P72	—	—	—	—	—	—	—	KR2	—	—	(TRDIOA1)	—	—	—	SO21/ (TxD1)	—	—
25	P71	—	—	—	—	—	—	—	KR1	(TI01)/ (TO01)	—	(TRDIOD0)	—	—	—	SI21/ SDA21	—	—
26	P70	—	—	—	—	—	—	—	KR0	—	—	(TRDIOB0)	—	—	—	SCK21/ SCL21	—	—
27	P30	—	—	—	—	(VCOUT1)	—	INTP3	—	—	TRJO0	—	—	—	RTC1HZ	SCK00/ SCL00	—	—
28	P50	—	TOOLRxD	—	—	(VCOUT3)	—	INTP1	—	(TI03)	(TRJO0)	—	TRGIOA	—	—	SI00/ RxD0/ SDA00	—	DALIRxD0
29	P51	—	TOOLTxD	—	—	(VCOUT2)	—	INTP2	—	—	—	—	TRGIOB	—	—	SO00/ TxD0	—	DALITxD0

Table 1 - 10 Multiplexed Pin Functions of the 52-pin Products (2/2)

Pin Number	I/O		Power supply, system clock, and debugging	Analog Circuits				HMIs		Timers						Communications Interfaces		
52LQFP	Digital port	Controlled current drive port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Programmable gain amplifier (PGA)	Interrupt (INTP)	Key interrupt (KR)	Timer array unit (TAU)	Timer RJ (TRJ)	Timer RD2 (TRD2)	Timer RG2 (TRG2)	16-bit timers KB30, KB31, and KB32 (TMKB3)	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Digital addressable lighting interface (DALI)
30	P17	CCD01	—	ANI27	—	—	—	—	—	TI02/ TO02	—	TRDIOA0/ TRDCLK	(TRGIOA)	TKBO21	—	(TxD0)	—	—
31	P16	CCD00	—	ANI26	—	IVREF0	—	INTP5	—	TI01/ TO01	—	TRDIOC0	(TRGIOB)	TKBO20	—	(RxD0)	—	—
32	P15	—	PCLBUZ1	ANI25	—	VCOUT1	—	—	—	—	—	TRDIOB0	—	TKBO11	—	SCK20/ SCL20	(SDAA0)	—
33	P14	—	—	ANI24	—	VCOUT0	—	—	—	—	—	TRDIOD0	—	TKBO10	—	SI20/ RxD2/ SDA20	(SCLA0)	—
34	P13	—	—	ANI23	—	—	—	—	—	—	—	TRDIOA1/ (TRDIOC0)	—	TKBO01	—	SO20/ TxD2	—	—
35	P12	—	—	ANI22	—	IVREF1	—	—	—	—	—	TRDIOB1	—	TKBO00	—	SO11/ (TxD0_1)	—	—
36	P11	CCD07	—	ANI21	—	VCOUT3	PGAO	—	—	(TO03)	—	TRDIOC1	—	—	—	SI11/ SDA11/ (RxD0_1)	—	—
37	P10	CCD06	—	ANI20	—	VCOUT2	—	—	—	—	—	TRDIOD1	—	—	—	SCK11/ SCL11	—	—
38	P146	—	—	ANI28	—	—	—	—	—	—	—	—	—	—	—	—	—	—
39	P147	—	—	ANI18	ANO2	IVCMP3	PGAI3	—	—	—	—	—	—	—	—	—	—	—
40	P27	—	—	ANI7	—	—	—	—	—	—	—	—	—	—	—	—	—	—
41	P26	—	—	ANI6	—	—	—	—	—	—	—	—	—	—	—	—	—	—
42	P25	—	—	ANI5	—	—	—	—	—	—	—	—	—	—	—	—	—	—
43	P24	—	—	ANI4	—	—	—	—	—	—	—	—	—	—	—	—	—	—
44	P23	—	—	ANI3	ANO1		PGAGND	—	—	—	—	—	—	—	—	—	—	—
45	P22	—	—	ANI2	ANO0		PGAI4	—	—	—	—	—	—	—	—	—	—	—
46	P21	—	—	ANI1/ AVREFM	—	—	—	(INTP7)	—	—	—	—	—	—	—	—	—	—
47	P20	—	—	ANI0/ AVREFP	—	—	—	(INTP6)	—	—	—	—	—	—	—	—	—	—
48	P130	—	—		—	—	—	—	—	—	—	—	—	—	—	—	—	—
49	P03	—	—	ANI16	—	—	—	—	—	(TI00)	—	—	—	—	—	RxD1	—	(DALIRxD0)
50	P02	—	—	ANI17	—	—	—	—	—	—	—	—	—	—	—	TxD1	—	(DALITxD0)
51	P01	—	—	ANI30	—	IVCMP2	PGAI2	—	—	TO00	TRJIO0	—	TRGCLKB	—	—	—	—	—
52	P00	—	—	ANI29	—	IVCMP1	PGAI1	—	—	TI00	(TRJO0)	—	TRGCLKA	—	—	—	—	—

1.3.10 64-pin products

- 64-pin plastic LQFP (12 × 12 mm, 0.65-mm pitch)
- 64-pin plastic LFQFP (10 × 10 mm, 0.50-mm pitch)



Caution 1. Connect the EVSS0 pin to the same ground as the VSS pin.

Caution 2. Make sure that the voltage on the VDD pin is no less than that on the EVDD0 pin.

Caution 3. Connect the REGC pin to VSS via a capacitor (0.47 to 1 μF).

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

Remark 2. When using the microcontroller for an application where the noise generated inside the microcontroller must be reduced, it is recommended to supply separate powers to the VDD and EVDD0 pins and connect the VSS and EVSS0 pins to separate ground lines.

Remark 3. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register x (PIORx). Refer to **Figure 7 - 8 Format of Peripheral I/O Redirection Register 0 (PIOR0)** to **Figure 7 - 11 Format of Peripheral I/O Redirection Register 3 (PIOR3)**.

Table 1 - 11 Multiplexed Pin Functions of the 64-pin Products (1/3)

Pin Number	I/O			Analog Circuits				HMIs		Timers						Communications Interfaces		
64LQFP, 64LFQFP	Digital port	Controlled current drive port	Power supply, system clock, and debugging	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Programmable gain amplifier (PGA)	Interrupt (INTP)	Key interrupt (KR)	Timer array unit (TAU)	Timer RJ (TRJ)	Timer RD2 (TRD2)	Timer RG2 (TRG2)	16-bit timers KB30, KB31, and KB32 (TMKB3)	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Digital addressable lighting interface (DALI)
1	P120	—	—	ANI19	—	IVCMP0	PGAI0	—	—	—	—	—	TRGIDZ/ TRGTRG	—	—	—	—	—
2	P43	—	—	—	—	—	—	(INTP9)	—	—	—	—	—	—	—	—	—	—
3	P42	—	—	—	—	—	—	(INTP8)	—	—	—	—	—	—	—	—	—	—
4	P41	—	—	—	—	—	—	—	—	—	(TRJIO0)	—	—	—	—	—	—	—
5	P40	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
6	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
7	P124	—	XT2/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
8	P123	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
9	P137	—	—	—	—	—	—	INTP0	—	—	—	—	—	—	—	—	—	—
10	P122	—	X2/EXCLK	—	—	—	—	INTP20	—	(TO00)	—	—	—	—	—	—	—	—
11	P121	—	X1/VBAT	—	—	—	—	INTP21	—	(TI02)/ (TO02)	—	—	—	—	—	—	—	—
12	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
13	—	—	V_{SS}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
14	—	—	EV_{SS0}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
15	—	—	V_{DD}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
16	—	—	EV_{DD0}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
17	P60	CCD04	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SCLA0	—
18	P61	CCD05	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDAA0	—
19	P62	CCD02	—	—	—	—	—	—	—	—	—	—	—	—	—	$\overline{\text{SSI00}}$	—	—
20	P63	CCD03	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
21	P31	—	(PCLBUZ0)	—	—	(VCOUT0)	—	INTP4	—	TI03/ TO03	(TRJIO0)	—	—	—	—	—	—	—
22	P77	—	—	—	—	—	—	INTP11	KR7	—	—	—	—	—	—	(TxD2)	—	—
23	P76	—	—	—	—	—	—	INTP10	KR6	—	—	—	—	—	—	(RxD2)	—	—
24	P75	—	—	—	—	—	—	INTP9	KR5	—	—	(TRDIOD1)	—	—	—	SCK01/ SCL01	—	—
25	P74	—	—	—	—	—	—	INTP8	KR4	—	—	(TRDIOB1)	—	—	—	SI01/ SDA01	—	—
26	P73	—	—	—	—	—	—	—	KR3	—	—	(TRDIOC1)	—	—	—	SO01/ (RxD1)	—	—
27	P72	—	—	—	—	—	—	—	KR2	—	—	(TRDIOA1)	—	—	—	SO21/ (TxD1)	—	—
28	P71	—	—	—	—	—	—	—	KR1	(TI01)/ (TO01)	—	(TRDIOD0)	—	—	—	SI21/ SDA21	—	—
29	P70	—	—	—	—	—	—	—	KR0	—	—	(TRDIOB0)	—	—	—	SCK21/ SCL21	—	—
30	P06	—	—	—	—	—	—	(INTP11)	—	—	(TRJIO0)	—	—	—	—	—	—	—
31	P05	—	—	—	—	—	—	(INTP10)	—	—	—	—	—	—	—	—	—	—

Table 1 - 11 Multiplexed Pin Functions of the 64-pin Products (2/3)

Pin Number	I/O			Analog Circuits				HMIs		Timers						Communications Interfaces		
64LQFP, 64LFQFP	Digital port	Controlled current drive port	Power supply, system clock, and debugging	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Programmable gain amplifier (PGA)	Interrupt (INTP)	Key interrupt (KR)	Timer array unit (TAU)	Timer RJ (TRJ)	Timer RD2 (TRD2)	Timer RG2 (TRG2)	16-bit timers KB30, KB31, and KB32 (TMKB3)	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Digital addressable lighting interface (DALI)
32	P30	—	—	—	—	(VCOUT1)	—	INTP3	—	—	TRJO0	—	—	—	RTC1HZ	SCK00/ SCL00	—	—
33	P50	—	TOOLRxD	—	—	(VCOUT3)	—	INTP1	—	(TI03)	(TRJO0)	—	TRGIOA	—	—	SI00/ RxD0/ SDA00	—	DALIRxD0
34	P51	—	TOOLTxD	—	—	(VCOUT2)	—	INTP2	—	—	—	—	TRGIOB	—	—	SO00/ TxD0	—	DALITxD0
35	P52	—	—	—	—	—	—	(INTP1)	—	—	—	—	—	—	—	—	—	—
36	P53	—	—	—	—	—	—	(INTP2)	—	—	—	—	—	—	—	—	—	—
37	P54	—	—	—	—	—	—	(INTP3)	—	—	—	—	—	—	—	—	—	—
38	P55	—	(PCLBUZ1)	—	—	—	—	(INTP4)	—	—	—	—	—	—	—	(SCK00)	—	—
39	P17	CCD01	—	ANI27	—	—	—	—	—	TI02/ TO02	—	TRDIOA0/ TRDCLK	(TRGIOA)	TKBO21	—	(SO00)/ (TxD0)	—	—
40	P16	CCD00	—	ANI26	—	IVREF0	—	INTP5	—	TI01/ TO01	—	TRDIOC0	(TRGIOB)	TKBO20	—	(SI00)/ (RxD0)	—	—
41	P15	—	—	ANI25	—	VCOUT1	—	—	—	—	—	TRDIOB0	—	TKBO11	—	SCK20/ SCL20	(SDAA0)	—
42	P14	—	—	ANI24	—	VCOUT0	—	—	—	—	—	TRDIOD0	—	TKBO10	—	SI20/ RxD2/ SDA20	(SCLA0)	—
43	P13	—	—	ANI23	—	—	—	—	—	—	—	TRDIOA1/ (TRDIOC0)	—	TKBO01	—	SO20/ TxD2	—	—
44	P12	—	—	ANI22	—	IVREF1	—	(INTP5)	—	—	—	TRDIOB1	—	TKBO00	—	SO11/ (TxD0_ 1)	—	—
45	P11	CCD07	—	ANI21	—	VCOUT3	PGAO	—	—	(TO03)	—	TRDIOC1	—	—	—	SI11/ SDA11/ (RxD0_ 1)	—	—
46	P10	CCD06	—	ANI20	—	VCOUT2	—	—	—	—	—	TRDIOD1	—	—	—	SCK11/ SCL11	—	—
47	P146	—	—	ANI28	—	—	—	—	—	—	—	—	—	—	—	—	—	—
48	P147	—	—	ANI18	ANO2	IVCMP3	PGAI3	—	—	—	—	—	—	—	—	—	—	—
49	P27	—	—	ANI7	—	—	—	—	—	—	—	—	—	—	—	—	—	—
50	P26	—	—	ANI6	—	—	—	—	—	—	—	—	—	—	—	—	—	—
51	P25	—	—	ANI5	—	—	—	—	—	—	—	—	—	—	—	—	—	—
52	P24	—	—	ANI4	—	—	—	—	—	—	—	—	—	—	—	—	—	—
53	P23	—	—	ANI3	ANO1	—	PGAGND	—	—	—	—	—	—	—	—	—	—	—
54	P22	—	—	ANI2	ANO0	—	PGAI4	—	—	—	—	—	—	—	—	—	—	—
55	P21	—	—	ANI1/ AV_{REFM}	—	—	—	(INTP7)	—	—	—	—	—	—	—	—	—	—
56	P20	—	—	ANI0/ AV_{REFP}	—	—	—	(INTP6)	—	—	—	—	—	—	—	—	—	—
57	P130	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
58	P04	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SCK10/ SCL10	—	—

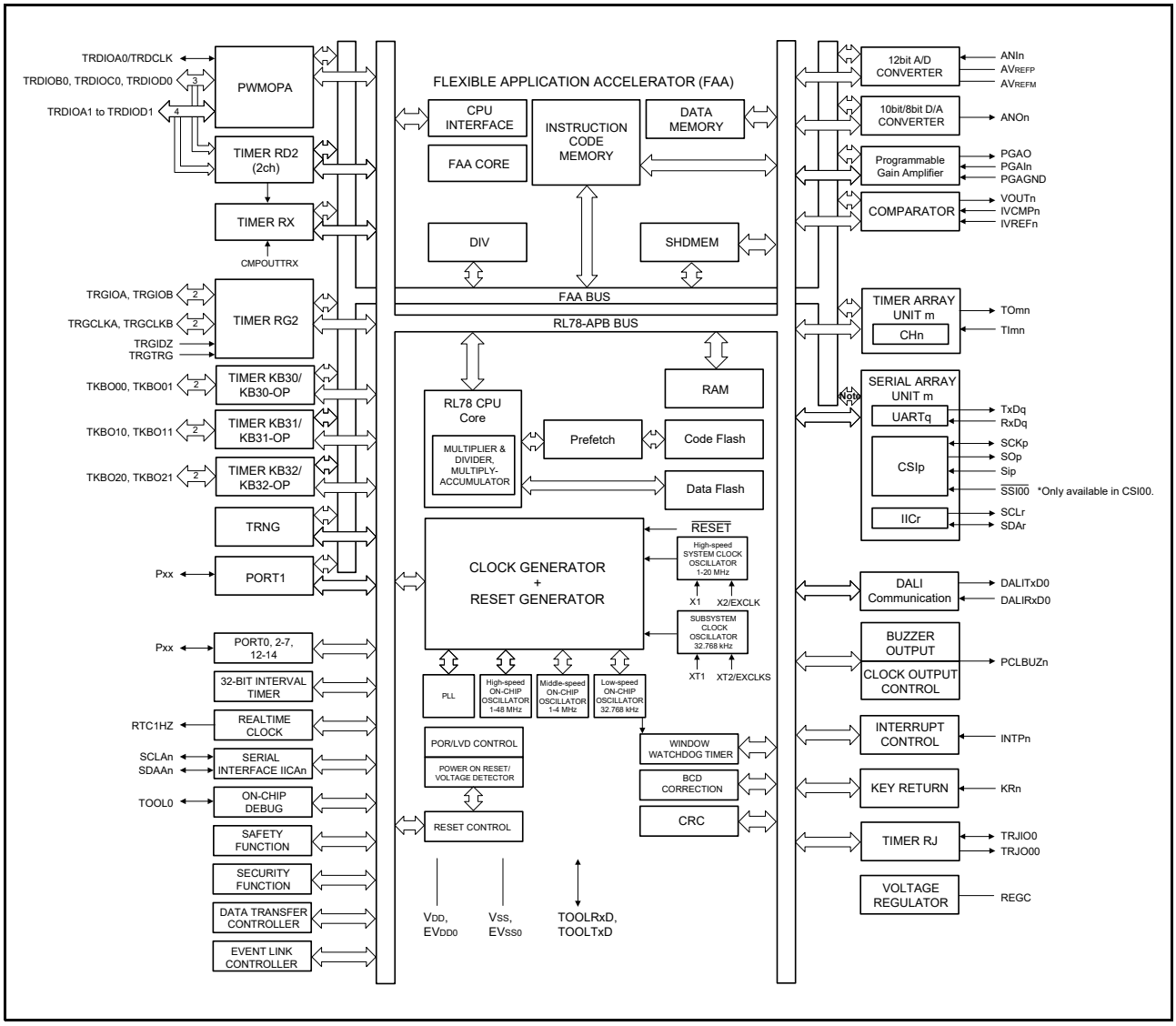
Table 1 - 11 Multiplexed Pin Functions of the 64-pin Products (3/3)

Pin Number	I/O		Power supply, system clock, and debugging	Analog Circuits				HMIs		Timers						Communications Interfaces		
64LQFP, 64LFQFP	Digital port	Controlled current drive port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Programmable gain amplifier (PGA)	Interrupt (INTP)	Key interrupt (KR)	Timer array unit (TAU)	Timer RJ (TRJ)	Timer RD2 (TRD2)	Timer RG2 (TRG2)	16-bit timers KB30, KB31, and KB32 (TMKB3)	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Digital addressable lighting interface (DALI)
59	P03	—	—	ANI16	—	—	—	—	—	(TI00)	—	—	—	—	—	SI10/ RxD1/ SDA10	—	(DALIRxD0)
60	P02	—	—	ANI17	—	—	—	—	—	—	—	—	—	—	—	SO10/ TxD1	—	(DALITxD0)
61	P01	—	—	ANI30	—	IVCMP2	PGAI2	—	—	TO00	TRJIO0	—	TRGCLKB	—	—	—	—	—
62	P00	—	—	ANI29	—	IVCMP1	PGAI1	—	—	TI00	(TRJO0)	—	TRGCLKA	—	—	—	—	—
63	P141	—	PCLBUZ1	—	—	—	—	INTP7	—	—	—	—	—	—	—	—	—	—
64	P140	—	PCLBUZ0	—	—	—	—	INTP6	—	—	—	—	—	—	—	—	—	—

1.4 Pin Identification

ANI0 to ANI7, ANI16 to ANI30	: Analog input
ANO0 to ANO2	: Analog output
AV_{REFM}	: Analog reference voltage minus
AV_{REFP}	: Analog reference voltage plus
CCD00 to CCD07	: Controlled current drive output
DALIRxD0	: DALI receive data
DALITxD0	: DALI transmit data
EV_{DD0}	: Power supply for port
EV_{SS0}	: Ground for port
EXCLK	: External clock input (main system clock)
EXCLKS	: External clock input (subsystem clock)
INTP0 to INTP11, INTP20, INTP21	: Interrupt request from peripheral
IVCMP0 to IVCMP3	: Comparator input
IVREF0, IVREF1	: Comparator reference input
KR0 to KR7	: Key return
P00 to P06	: Port 0
P10 to P17	: Port 1
P20 to P27	: Port 2
P30, P31	: Port 3
P40 to P43	: Port 4
P50 to P55	: Port 5
P60 to P63	: Port 6
P70 to P77	: Port 7
P120 to P124	: Port 12
P130, P137	: Port 13
P140, P141, P146, P147	: Port 14
PCLBUZ0, PCLBUZ1	: Programmable clock output/buzzer output
PGAGND	: PGA ground
PGAI0 to PGAI4	: PGA input
PGAO	: PGA output
REGC	: Regulator capacitance
$\overline{\text{RESET}}$	: Reset
RTC1HZ	: Realtime clock correction clock (1 Hz) output
RxD0 to RxD2	: Receive data
SCLA0, SCK00, SCK01, SCK10, SCK11, SCK20, SCK21	: Serial clock input/output
SCL00, SCL01, SCL10, SCL11, SCL20, SCL21	: Serial clock output
SDAA0, SDA00, SDA01, SDA10, SDA11, SDA20, SDA21	: Serial data input/output
SI00, SI01, SI10, SI11, SI20, SI21, SI30, SI31	: Serial data input
SO00, SO01, SO10, SO11, SO20, SO21	: Serial data output
$\overline{\text{SSI00}}$	: Serial interface chip select input
TI00 to TI03	: Timer input
TKBO00, TKBO01, TKBO10, TKBO11, TKBO20, TKBO21	: Timer KB3 output
TO00 to TO03	: Timer output
TOOL0	: Data input/output for tool
TOOLRxD, TOOLTxD	: Data input/output for external device
TRDCLK	: Timer RD2 external input clock
TRDIOA0, TRDIOB0, TRDIOC0, TRDIOD0, TRDIOA1, TRDIOB1, TRDIOC1, TRDIOD1	: Timer RD2 input/output
TRGIOA, TRGIOB	: Timer RG2 input/output
TRGCLKA, TRGCLKB	: Timer RG2 external input clock
TRGIDZ, TRGTRG	: Timer RG2 external trigger input
TRJIO0	: Timer RJ input/output
TRJO0	: Timer RJ output
TxD0 to TxD2	: Transmit data
VBAT	: Battery backup power supply
VCOUT0 to VCOUT3	: Comparator output
V_{DD}	: Power supply
V_{SS}	: Ground
X1, X2	: Crystal oscillator (main system clock)
XT1, XT2	: Crystal oscillator (subsystem clock)

1.5 Block Diagram



Note Serial array unit 0 is only connected to the FAA bus.

Caution **The key return function is only incorporated in the 40- to 128-pin products.**

Remark m: Unit number, n: Channel number, p: CSI number, q: UART number, r: Simplified I^2C number, xx: Port number

1.6 Outline of Functions

[20-, 24-, 25-, 30-, and 32-pin products]

Caution This outline describes the functions at the time when peripheral I/O redirection register x (PIORx) is set to 00H.

(1/3)

Item		20-pin	24-pin	25-pin	30-pin	32-pin
		R7F101G6x	R7F101G7x	R7F101G8x	R7F101GAx	R7F101GBx
Code flash memory		64 or 128 Kbytes				
Data flash memory		4 Kbytes				
RAM		12 Kbytes				
Address space		1 Mbyte				
CPU/ peripheral hardware clock frequency (f_{CLK})	Main system clock	HS (high-speed main) mode: 1 to 48 MHz (V_{DD} = 2.4 to 5.5 V) HS (high-speed main) mode: 1 to 32 MHz (V_{DD} = 1.8 to 5.5 V) HS (high-speed main) mode: 1 to 4 MHz**Note 1** (V_{DD} = 1.6 to 5.5 V) LS (low-speed main) mode: 1 to 24 MHz (V_{DD} = 1.8 to 5.5 V) LS (low-speed main) mode: 1 to 4 MHz**Note 1** (V_{DD} = 1.6 to 5.5 V) LP (low-power main) mode: 1 to 2 MHz**Note 2** (V_{DD} = 1.6 to 5.5 V)				
	Subsystem clock	SUB mode: 32.768 kHz (V_{DD} = 1.6 to 5.5 V)				
Main system clock	High-speed system clock (f_{MX})	1 to 20 MHz				
	High-speed on-chip oscillator clock (f_{IH})	1 MHz, 2 MHz, 3 MHz, 4 MHz, 6 MHz, 8 MHz, 12 MHz, 16 MHz, 24 MHz, 32 MHz, 48 MHz, 64 MHz				
	Middle-speed on-chip oscillator clock (f_{IM})	1 MHz, 2 MHz, 4 MHz				
	PLL clock	16 MHz, 32 MHz**Note 3** (V_{DD} = 1.8 to 5.5 V) 24 MHz, 48 MHz**Note 4** (V_{DD} = 2.4 to 5.5 V)				
Subsystem clock	Subsystem clock oscillator clock (f_{SX}, f_{SXR})	32.768 kHz (V_{DD} = 2.4 to 5.5 V)				
	Low-speed on-chip oscillator clock (f_{IL})	32.768 kHz (typ.)				
General-purpose registers		8 bits × 32 registers (8 bits × 8 registers × 4 banks)				
Minimum instruction execution time		0.03125 μs (at the 32-MHz operation with the high-speed on-chip oscillator clock (f_{IH})) 0.02083 μs (at the 48-MHz operation with the high-speed on-chip oscillator clock (f_{IH}))**Note 5** 0.03125 μs (PLL clock: f_{PLL} = 64 MHz, f_{IH} = 16 or 32 MHz**Note 3**) 0.02083 μs (PLL clock: f_{PLL} = 96 MHz, f_{IH} = 24 or 48 MHz**Note 4**)**Note 5**				
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.				

(2/3)

Item		20-pin R7F101G6x	24-pin R7F101G7x	25-pin R7F101G8x	30-pin R7F101GAx	32-pin R7F101GBx
FAA core		• Multiplication: 32-bit signed × 32-bit signed → 32-bit signed • Results of 64-bit multiplication can be right-shifted by a desired number of bits. • Addition: 32-bit signed + 32-bit signed → 32-bit signed (internally calculated with 33-bit precision) • Subtraction: 32-bit signed - 32-bit signed → 32-bit signed (internally calculated with 33-bit precision) • Limit operation: Operation parameter registers (33 bits × 4) in which upper and lower limits can be set. • Operation parameter registers (32 bits × 6) • Address pointer registers (12 bits × 6) • On-chip code RAM: 4 Kbytes • On-chip data RAM: 2 Kbytes • Multiple interrupts available • A 32-byte shared memory is included for sharing of data by the RL78 CPU and FAA core.				
I/O port	Total number of pins	16	20	21	26	28
	CMOS I/O	15 (N-ch open drain I/O [withstand voltage of VDD]: 7)	19 (N-ch open drain I/O [withstand voltage of VDD]: 8)		23 (N-ch open drain I/O [withstand voltage of VDD]: 11)	25 (N-ch open drain I/O [withstand voltage of VDD]: 11)
	CMOS input	1				
	CMOS output	—		1	—	
	N-ch open drain I/O (withstand voltage: 6 V)	—			2	
	Controlled current drive port	2	4		6	7
Timers	16-bit timers TAU, timer RJ, timer RD2, timer RX, timer RG2	9 channels in total: 4-channel timer array unit (TAU) 1-channel timer RJ 2-channel timer RD2 with PWMOPA 1-channel timer RG2 1-channel timer RX				
	16-bit timer Timer KB3	2 channels (PWM outputs: 4)	3 channels (PWM outputs: 6)			
	Watchdog timer	1 channel				
	Realtime clock (RTC)	1 channel				
	32-bit interval timer (TML32)	1 channel in 32-bit counter mode, 2 channels in 16-bit counter mode, 4 channels in 8-bit counter mode				
	Timer outputs	11 (PWM outputs: 10Note 6), 19 (PWM outputs: 11Note 6)Note 7	17 (PWM outputs: 14Note 6), 22 (PWM outputs: 14Note 6)Note 7			
	RTC output	—			1	
Clock output/buzzer output		1			2	
		• 3.91 kHz, 7.81 kHz, 15.63 kHz, 2 MHz, 4 MHz, 8 MHz, 16 MHz (Main system clock: fMAIN = 32 MHz) • 256 Hz, 512 Hz, 1.024 kHz, 2.048 kHz, 4.096 kHz, 8.192 kHz, 16.384 kHz, 32.768 kHz (Low-speed peripheral clock: fSXP = 32.768 kHz)				
8-/10-/12-bit resolution A/D converter		12 channels	13 channels	16 channels		
	3-channel simultaneous sampling	2 channels				

(3/3)

Item		20-pin	24-pin	25-pin	30-pin	32-pin
		R7F101G6x	R7F101G7x	R7F101G8x	R7F101GAx	R7F101GBx
8-/10-bit D/A converter		2 to 3 channels				
	DAC outputs (ANOx)	2	3			
Programmable gain amplifier (PGA)		1 channel				
Comparator module		3 channels	4 channels			
Serial interfaces		[20-pin products] • Simplified SPI (CSI): 1 channel, simplified I²C: 1 channel, UART: 1 channel • Simplified SPI (CSI): 1 channel, simplified I²C: 1 channel, UART: 1 channel [24- and 25-pin products] • UART (supporting LIN-bus): 1 channel • Simplified SPI (CSI): 1 channel, simplified I²C: 1 channel, UART: 1 channel • Simplified SPI (CSI): 1 channel, simplified I²C: 1 channel, UART: 1 channel [30- and 32-pin products] • Simplified SPI (CSI): 1 channel, simplified I²C: 1 channel, UART (UART supporting LIN-bus): 1 channel • Simplified SPI (CSI): 1 channel, simplified I²C: 1 channel, UART: 1 channel • Simplified SPI (CSI): 1 channel, simplified I²C: 1 channel, UART: 1 channel				
	I²C bus	—			1	
	I²C (SM/PM) bus	—			1	
	DALI	—			1	
Data transfer controller (DTC)		42 sources	47 sources		52 sources	
Vectored interrupt sources	Internal	46	55			
	External	6	8		12	
Key interrupt		—				
Reset		• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-reset • Internal reset by voltage detectors (LVD0 and LVD1) • Internal reset by illegal instruction execution**Note 8** • Internal reset by RAM parity error • Internal reset by illegal-memory access				
Power-on-reset circuit		• Power-on reset: 1.51 V (typ.) • Power-down reset: 1.50 V (typ.)				
Voltage detector	LVD0	Detection voltage • Rising edge: 1.69 to 3.96 V (6 stages) Falling edge: 1.65 to 3.88 V (6 stages)				
	LVD1	Detection voltage • Rising edge: 1.67 to 4.16 V (18 stages) Falling edge: 1.63 to 4.08 V (18 stages)				
On-chip debugging		Available (tracing supported)				
Power supply voltage		V_{DD} = 1.6 to 5.5 V (2D: Consumer applications, 3C: Industrial applications) V_{DD} = 2.7 to 5.5 V (4C: Industrial applications)				
Operating ambient temperature		T_A = –40 to +85°C (2D: Consumer applications), T_A = –40 to +105°C (3C: Industrial applications), T_A = –40 to +125°C (4C: Industrial applications)				

Note 1. Overwrite the flash memory during operation at a frequency of no higher than 4 MHz.

Note 2. When the flash memory is to be overwritten, switch to HS (high-speed main) mode or LS (low-speed main) mode.

Note 3. Applicable when the PLL clock frequency is 64 MHz. Select f_{PLL}/2 (32 MHz) or f_{PLL}/4 (16 MHz) as the system clock.

Note 4. Applicable when the PLL clock frequency is 96 MHz. Select f_{PLL}/2 (32 MHz) or f_{PLL}/4 (16 MHz) as the system clock.

Note 5. This applies when the prefetch buffer is enabled. For details on the operation of the prefetch buffer, refer to **Section 8 Operation State Control**.

Note 6. The number of PWM outputs varies depending on the setting of channels in use (the number of masters and slaves). For details, see **10.9.3 Operation for the multiple PWM output function**.

Note 7. This applies when the setting of the PIOR0 bit is 1.

Note 8. In normal operation, executing the instruction code FFH triggers an internal reset, but this is not the case during emulation by the in-circuit emulator or on-chip debugging emulator.

[40-, 44-, 48-, 52-, and 64-pin products]

Caution This outline describes the functions at the time when peripheral I/O redirection register x (PIORx) is set to 00H.

(1/3)

Item		40-pin	44-pin	48-pin	52-pin	64-pin
		R7F101GEx	R7F101GFx	R7F101GGx	R7F101GJx	R7F101GLx
Code flash memory		64 or 128 Kbytes				
Data flash memory		4 Kbytes				
RAM		12 Kbytes				
Address space		1 Mbyte				
CPU/ peripheral hardware clock frequency (f_{CLK})	Main system clock	HS (high-speed main) mode: 1 to 48 MHz (V_{DD} = 2.4 to 5.5 V) HS (high-speed main) mode: 1 to 32 MHz (V_{DD} = 1.8 to 5.5 V) HS (high-speed main) mode: 1 to 4 MHz**Note 1** (V_{DD} = 1.6 to 5.5 V) LS (low-speed main) mode: 1 to 24 MHz (V_{DD} = 1.8 to 5.5 V) LS (low-speed main) mode: 1 to 4 MHz**Note 1** (V_{DD} = 1.6 to 5.5 V) LP (low-power main) mode: 1 to 2 MHz**Note 2** (V_{DD} = 1.6 to 5.5 V)				
	Subsystem clock	SUB mode: 32.768 kHz (V_{DD} = 1.6 to 5.5 V)				
Main system clock	High-speed system clock (f_{MX})	1 to 20 MHz				
	High-speed on-chip oscillator clock (f_{IH})	1 MHz, 2 MHz, 3 MHz, 4 MHz, 6 MHz, 8 MHz, 12 MHz, 16 MHz, 24 MHz, 32 MHz, 48 MHz, 64 MHz				
	Middle-speed on-chip oscillator clock (f_{IM})	1 MHz, 2 MHz, 4 MHz				
	PLL clock	16 MHz, 32 MHz**Note 3** (V_{DD} = 1.8 to 5.5 V) 24 MHz, 48 MHz**Note 4** (V_{DD} = 2.4 to 5.5 V)				
Subsystem clock	Subsystem clock oscillator clock (f_{SX}, f_{SXR})	32.768 kHz (V_{DD} = 1.6 to 5.5 V)				
	Low-speed on-chip oscillator clock (f_{IL})	32.768 kHz (typ.)				
General-purpose registers		8 bits × 32 registers (8 bits × 8 registers × 4 banks)				
Minimum instruction execution time		0.03125 μs (at the 32-MHz operation with the high-speed on-chip oscillator clock (f_{IH})) 0.02083 μs (at the 48-MHz operation with the high-speed on-chip oscillator clock (f_{IH}))**Note 5** 0.03125 μs (PLL clock: f_{PLL} = 64 MHz, f_{IH} = 16 or 32 MHz**Note 3**) 0.02083 μs (PLL clock: f_{PLL} = 96 MHz, f_{IH} = 24 or 48 MHz**Note 4**)**Note 5**				
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.				
FAA core		• Multiplication: 32-bit signed × 32-bit signed → 32-bit signed • Results of 64-bit multiplication can be right-shifted by a desired number of bits. • Addition: 32-bit signed + 32-bit signed → 32-bit signed (internally calculated with 33-bit precision) • Subtraction: 32-bit signed - 32-bit signed → 32-bit signed (internally calculated with 33-bit precision) • Limit operation: Operation parameter registers (33 bits × 4) in which upper and lower limits can be set. • Operation parameter registers (32 bits × 6) • Address pointer registers (12 bits × 6) • On-chip code RAM: 4 Kbytes • On-chip data RAM: 2 Kbytes • Multiple interrupts available • A 32-byte shared memory is included for sharing of data by the RL78 CPU and FAA core.				

(2/3)

Item		40-pin	44-pin	48-pin	52-pin	64-pin
		R7F101GEx	R7F101GFx	R7F101GGx	R7F101GJx	R7F101GLx
I/O port	Total number of pins	36	40	44	48	58
	CMOS I/O	31 (N-ch open drain I/O [withstand voltage of VDD]: 14)	35 (N-ch open drain I/O [withstand voltage of VDD]: 14)	38 (N-ch open drain I/O [withstand voltage of VDD]: 15)	42 (N-ch open drain I/O [withstand voltage of VDD]: 17)	52 (N-ch open drain I/O [withstand voltage of VDD]: 19)
	CMOS input	3				
	CMOS output	—		1		
	N-ch open drain I/O (withstand voltage: 6 V)	2				
	Controlled current drive port	7	8			
Timers	16-bit timers TAU, timer RJ, timer RD2, timer RX, timer RG2	9 channels in total: 4-channel timer array unit (TAU) 1-channel timer RJ 2-channel timer RD2 with PWMOPA 1-channel timer RG2 1-channel timer RX				
	16-bit timer Timer KB3	3 channels (PWM outputs: 6)				
	Watchdog timer	1 channel				
	Realtime clock (RTC)	1 channel				
	32-bit interval timer (TML32)	1 channel in 32-bit counter mode, 2 channels in 16-bit counter mode, 4 channels in 8-bit counter mode				
	Timer outputs	17 (PWM outputs: 14Note 6), 22 (PWM outputs: 14Note 6)Note 7				
	RTC output	1				
Clock output/buzzer output		2				
		• 3.91 kHz, 7.81 kHz, 15.63 kHz, 2 MHz, 4 MHz, 8 MHz, 16 MHz (Main system clock: fMAIN = 32 MHz) • 256 Hz, 512 Hz, 1.024 kHz, 2.048 kHz, 4.096 kHz, 8.192 kHz, 16.384 kHz, 32.768 kHz (Low-speed peripheral clock: fSXP = 32.768 kHz)				
8-/10-/12-bit resolution A/D converter		19 channels	21 channels		23 channels	
	3-channel simultaneous sampling	2 channels				
8-/10-bit D/A converter		2 to 3 channels				
	DAC outputs (ANOx)	3				
Programmable gain amplifier (PGA)		1 channel				
Comparator module		4 channels				

(3/3)

Item		40-pin	44-pin	48-pin	52-pin	64-pin
		R7F101GEx	R7F101GFx	R7F101GGx	R7F101GJx	R7F101GLx
Serial interfaces		[40- and 44-pin products] • Simplified SPI (CSI): 1 channel, simplified I²C: 1 channel, UART (UART supporting LIN-bus): 1 channel • Simplified SPI (CSI): 1 channel, simplified I²C: 1 channel, UART: 1 channel • Simplified SPI (CSI): 2 channels, simplified I²C: 2 channels, UART: 1 channel [48- and 52-pin products] • Simplified SPI (CSI): 2 channels, simplified I²C: 2 channels, UART (UART supporting LIN-bus): 1 channel • Simplified SPI (CSI): 1 channel, simplified I²C: 1 channel, UART: 1 channel • Simplified SPI (CSI): 2 channels, simplified I²C: 2 channels, UART: 1 channel [64-pin products] • Simplified SPI (CSI): 2 channels, simplified I²C: 2 channels, UART (UART supporting LIN-bus): 1 channel • Simplified SPI (CSI): 2 channels, simplified I²C: 2 channels, UART: 1 channel • Simplified SPI (CSI): 2 channels, simplified I²C: 2 channels, UART: 1 channel				
	I²C bus	1				
	I²C (SM/PM) bus	1				
	DALI	1				
Data transfer controller (DTC)		53 sources				
Vectored interrupt sources	Internal	55				
	External	13		15		
Key interrupt		4		6		8
Reset		• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-reset • Internal reset by voltage detectors (LVD0 and LVD1) • Internal reset by illegal instruction execution**Note 8** • Internal reset by RAM parity error • Internal reset by illegal-memory access				
Power-on-reset circuit		• Power-on reset: 1.51 V (typ.) • Power-down reset: 1.50 V (typ.)				
Voltage detector	LVD0	Detection voltage • Rising edge: 1.69 to 3.96 V (6 stages) • Falling edge: 1.65 to 3.88 V (6 stages)				
	LVD1	Detection voltage • Rising edge: 1.67 to 4.16 V (18 stages) • Falling edge: 1.63 to 4.08 V (18 stages)				
On-chip debugging		Available (tracing supported)				
Power supply voltage		V_{DD} = 1.6 to 5.5 V (2D: Consumer applications, 3C: Industrial applications) V_{DD} = 2.7 to 5.5 V (4C: Industrial applications)				
Operating ambient temperature		T_A = –40 to +85°C (2D: Consumer applications), T_A = –40 to +105°C (3C: Industrial applications), T_A = –40 to +125°C (4C: Industrial applications)				

Note 1. Overwrite the flash memory during operation at a frequency of no higher than 2 MHz.

Note 2. When the flash memory is to be overwritten, switch to HS (high-speed main) mode or LS (low-speed main) mode.

Note 3. Applicable when the PLL clock frequency is 64 MHz. Select f_{PLL}/2 (32 MHz) or f_{PLL}/4 (16 MHz) as the system clock.

Note 4. Applicable when the PLL clock frequency is 96 MHz. Select f_{PLL}/2 (32 MHz) or f_{PLL}/4 (16 MHz) as the system clock.

Note 5. This applies when the prefetch buffer is enabled. For details on the operation of the prefetch buffer, refer to **Section 8 Operation State Control**.

Note 6. The number of PWM outputs varies depending on the setting of channels in use (the number of masters and slaves). For details, see **10.9.3 Operation for the multiple PWM output function**.

Note 7. This applies when the setting of the PIOR0 bit is 1.

Note 8. In normal operation, executing the instruction code FFH triggers an internal reset, but this is not the case during emulation by the in-circuit emulator or on-chip debugging emulator.

Section 2 Pin Functions

2.1 Functions of Port Pins

Pin I/O buffer power supplies depend on the product. The relationship between these power supplies and the pins is shown below.

Table 2 - 1 Pin I/O Buffer Power Supplies

1. 20-, 24-, 25-, 30-, 32-, 40-, 44-, 48-, and 52-pin products

Power Supply	Corresponding Pins
V_{DD}	All pins

2. 64-pin products

Power Supply	Corresponding Pins
EV_{DD0}	• Port pins other than P20 to P27, P121 to P124, and P137
V_{DD}	• P20 to P27, P121 to P124, and P137 • $\overline{RESET}$, REGC

2.1.1 20-pin products

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-18-1	I/O	Analog function	P00/ANI29/IVCMP1/ PGAI1/TI00/TRGCLKA/ (TRJO0)/TxD1/(DALITxD0)	Port 0. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P01 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P00 as an output. P00 and P01 can be set for the analog pin functions**Note**.
P01	8-18-2			P01/ANI30/IVCMP2/PGAI2/ TO00/TRGCLKB/TRJIO0/ RxD1/(DALIRxD0)/(TI00)	
P10	8-41-1	I/O	Analog function	P10/ANI20/CCD06/SCK11/ SCL11/TRDIOD1/VCOUT2/ (TxD2)	Port 1. 6-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, P14, and P15 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P10 to P15 as outputs. P10 and P11 can be set as controlled current drive port pins. P10 to P14 can be set for the analog pin functions**Note**.
P11	8-42-1			P11/ANI21/PGAO/CCD07/ SI11/SDA11/TRDIOC1/ (TO03)/(RxD2)/TOOLRxD/ (TI03)	
P12	7-8-2			P12/ANI22/SO11/ TRDIOB1/TKBO00/ IVREF1/TOOLTxD	
P13	7-8-1			P13/ANI23/TxD2/SO20/ TRDIOA1/TKBO01/ (TRDIOC0)	
P14	8-3-8			P14/RxD2/SI20/SDA20/ TRDIOD0/TKBO10/ VCOUT0/(SCLA0)/ANI24	
P15	8-3-8			P15/SCK20/SCL20/ TRDIOB0/TKBO11/ VCOUT1/(SDAA0)/ PCLBUZ1/(TI01)/(TO01)	
P20	4-3-5	I/O	Analog function	P20/ANI0/AVREFP/(INTP6)	Port 2. 4-bit I/O port. Input or output can be specified in 1-bit units. P20 to P23 can be set for the analog pin functions**Note**.
P21	4-3-5			P21/ANI1/AVREFM/(INTP7)	
P22	4-8-2			P22/ANI2/ANO0/PGAI4	
P23	4-8-2			P23/ANI3/ANO1/PGAGND	
P40	7-1-3	I/O	Input port	P40/TOOL0	Port 4. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P120	7-3-3	I/O	Analog function	P120/ANI19/IVCMP0/ PGAI0/TRGIDZ/TRGTRG/ (TI02)/(TO02)	Port 12. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P120 can be set for the analog pin function**Note**.
P122	7-2-1			P122/EXCLK/INTP20/ (TO00)/EXCLKS	
P137	2-1-3	Input	Input port	P137/INTP0	Port 13. 1-bit input-only port.
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

(Note and Remark are listed on the next page.)

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register xx (PMCAxx).

Remark Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register x (PIORx). For details, see **Figure 7 - 8 Format of Peripheral I/O Redirection Register 0 (PIOR0)** to **Figure 7 - 11 Format of Peripheral I/O Redirection Register 3 (PIOR3)**.

2.1.2 24-pin products

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-18-1	I/O	Analog function	P00/ANI29/IVCMP1/ PGAI1/TI00/TRGCLKA/ (TRJO0)/TxD1/ (DALITxD0)	Port 0. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P01 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P00 as an output. P00 and P01 can be set for the analog pin functions**Note**.
P01	8-18-2			P01/ANI30/IVCMP2/ PGAI2/TO00/TRGCLKB/ TRJIO0/RxD1/ (DALIRxD0)/(TI00)	
P10	8-41-1	I/O	Analog function	P10/ANI20/CCD06/SCK11/ SCL11/TRDIOD1/VCOUT2/ (TxD2)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P14 to P17 as inputs. N-ch open-drain output (withstand voltage of EVDD) can be set for P10 to P15 and P17 as outputs. P10, P11, P16, and P17 can be set as controlled current drive port pins. P10 to P13 and P16 can be set for the analog pin functions**Note**.
P11	8-42-1			P11/ANI21/PGAO/CCD07/ SI11/SDA11/TRDIOC1/ (TO03)/(RxD2)/TOOLRxD/ (TI03)/VCOUT3	
P12	7-8-2			P12/ANI22/SO11/ TRDIOB1/TKBO00/ IVREF1/TOOLTxD	
P13	7-8-1			P13/ANI23/TxD2/SO20/ TRDIOA1/TKBO01/ (TRDIOC0)	
P14	8-3-8			P14/RxD2/SI20/SDA20/ TRDIOD0/TKBO10/ VCOUT0/(SCLA0)	
P15	8-3-8			P15/SCK20/SCL20/ TRDIOB0/TKBO11/ VCOUT1/(SDAA0)/ PCLBUZ1/(TI01)/(TO01)	
p16	8-41-2			P16/ANI26/CCD00/TI01/ TO01/INTP5/TRDIOC0/ TKBO20/IVREF0/(RxD0)/ (TRGIOB)/(TRDIOA1)	
p17	8-41-1			P17/CCD01/TI02/TO02/ TRDIOA0/TRDCLK/ TKBO21/(TxD0)/(TRGIOA)/ (TRDIOD0)	
P20	4-3-5	I/O	Analog function	P20/ANI0/AVREFP/(INTP6)	Port 2. 4-bit I/O port. Input or output can be specified in 1-bit units. P20 to P23 can be set for the analog pin functions**Note**.
P21	4-3-5			P21/ANI1/AVREFM/(INTP7)	
P22	4-8-2			P22/ANI2/ANO0/PGAI4	
P23	4-8-2			P23/ANI3/ANO1/PGAGND	
P40	7-1-3	I/O	Input port	P40/TOOL0	Port 4. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P120	7-3-3	I/O	Analog function	P120/ANI19/IVCMP0/ PGAI0/TRGIDZ/TRGTRG/ (TI02)/(TO02)	Port 12. 3-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P120 can be set for the analog pin function**Note**.
P121	7-2-1		Input port	P121/X1/INTP21/ XT1	
P122	7-2-1			P122/X2/EXCLK/INTP20/ (TO00)/XT2/EXCLKS	
P137	2-1-3	Input	Input port	P137/INTP0	Port 13. 1-bit input-only port.
P147	7-18-2	I/O	Analog function	P147/ANI18/ANO2/ IVCMP3/PGAI3	Port 14. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function**Note**.
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to V_{DD}, either directly or via a resistor.

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register xx (PMCAxx).

Remark Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register x (PIORx). For details, see **Figure 7 - 8 Format of Peripheral I/O Redirection Register 0 (PIOR0)** to **Figure 7 - 11 Format of Peripheral I/O Redirection Register 3 (PIOR3)**.

2.1.3 25-pin products

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-18-1	I/O	Analog function	P00/ANI29/IVCMP1/PGAI1/TI00/TRGCLKA/(TRJO0)/TxD1/(DALITxD0)	Port 0. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P01 as an input. N-ch open-drain output (withstand voltage of V_{DD}) can be set for P00 as an output. P00 and P01 can be set for the analog pin functions**Note**.
P01	8-18-2			P01/ANI30/IVCMP2/PGAI2/TO00/TRGCLKB/TRJIO0/RxD1/(DALIRxD0)/(TI00)	
P10	8-41-1	I/O	Analog function	P10/ANI20/CCD06/SCK11/SCL11/TRDIOD1/VCOUT2/(TxD2)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P14 to P17 as inputs. N-ch open-drain output (withstand voltage of EV_{DD}) can be set for P10 to P15 and P17 as outputs. P10, P11, P16, and P17 can be set as controlled current drive port pins. P10 to P17 can be set for the analog pin functions**Note**.
P11	8-42-1			P11/ANI21/PGAO/CCD07/SI11/SDA11/TRDIOC1/(TO03)/(RxD2)/TOOLRxD/(TI03)/VCOUT3	
P12	7-8-2			P12/ANI22/SO11/TRDIOB1/TKBO00/IVREF1/TOOLTxD	
P13	7-8-1			P13/ANI23/TxD2/SO20/TRDIOA1/TKBO01/(TRDIOC0)	
P14	8-3-8			P14/RxD2/SI20/SDA20/TRDIOD0/TKBO10/VCOUT0/(SCLA0)/ANI24	
P15	8-3-8			P15/SCK20/SCL20/TRDIOB0/TKBO11/VCOUT1/(SDAA0)/ANI25/PCLBUZ1/(TI01)/(TO01)	
p16	8-41-2			P16/ANI26/CCD00/TI01/TO01/INTP5/TRDIOC0/TKBO20/IVREF0/(RxD0)/(TRGIOB)/(TRDIOA1)	
p17	8-41-1			P17/CCD01/TI02/TO02/TRDIOA0/TRDCLK/TKBO21/(TxD0)/(TRGIOA)/ANI27/(TRDIOD0)	
P20	4-3-5	I/O	Analog function	P20/ANI0/AV_{REFP}/(INTP6)	Port 2. 4-bit I/O port. Input or output can be specified in 1-bit units. P20 to P23 can be set for the analog pin functions**Note**.
P21	4-3-5			P21/ANI1/AV_{REFM}/(INTP7)	
P22	4-8-2			P22/ANI2/ANO0/PGAI4	
P23	4-8-2			P23/ANI3/ANO1/PGAGND	
P40	7-1-3	I/O	Input port	P40/TOOL0	Port 4. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P120	7-3-3	I/O	Analog function	P120/ANI19/IVCMP0/ PGAI0/TRGIDZ/TRGTRG/ (TI02)/(TO02)	Port 12. 3-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P120 can be set for the analog pin function**Note**.
P121	7-2-1		Input port	P121/X1/INTP21/ XT1	
P122	7-2-1			P122/X2/EXCLK/INTP20/ (TO00)/XT2/EXCLKS	
p130	1-1-1	Output	Output port	P130	Port 13. 1-bit output-only port and 1-bit input-only port.
P137	2-1-3	Input	Input port	P137/INTP0	
P147	7-18-2	I/O	Analog function	P147/ANI18/ANO2/ IVCMP3/PGAI3	Port 14. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function**Note**.
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to V_{DD}, either directly or via a resistor.

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register xx (PMCAxx).

Remark Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register x (PIORx). For details, see **Figure 7 - 8 Format of Peripheral I/O Redirection Register 0 (PIOR0)** to **Figure 7 - 11 Format of Peripheral I/O Redirection Register 3 (PIOR3)**.

2.1.4 30-pin products

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-18-1	I/O	Analog function	P00/ANI29/IVCMP1/PGAI1/TI00/TRGCLKA/(TRJO0)/TxD1/(DALITxD0)	Port 0. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P01 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P00 as an output. P00 and P01 can be set for the analog pin functions**Note**.
P01	8-18-2			P01/ANI30/IVCMP2/PGAI2/TO00/TRGCLKB/TRJIO0/RxD1/(DALIRxD0)/(TI00)	
P10	8-41-1	I/O	Analog function	P10/ANI20/CCD06/SCK11/SCL11/TRDIOD1/VCOUT2/(TxD2)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P14 to P17 as inputs. N-ch open-drain output (withstand voltage of EVDD) can be set for P10 to P15 and P17 as outputs. P10, P11, P16, and P17 can be set as controlled current drive port pins. P10 to P17 can be set for the analog pin functions**Note**.
P11	8-42-1			P11/ANI21/PGAO/CCD07/SI11/SDA11/TRDIOC1/(TO03)/(RxD2)/VCOUT3	
P12	7-8-2			P12/ANI22/SO11/TRDIOB1/TKBO00/IVREF1	
P13	7-8-1			P13/ANI23/TxD2/SO20/TRDIOA1/TKBO01/(TRDIOC0)	
P14	8-3-8			P14/RxD2/SI20/SDA20/TRDIOD0/TKBO10/VCOUT0/(SCLA0)/ANI24	
P15	8-3-8			P15/SCK20/SCL20/TRDIOB0/TKBO11/VCOUT1/(SDAA0)/ANI25/PCLBUZ1/(TI01)/(TO01)	
P16	8-41-2			P16/ANI26/CCD00/TI01/TO01/INTP5/TRDIOC0/TKBO20/IVREF0/(RxD0)/(TRGIOB)/(TRDIOA1)	
P17	8-41-1			P17/CCD01/TI02/TO02/TRDIOA0/TRDCLK/TKBO21/(TxD0)/(TRGIOA)/ANI27/(TRDIOD0)	
P20	4-3-5	I/O	Analog function	P20/ANI0/AVREFP/(INTP6)	Port 2. 4-bit I/O port. Input or output can be specified in 1-bit units. P20 to P23 can be set for the analog pin functions**Note**.
P21	4-3-5			P21/ANI1/AVREFM/(INTP7)	
P22	4-8-2			P22/ANI2/ANO0/PGAI4	
P23	4-8-2			P23/ANI3/ANO1/PGAGND	
P30	8-1-13	I/O	Input port	P30/INTP3/RTC1HZ/SCK00/SCL00/TRJO0/(VCOUT1)/(TxD1)/(TRDIOB1)	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P30 as an input. N-ch open-drain output (withstand voltage of EVDD) can be set for P30 as an output.
P31	7-1-3			P31/TI03/TO03/INTP4/(TRJIO0)/(VCOUT0)/PCLBUZ0/SSI00/(RxD1)	

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P40	7-1-3	I/O	Input port	P40/TOOL0	Port 4. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P50	8-1-13	I/O	Input port	P50/INTP1/SI00/RxD0/ TOOLRxD/DALIRxD0/ SDA00/TRGIOA/(TRJO0)/ (TI03)/(VCOUT3)/ (TRDIOC1)	Port 5. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P50 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P50 and P51 as outputs.
P51	7-1-13			P51/INTP2/SO00/TxD0/ TOOLTxD/DALITxD0/ TRGIOB/(VCOUT2)/ (TRDIOD1)	
P60	12-38-2	I/O	Input port	P60/CCD04/SCLA0	Port 6. 2-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 and P61 are N-ch open-drain (withstand voltage of 6 V). P60 and P61 can be set as controlled current drive port pins.
P61	12-38-2			P61/CCD05/SDAA0	
P120	7-3-3	I/O	Analog function	P120/ANI19/IVCMP0/ PGAI0/TRGIDZ/TRGTRG/ (TI02)/(TO02)	Port 12. 3-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P120 can be set for the analog pin function**Note**.
P121	7-2-1		Input port	P121/X1/INTP21/XT1	
P122	7-2-1			P122/X2/EXCLK/INTP20/ (TO00)/XT2/EXCLKS	
P137	2-1-3	Input	Input port	P137/INTP0	Port 13. 1-bit input-only port.
P147	7-18-2	I/O	Analog function	P147/ANI18/ANO2/ IVCMP3/PGAI3	Port 14. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function**Note**.
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register xx (PMCAxx).

Remark Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register x (PIORx). For details, see **Figure 7 - 8 Format of Peripheral I/O Redirection Register 0 (PIOR0)** to **Figure 7 - 11 Format of Peripheral I/O Redirection Register 3 (PIOR3)**.

2.1.5 32-pin products

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-18-1	I/O	Analog function	P00/ANI29/IVCMP1/ PGAI1/TI00/TRGCLKA/ (TRJO0)/TxD1/ (DALITxD0)	Port 0. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P01 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P00 as an output. P00 and P01 can be set for the analog pin functions**Note**.
P01	8-18-2			P01/ANI30/IVCMP2/ PGAI2/TO00/TRGCLKB/ TRJIO0/RxD1/ (DALIRxD0)/(TI00)	
P10	8-41-1	I/O	Analog function	P10/ANI20/CCD06/ SCK11/SCL11/TRDIOD1/ VCOUT2/(TxD2)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P14 to P17 as inputs. N-ch open-drain output (withstand voltage of EVDD) can be set for P10 to P15, and P17 as outputs. P10, P11, P16, and P17 can be set as controlled current drive port pins. P10 to P17 can be set for the analog pin functions**Note**.
P11	8-42-1			P11/ANI21/PGAO/CCD07/ SI11/SDA11/TRDIOC1/ (TO03)/(RxD2)/VCOUT3	
P12	7-8-2			P12/ANI22/SO11/ TRDIOB1/TKBO00/ IVREF1	
P13	7-8-1			P13/ANI23/TxD2/SO20/ TRDIOA1/TKBO01/ (TRDIOC0)	
P14	8-3-8			P14/RxD2/SI20/SDA20/ TRDIOD0/TKBO10/ VCOUT0/(SCLA0)/ANI24	
P15	8-3-8			P15/SCK20/SCL20/ TRDIOB0/TKBO11/ VCOUT1/(SDAA0)/ANI25/ PCLBUZ1/(TI01)/(TO01)	
P16	8-41-2			P16/ANI26/CCD00/TI01/ TO01/INTP5/TRDIOC0/ TKBO20/IVREF0/(RxD0)/ (TRGIOB)/(TRDIOA1)	
P17	8-41-1			P17/CCD01/TI02/TO02/ TRDIOA0/TRDCLK/ TKBO21/(TxD0)/ (TRGIOA)/ANI27/ (TRDIOD0)	
P20	4-3-5	I/O	Analog function	P20/ANI0/AVREFP/(INTP6)	Port 2. 4-bit I/O port. Input or output can be specified in 1-bit units. P20 to P23 can be set for the analog pin functions**Note**.
P21	4-3-5			P21/ANI1/AVREFM/(INTP7)	
P22	4-8-2			P22/ANI2/ANO0/PGAI4	
P23	4-8-2			P23/ANI3/ANO1/PGAGND	
P30	8-1-13	I/O	Input port	P30/INTP3/RTC1HZ/ SCK00/SCL00/TRJO0/ (VCOUT1)/(TxD1)/ (TRDIOB1)	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P30 as an input. N-ch open-drain output (withstand voltage of EVDD) can be set for P30 as an output.
P31	7-1-3			P31/TI03/TO03/INTP4/ (TRJIO0)/(VCOUT0)/ PCLBUZ0/(RxD1)	

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P40	7-1-3	I/O	Input port	P40/TOOL0	Port 4. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P50	8-1-13	I/O	Input port	P50/INTP1/SI00/RxD0/ TOOLRxD/DALIRxD0/ SDA00/TRGIOA/(TRJO0)/ (TI03)/(VCOUT3)/ (TRDIOC1)	Port 5. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P50 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P50 and P51 as outputs.
P51	7-1-13			P51/INTP2/SO00/TxD0/ TOOLTxD/DALITxD0/ TRGIOB/(VCOUT2)/ (TRDIOD1)	
P60	12-38-2	I/O	Input port	P60/CCD04/SCLA0	Port 6. 3-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 and P61 are N-ch open-drain (withstand voltage of 6 V). For P62, use of an on-chip pull-up resistor can be specified by a software setting. P60 to P62 can be set as controlled current drive port pins.
P61	12-38-2			P61/CCD05/SDAA0	
P62	7-38-3			P62/CCD02/$\overline{\text{SSI00}}$	
P70	7-1-7	I/O	Input port	P70/(TRDIOB0)	Port 7. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P120	7-3-3	I/O	Analog function	P120/ANI19/IVCMP0/ PGAI0/TRGIDZ/TRGTRG/ (TI02)/(TO02)	Port 12. 3-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P120 can be set for the analog pin function[Note].
P121	7-2-1		Input port	P121/X1/INTP21/ XT1	
P122	7-2-1			P122/X2/EXCLK/INTP20/ (TO00)/XT2/EXCLKS	
P137	2-1-3	Input	Input port	P137/INTP0	Port 13. 1-bit input-only port.
P147	7-18-2	I/O	Analog function	P147/ANI18/ANO2/ IVCMP3/PGAI3	Port 14. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function[Note].
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register xx (PMCAxx).

Remark Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register x (PIORx). For details, see **Figure 7 - 8 Format of Peripheral I/O Redirection Register 0 (PIOR0)** to **Figure 7 - 11 Format of Peripheral I/O Redirection Register 3 (PIOR3)**.

2.1.6 40-pin products

(1/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-18-1	I/O	Analog function	P00/ANI29/IVCMP1/ PGAI1/TI00/TRGCLKA/ (TRJO0)/TxD1/ (DALITxD0)	Port 0. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P01 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P00 as an output. P00 and P01 can be set for the analog pin functions**Note**.
P01	8-18-2			P01/ANI30/IVCMP2/ PGAI2/TO00/TRGCLKB/ TRJIO0/RxD1/ (DALIRxD0)/(TI00)	
P10	8-41-1	I/O	Analog function	P10/ANI20/CCD06/ SCK11/SCL11/TRDIOD1/ VCOUT2/(TxD2)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P14 to P17 as inputs. N-ch open-drain output (withstand voltage of EVDD) can be set for P10 to P15 and P17 as outputs. P10, P11, P16, and P17 can be set as controlled current drive port pins. P10 to P17 can be set for the analog pin functions**Note**.
P11	8-42-1			P11/ANI21/PGAO/CCD07/ SI11/SDA11/TRDIOC1/ (TO03)/(RxD2)/VCOUT3	
P12	7-8-2			P12/ANI22/SO11/ TRDIOB1/TKBO00/ IVREF1	
P13	7-8-1			P13/ANI23/TxD2/SO20/ TRDIOA1/TKBO01/ (TRDIOC0)	
P14	8-3-8			P14/RxD2/SI20/SDA20/ TRDIOD0/TKBO10/ VCOUT0/(SCLA0)/ANI24	
P15	8-3-8			P15/SCK20/SCL20/ TRDIOB0/TKBO11/ VCOUT1/(SDAA0)/ANI25/ PCLBUZ1	
P16	8-41-2			P16/ANI26/CCD00/TI01/ TO01/INTP5/TRDIOC0/ TKBO20/IVREF0/(RxD0)/ (TRGIOB)	
P17	8-41-1			P17/CCD01/TI02/TO02/ TRDIOA0/TRDCLK/ TKBO21/(TxD0)/ (TRGIOA)/ANI27	
P20	4-3-5	I/O	Analog function	P20/ANI0/AVREFP/(INTP6)	Port 2. 7-bit I/O port. Input or output can be specified in 1-bit units. P20 to P26 can be set for the analog pin functions**Note**.
P21	4-3-5			P21/ANI1/AVREFM/(INTP7)	
P22	4-8-2			P22/ANI2/ANO0/PGAI4	
P23	4-8-2			P23/ANI3/ANO1/PGAGND	
P24	4-3-5			P24/ANI4	
P25	4-3-5			P25/ANI5	
P26	4-3-5			P26/ANI6	

(2/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P30	8-1-13	I/O	Input port	P30/INTP3/RTC1HZ/ SCK00/SCL00/TRJO0/ (VCOUT1)/(TRDIOB1)	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P30 as an input. N-ch open-drain output (withstand voltage of EV_{DD}) can be set for P30 as an output.
P31	7-1-3			P31/TI03/TO03/INTP4/ (TRJIO0)/(VCOUT0)/ PCLBUZ0	
P40	7-1-3	I/O	Input port	P40/TOOL0	Port 4. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P50	8-1-13	I/O	Input port	P50/INTP1/SI00/RxD0/ TOOLRxD/DALIRxD0/ SDA00/TRGIOA/(TRJO0)/ (TI03)/(VCOUT3)	Port 5. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P50 as an input. N-ch open-drain output (withstand voltage of V_{DD}) can be set for P50 and P51 as outputs.
P51	7-1-13			P51/INTP2/SO00/TxD0/ TOOLTxD/DALITxD0/ TRGIOB/(VCOUT2)/ (TRDIOD1)	
P60	12-38-2	I/O	Input port	P60/CCD04/SCLA0	Port 6. 3-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 and P61 are N-ch open-drain (withstand voltage of 6 V). For P62, use of an on-chip pull-up resistor can be specified by a software setting. P60 to P62 can be set as controlled current drive port pins.
P61	12-38-2			P61/CCD05/SDAA0	
P62	7-38-3			P62/CCD02/$\overline{SSI00}$	
P70	7-1-7	I/O	Input port	P70/(TRDIOB0)/KR0/ SCK21/SCL21	Port 7. 4-bit I/O port. A TTL input buffer can be set for P73 as an input. N-ch open-drain output (withstand voltage of V_{DD}) can be set for P71 to P73 as outputs. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P71	7-1-13			P71/KR1/SI21/SDA21/ (TI01)/(TO01)/(TRDIOD0)	
P72	7-1-13			P72/KR2/SO21/(TxD1)/ (TRDIOA1)	
P73	8-1-13			P73/KR3/(RxD1)/ (TRDIOC1)	
P120	7-3-3	I/O	Analog function	P120/ANI19/IVCMP0/ PGAI0/TRGIDZ/TRGTRG	Port 12. 3-bit I/O port and 2-bit input-only port. For P120 to P122, input or output can be specified in 1-bit units. For P120 to P122, use of an on-chip pull-up resistor can be specified by a software setting at input port. P120 can be set for the analog pin function[Note].
P121	7-2-1		Input port	P121/VBAT/X1/INTP21/ (TI02)/(TO02)	
P122	7-2-1			P122/X2/EXCLK/INTP20/ (TO00)	
P123	2-2-1	Input		P123/XT1	
P124	2-2-1			P124/XT2/EXCLKS	
P137	2-1-3	Input	Input port	P137/INTP0	Port 13. 1-bit input-only port.
P147	7-18-2	I/O	Analog function	P147/ANI18/ANO2/ IVCMP3/PGAI3	Port 14. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function[Note].

(3/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to V_{DD}, either directly or via a resistor.

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register xx (PMCAxx).

Remark Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register x (PIORx). For details, see **Figure 7 - 8 Format of Peripheral I/O Redirection Register 0 (PIOR0)** to **Figure 7 - 11 Format of Peripheral I/O Redirection Register 3 (PIOR3)**.

2.1.7 44-pin products

(1/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-18-1	I/O	Analog function	P00/ANI29/IVCMP1/ PGAI1/TI00/TRGCLKA/ (TRJO0)/TxD1/ (DALITxD0)	Port 0. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P01 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P00 as an output. P00 and P01 can be set for the analog pin functions**Note**.
P01	8-18-2			P01/ANI30/IVCMP2/ PGAI2/TO00/TRGCLKB/ TRJIO0/RxD1/ (DALIRxD0)/(TI00)	
P10	8-41-1	I/O	Analog function	P10/ANI20/CCD06/ SCK11/SCL11/TRDIOD1/ VCOUT2/(TxD2)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P14 to P17 as inputs. N-ch open-drain output (withstand voltage of EVDD) can be set for P10 to P15 and P17 as outputs. P10, P11, P16, and P17 can be set as controlled current drive port pins. P10 to P17 can be set for the analog pin functions**Note**.
P11	8-42-1			P11/ANI21/PGAO/CCD07/ SI11/SDA11/TRDIOC1/ (TO03)/(RxD2)/VCOUT3	
P12	7-8-2			P12/ANI22/SO11/ TRDIOB1/TKBO00/ IVREF1	
P13	7-8-1			P13/ANI23/TxD2/SO20/ TRDIOA1/TKBO01/ (TRDIOC0)	
P14	8-3-8			P14/RxD2/SI20/SDA20/ TRDIOD0/TKBO10/ VCOUT0/(SCLA0)/ANI24	
P15	8-3-8			P15/SCK20/SCL20/ TRDIOB0/TKBO11/ VCOUT1/(SDAA0)/ANI25/ PCLBUZ1	
P16	8-41-2			P16/ANI26/CCD00/TI01/ TO01/INTP5/TRDIOC0/ TKBO20/IVREF0/(RxD0)/ (TRGIOB)	
P17	8-41-1			P17/CCD01/TI02/TO02/ TRDIOA0/TRDCLK/ TKBO21/(TxD0)/ (TRGIOA)/ANI27	
P20	4-3-5	I/O	Analog function	P20/ANI0/AVREFP/(INTP6)	Port 2. 8-bit I/O port. Input or output can be specified in 1-bit units. P20 to P27 can be set for the analog pin functions**Note**.
P21	4-3-5			P21/ANI1/AVREFM/(INTP7)	
P22	4-8-2			P22/ANI2/ANO0/PGAI4	
P23	4-8-2			P23/ANI3/ANO1/PGAGND	
P24	4-3-5			P24/ANI4	
P25	4-3-5			P25/ANI5	
P26	4-3-5			P26/ANI6	
P27	4-3-5			P27/ANI7	

(2/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P30	8-1-13	I/O	Input port	P30/INTP3/RTC1HZ/ SCK00/SCL00/TRJO0/ (VCOUT1)/(TRDIOB1)	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P30 as an input. N-ch open-drain output (withstand voltage of EV_{DD}) can be set for P30 as an output.
P31	7-1-3			P31/TI03/TO03/INTP4/ (TRJIO0)/(VCOUT0)/ PCLBUZ0	
P40	7-1-3	I/O	Input port	P40/TOOL0	Port 4. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P41	7-1-3			P41/(TRJIO0)	
P50	8-1-13	I/O	Input port	P50/INTP1/SI00/RxD0/ TOOLRxD/DALIRxD0/ SDA00/TRGIOA/(TRJO0)/ (TI03)/(VCOUT3)	Port 5. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P50 as an input. N-ch open-drain output (withstand voltage of V_{DD}) can be set for P50 and P51 as outputs.
P51	7-1-13			P51/INTP2/SO00/TxD0/ TOOLTxD/DALITxD0/ TRGIOB/(VCOUT2)/ (TRDIOD1)	
P60	12-38-2	I/O	Input port	P60/CCD04/SCLA0	Port 6. 4-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 and P61 are N-ch open-drain (withstand voltage of 6 V). For P62 and P63, use of an on-chip pull-up resistor can be specified by a software setting. P60 to P63 can be set as controlled current drive port pins.
P61	12-38-2			P61/CCD05/SDAA0	
P62	7-38-3			P62/CCD02/$\overline{SSI00}$	
P63	7-38-3			P63/CCD03	
P70	7-1-7	I/O	Input port	P70/(TRDIOB0)/KR0/ SCK21/SCL21	Port 7. 4-bit I/O port. A TTL input buffer can be set for P73 as an input. N-ch open-drain output (withstand voltage of V_{DD}) can be set for P71 to P73 as outputs. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P71	7-1-13			P71/KR1/SI21/SDA21/ (TI01)/(TO01)/(TRDIOD0)	
P72	7-1-13			P72/KR2/SO21/(TxD1)/ (TRDIOA1)	
P73	8-1-13			P73/KR3/(RxD1)/ (TRDIOC1)	
P120	7-3-3	I/O	Analog function	P120/ANI19/IVCMP0/ PGAI0/TRGIDZ/TRGTRG	Port 12. 3-bit I/O port and 2-bit input-only port. For P120 to P122, input or output can be specified in 1-bit units. For P120 to P122, use of an on-chip pull-up resistor can be specified by a software setting at input port. P120 can be set for the analog pin function[Note].
P121	7-2-1		Input port	P121/VBAT/X1/INTP21/ (TI02)/(TO02)	
P122	7-2-1			P122/X2/EXCLK/INTP20/ (TO00)	
P123	2-2-1	Input		P123/XT1	
P124	2-2-1			P124/XT2/EXCLKS	
P137	2-1-3	Input	Input port	P137/INTP0	Port 13. 1-bit input-only port.
P146	7-3-3	I/O	Analog function	P146/ANI28	Port 14. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P146 and P147 can be set for the analog pin functions[Note].
P147	7-18-2			P147/ANI18/ANO2/ IVCMP3/PGAI3	

(3/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to V_{DD}, either directly or via a resistor.

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register xx (PMCAxx).

Remark Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register x (PIORx). For details, see **Figure 7 - 8 Format of Peripheral I/O Redirection Register 0 (PIOR0)** to **Figure 7 - 11 Format of Peripheral I/O Redirection Register 3 (PIOR3)**.

2.1.8 48-pin products

(1/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-18-1	I/O	Analog function	P00/ANI29/IVCMP1/ PGAI1/TI00/TRGCLKA/ (TRJO0)/TxD1/ (DALITxD0)	Port 0. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P01 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P00 as an output. P00 and P01 can be set for the analog pin functions**Note**.
P01	8-18-2			P01/ANI30/IVCMP2/ PGAI2/TO00/TRGCLKB/ TRJIO0/RxD1/ (DALIRxD0)/(TI00)	
P10	8-41-1	I/O	Analog function	P10/ANI20/CCD06/ SCK11/SCL11/TRDIOD1/ VCOUT2/(TxD2)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P14 to P17 as inputs. N-ch open-drain output (withstand voltage of EVDD) can be set for P10 to P15 and P17 as outputs. P10, P11, P16, and P17 can be set as controlled current drive port pins. P10 to P17 can be set for the analog pin functions**Note**.
P11	8-42-1			P11/ANI21/PGAO/CCD07/ SI11/SDA11/TRDIOC1/ (TO03)/(RxD0_1)/(RxD2)/ VCOUT3	
P12	7-8-2			P12/ANI22/SO11/ TRDIOB1/TKBO00/ IVREF1/(TxD0_1)	
P13	7-8-1			P13/ANI23/TxD2/SO20/ TRDIOA1/TKBO01/ (TRDIOC0)	
P14	8-3-8			P14/RxD2/SI20/SDA20/ TRDIOD0/TKBO10/ VCOUT0/(SCLA0)/ANI24	
P15	8-3-8			P15/SCK20/SCL20/ TRDIOB0/TKBO11/ VCOUT1/(SDAA0)/ANI25/ PCLBUZ1	
P16	8-41-2			P16/ANI26/CCD00/TI01/ TO01/INTP5/TRDIOC0/ TKBO20/IVREF0/(RxD0)/ (TRGIOB)	
P17	8-41-1			P17/CCD01/TI02/TO02/ TRDIOA0/TRDCLK/ TKBO21/(TxD0)/ (TRGIOA)/ANI27	
P20	4-3-5	I/O	Analog function	P20/ANI0/AVREFP/(INTP6)	Port 2. 8-bit I/O port. Input or output can be specified in 1-bit units. P20 to P27 can be set for the analog pin functions**Note**.
P21	4-3-5			P21/ANI1/AVREFM/(INTP7)	
P22	4-8-2			P22/ANI2/ANO0/PGAI4	
P23	4-8-2			P23/ANI3/ANO1/PGAGND	
P24	4-3-5			P24/ANI4	
P25	4-3-5			P25/ANI5	
P26	4-3-5			P26/ANI6	
P27	4-3-5			P27/ANI7	

(2/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P30	8-1-13	I/O	Input port	P30/INTP3/RTC1HZ/ SCK00/SCL00/TRJO0/ (VCOUT1)	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P30 as an input. N-ch open-drain output (withstand voltage of EV_{DD}) can be set for P30 as an output.
P31	7-1-3			P31/TI03/TO03/INTP4/ (TRJIO0)/(VCOUT0)/ (PCLBUZ0)	
P40	7-1-3	I/O	Input port	P40/TOOL0	Port 4. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P41	7-1-3			P41/(TRJIO0)	
P50	8-1-13	I/O	Input port	P50/INTP1/SI00/RxD0/ TOOLRxD/DALIRxD0/ SDA00/TRGIOA/(TRJO0)/ (TI03)/(VCOUT3)	Port 5. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P50 as an input. N-ch open-drain output (withstand voltage of V_{DD}) can be set for P50 and P51 as outputs.
P51	7-1-13			P51/INTP2/SO00/TxD0/ TOOLTxD/DALITxD0/ TRGIOB/(VCOUT2)	
P60	12-38-2	I/O	Input port	P60/CCD04/SCLA0	Port 6. 4-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 and P61 are N-ch open-drain (withstand voltage of 6 V). For P62 and P63, use of an on-chip pull-up resistor can be specified by a software setting. P60 to P63 can be set as controlled current drive port pins.
P61	12-38-2			P61/CCD05/SDAA0	
P62	7-38-3			P62/CCD02/$\overline{SSI00}$	
P63	7-38-3			P63/CCD03	
P70	7-1-7	I/O	Input port	P70/(TRDIOB0)/KR0/ SCK21/SCL21	Port 7. 6-bit I/O port. A TTL input buffer can be set for P73 as an input. N-ch open-drain output (withstand voltage of V_{DD}) can be set for P71 to P74 as outputs. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P71	7-1-13			P71/KR1/SI21/SDA21/ (TI01)/(TO01)/(TRDIOD0)	
P72	7-1-13			P72/KR2/SO21/(TxD1)/ (TRDIOA1)	
P73	8-1-13			P73/KR3/(RxD1)/ (TRDIOC1)/SO01	
P74	7-1-13			P74/KR4/INTP8/SI01/ SDA01/(TRDIOB1)	
P75	7-1-7			P75/KR5/INTP9/SCK01/ SCL01/(TRDIOD1)	
P120	7-3-3	I/O	Analog function	P120/ANI19/IVCMP0/ PGAI0/TRGIDZ/TRGTRG	Port 12. 3-bit I/O port and 2-bit input-only port. For P120 to P122, input or output can be specified in 1-bit units. For P120 to P122, use of an on-chip pull-up resistor can be specified by a software setting at input port. P120 can be set for the analog pin function**Note**.
P121	7-2-1		Input port	P121/VBAT/X1/INTP21/ (TI02)/(TO02)	
P122	7-2-1			P122/X2/EXCLK/INTP20/ (TO00)	
P123	2-2-1	Input		P123/XT1	
P124	2-2-1			P124/XT2/EXCLKS	
P130	1-1-1	Output	Output port	P130	Port 13. 1-bit output-only port and 1-bit input-only port.
P137	2-1-3	Input	Input port	P137/INTP0	

(3/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P140	7-1-3	I/O	Input port	P140/PCLBUZ0/INTP6	Port 14. 3-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P146 and P147 can be set for the analog pin functions**Note**.
P146	7-3-3		Analog function	P146/ANI28	
P147	7-18-2			P147/ANI18/ANO2/ IVCMP3/PGAI3	
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register xx (PMCAxx).

Remark Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register x (PIORx). For details, see **Figure 7 - 8 Format of Peripheral I/O Redirection Register 0 (PIOR0)** to **Figure 7 - 11 Format of Peripheral I/O Redirection Register 3 (PIOR3)**.

2.1.9 52-pin products

(1/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-18-1	I/O	Analog function	P00/ANI29/IVCMP1/ PGAI1/TI00/TRGCLKA/ (TRJO0)	Port 0. 4-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P01 and P03 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P00, P02, and P03 as outputs. P00 to P03 can be set for the analog pin functions**Note**.
P01	8-18-2			P01/ANI30/IVCMP2/ PGAI2/TO00/TRGCLKB/ TRJIO0	
P02	7-9-6			P02/ANI17/TxD1/ (DALITxD0)	
P03	8-3-6			P03/ANI16/RxD1/ (DALIRxD0)/(TI00)	
P10	8-41-1	I/O	Analog function	P10/ANI20/CCD06/ SCK11/SCL11/TRDIOD1/ VCOUT2	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P14 to P17 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P10 to P15 and P17 as outputs. P10, P11, P16, and P17 can be set as controlled current drive port pins. P10 to P17 can be set for the analog pin functions**Note**.
P11	8-42-1			P11/ANI21/PGAO/CCD07/ SI11/SDA11/TRDIOC1/ (TO03)/(RxD0_1)/ VCOUT3	
P12	7-8-2			P12/ANI22/SO11/ TRDIOB1/TKBO00/ IVREF1/(TxD0_1)	
P13	7-8-1			P13/ANI23/TxD2/SO20/ TRDIOA1/TKBO01/ (TRDIOC0)	
P14	8-3-8			P14/RxD2/SI20/SDA20/ TRDIOD0/TKBO10/ VCOUT0/(SCLA0)/ANI24	
P15	8-3-8			P15/SCK20/SCL20/ TRDIOB0/TKBO11/ VCOUT1/(SDAA0)/ANI25/ PCLBUZ1	
P16	8-41-2			P16/ANI26/CCD00/TI01/ TO01/INTP5/TRDIOC0/ TKBO20/IVREF0/(RxD0)/ (TRGIOB)	
P17	8-41-1			P17/CCD01/TI02/TO02/ TRDIOA0/TRDCLK/ TKBO21/(TxD0)/ (TRGIOA)/ANI27	
P20	4-3-5	I/O	Analog function	P20/ANI0/AVREFP/(INTP6)	Port 2. 8-bit I/O port. Input or output can be specified in 1-bit units. P20 to P27 can be set for the analog pin functions**Note**.
P21	4-3-5			P21/ANI1/AVREFM/(INTP7)	
P22	4-8-2			P22/ANI2/ANO0/PGAI4	
P23	4-8-2			P23/ANI3/ANO1/PGAGND	
P24	4-3-5			P24/ANI4	
P25	4-3-5			P25/ANI5	
P26	4-3-5			P26/ANI6	
P27	4-3-5			P27/ANI7	

(2/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P30	8-1-13	I/O	Input port	P30/INTP3/RTC1HZ/ SCK00/SCL00/TRJO0/ (VCOUT1)	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P30 as an input. N-ch open-drain output (withstand voltage of EV_{DD}) can be set for P30 as an output.
P31	7-1-3			P31/TI03/TO03/INTP4/ (TRJIO0)/(VCOUT0)/ (PCLBUZ0)	
P40	7-1-3	I/O	Input port	P40/TOOL0	Port 4. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P41	7-1-3			P41/(TRJIO0)	
P50	8-1-13	I/O	Input port	P50/INTP1/SI00/RxD0/ TOOLRxD/DALIRxD0/ SDA00/TRGIOA/(TRJO0)/ (TI03)/(VCOUT3)	Port 5. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P50 as an input. N-ch open-drain output (withstand voltage of V_{DD}) can be set for P50 and P51 as outputs.
P51	7-1-13			P51/INTP2/SO00/TxD0/ TOOLTxD/DALITxD0/ TRGIOB/(VCOUT2)	
P60	12-38-2	I/O	Input port	P60/CCD04/SCLA0	Port 6. 4-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 and P61 are N-ch open-drain (withstand voltage of 6 V). For P62 and P63, use of an on-chip pull-up resistor can be specified by a software setting. P60 to P63 can be set as controlled current drive port pins.
P61	12-38-2			P61/CCD05/SDAA0	
P62	7-38-3			P62/CCD02/$\overline{\text{SSI00}}$	
P63	7-38-3			P63/CCD03	
P70	7-1-7	I/O	Input port	P70/(TRDIOB0)/KR0/ SCK21/SCL21	Port 7. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P73 as an input. N-ch open-drain output (withstand voltage of EV_{DD}) can be set for P71 to P74 as outputs.
P71	7-1-13			P71/KR1/SI21/SDA21/ (TI01)/(TO01)/(TRDIOD0)	
P72	7-1-13			P72/KR2/SO21/(TxD1)/ (TRDIOA1)	
P73	8-1-13			P73/KR3/(RxD1)/ (TRDIOC1)/SO01	
P74	7-1-13			P74/KR4/INTP8/SI01/ SDA01/(TRDIOB1)	
P75	7-1-7			P75/KR5/INTP9/SCK01/ SCL01/(TRDIOD1)	
P76	7-1-3			P76/KR6/INTP10/(RxD2)	
P77	7-1-3			P77/KR7/INTP11/(TxD2)	
P120	7-3-3	I/O	Analog function	P120/ANI19/IVCMP0/ PGAI0/TRGIDZ/TRGTRG	Port 12. 3-bit I/O port and 2-bit input-only port. For P120 to P122, input or output can be specified in 1-bit units. For P120 to P122, use of an on-chip pull-up resistor can be specified by a software setting at input port. P120 can be set for the analog pin function**Note**.
P121	7-2-1		Input port	P121/VBAT/X1/INTP21/ (TI02)/(TO02)	
P122	7-2-1			P122/X2/EXCLK/INTP20/ (TO00)	
P123	2-2-1	Input		P123/XT1	
P124	2-2-1			P124/XT2/EXCLKS	

(3/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P130	1-1-1	Output	Output port	P130	Port 13. 1-bit output-only port and 1-bit input-only port.
P137	2-1-3	Input	Input port	P137/INTP0	
P140	7-1-3	I/O	Input port	P140/PCLBUZ0/INTP6	Port 14. 3-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P146 and P147 can be set for the analog pin functions**Note**.
P146	7-3-3		Analog function	P146/ANI28	
P147	7-18-2			P147/ANI18/ANO2/ IVCMP3/PGAI3	
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register xx (PMCAxx).

Remark Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register x (PIORx). For details, see **Figure 7 - 8 Format of Peripheral I/O Redirection Register 0 (PIOR0)** to **Figure 7 - 11 Format of Peripheral I/O Redirection Register 3 (PIOR3)**.

2.1.10 64-pin products

(1/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-18-1	I/O	Analog function	P00/ANI29/IVCMP1/ PGAI1/TI00/TRGCLKA/ (TRJO0)	Port 0. 7-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P01, P03, and P04 as inputs. N-ch open-drain output (withstand voltage of EV_{DD}) can be set for P00 and P02 to P04 as outputs. P00 to P03 can be set for the analog pin functions[Note].
P01	8-18-2			P01/ANI30/IVCMP2/ PGAI2/TO00/TRGCLKB/ TRJIO0	
P02	7-9-6			P02/ANI17/TxD1/ (DALITxD0)/SO10	
P03	8-3-6			P03/ANI16/RxD1/ (DALIRxD0)/(TI00)/SI10/ SDA10	
P04	8-1-10		Input port	P04/SCK10/SCL10	
P05	7-1-3			P05/(INTP10)	
P06	7-1-3			P06/(INTP11)/(TRJIO0)	
P10	8-41-1	I/O	Analog function	P10/ANI20/CCD06/ SCK11/SCL11/TRDIOD1/ VCOUT2	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P14 to P17 as inputs. N-ch open-drain output (withstand voltage of EV_{DD}) can be set for P10 to P15 and P17 as outputs. P10, P11, P16, and P17 can be set as controlled current drive port pins. P10 to P17 can be set for the analog pin functions[Note].
P11	8-42-1			P11/ANI21/PGAO/CCD07/ SI11/SDA11/TRDIOC1/ (TO03)/(RxD0_1)/ VCOUT3	
P12	7-8-2			P12/ANI22/SO11/ TRDIOB1/TKBO00/ IVREF1/(INTP5)/(TxD0_1)	
P13	7-8-1			P13/ANI23/TxD2/SO20/ TRDIOA1/TKBO01/ (TRDIOC0)	
P14	8-3-8			P14/RxD2/SI20/SDA20/ TRDIOD0/TKBO10/ VCOUT0/(SCLA0)/ANI24	
P15	8-3-8			P15/SCK20/SCL20/ TRDIOB0/TKBO11/ VCOUT1/(SDAA0)/ANI25	
P16	8-41-2			P16/ANI26/CCD00/TI01/ TO01/INTP5/TRDIOC0/ TKBO20/IVREF0/(RxD0)/ (TRGIOB)/(SI00)	
P17	8-41-1			P17/CCD01/TI02/TO02/ TRDIOA0/TRDCLK/ TKBO21/(TxD0)/ (TRGIOA)/ANI27/(SO00)	

(2/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P20	4-3-5	I/O	Analog function	P20/ANI0/AVREFP/(INTP6)	Port 2. 8-bit I/O port. Input or output can be specified in 1-bit units. P20 to P27 can be set for the analog pin functions[Note].
P21	4-3-5			P21/ANI1/AVREFM/(INTP7)	
P22	4-8-2			P22/ANI2/ANO0/PGAI4	
P23	4-8-2			P23/ANI3/ANO1/PGAGND	
P24	4-3-5			P24/ANI4	
P25	4-3-5			P25/ANI5	
P26	4-3-5			P26/ANI6	
P27	4-3-5			P27/ANI7	
P30	8-1-13	I/O	Input port	P30/INTP3/RTC1HZ/ SCK00/SCL00/TRJO0/ (VCOUT1)	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P30 as an input. N-ch open-drain output (withstand voltage of EVDD) can be set for P30 as an output.
P31	7-1-3			P31/TI03/TO03/INTP4/ (TRJIO0)/(VCOUT0)/ (PCLBUZ0)	
P40	7-1-3	I/O	Input port	P40/TOOL0	Port 4. 4-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P41	7-1-3			P41/(TRJIO0)	
P42	7-1-3			P42/(INTP8)	
P43	7-1-3			P43/(INTP9)	
P50	8-1-13	I/O	Input port	P50/INTP1/SI00/RxD0/ TOOLRxD/DALIRxD0/ SDA00/TRGIOA/(TRJO0)/ (TI03)/(VCOUT3)	Port 5. 6-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P50 and P55 as inputs. N-ch open-drain output (withstand voltage of EVDD) can be set for P50, P51, and P55 as outputs.
P51	7-1-13			P51/INTP2/SO00/TxD0/ TOOLTxD/DALITxD0/ TRGIOB/(VCOUT2)	
P52	7-1-3			P52/(INTP1)	
P53	7-1-3			P53/(INTP2)	
P54	7-1-3			P54/(INTP3)	
P55	8-1-10			P55/(PCLBUZ1)/(SCK00)/ (INTP4)	
P60	12-38-2	I/O	Input port	P60/CCD04/SCLA0	Port 6. 4-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 and P61 are N-ch open-drain (withstand voltage of 6 V). For P62 and P63, use of an on-chip pull-up resistor can be specified by a software setting. P60 to P63 can be set as controlled current drive port pins.
P61	12-38-2			P61/CCD05/SDAA0	
P62	7-38-3			P62/CCD02/$\overline{\text{SSI00}}$	
P63	7-38-3			P63/CCD03	

(3/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P70	7-1-7	I/O	Input port	P70/(TRDIOB0)/KR0/ SCK21/SCL21	Port 7. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P73 as an input. N-ch open-drain output (withstand voltage of EVDD) can be set for P71 to P74 as outputs.
P71	7-1-13			P71/KR1/SI21/SDA21/ (TI01)/(TO01)/(TRDIOD0)	
P72	7-1-13			P72/KR2/SO21/(TxD1)/ (TRDIOA1)	
P73	8-1-13			P73/KR3/(RxD1)/ (TRDIOC1)/SO01	
P74	7-1-13			P74/KR4/INTP8/SI01/ SDA01/(TRDIOB1)	
P75	7-1-7			P75/KR5/INTP9/SCK01/ SCL01/(TRDIOD1)	
P76	7-1-3			P76/KR6/INTP10/(RxD2)	
P77	7-1-3			P77/KR7/INTP11/(TxD2)	
P120	7-3-3	I/O	Analog function	P120/ANI19/IVCMP0/ PGAI0/TRGIDZ/TRGTRG	Port 12. 3-bit I/O port and 2-bit input-only port. For P120 to P122, input or output can be specified in 1-bit units. For P120 to P122, use of an on-chip pull-up resistor can be specified by a software setting at input port. P120 can be set for the analog pin function[Note].
P121	7-2-1		Input port	P121/VBAT/X1/INTP21/ (TI02)/(TO02)	
P122	7-2-1			P122/X2/EXCLK/INTP20/ (TO00)	
P123	2-2-1	Input		P123/XT1	
P124	2-2-1			P124/XT2/EXCLKS	
P130	1-1-1	Output	Output port	P130	Port 13. 1-bit output-only port and 1-bit input-only port.
P137	2-1-3	Input	Input port	P137/INTP0	
P140	7-1-3	I/O	Input port	P140/PCLBUZ0/INTP6	Port 14. 4-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P146 and P147 can be set for the analog pin functions[Note].
P141	7-1-3			P141/PCLBUZ1/INTP7	
P146	7-3-3		Analog function	P146/ANI28	
P147	7-18-2			P147/ANI18/ANO2/ IVCMP3/PGAI3	
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register xx (PMCAxx).

Remark Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register x (PIORx). For details, see **Figure 7 - 8 Format of Peripheral I/O Redirection Register 0 (PIOR0)** to **Figure 7 - 11 Format of Peripheral I/O Redirection Register 3 (PIOR3)**.

2.2 Pin Functions Other Than Port Pin Functions

2.2.1 Functions for each product

(1/5)

Function Name	64-pin	52-pin	48-pin	44-pin	40-pin	32-pin	30-pin	25-pin	24-pin	20-pin
ANI0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ANI1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ANI2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ANI3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ANI4	✓	✓	✓	✓	✓	—	—	—	—	—
ANI5	✓	✓	✓	✓	✓	—	—	—	—	—
ANI6	✓	✓	✓	✓	✓	—	—	—	—	—
ANI7	✓	✓	✓	✓	—	—	—	—	—	—
ANI16	✓	✓	—	—	—	—	—	—	—	—
ANI17	✓	✓	—	—	—	—	—	—	—	—
ANI18	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
ANI19	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ANI20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ANI21	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ANI22	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ANI23	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ANI24	✓	✓	✓	✓	✓	✓	✓	✓	—	✓
ANI25	✓	✓	✓	✓	✓	✓	✓	✓	—	—
ANI26	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
ANI27	✓	✓	✓	✓	✓	✓	✓	✓	—	—
ANI28	✓	✓	✓	✓	—	—	—	—	—	—
ANI29	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ANI30	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ANO0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ANO1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ANO2	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
IVCMP0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
IVCMP1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
IVCMP2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
IVCMP3	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
IVREF0	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
IVREF1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
VCOUT0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
VCOUT1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
VCOUT2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

(2/5)

Function Name	64-pin	52-pin	48-pin	44-pin	40-pin	32-pin	30-pin	25-pin	24-pin	20-pin
VCOUT3	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
PGAI0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PGAI1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PGAI2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PGAI3	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
PGAI4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PGAGND	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PGAO	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CCD00	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
CCD01	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
CCD02	✓	✓	✓	✓	✓	✓	—	—	—	—
CCD03	✓	✓	✓	✓	—	—	—	—	—	—
CCD04	✓	✓	✓	✓	✓	✓	✓	—	—	—
CCD05	✓	✓	✓	✓	✓	✓	✓	—	—	—
CCD06	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CCD07	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP1	✓	✓	✓	✓	✓	✓	✓	—	—	—
INTP2	✓	✓	✓	✓	✓	✓	✓	—	—	—
INTP3	✓	✓	✓	✓	✓	✓	✓	—	—	—
INTP4	✓	✓	✓	✓	✓	✓	✓	—	—	—
INTP5	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
INTP6	✓	✓	✓	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)
INTP7	✓	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)
INTP8	✓	✓	✓	—	—	—	—	—	—	—
INTP9	✓	✓	✓	—	—	—	—	—	—	—
INTP10	✓	✓	—	—	—	—	—	—	—	—
INTP11	✓	✓	—	—	—	—	—	—	—	—
INTP20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP21	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
KR0	✓	✓	✓	✓	✓	—	—	—	—	—
KR1	✓	✓	✓	✓	✓	—	—	—	—	—
KR2	✓	✓	✓	✓	✓	—	—	—	—	—
KR3	✓	✓	✓	✓	✓	—	—	—	—	—
KR4	✓	✓	✓	—	—	—	—	—	—	—
KR5	✓	✓	✓	—	—	—	—	—	—	—
KR6	✓	✓	—	—	—	—	—	—	—	—
KR7	✓	✓	—	—	—	—	—	—	—	—
PCLBUZ0	✓	✓	✓	✓	✓	✓	✓	—	—	—

(3/5)

Function Name	64-pin	52-pin	48-pin	44-pin	40-pin	32-pin	30-pin	25-pin	24-pin	20-pin
PCLBUZ1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
REGC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
RTC1HZ	✓	✓	✓	✓	✓	✓	✓	—	—	—
$\overline{\text{RESET}}$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
RxD0	✓	✓	✓	✓	✓	✓	✓	(✓)	(✓)	—
RxD0_1	✓	✓	✓	—	—	—	—	—	—	—
RxD1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
RxD2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TxD0	✓	✓	✓	✓	✓	✓	✓	(✓)	(✓)	—
TxD0_1	✓	✓	✓	—	—	—	—	—	—	—
TxD1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TxD2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SCK00	✓	✓	✓	✓	✓	✓	✓	—	—	—
SCK01	✓	✓	✓	—	—	—	—	—	—	—
SCK10	✓	—	—	—	—	—	—	—	—	—
SCK11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SCK20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SCK21	✓	✓	✓	✓	✓	—	—	—	—	—
SCL00	✓	✓	✓	✓	✓	✓	✓	—	—	—
SCL01	✓	✓	✓	—	—	—	—	—	—	—
SCL10	✓	—	—	—	—	—	—	—	—	—
SCL11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SCL20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SCL21	✓	✓	✓	✓	✓	—	—	—	—	—
SDA00	✓	✓	✓	✓	✓	✓	✓	—	—	—
SDA01	✓	✓	✓	—	—	—	—	—	—	—
SDA10	✓	—	—	—	—	—	—	—	—	—
SDA11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SDA20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SDA21	✓	✓	✓	✓	✓	—	—	—	—	—
SI00	✓	✓	✓	✓	✓	✓	✓	—	—	—
SI01	✓	✓	✓	—	—	—	—	—	—	—
SI10	✓	—	—	—	—	—	—	—	—	—
SI11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SI20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SI21	✓	✓	✓	✓	✓	—	—	—	—	—
$\overline{\text{SSI00}}$	✓	✓	✓	✓	✓	✓	✓	—	—	—
SO00	✓	✓	✓	✓	✓	✓	✓	—	—	—
SO01	✓	✓	✓	—	—	—	—	—	—	—

(4/5)

Function Name	64-pin	52-pin	48-pin	44-pin	40-pin	32-pin	30-pin	25-pin	24-pin	20-pin
SO10	✓	—	—	—	—	—	—	—	—	—
SO11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SO20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SO21	✓	✓	✓	✓	✓	—	—	—	—	—
SCLA0	✓	✓	✓	✓	✓	✓	✓	(✓)	(✓)	(✓)
SDAA0	✓	✓	✓	✓	✓	✓	✓	(✓)	(✓)	(✓)
DALIRxD0	✓	✓	✓	✓	✓	✓	✓	(✓)	(✓)	(✓)
DALITxD0	✓	✓	✓	✓	✓	✓	✓	(✓)	(✓)	(✓)
TI00	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TI01	✓	✓	✓	✓	✓	✓	✓	✓	✓	(✓)
TI02	✓	✓	✓	✓	✓	✓	✓	✓	✓	(✓)
TI03	✓	✓	✓	✓	✓	✓	✓	(✓)	(✓)	(✓)
TO00	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TO01	✓	✓	✓	✓	✓	✓	✓	✓	✓	(✓)
TO02	✓	✓	✓	✓	✓	✓	✓	✓	✓	(✓)
TO03	✓	✓	✓	✓	✓	✓	✓	(✓)	(✓)	(✓)
TRJIO0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TRJO0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TRDCLK	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
TRDIOA0	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
TRDIOA1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TRDIOB0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TRDIOB1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TRDIOC0	✓	✓	✓	✓	✓	✓	✓	✓	✓	(✓)
TRDIOC1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TRDIOD0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TRDIOD1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TRGIOA	✓	✓	✓	✓	✓	✓	✓	(✓)	(✓)	—
TRGIOB	✓	✓	✓	✓	✓	✓	✓	(✓)	(✓)	—
TRGCLKA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TRGCLKB	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TRGIDZ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TRGTRG	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TKBO00	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TKBO01	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TKBO10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TKBO11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TKBO20	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
TKBO21	✓	✓	✓	✓	✓	✓	✓	✓	✓	—

(5/5)

Function Name	64-pin	52-pin	48-pin	44-pin	40-pin	32-pin	30-pin	25-pin	24-pin	20-pin
X1	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
X2	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
EXCLK	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
XT1	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
XT2	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
EXCLKS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
V_{DD}	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EV_{DD0}	✓	—	—	—	—	—	—	—	—	—
VBAT	✓	✓	✓	✓	✓	—	—	—	—	—
AV_{REFP}	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
AV_{REFM}	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
V_{SS}	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EV_{SS0}	✓	—	—	—	—	—	—	—	—	—
TOOLRxD	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TOOLTxD	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TOOL0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Remark Functions with a check mark enclosed in parentheses in the above table are only available when the bit corresponding to a given function in the peripheral I/O redirection register x (PIORx) is set to 1.

2.2.2 Description of pin functions

(1/2)

Function Name	I/O	Function
ANI0 to ANI7, ANI16 to ANI30	Input	Analog voltage inputs for the A/D converter (see **Figure 20 - 63 Connections of VDD, AVREFP, and Analog Input Pins**)
ANO0, ANO1	Output	D/A converter outputs
IVCMP0 to IVCMP3	Input	Analog voltage inputs for the comparator
IVREF0, IVREF1	Input	Reference voltage inputs for the comparator
VCOUT0 to VCOUT3	Output	Comparator outputs
PGAI0 to PGAI4	Input	Voltage inputs for the PGA
PGAGND	Input	Reference voltage input for the PGA
PGAO	Output	Voltage output for the PGA
CCD00 to CCD07	Output	Controlled current drive port pins
INTP0 to INTP11, INTP20, INTP21	Input	External interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.
KR0 to KR7	Input	Key interrupt inputs
PCLBUZ0, PCLBUZ1	Output	Clock outputs/buzzer outputs
REGC	—	Pin for connecting a regulator output stabilization capacitor for internal operation. Connect this 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.
RTC1HZ	Output	Realtime clock correction clock (1 Hz) output
$\overline{\text{RESET}}$	Input	This is the active-low system reset input pin. When an external reset signal is not used, connect this pin directly or via a resistor to VDD.
RxD0 to RxD2	Input	Serial data input pins for serial interfaces UART0, UART1, and UART2
TxD0 to TxD2	Output	Serial data output pins for serial interfaces UART0, UART1, and UART2
SCK00, SCK01, SCK10, SCK11, SCK20, SCK21	I/O	Serial clock I/O pins for serial interfaces CSI00, CSI01, CSI10, CSI11, CSI20, and CSI21
SCL00, SCL01, SCL10, SCL11, SCL20, SCL21	Output	Serial clock output pins for serial interfaces IIC00, IIC01, IIC10, IIC11, IIC20, and IIC21
SDA00, SDA01, SDA10, SDA11, SDA20, SDA21	I/O	Serial data I/O pins for serial interfaces IIC00, IIC01, IIC10, IIC11, IIC20, and IIC21
SI00, SI01, SI10, SI11, SI20, SI21	Input	Serial data input pins for serial interfaces CSI00, CSI01, CSI10, CSI11, CSI20, and CSI21
$\overline{\text{SSI00}}$	Input	Chip select input pin for the CSI00 serial interface
SO00, SO01, SO10, SO11, SO20, SO21	Output	Serial data output pins for serial interfaces CSI00, CSI01, CSI10, CSI11, CSI20, and CSI21
SCLA0	I/O	Clock I/O pin for the IICA0 serial interface
SDAA0	I/O	Serial data I/O pin for the IICA0 serial interface
DALIRxD0	Input	DALI-2 serial data input pin
DALITxD0	Output	DALI-2 serial data output pin
TI00 to TI03	Input	Pins for inputting an external counting clock/capture trigger to 16-bit timers 00 to 03
TO00 to TO03	Output	Timer output pins for 16-bit timers 00 to 03
TRJIO0	I/O	Timer I/O pin for the RJ timer
TRJO0	Output	Timer output pin for the RJ timer
TRDCLK	Input	External clock input for the RD2 timer

(2/2)

Function Name	I/O	Function
TRDIOA0, TRDIOB0, TRDIOC0, TRDIOD0, TRDIOA1, TRDIOB1, TRDIOC1, TRDIOD1	I/O	Timer I/O pins for the RD2 timer
TRGIOA, TRGIOB	I/O	Timer I/O pins for the RG2 timer
TRGCLKA, TRGCLKB	Input	External clock input for the RG2 timer
TRGIDZ, TRGTRG	Input	External trigger input for the RG2 timer
TKBO00, TKBO01, TKBO10, TKBO11, TKBO20, TKBO21	Output	Timer output pins for 16-bit timers KB0 to KB2
X1, X2	—	Resonator connection for the main system clock
EXCLK	Input	External clock input for the main system clock
XT1, XT2	—	Resonator connection for the subsystem clock
EXCLKS	Input	External clock input for the subsystem clock
V_{DD}	—	<20-, 24-, 25-, 30-, 32-, 40-, 44-, 48-, and 52-pin products> Positive power supply for all pins <64-pin products> Positive power supply for P20 to P27, P121 to P124, P137, and pins in use for functions other than port functions
EV_{DD0}	—	Positive power supply for port pins (other than P20 to P27, P121 to P124, and P137)
VBAT	—	Power supply pin for battery backup
AV_{REFP}	Input	Positive reference voltage input of the A/D converter
AV_{REFM}	Input	Negative reference voltage input of the A/D converter
V_{SS}	—	<20-, 24-, 25-, 30-, 32-, 40-, 44-, 48-, and 52-pin products> Ground voltage for all pins <64-pin products> Ground voltage for P20 to P27, P121 to P124, P137, and pins in use for functions other than port functions
EV_{SS0}	—	Ground voltage for port pins (other than P20 to P27, P121 to P124, and P137)
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 or debugger

Caution The relationship between the voltage on P40/TOOL0 and the operating mode after release from the reset state is as follows.

P40/TOOL0	Operating Mode
EV_{DD}	Normal operating mode
0 V	Flash memory programming mode

For details, see **39.4 Programming Method**.

Remark As a measure against noise and latch up, connect a bypass capacitor (about 0.1 μF) with relatively thick wire at the shortest distance from the pins in a line from V_{DD} to V_{SS}.

2.2.3 VBAT pin

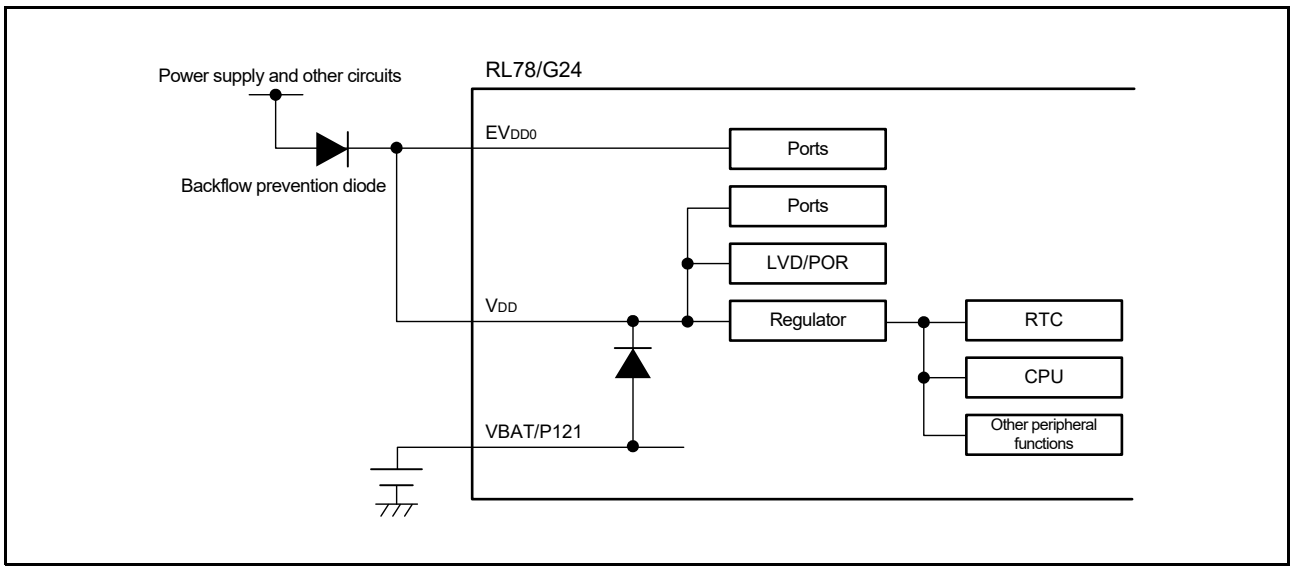
2.2.3.1 Function of the VBAT pin

The VBAT pin is used to connect the battery for use in backing up. Connecting the VBAT pin to a battery for use in backing up enables the supply of power from the VBAT pin when the power supply to the V_{DD} pin is shut off.
The main purpose of the VBAT pin is to continue operation of the realtime clock (RTC).

2.2.3.2 Connecting the VBAT pin to the battery for use in backing up

Figure 2 - 1 shows an example of the connection of the VBAT pin.
The VBAT pin supplies power to the V_{DD} pin through an internal diode. The internal diode for use with the VBAT pin is always connected to the V_{DD} pin. If preventing the backward flow of current, that is, current flowing from the VBAT pin to the power supply or other circuits that are connected to the V_{DD} pin through the diode, is required, externally connect a backflow prevention diode to the V_{DD} and EV_{DD0} pins.
The allowed range of input voltage on the VBAT pin is 2.7 V to 5.5 V. The input voltage on the VBAT pin falling below 2.7 V while power is being supplied from the VBAT pin may lead to the generation of a POR reset due to the fall in the voltage across the diode. The maximum current that can be supplied through the VBAT pin is 150 μA.

Figure 2 - 1 Example of the Connection of the VBAT Pin

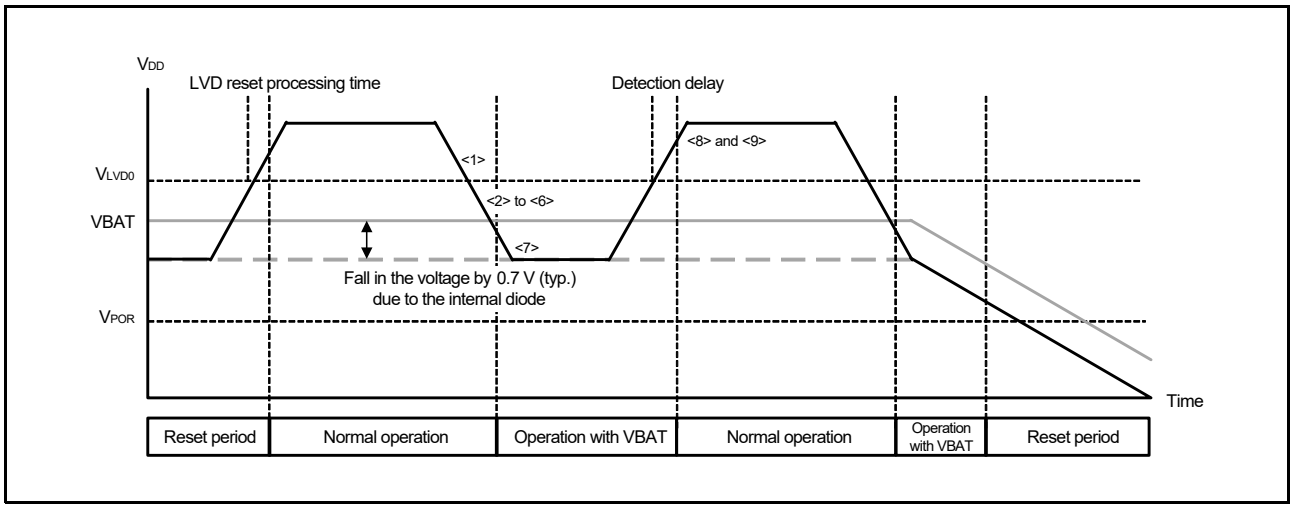


2.2.3.3 Using the VBAT pin

How to make the initial settings for the VBAT pin and an example of the procedure for switching the power supply pin to the VBAT pin are described below. This processing is to be completed before the voltage on the V_{DD} pin falls below that supplied from the VBAT pin.
In addition, **Figure 2 - 2** shows the state transitions in switching the power supply pin between the V_{DD} and VBAT pins.

1. Making the initial settings for the VBAT pin
   Set the P121 pin to X1 oscillation mode (by setting the EXCLK and OSCSEL bits of the CMC register to 0 and 1, respectively, and the MSTOP bit of the CSC register to 0) in the initial settings.
2. Example of the procedure for switching the power supply pin to the VBAT pin
   <1> Use LVD0 in the interrupt mode to generate an interrupt request when the power supply voltage (V_{DD}) falls. The switching process starts in response to this interrupt request signal.
   <2> Disable interrupts other than LVD0.
   <3> Stop the operation of all peripheral functions other than the realtime clock (RTC).
   <4> Change the output settings for port pins so that no current flows through them.
   <5> Clear the interrupt request flag of LVD0.
   <6> After having confirmed that the value of the LVD0F bit is 1 (V_{DD} < detection voltage), if the CPU is operating with the main system clock, place this LSI chip in the STOP mode. If the CPU is operating with the subsystem clock, place the chip in the HALT mode.
   <7> Keep the chip in the above state until LVD0 generates an interrupt request.
   <8> Supplying the power supply voltage (V_{DD}) back to the chip makes LVD0 generate an interrupt request, thereby releasing the chip from the STOP or HALT mode.
   <9> After having confirmed that the value of the LVD0F bit is 0 (V_{DD} ≥ detection voltage), make the settings to return the peripheral functions to normal operation while the power supply voltage (V_{DD}) is being supplied.

Figure 2 - 2 State Transitions in Switching the Power Supply Pin between the V_{DD} and VBAT Pins



When switching between the V_{DD} and VBAT pins is too frequent, waiting for the fluctuations in voltage to settle between steps <4> and <5> in the procedure can prevent excessive switching between the V_{DD} and VBAT pins.

Caution 1. Operation with the main system clock is prohibited while the battery is supplying power.
Caution 2. Make the setting to stop counting by the WDT when the battery is to supply power.
Moreover, input of the low level on the reset pin is prohibited. This is because attempting to do so leads to starting operation with the main system clock after release from the reset state, which requires current exceeding 150 µA, the maximum allowable current through the VBAT pin.
Caution 3. When the battery is to be used for the supply of power, set the P121 pin to the X1 oscillation mode; that is, do not set the P121 pin to the input or output mode.

2.3 Connection of Unused Pins

Table 2 - 2 shows the connections of unused pins.

Remark The pins mounted depend on the product. Refer to **1.3 Pin Configuration (Top View)** and **2.1 Functions of Port Pins**.

Table 2 - 2 Connections of Unused Pins

Pin Name	I/O	Recommended Connection of Unused Pins
P00 to P06	I/O	Input: Independently connect the pins to EV_{DD0} or EV_{SS0} via resistors. Output: Leave the pins open-circuit.
P10 to P17		
P20 to P27		Input: Independently connect the pins to V_{DD} or V_{SS} via resistors. Output: Leave the pins open-circuit.
P30, P31		Input: Independently connect the pins to EV_{DD0} or EV_{SS0} via resistors. Output: Leave the pins open-circuit.
P40/TOOL0		Input: Independently connect the pin to EV_{DD0} via a resistor or leave the pin open-circuit. Output: Leave the pin open-circuit.
P41 to P43		Input: Independently connect the pins to EV_{DD0} or EV_{SS0} via resistors. Output: Leave the pins open-circuit.
P50 to P55		
P60 to P63		Input: Independently connect the pins to EV_{DD0} or EV_{SS0} via resistors. Output: Set the port's output latch to 0 and leave the pins open-circuit, or set the port's output latch to 1 and independently connect the pins to EV_{DD0} or EV_{SS0} via resistors.
P70 to P77		Input: Independently connect the pins to EV_{DD0} or EV_{SS0} via resistors. Output: Leave the pins open-circuit.
P120		
P121, P122		Input: Independently connect the pins to V_{DD} or V_{SS} via resistors. Output: Leave the pins open-circuit.
P123, P124	Input	Set the EXCLKS bit to 0 and the OSCSELS bit to 1 in the clock operation mode control register (CMC), set the XTSTOP bit in the clock operation status control register (CSC) to 1, and leave the pins open-circuit[Note]. Alternatively, provide the pins with independent connections to V_{DD} or V_{SS} via resistors.
P130	Output	Leave the pin open-circuit.
P137	Input	Set the PDIDIS137 bit in port digital input disable register 13 (PDIDIS13) to 1, and leave the pin open-circuit. Alternatively, provide the pin with an independent connection to V_{DD} or V_{SS} via a resistor.
P140, P141, P146, P147	I/O	Input: Independently connect the pins to EV_{DD0} or EV_{SS0} via resistors. Output: Leave the pins open-circuit.
$\overline{\text{RESET}}$	Input	Connect the pin directly or via a resistor to V_{DD}.
REGC	—	Connect the pin to V_{SS} via a capacitor (0.47 to 1 μF).

Note When the low-speed on-chip oscillator clock (f_{IL}) is selected for the CPU/peripheral hardware clock frequency (f_{CLK}), the current may increase approximately by 1 μA.

Remark For products that do not have an EV_{DD0} or EV_{SS0}, read EV_{DD0} as V_{DD}, and EV_{SS0} as V_{SS}.

2.4 Block Diagrams of Pins

Figures 2 - 3 to **2 - 28** show the block diagrams of the pins described in **2.1.1 20-pin products** to **2.1.10 64-pin products**.

Figure 2 - 3 Pin Block Diagram for Pin Type 1-1-1

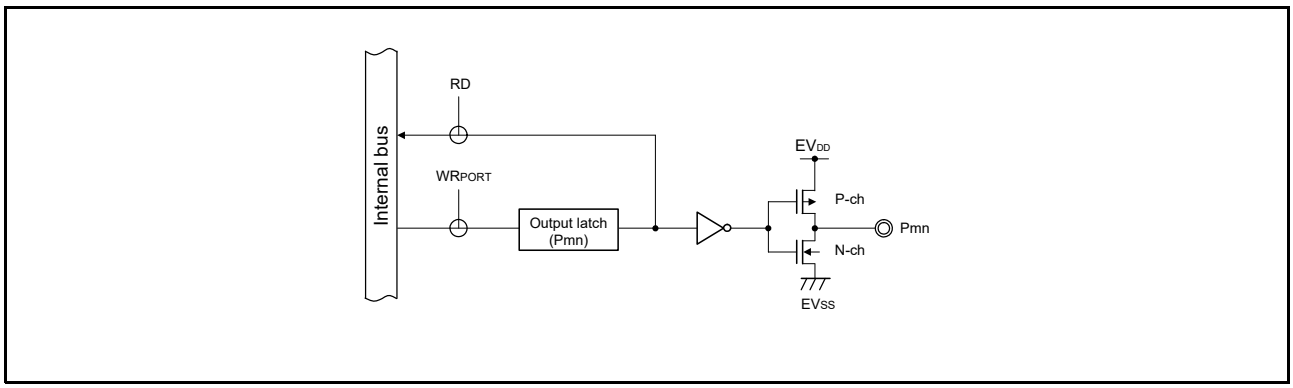


Figure 2 - 4 Pin Block Diagram for Pin Type 2-1-1

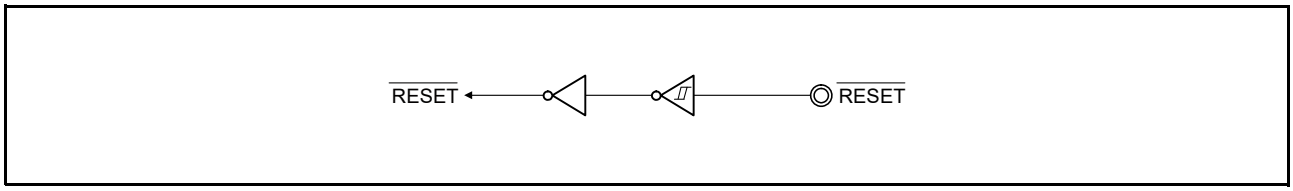
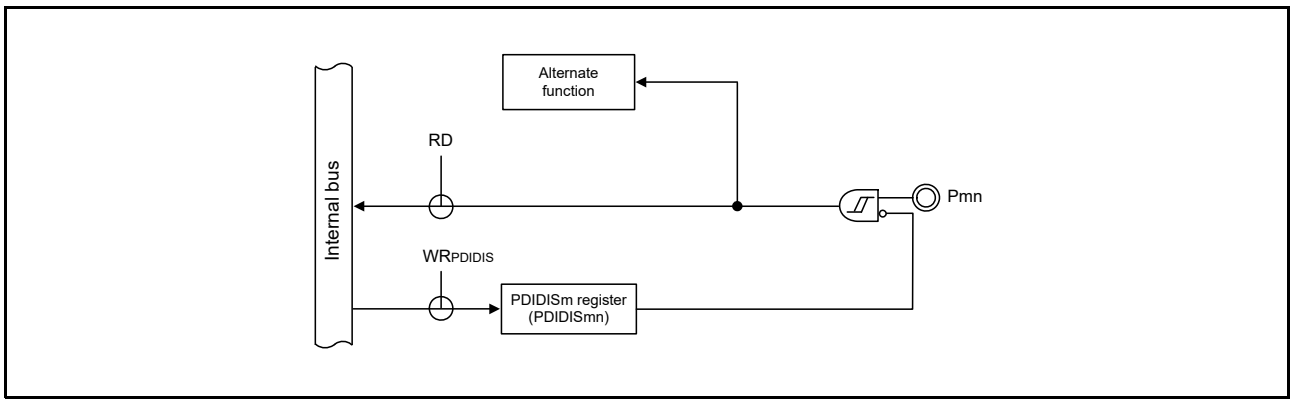
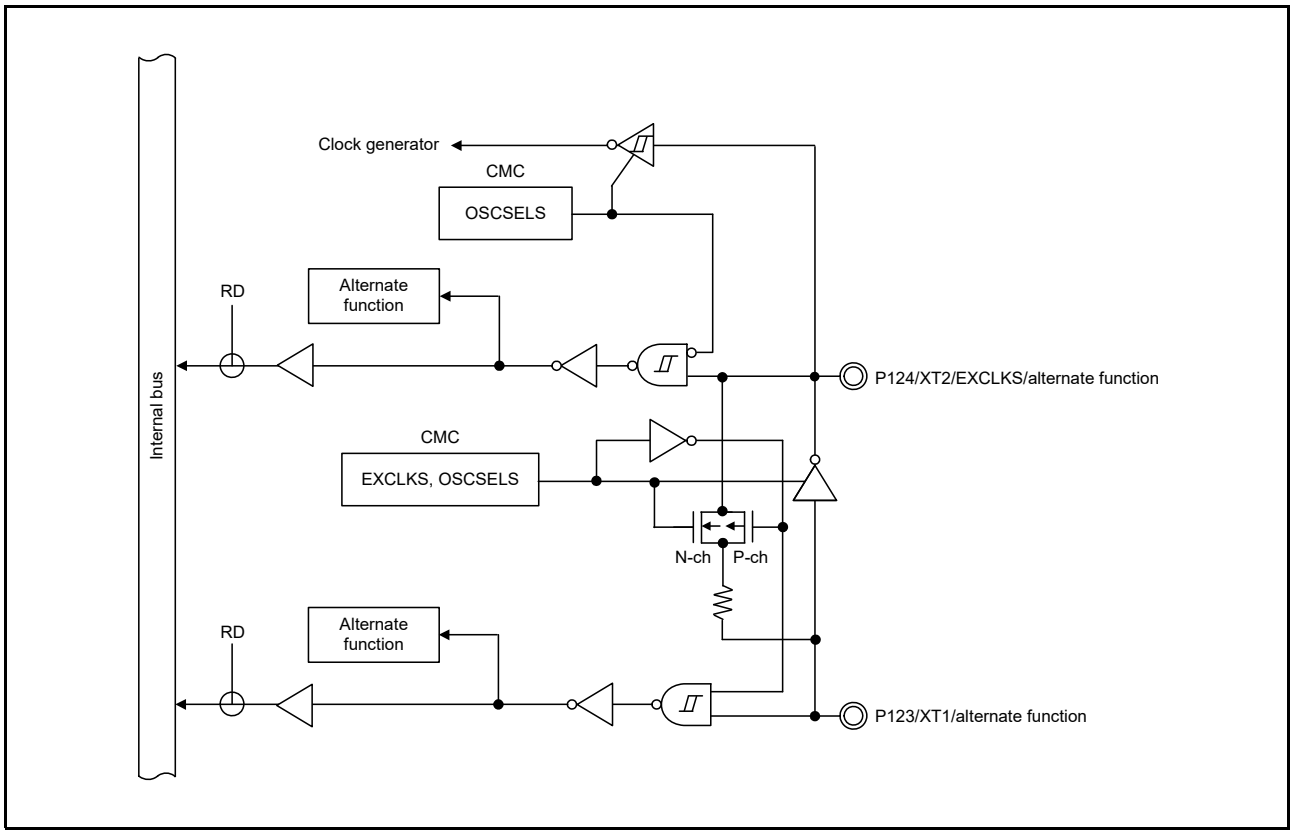


Figure 2 - 5 Pin Block Diagram for Pin Type 2-1-3



Remark For alternate functions, see **2.1 Functions of Port Pins**.

Figure 2 - 6 Pin Block Diagram for Pin Type 2-2-1



Remark For alternate functions, see **2.1 Functions of Port Pins**.

Figure 2 - 7 Pin Block Diagram for Pin Type 4-3-5

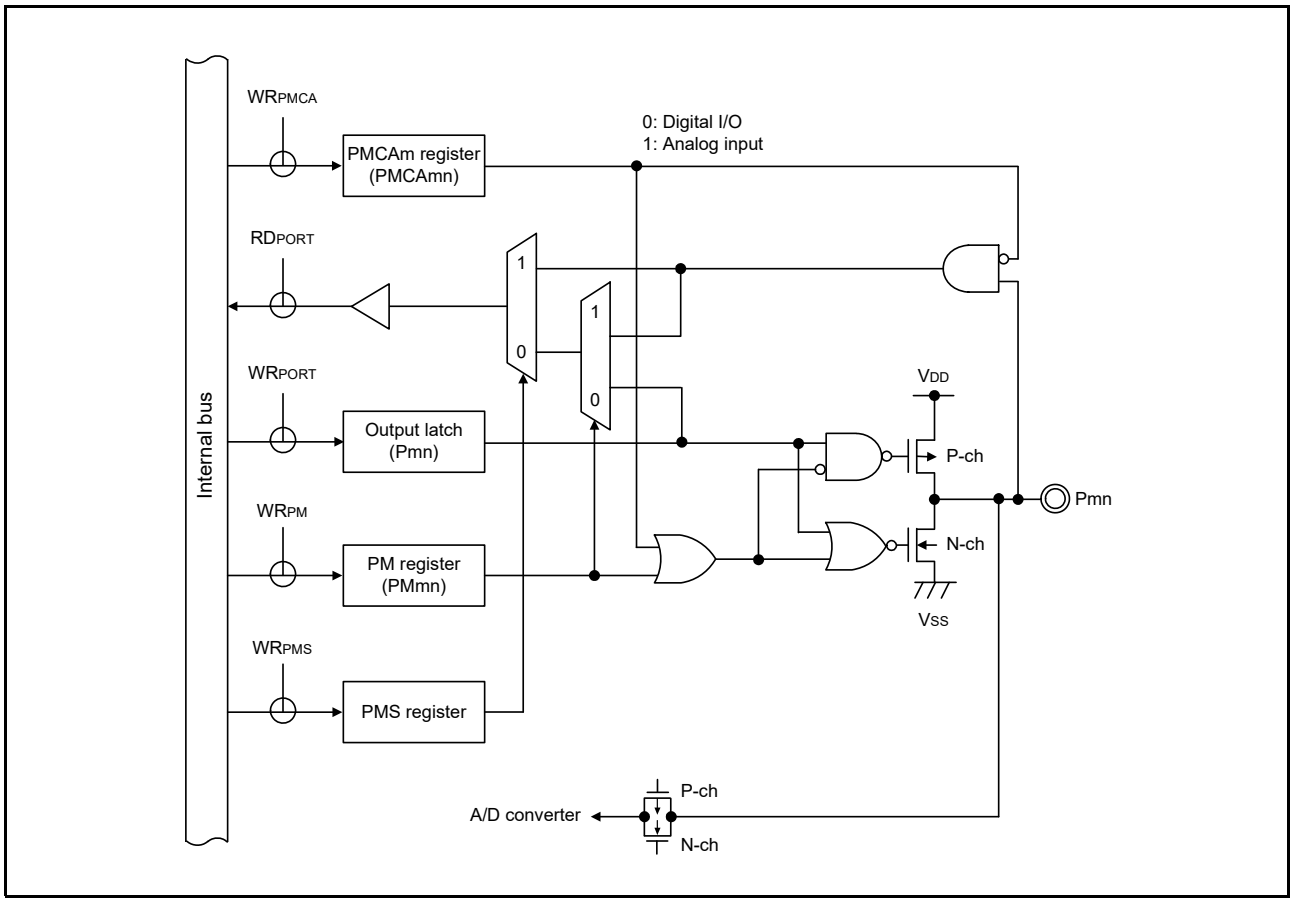


Figure 2 - 8 Pin Block Diagram for Pin Type 4-8-2

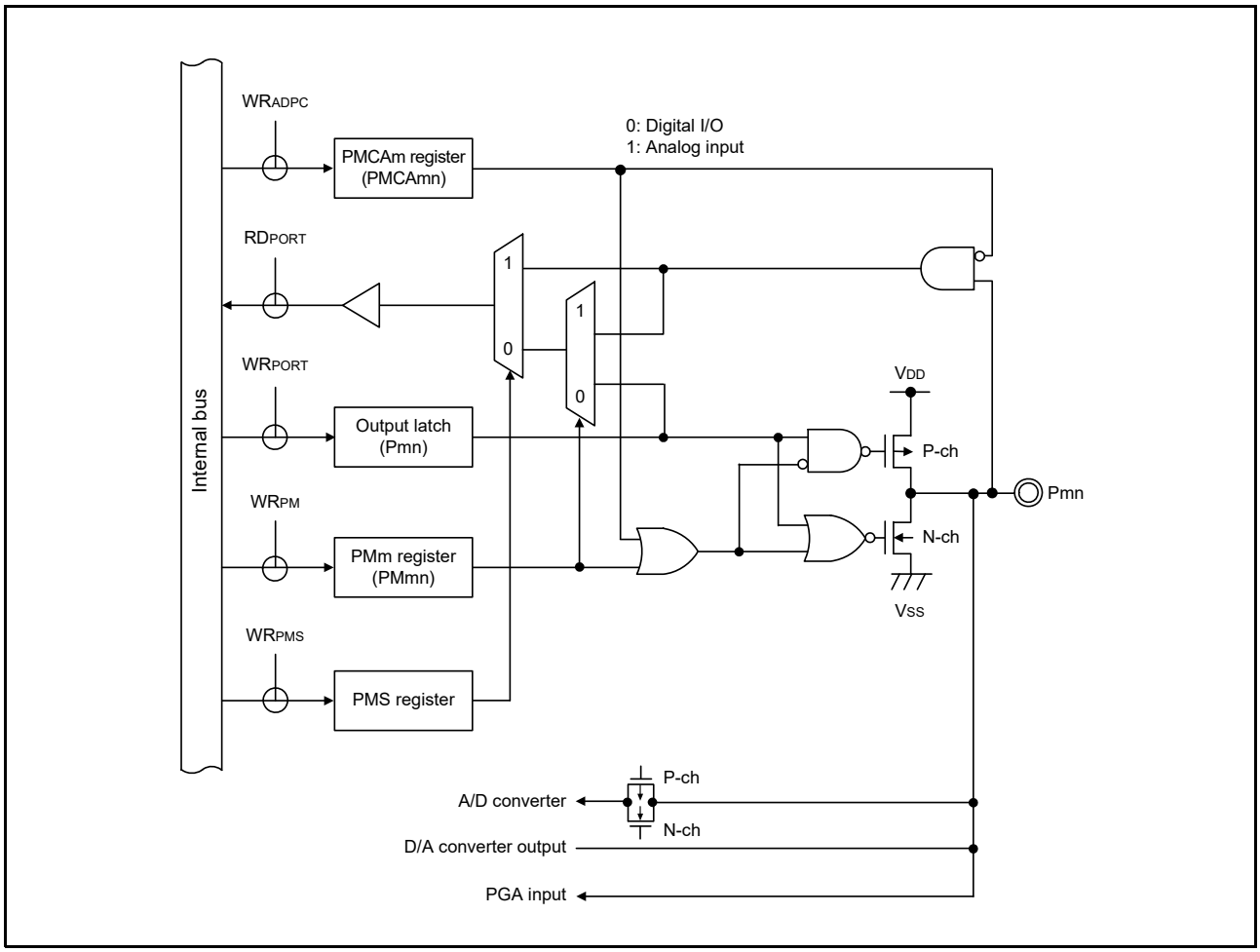
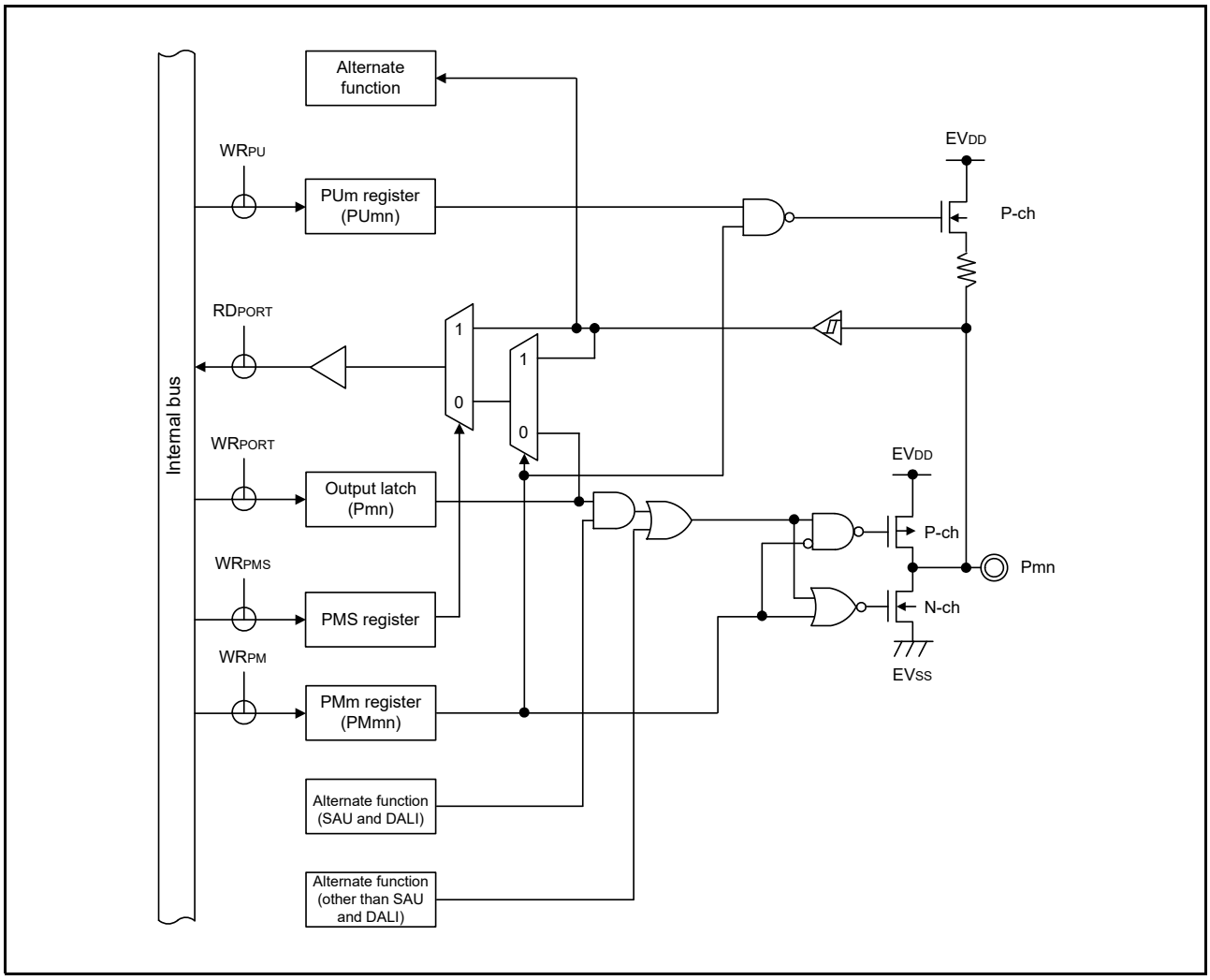


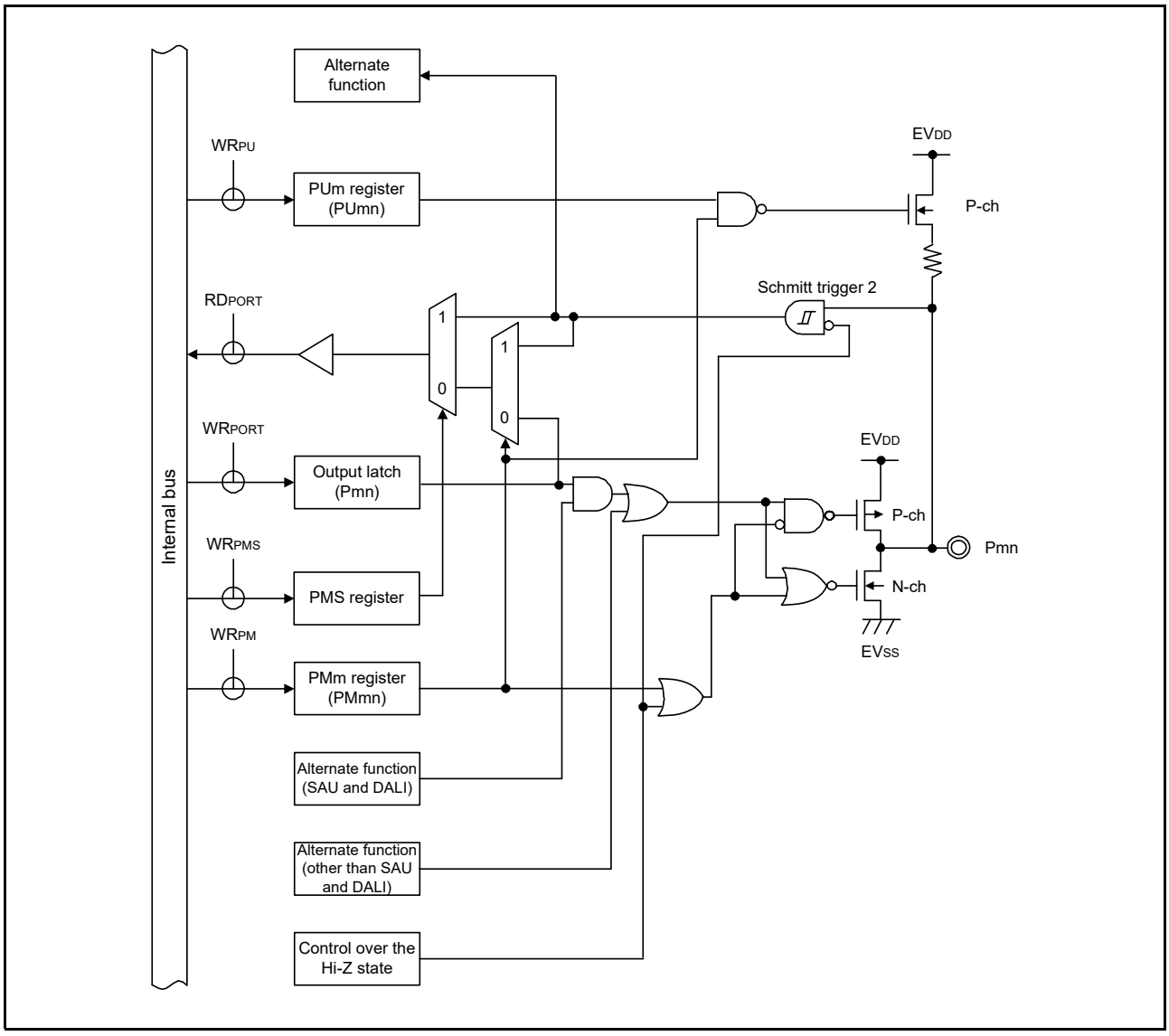
Figure 2 - 9 Pin Block Diagram for Pin Type 7-1-3



Remark 1. For alternate functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit, DALI: Digital addressable lighting interface

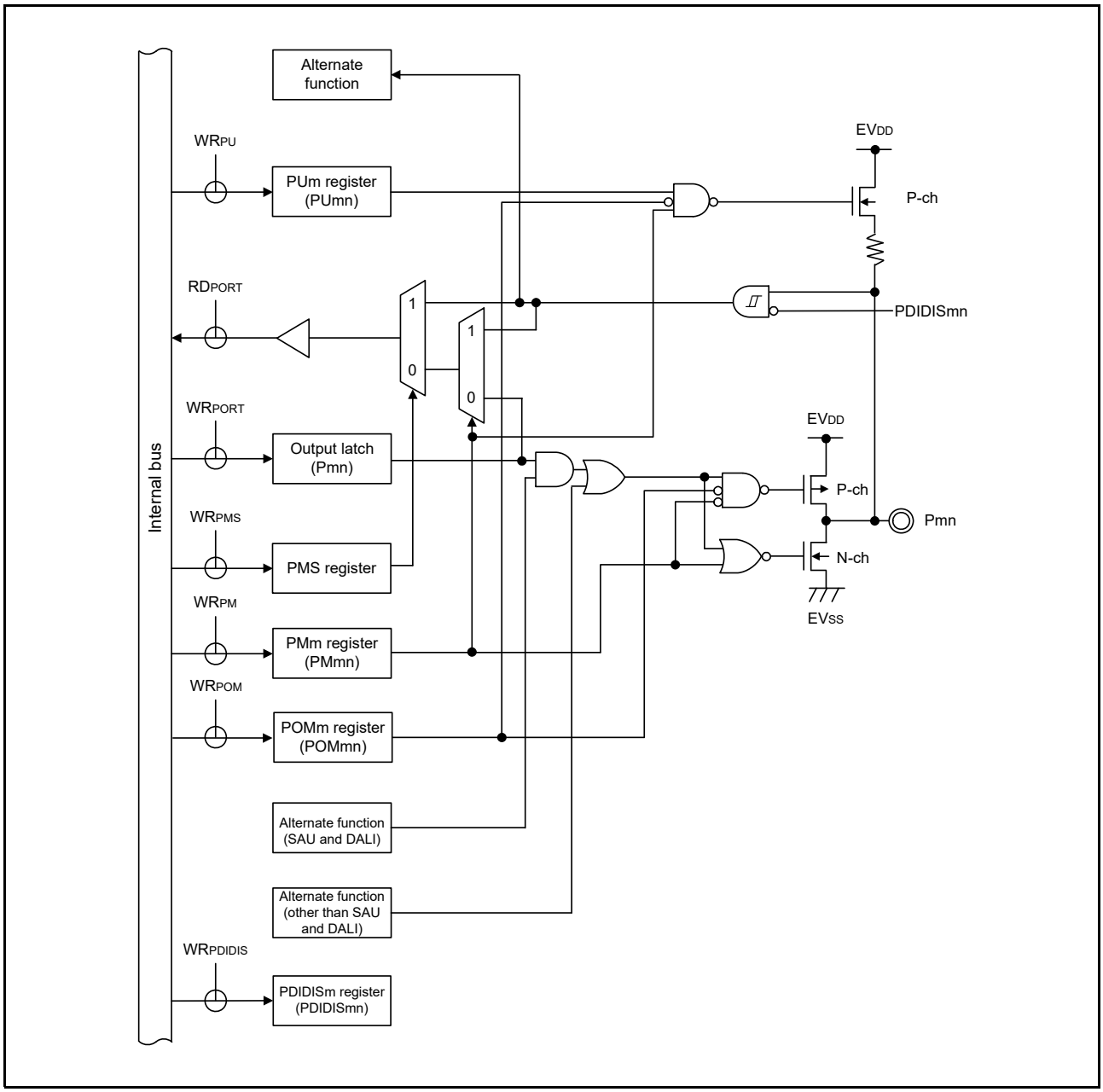
Figure 2 - 10 Pin Block Diagram for Pin Type 7-1-7



Remark 1. For alternate functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit, DALI: Digital addressable lighting interface

Figure 2 - 11 Pin Block Diagram for Pin Type 7-1-11

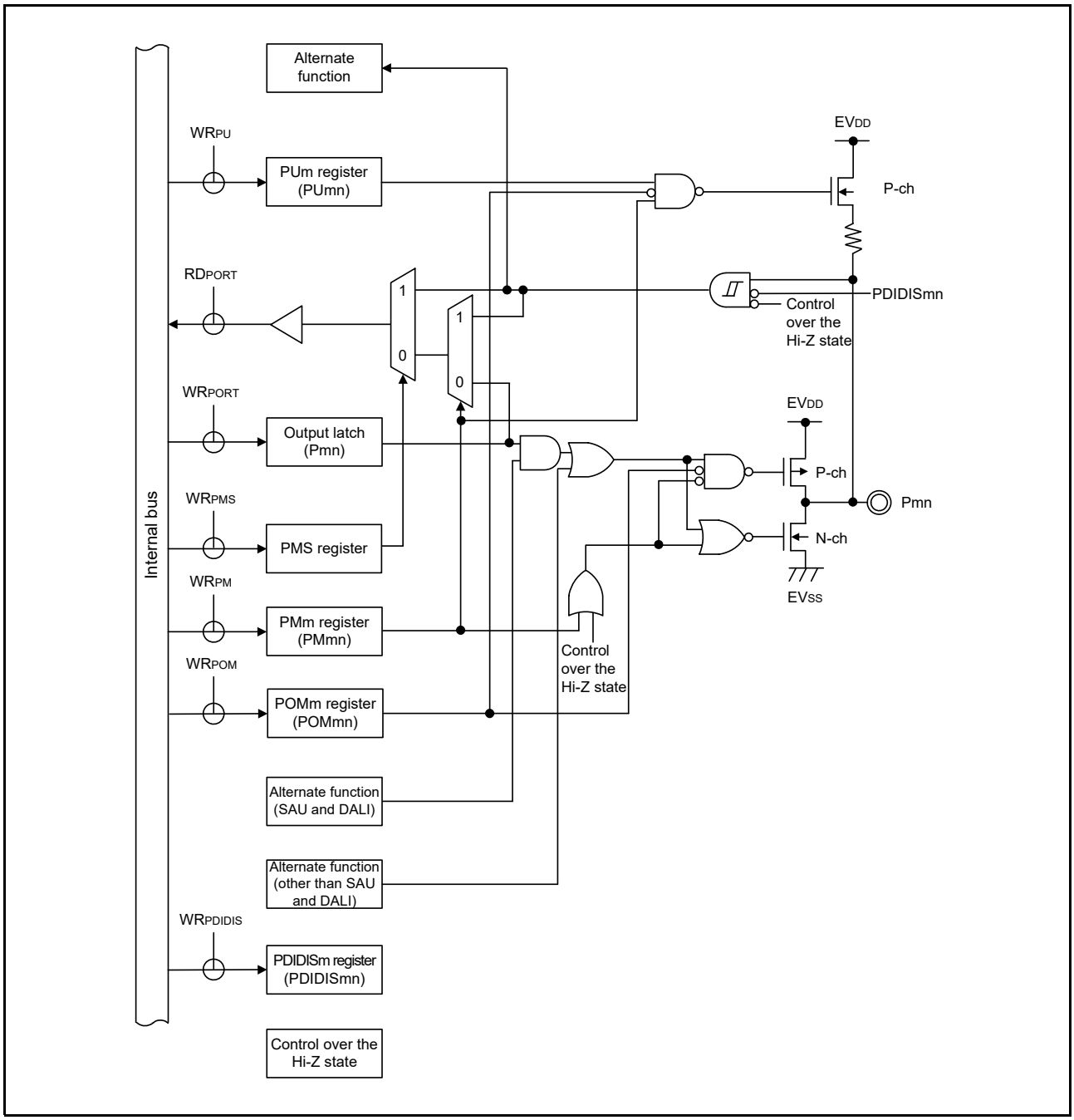


Caution The input buffer is enabled even if the type 7-1-11 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register m (POMm). This may lead to a through current flowing through the type 7-1-11 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.

Remark 1. For alternate functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit, DALI: Digital addressable lighting interface

Figure 2 - 12 Pin Block Diagram for Pin Type 7-1-13

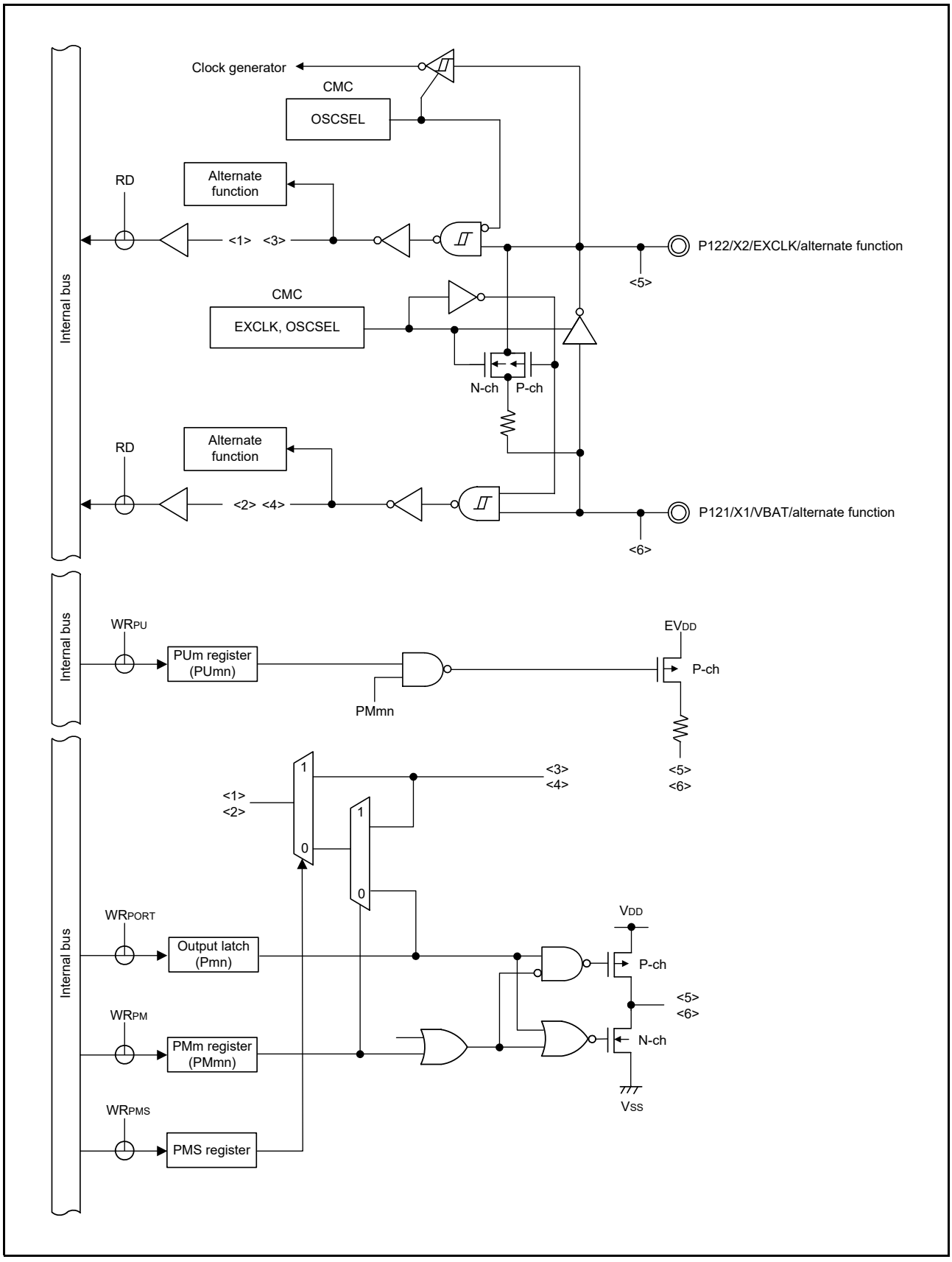


Caution **The input buffer is enabled even if the type 7-1-13 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register m (POMm). This may lead to a through current flowing through the type 7-1-13 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.**

Remark 1. For alternate functions, see **2.1 Functions of Port Pins**.

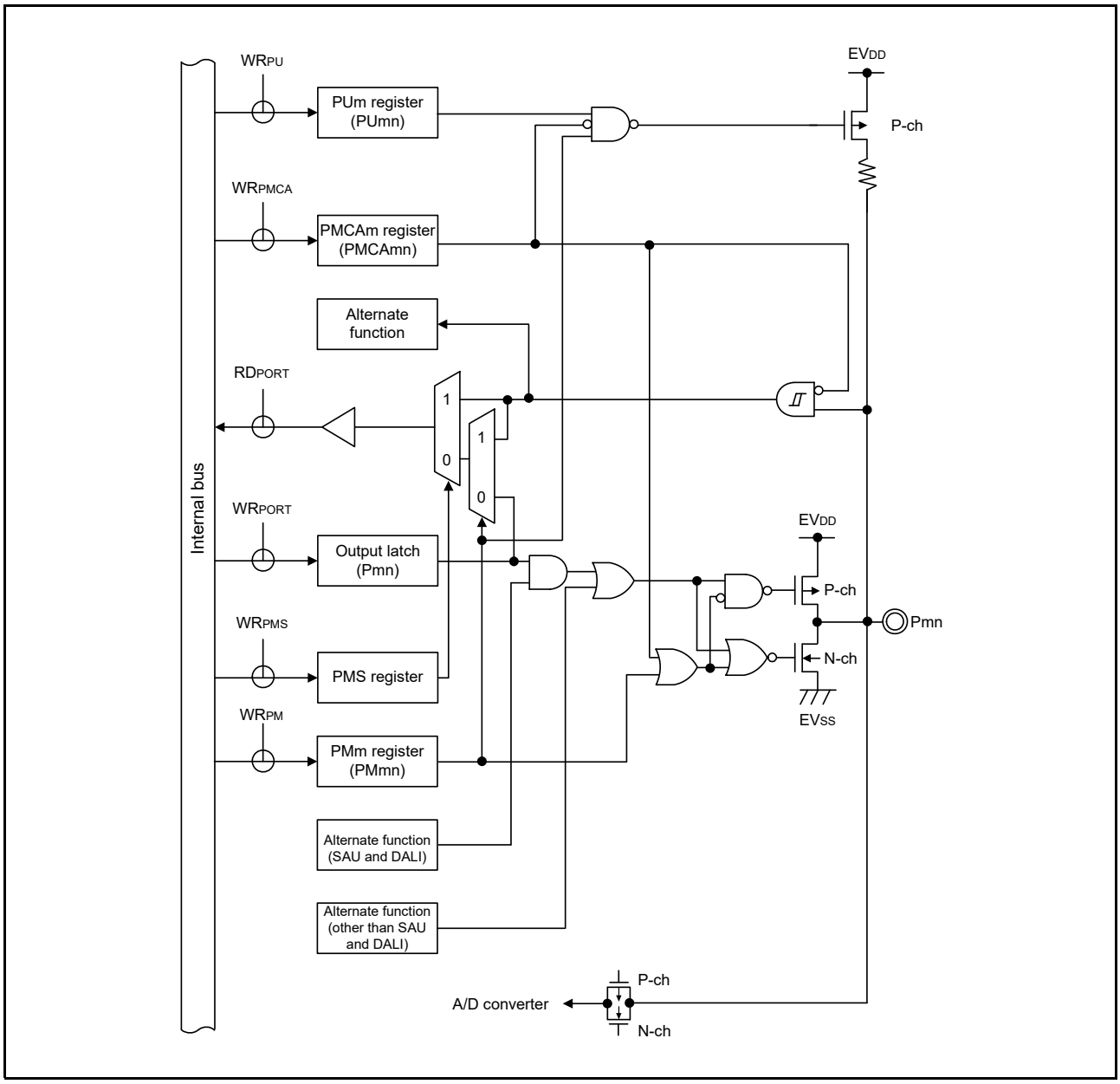
Remark 2. SAU: Serial array unit, DALI: Digital addressable lighting interface

Figure 2 - 13 Pin Block Diagram for Pin Type 7-2-1



Remark For alternate functions, see **2.1 Functions of Port Pins**.

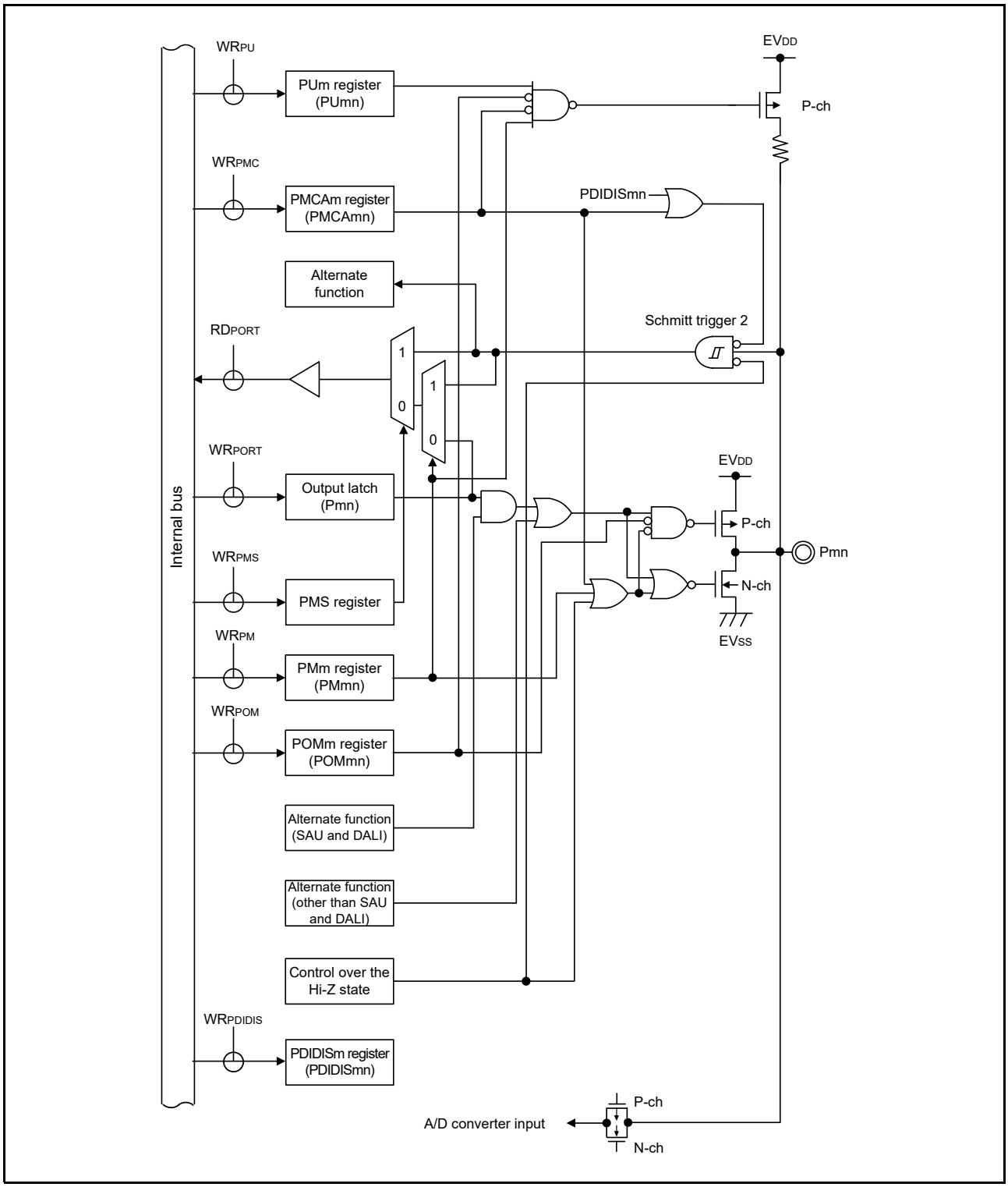
Figure 2 - 14 Pin Block Diagram for Pin Type 7-3-3



Remark 1. For alternate functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit, DALI: Digital addressable lighting interface

Figure 2 - 15 Pin Block Diagram for Pin Type 7-8-1

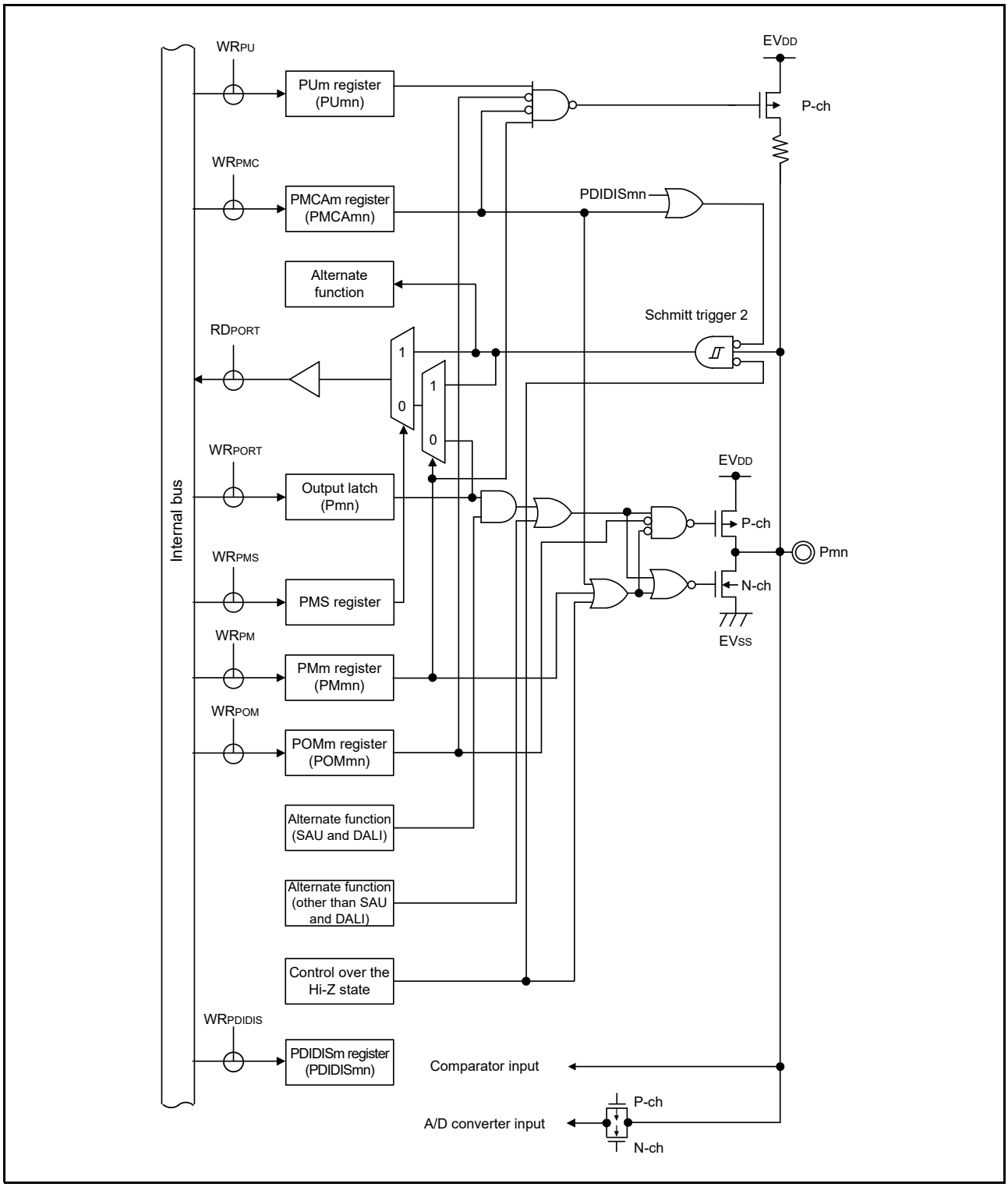


Caution The input buffer is enabled even if the type 7-8-1 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register m (POMm). This may lead to a through current flowing through the type 7-8-1 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.

Remark 1. For alternate functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit, DALI: Digital addressable lighting interface

Figure 2 - 16 Pin Block Diagram for Pin Type 7-8-2

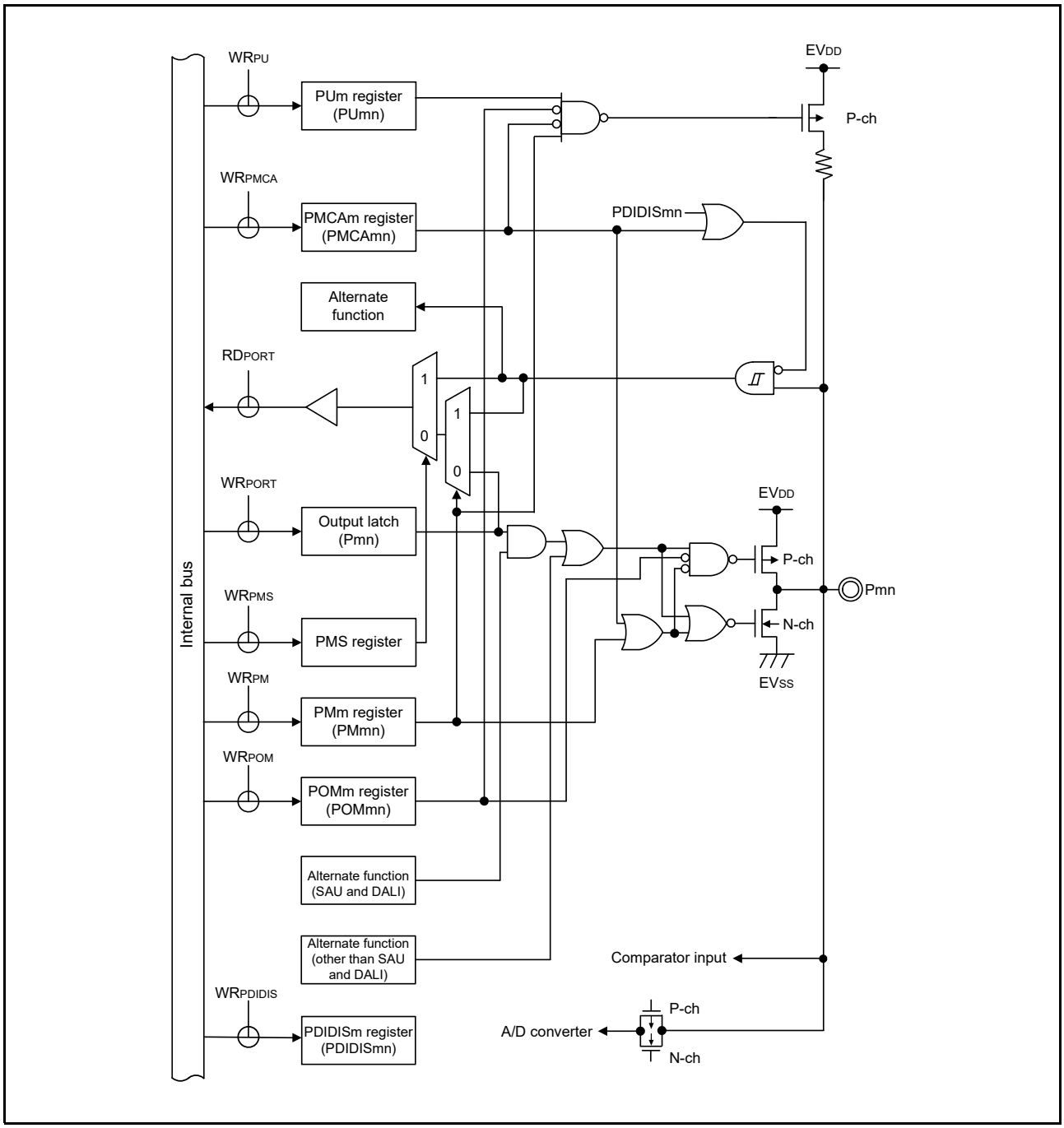


Caution **The input buffer is enabled even if the type 7-8-2 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register m (POMm). This may lead to a through current flowing through the type 7-8-2 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.**

Remark 1. For alternate functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit, DALI: Digital addressable lighting interface

Figure 2 - 17 Pin Block Diagram for Pin Type 7-9-6

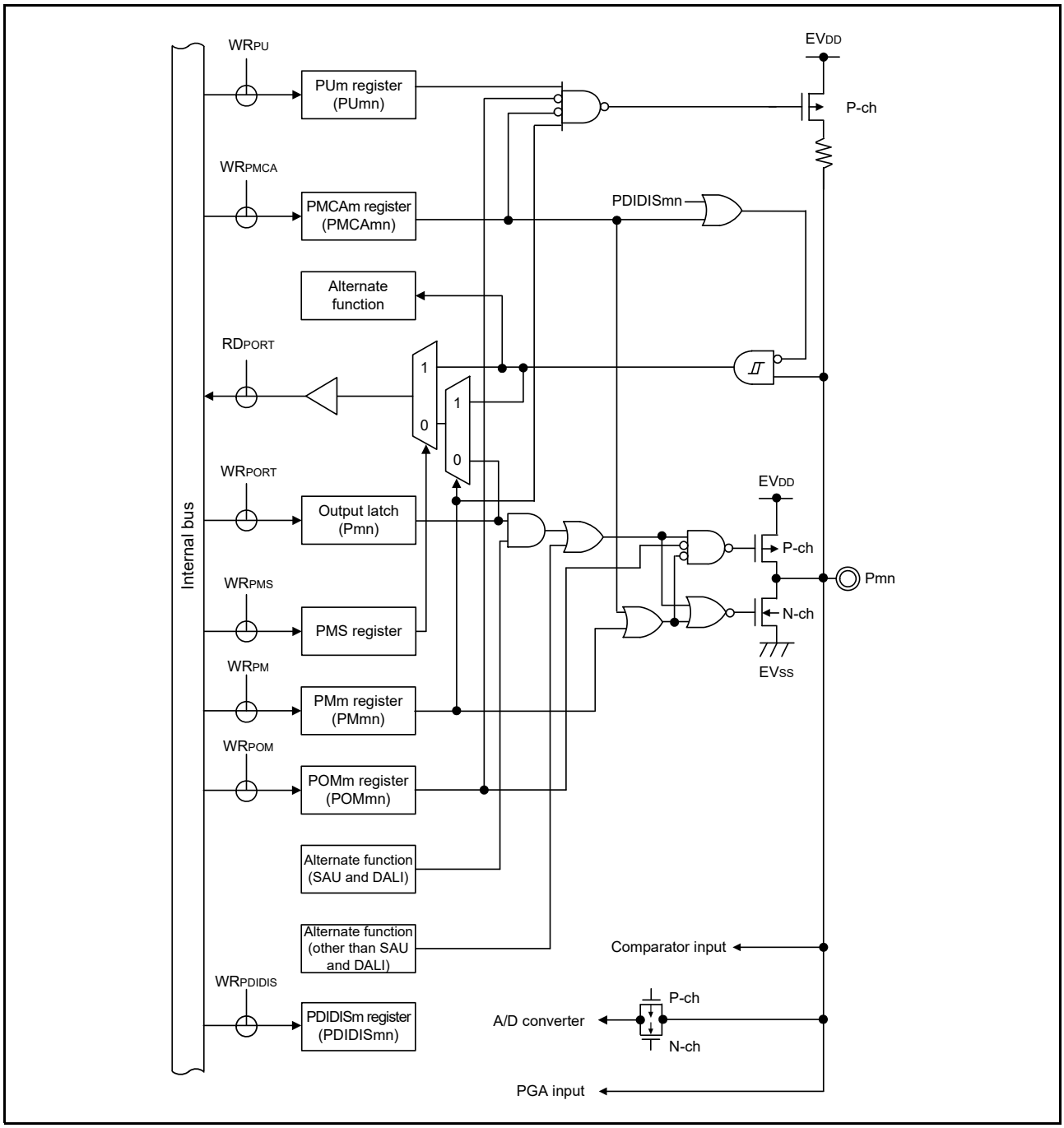


Caution The input buffer is enabled even if the type 7-9-6 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register m (POMm). This may lead to a through current flowing through the type 7-9-6 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.

Remark 1. For alternate functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit, DALI: Digital addressable lighting interface

Figure 2 - 18 Pin Block Diagram for Pin Type 7-18-1

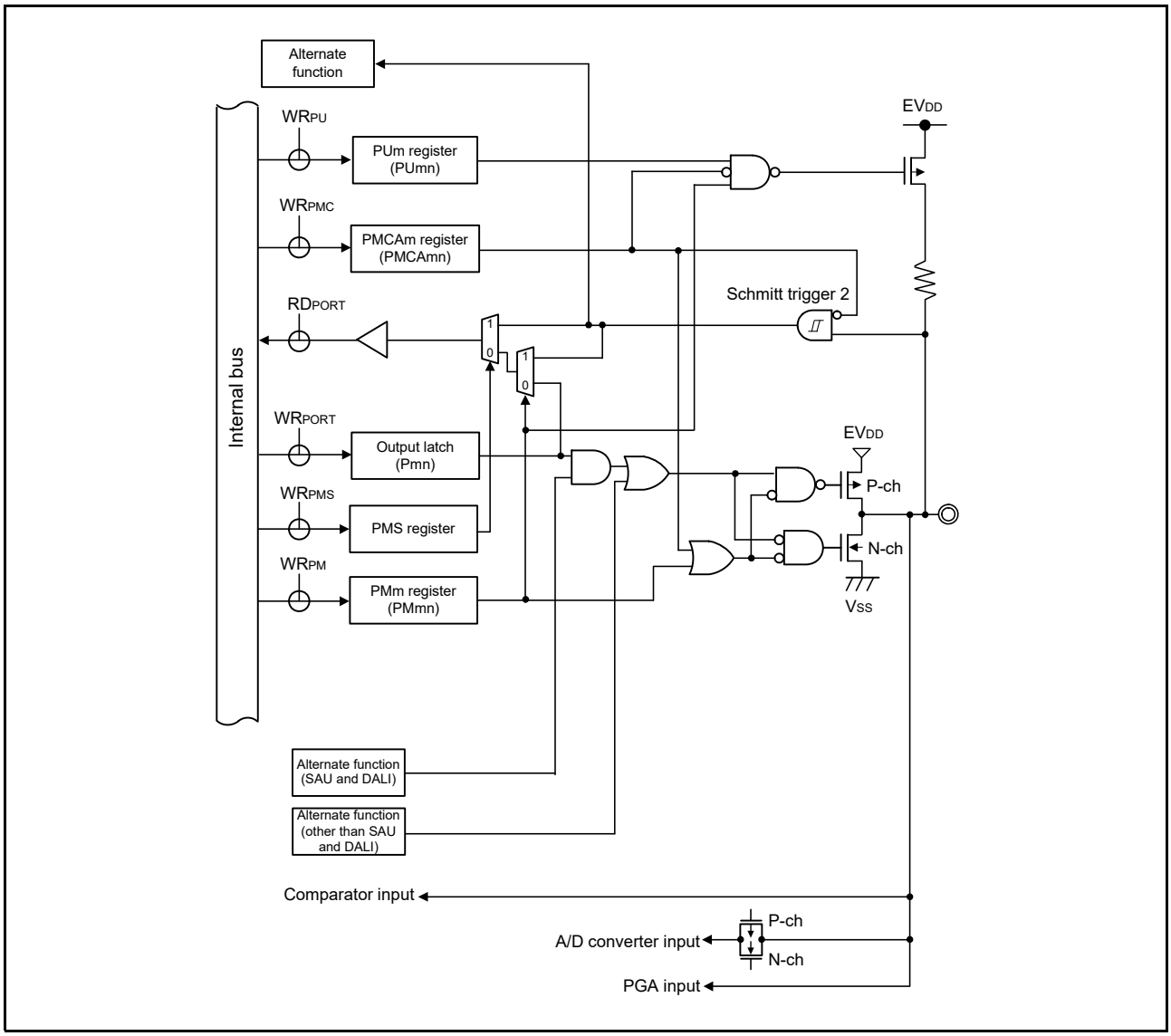


Caution **The input buffer is enabled even if the type 7-18-1 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register m (POMm). This may lead to a through current flowing through the type 7-18-1 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.**

Remark 1. For alternate functions, see **2.1 Functions of Port Pins**.

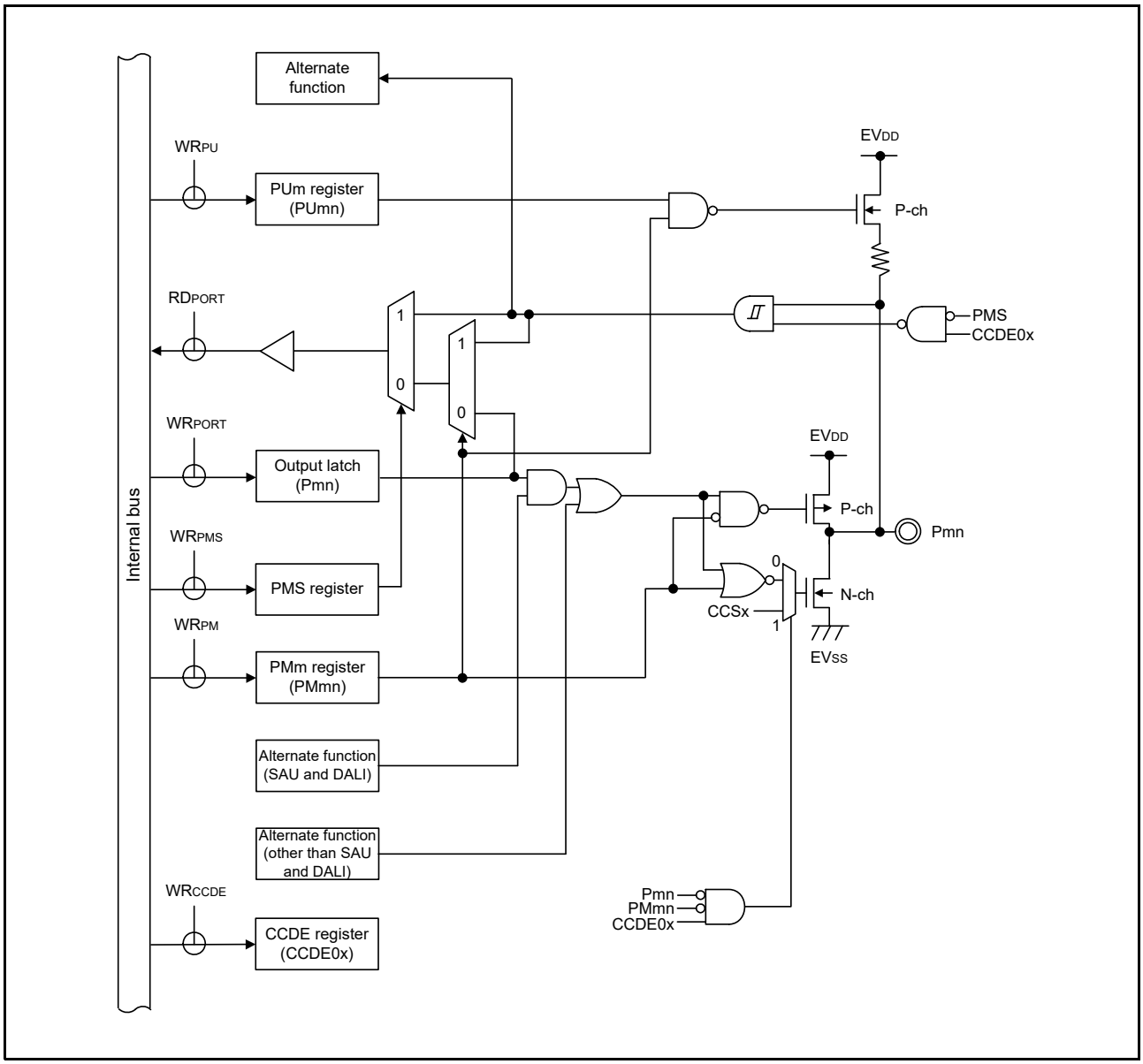
Remark 2. SAU: Serial array unit, DALI: Digital addressable lighting interface

Figure 2 - 19 Pin Block Diagram for Pin Type 7-18-2



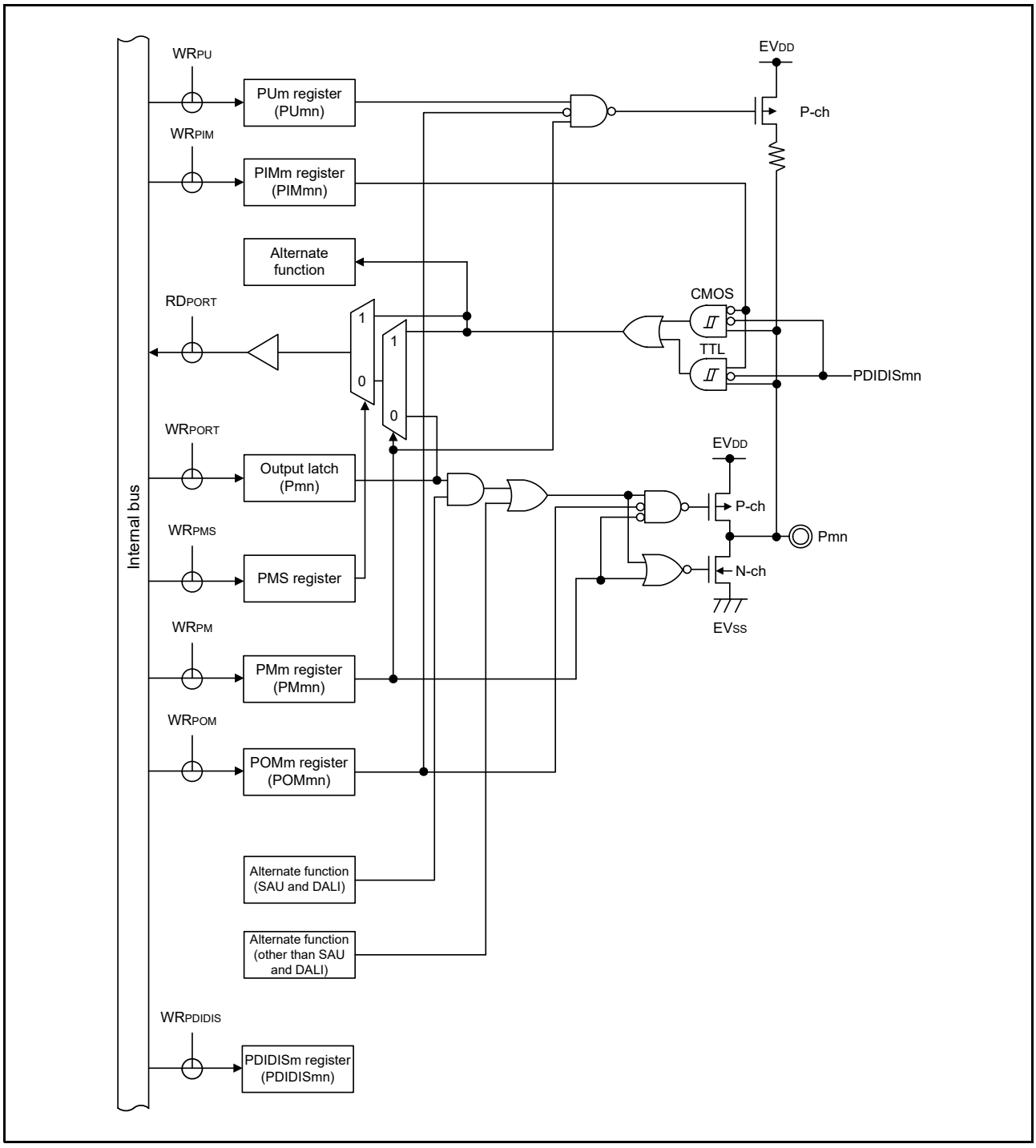
Remark SAU: Serial array unit, DALI: Digital addressable lighting interface

Figure 2 - 20 Pin Block Diagram for Pin Type 7-38-3



Remark SAU: Serial array unit, DALI: Digital addressable lighting interface

Figure 2 - 21 Pin Block Diagram for Pin Type 8-1-10



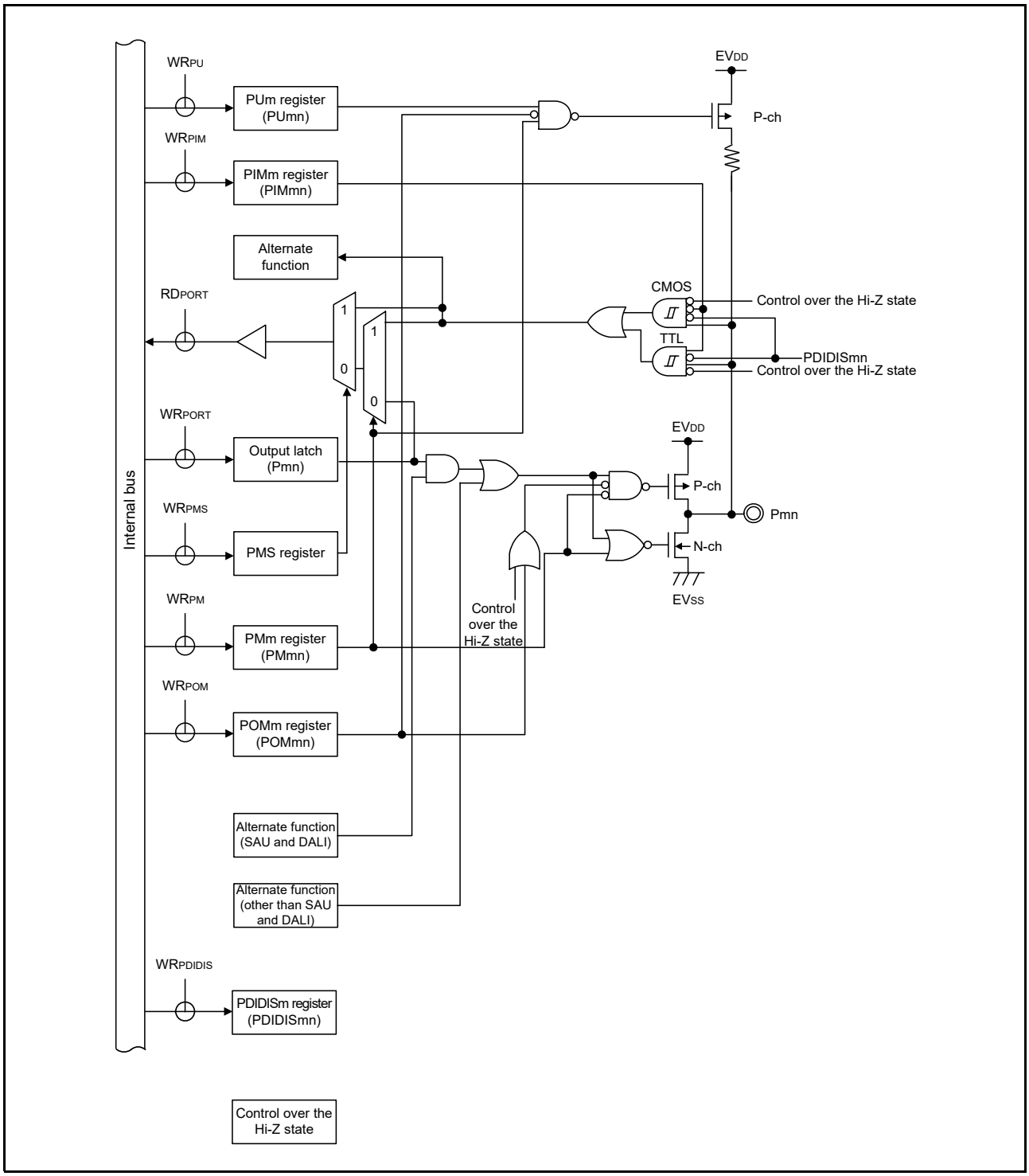
Caution 1. **The input buffer is enabled even if the type 8-1-10 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register m (POMm). This may lead to a through current flowing through the type 8-1-10 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.**

Caution 2. **When the type 8-1-10 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMx) and is driven high, a through current may flow through the type 8-1-10 pin due to the configuration of the TTL input buffer. Drive the type 8-1-10 pin low to prevent the through current.**

Remark 1. For alternate functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit, DALI: Digital addressable lighting interface

Figure 2 - 22 Pin Block Diagram for Pin Type 8-1-13

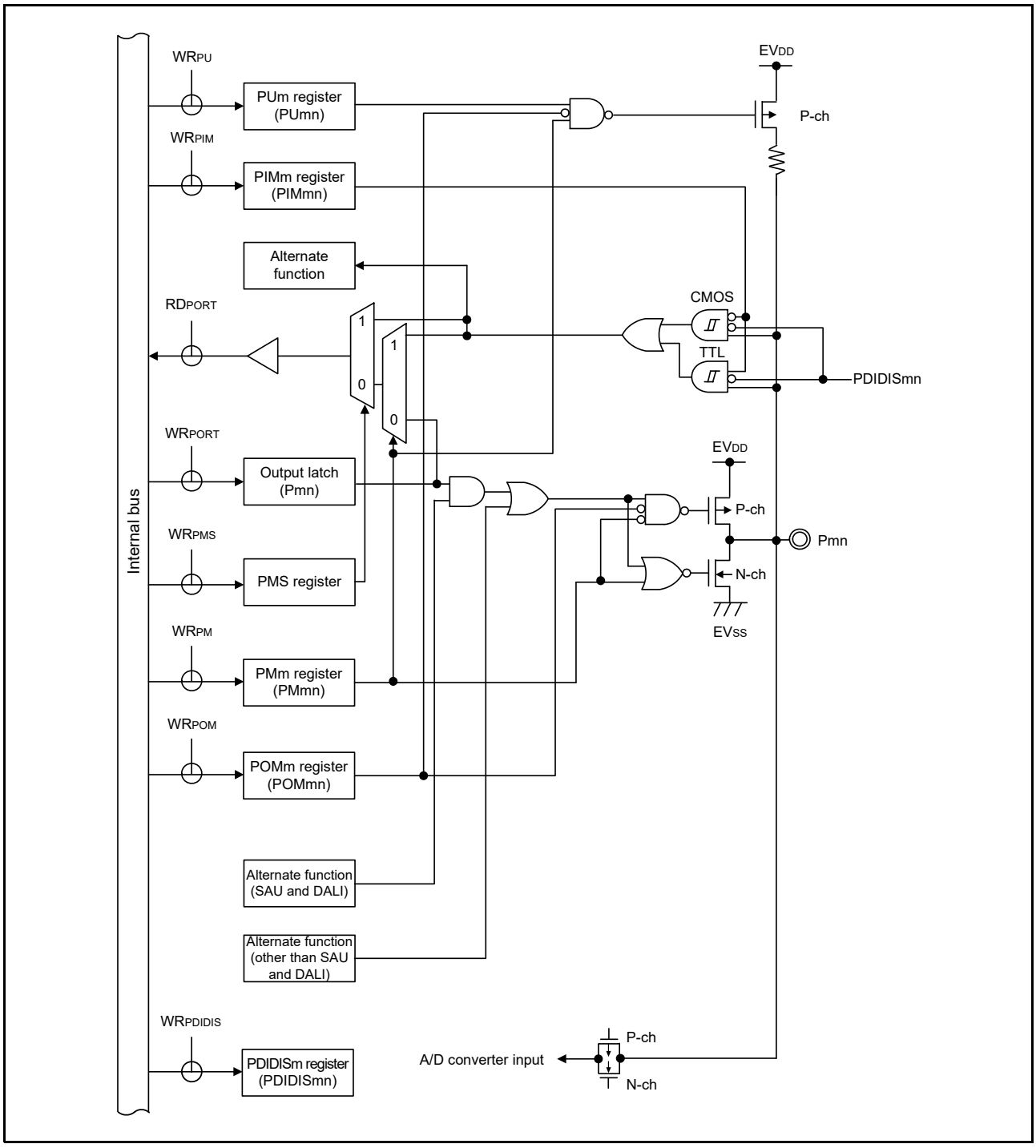


Caution **The input buffer is enabled even if the type 8-1-13 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register m (POMm). This may lead to a through current flowing through the type 8-1-13 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.**

Remark 1. For alternate functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit, DALI: Digital addressable lighting interface

Figure 2 - 23 Pin Block Diagram for Pin Type 8-3-6



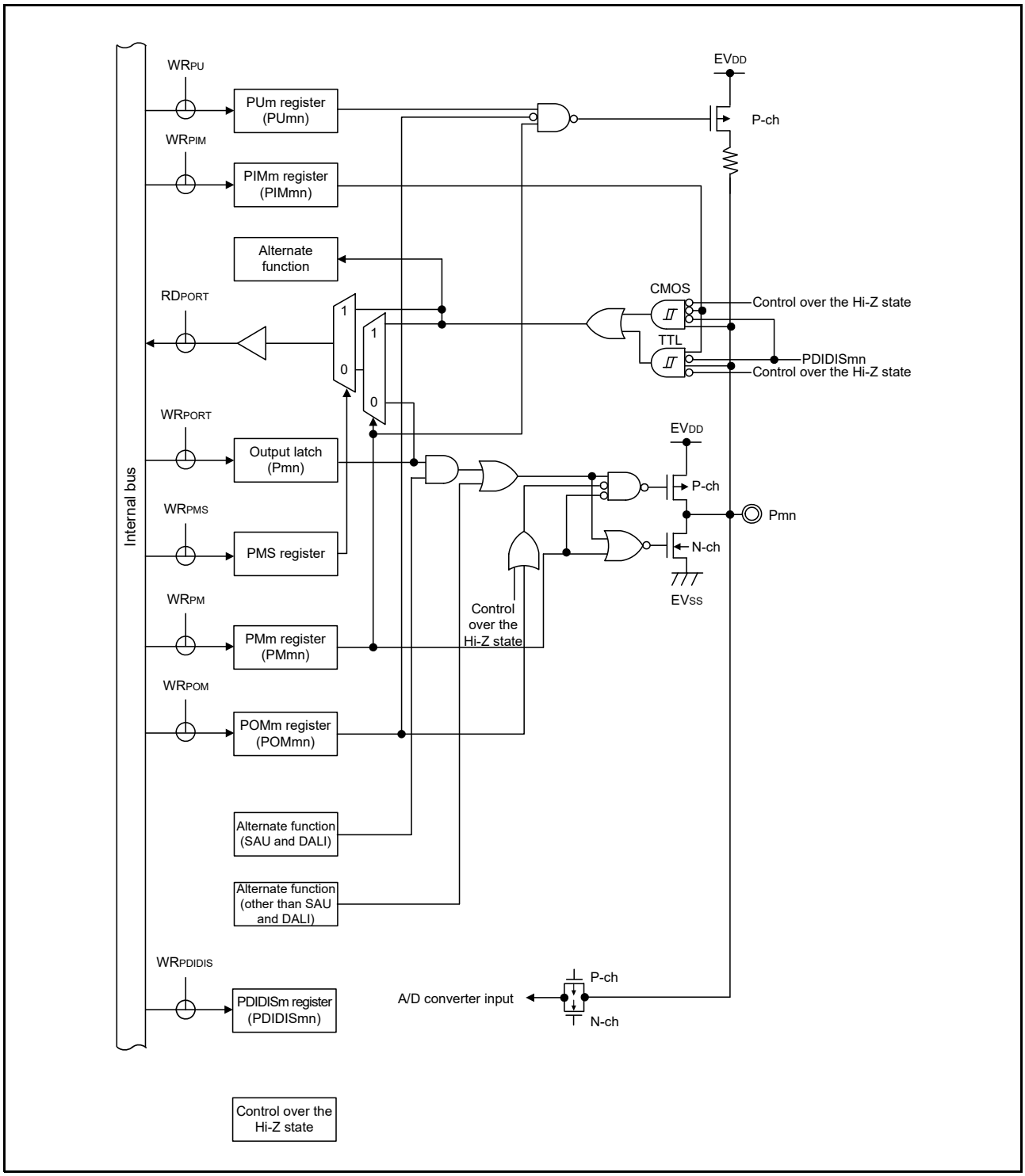
Caution 1. The input buffer is enabled even if the type 8-3-6 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register m (POMm). This may lead to a through current flowing through the type 8-3-6 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.

Caution 2. When the type 8-3-6 pin is set to TTL input buffer by the corresponding bit in the port input mode register m (PIMm) and is driven high, a through current may flow through the type 8-3-6 pin due to the configuration of the TTL input buffer. Drive the type 8-3-6 pin low to prevent the through current.

Remark 1. For alternate functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit, DALI: Digital addressable lighting interface

Figure 2 - 24 Pin Block Diagram for Pin Type 8-3-8

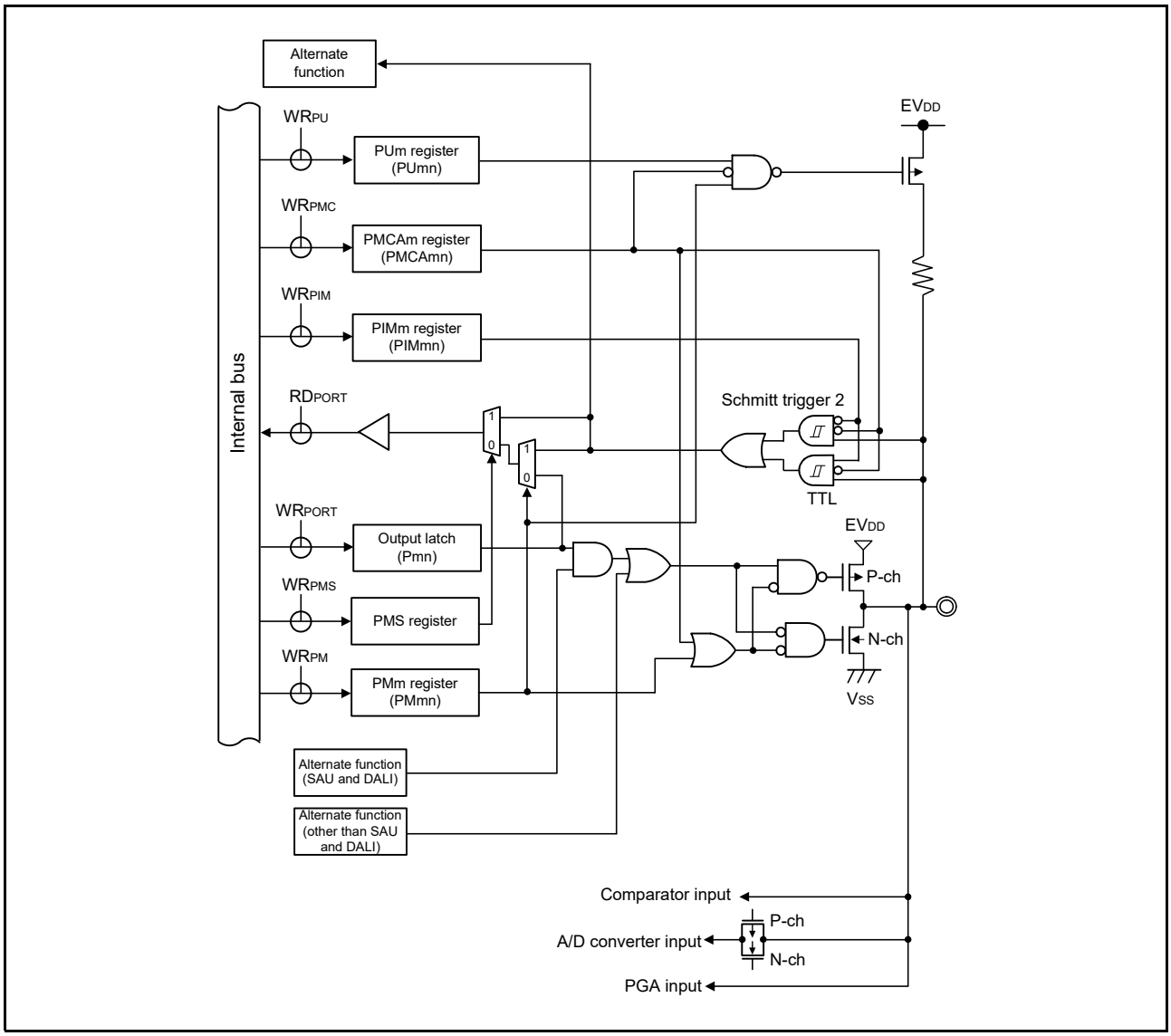


Caution **The input buffer is enabled even if the type 8-3-8 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register m (POMm). This may lead to a through current flowing through the type 8-3-8 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.**

Remark 1. For alternate functions, see **2.1 Functions of Port Pins**.

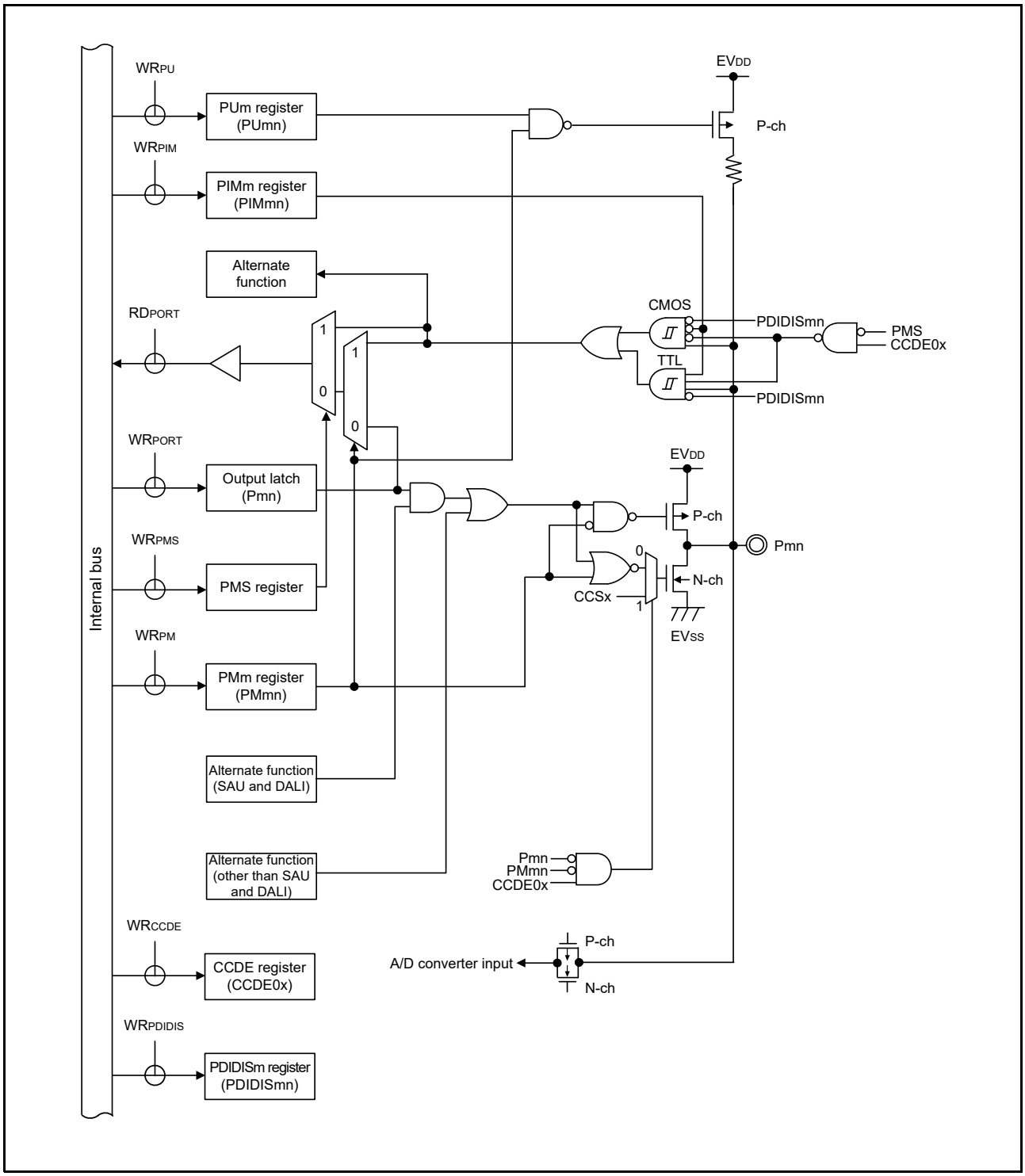
Remark 2. SAU: Serial array unit, DALI: Digital addressable lighting interface

Figure 2 - 25 Pin Block Diagram for Pin Type 8-18-2



Remark SAU: Serial array unit, DALI: Digital addressable lighting interface

Figure 2 - 26 Pin Block Diagram for Pin Type 8-41-1

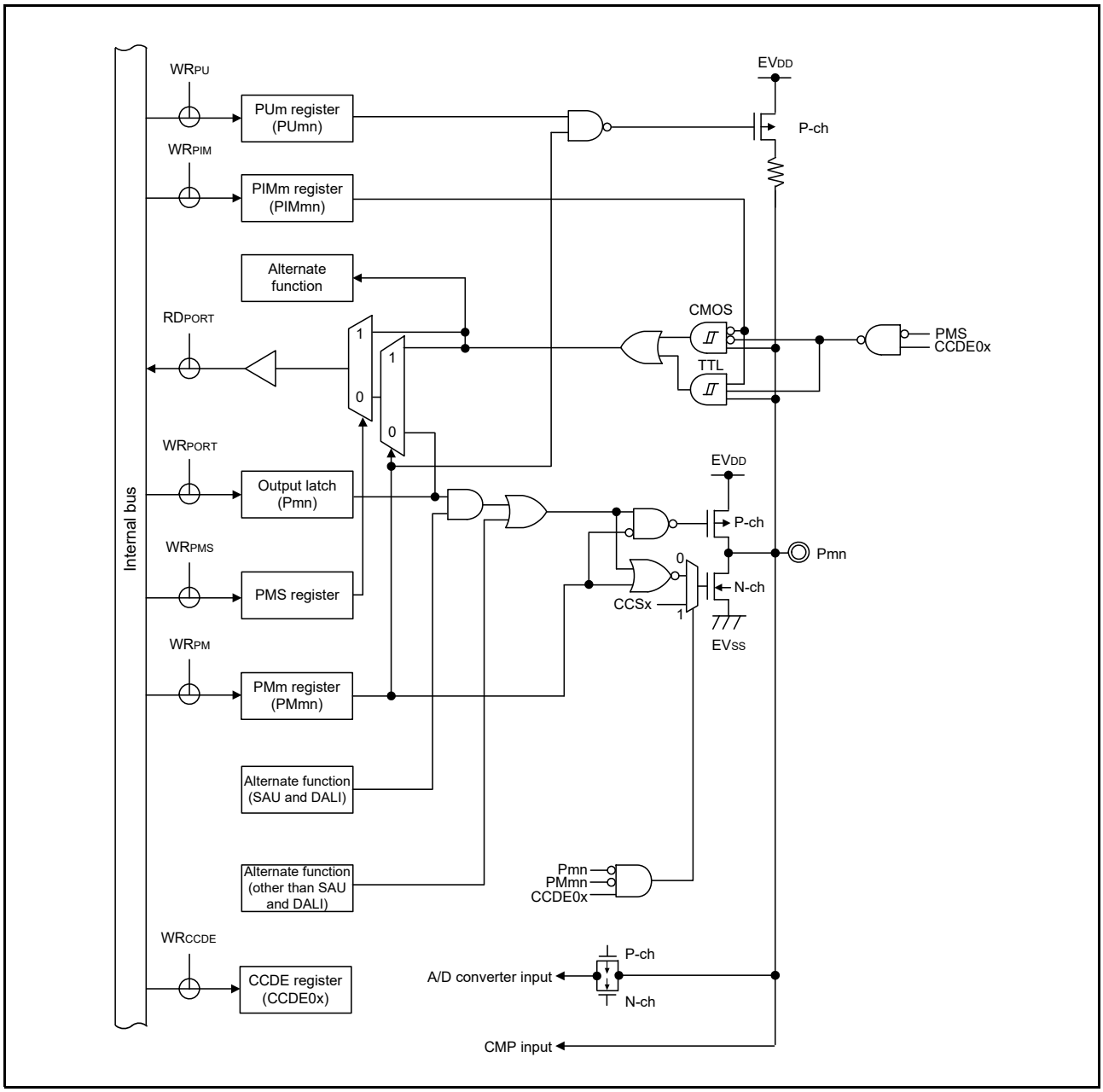


Caution **The input buffer is enabled even if the type 8-41-1 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register m (POMm). This may lead to a through current flowing through the type 8-41-1 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.**

Remark 1. For alternate functions, see **2.1 Functions of Port Pins**.

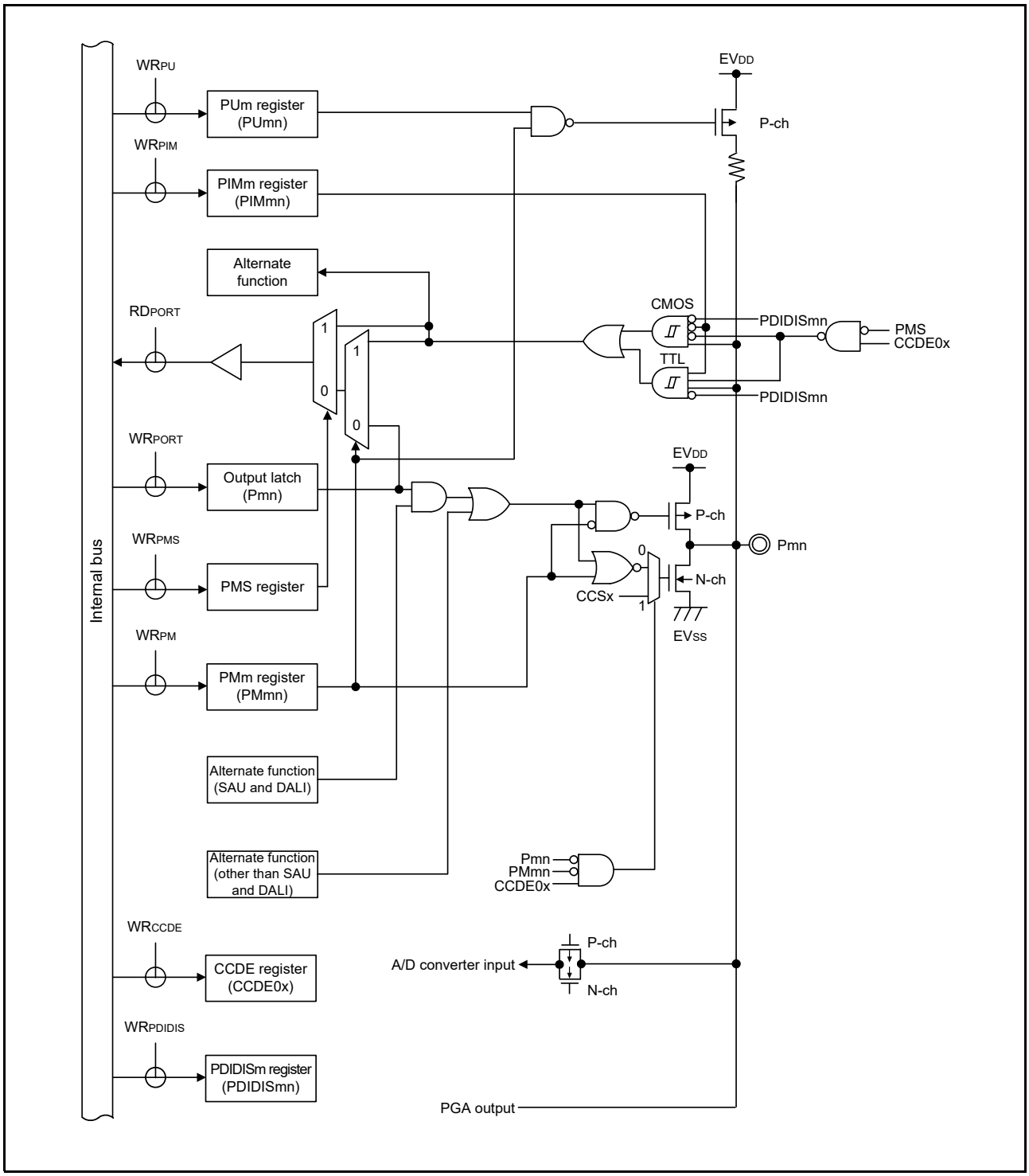
Remark 2. SAU: Serial array unit, DALI: Digital addressable lighting interface

Figure 2 - 27 Pin Block Diagram for Pin Type 8-41-2



Remark SAU: Serial array unit, DALI: Digital addressable lighting interface

Figure 2 - 28 Pin Block Diagram for Pin Type 8-42-1



Caution **The input buffer is enabled even if the type 8-42-1 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register m (POMm). This may lead to a through current flowing through the type 8-42-1 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.**

Remark 1. For alternate functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit, DALI: Digital addressable lighting interface

Section 3 CPU Architecture

The RL78/G24 is a microcontroller that has the RL78-S3 CPU core.
The CPU core in the RL78-S3 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.

- 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

The following multiply/divide instructions are available only in the RL78-S3 CPU core.
MULHU (unsigned 16-bit multiplication)
MULH (signed 16-bit multiplication)
DIVHU (unsigned 16-bit division)
DIVWU (unsigned 32-bit division)
MACHU (unsigned multiplication/accumulation (16 bits × 16 bits) + 32 bits)
MACH (signed multiplication/accumulation (16 bits × 16 bits) + 32 bits)

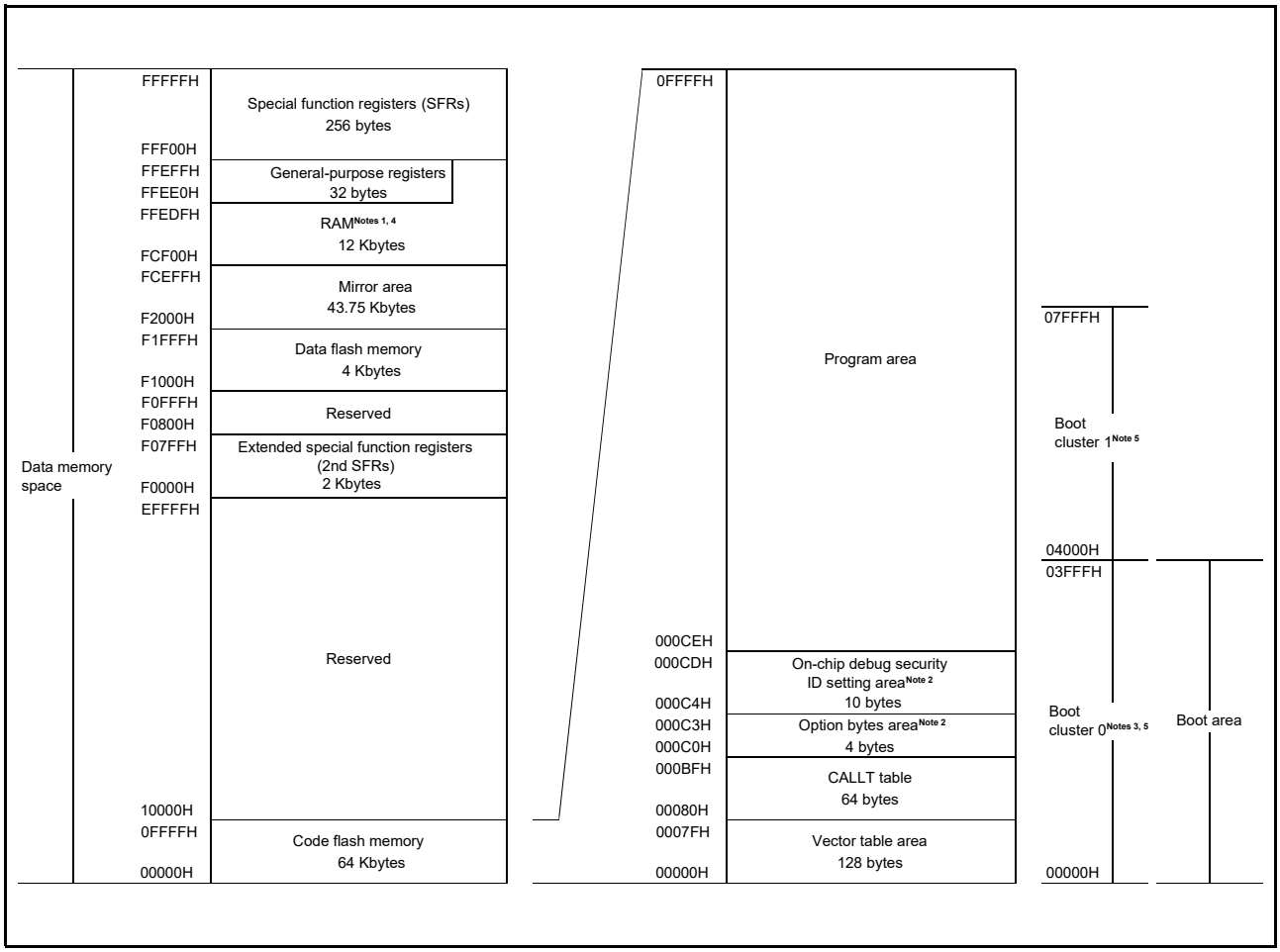
- Data allocation: Little endian

The RL78/G24 supports an OCD trace function.

3.1 Memory Space

Products in the RL78/G24 can access a 1 MB address space. For details, see **Figures 3 - 1** and **3 - 2**.

Figure 3 - 1 Memory Map (R7F101GxE (x = 6, 7, 8, A, B, E, F, G, J, L))



Note 1. Instructions can be executed from the RAM area excluding the general-purpose register area.

Note 2. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.
When boot swap is used: Set the option bytes to 000C0H to 000C3H and 040C0H to 040C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 040C4H to 040CDH.

Note 3. Rewriting of the boot area can be prohibited by a security setting. See **39.9 Security Settings**.

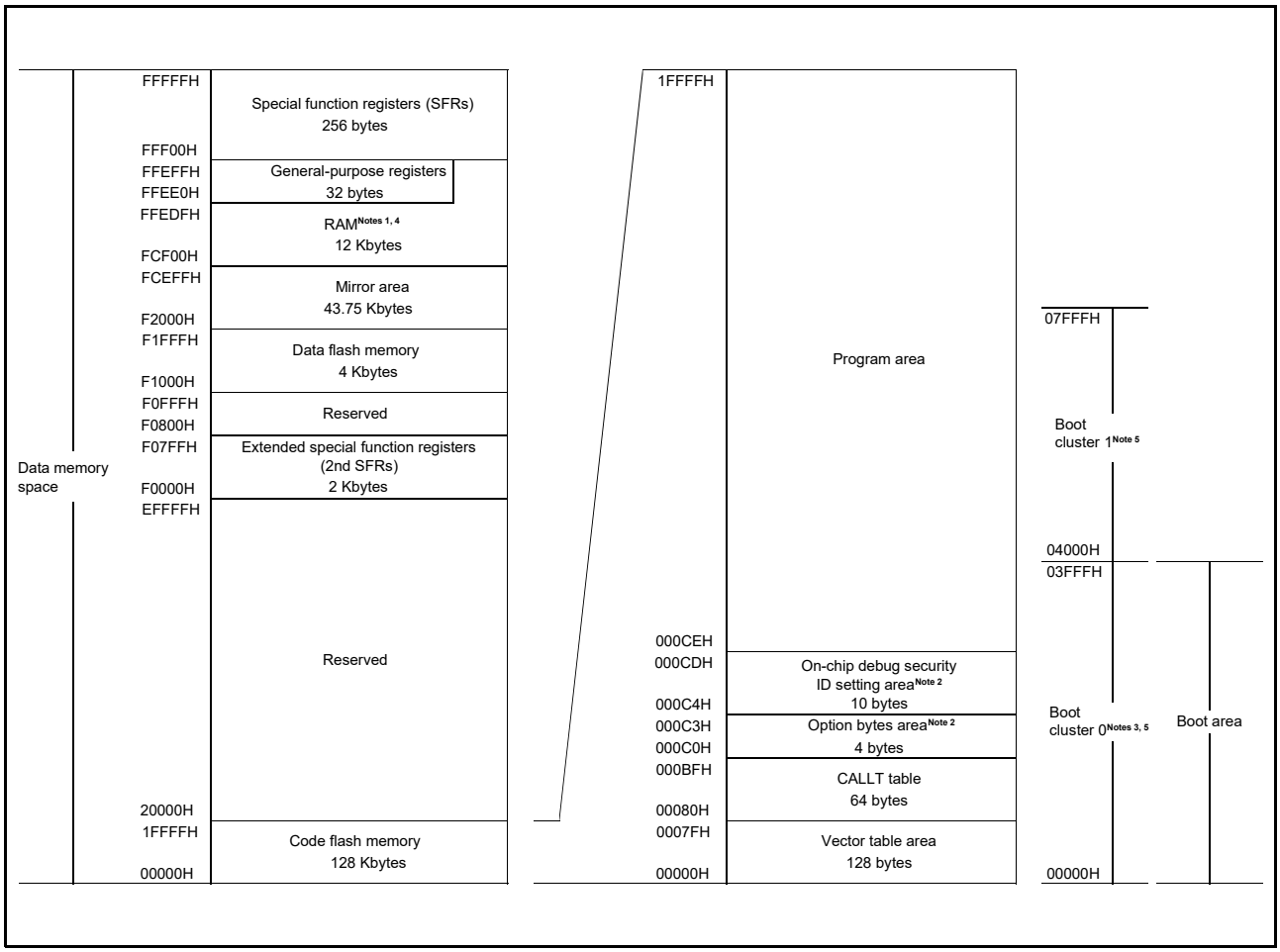
Note 4. When the FAA is in use (FAAEN = 1), FD800H to FE7FFH (4 Kbytes) and FE800H to FEFFFH (2 Kbytes) are used respectively as the instruction code memory and data memory areas, so these areas are not accessible from the CPU.

Note 5. Boot cluster 0 is selected as the boot area at the time of shipment. When boot swapping is applied, the boot area is swapped between boot cluster 0 and boot cluster 1. See **39.7 Boot Swap Function**.

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.**
RAM parity error resets become enabled (RPERDIS = 0) following a reset. For details, see 35.3.4 RAM parity error detection.

Remark The code flash memory area is divided into blocks, with each block being 2 Kbytes. For the correspondence between addresses and block numbers, see **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**.

Figure 3 - 2 Memory Map (R7F101GxG (x = 6, 7, 8, A, B, E, F, G, J, L))



Note 1. Instructions can be executed from the RAM area excluding the general-purpose register area.

Note 2. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.
When boot swap is used: Set the option bytes to 000C0H to 000C3H and 040C0H to 040C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 040C4H to 040CDH.

Note 3. Rewriting of the boot area can be prohibited by a security setting. See **39.9 Security Settings**.

Note 4. When the FAA is in use (FAAEN = 1), FD800H to FE7FFH (4 Kbytes) and FE800H to FEFFFH (2 Kbytes) are used respectively as the instruction code memory and data memory areas, so these areas are not accessible from the CPU.

Note 5. Boot cluster 0 is selected as the boot area at the time of shipment. When boot swapping is applied, the boot area is swapped between boot cluster 0 and boot cluster 1. See **39.7 Boot Swap Function**.

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.**
RAM parity error resets become enabled (RPERDIS = 0) following a reset. For details, see 35.3.4 RAM parity error detection.

Remark The code flash memory area is divided into blocks, with each block being 2 Kbytes. For the correspondence between addresses and block numbers, see **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**.

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

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

Address	Block Number	Address	Block Number
00000H to 007FFH	000H	10000H to 107FFH	020H
00800H to 00FFFH	001H	10800H to 10FFFH	021H
01000H to 017FFH	002H	11000H to 117FFH	022H
01800H to 01FFFH	003H	11800H to 11FFFH	023H
02000H to 027FFH	004H	12000H to 127FFH	024H
02800H to 02FFFH	005H	12800H to 12FFFH	025H
03000H to 037FFH	006H	13000H to 137FFH	026H
03800H to 03FFFH	007H	13800H to 13FFFH	027H
04000H to 047FFH	008H	14000H to 147FFH	028H
04800H to 04FFFH	009H	14800H to 14FFFH	029H
05000H to 057FFH	00AH	15000H to 157FFH	02AH
05800H to 05FFFH	00BH	15800H to 15FFFH	02BH
06000H to 067FFH	00CH	16000H to 167FFH	02CH
06800H to 06FFFH	00DH	16800H to 16FFFH	02DH
07000H to 077FFH	00EH	17000H to 177FFH	02EH
07800H to 07FFFH	00FH	17800H to 17FFFH	02FH
08000H to 087FFH	010H	18000H to 187FFH	030H
08800H to 08FFFH	011H	18800H to 18FFFH	031H
09000H to 097FFH	012H	19000H to 197FFH	032H
09800H to 09FFFH	013H	19800H to 19FFFH	033H
0A000H to 0A7FFH	014H	1A000H to 1A7FFH	034H
0A800H to 0AFFFH	015H	1A800H to 1AFFFH	035H
0B000H to 0B7FFH	016H	1B000H to 1B7FFH	036H
0B800H to 0BFFFH	017H	1B800H to 1BFFFH	037H
0C000H to 0C7FFH	018H	1C000H to 1C7FFH	038H
0C800H to 0CFFFH	019H	1C800H to 1CFFFH	039H
0D000H to 0D7FFH	01AH	1D000H to 1D7FFH	03AH
0D800H to 0DFFFH	01BH	1D800H to 1DFFFH	03BH
0E000H to 0E7FFH	01CH	1E000H to 1E7FFH	03CH
0E800H to 0EFFFH	01DH	1E800H to 1EFFFH	03DH
0F000H to 0F7FFH	01EH	1F000H to 1F7FFH	03EH
0F800H to 0FFFFH	01FH	1F800H to 1FFFFH	03FH

Remark R7F101GxE (x = 6, 7, 8, A, B, E, F, G, J, L): Block numbers 000H to 01FH
R7F101GxG (x = 6, 7, 8, A, B, E, F, G, J, L): Block numbers 000H to 03FH

3.1.1 Internal program memory space

The internal program memory holds the program and table data.
The RL78/G24 products incorporate the internal ROM (flash memory) with the capacity shown below.

Table 3 - 2 Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
R7F101GxE (x = 6, 7, 8, A, B, E, F, G, J, L)	Flash memory	65536 × 8 bits (00000H to 0FFFFH)
R7F101GxG (x = 6, 7, 8, A, B, E, F, G, J, L)		131072 × 8 bits (00000H to 1FFFFH)

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.
   To use the boot swap function, set a vector table also at 04000H to 0407FH.
   In **Table 3 - 3 Vector Table**, "✓" indicates an interrupt source which is supported. "—" indicates an interrupt source which is not supported.
   The vector table address can be changed to an address in RAM when self-programming is to proceed. For details, see **39.6.2.18 Interrupt vector change registers 0 and 1 (FLSIVC0, FLSIVC1)**.

Table 3 - 3 Vector Table (1/2)

Vector Table Address	Interrupt Source	64-pin	52-pin	48-pin	44-pin	40-pin	32-pin	30-pin	25-pin	24-pin	20-pin
00004H	INTWDTI	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00006H	INTLVI	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00008H	INTP0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0000AH	INTP1	✓	✓	✓	✓	✓	✓	✓	—	—	—
	INTAD1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0000CH	INTP2	✓	✓	✓	✓	✓	✓	✓	—	—	—
	INTAD2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0000EH	INTP3	✓	✓	✓	✓	✓	✓	✓	—	—	—
00010H	INTP4	✓	✓	✓	✓	✓	✓	✓	—	—	—
00012H	INTP5	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
00014H	INTST2/INTCSI20/INTIIC20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00016H	INTSR2/INTCSI21/INTIIC21	✓	✓	✓	✓	✓	Note 1	Note 1	Note 1	Note 1	Note 1
00018H	INTSRE2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0001AH	INTFAAE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0001CH	INTTIMEC0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0001EH	INTST0/INTCSI00/INTIIC00	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
00020H	INTTM00	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00022H	INTSRE0	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
	INTTM01H	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00024H	INTST1/INTCSI10/INTIIC10	✓	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2
00026H	INTSR1/INTCSI11/INTIIC11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00028H	INTSRE1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	INTTM03H	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0002AH	INTIICA0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0002CH	INTSR0/INTCSI01/INTIIC01	✓	✓	✓	Note 3	Note 3	Note 3	Note 3	Note 3	Note 3	—
0002EH	INTTM01	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00030H	INTTM02	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00032H	INTTM03	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00034H	INTAD0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00036H	INTRTC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00038H	INTITL	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0003AH	INTKR	✓	✓	✓	✓	✓	—	—	—	—	—
0003CH	INTTD	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	INTTRJ0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0003EH	INTTRD0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00040H	INTTRD1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00042H	INTRD	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	INTTRG	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 3 - 3 Vector Table (2/2)

Vector Table Address	Interrupt Source	64-pin	52-pin	48-pin	44-pin	40-pin	32-pin	30-pin	25-pin	24-pin	20-pin
00044H	INTED/INTCLD/INTBPD	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	INTTRX	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00046H	INTP20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00048H	INTP21	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
0004AH	INTP6	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	INTTMKBSTR10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0004CH	INTP7	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	INTTMKBSTP10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0004EH	INTP8	✓	✓	✓	—	—	—	—	—	—	—
	INTTMKBSTR11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00050H	INTP9	✓	✓	✓	—	—	—	—	—	—	—
	INTTMKBSTP11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00052H	INTFL	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00054H	INTP10	✓	✓	—	—	—	—	—	—	—	—
	INTCMP0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00056H	INTP11	✓	✓	—	—	—	—	—	—	—	—
	INTCMP1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00058H	INTFAATRAP	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0005AH	INTCMP2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0005CH	INTCMP3	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
0005EH	INTTMKB0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00060H	INTTMKB1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00062H	INTTMKB2	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
00064H	INTSDD	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	INTGCR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00066H	INTFED	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	INTPMC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00068H	INTTMKBSTR00	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0006AH	INTTMKBSTP00	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0006CH	INTTMKBSTR01	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0006EH	INTTMKBSTP01	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00070H	INTTMKBSTR20	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
00072H	INTTMKBSTP20	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
00074H	INTTMKBSTR21	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
00076H	INTTMKBSTP21	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
00078H	INTTIMEC1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0007AH	INTTIMEC2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0007CH	INTAD3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0007EH	BRK	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note 1. INTSR2 is only present in this product.

Note 2. INTST1 is only present in this product.
Note 3. INTSR0 is only present in this product.

2. CALLT instruction table area
   The 64-byte area 00080H to 000BFH can hold 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).
   To use the boot swap function, set a CALLT instruction table also at 04080H to 040BFH.

3. Option bytes area
   A 4-byte area of 000C0H to 000C3H can be used as an option bytes area. Set the option byte at 040C0H to 040C3H when the boot swap is used. For details, see **Section 38 Option Bytes**.

4. On-chip debug security ID setting area
   A 10-byte area of 000C4H to 000CDH and 040C4H to 040CDH 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 when the boot swap is not used and at 000C4H to 000CDH and at 040C4H to 040CDH when the boot swap is used. For details, see **Section 40 On-chip Debugging**.

3.1.2 Mirror area

In products of the RL78/G24, the code flash area from 00000H to 0FFFFH or 10000H to 1FFFFH is mirrored at F0000H to FFFFFH. The setting of the effective bit of the processor mode control register (PMC) determines which range of the code flash area is to be mirrored.

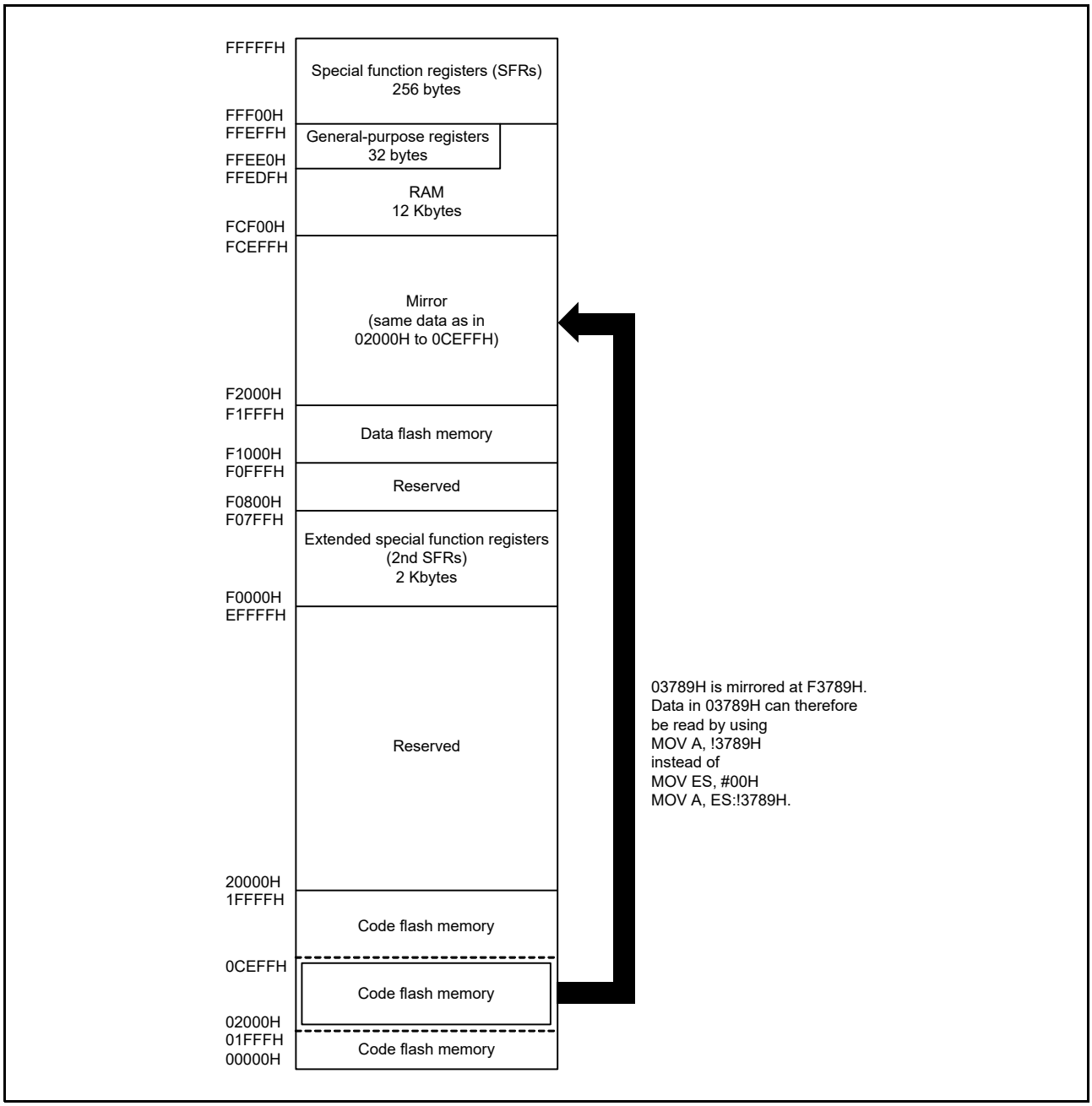
By reading data from F0000H to FFFFFH, an instruction that does not have the ES register as an operand can be used, and thus the contents of the code flash can be read with the shorter code. However, the code flash area is not mirrored to the special function registers (SFRs), extended special function registers (2nd SFRs), RAM, data flash memory, and use prohibited areas.

See **3.1 Memory Space** 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 R7F101GxG (x = 6, 7, 8, A, B, E, F, G, J, L) (Flash Memory: 128 Kbytes, RAM: 12 Kbytes)



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. The value of this register following a reset is 00H.

Figure 3 - 3 Format of Processor Mode Control Register (PMC)

Address: FFFFEH
After reset: 00H
R/W: 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 0FFFFH is mirrored to F0000H to FFFFFH
1	10000H to 1FFFFH is mirrored to F0000H to FFFFFH

Caution 1. After setting the PMC register, wait for at least one instruction and access the mirror area.

Caution 2. When boot swapping is executed while bit 0 (MAA) is 0, data at 03000H to 07FFFH are mirrored at F3000H to F7FFFH after boot swapping.

3.1.3 Internal data memory space

The RL78/G24 products incorporate the RAM with the capacity shown below.

Table 3 - 4 Internal RAM Capacity

Part Number	Internal RAM
R7F101GxE (x = 6, 7, 8, A, B, E, F, G, J, L)	12288 × 8 bits (FCF00H to FFEFFH)
R7F101GxG (x = 6, 7, 8, A, B, E, F, G, J, L)	12288 × 8 bits (FCF00H to FFEFFH)

The internal RAM can be used as a data area and a program area where instructions are fetched (the space to which the general-purpose registers are allocated cannot be used for instruction fetching). 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. The space to which the general-purpose registers are allocated (FFEE0H to FFEFFH) cannot be used for instruction fetching or the stack.

Caution 2. The internal RAM area in the following products cannot be used as stack memory when using the on-chip debugging trace function.
R7F101GxE (x = 6, 7, 8, A, B, E, F, G, J, L): FD300H to FD6FFH
R7F101GxG (x = 6, 7, 8, A, B, E, F, G, J, L): FD300H to FD6FFH

3.1.4 Special function register (SFR) area

The special function registers (SFRs) in the on-chip peripheral modules are allocated in the area FFF00H to FFFFFH. See **Table 3 - 5 List of Special Function Registers (SFRs)** in **3.2.4 Special function registers (SFRs)**.

Caution Only access the addresses to which SFRs are assigned.

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

The extended special function registers (2nd SFRs) in the on-chip peripheral modules are allocated in the area F0000H to F07FFH. See **Table 3 - 6 List of Extended Special Function Registers (2nd SFRs)** in **3.2.5 Extended special function registers (2nd SFRs)**.

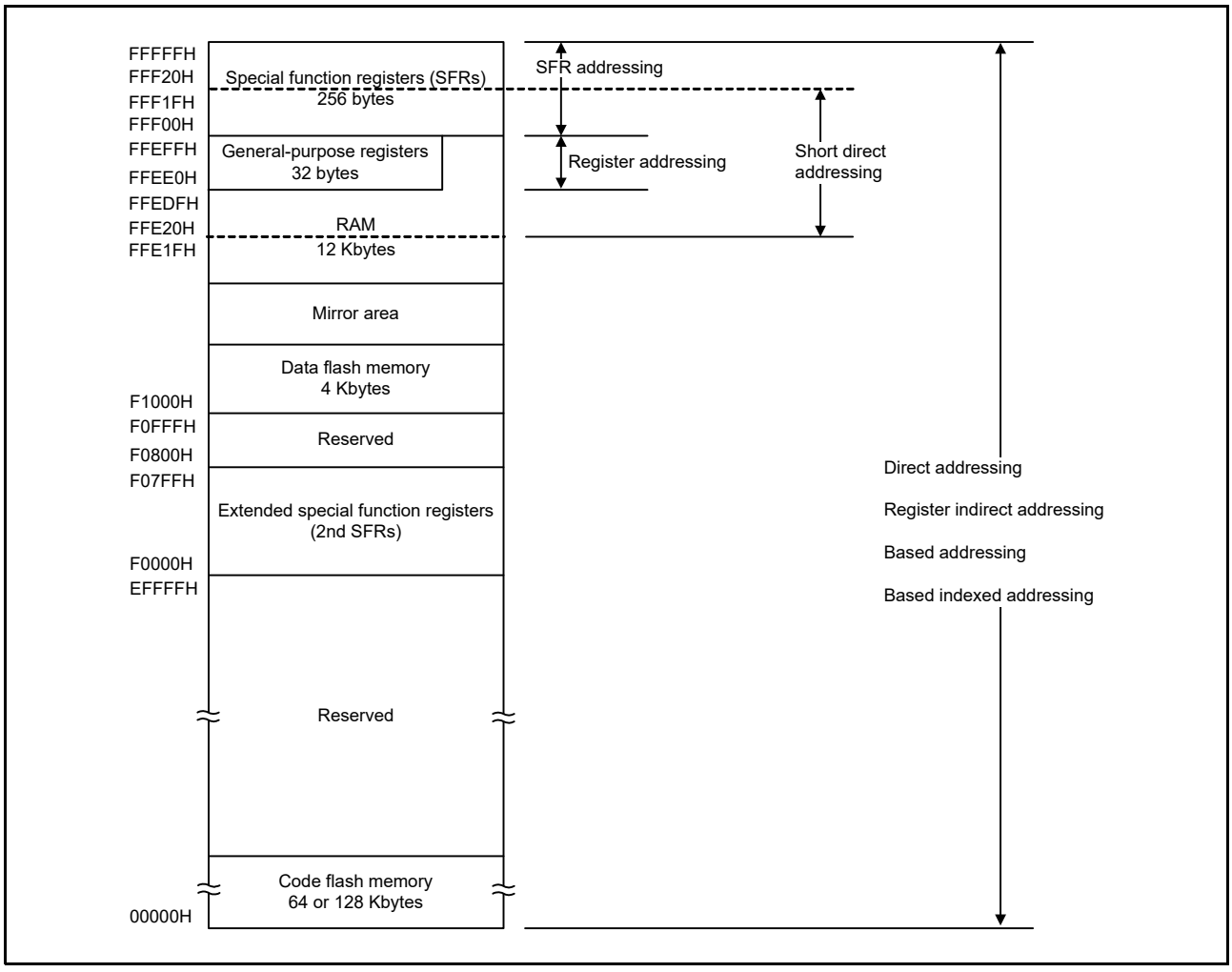
Caution 1. Only access the addresses to which extended SFRs are assigned.

Caution 2. In the area for the extended special function registers (2nd SFRs), timer RJ counter register 0 (TRJ0) is allocated to the address F0500H, the registers of the true random number generator (TRNG) are allocated to F0540H and F0542H, and the registers of the flexible application accelerator (FAA) are allocated to the address range from F0600H to F06A9H. The CPU is placed in the wait state and does not proceed to the next instruction during access to these registers. Accordingly, the CPU entering this state lengthens the number of clock cycles to execute an instruction by the number of cycles of waiting. Specifically, the CPU waits for one clock cycle during access (whether reading or writing) to timer RJ counter register 0 (TRJ0), the registers of the true random number generator (TRNG), or those of the flexible application accelerator (FAA).

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/G24, 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 (SFRs) and general-purpose registers are available for use. **Figure 3 - 4** shows 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



3.2 Processor Registers

The RL78/G24 products incorporate the following processor registers.

3.2.1 Control registers

The control registers control the program sequence, state, 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. The values of the reset vector table at addresses 0000H and 0001H are set in the program counter following a reset.

Figure 3 - 5 Format of Program Counter

Symbol	19																0
PC																	

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. The value of the PSW following a reset is 06H.

Figure 3 - 6 Format of Program Status Word

Symbol	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 the in-service priority flags (ISP[1:0]), an interrupt mask flag for various interrupt sources, and a priority specification flag.
   The IE flag is reset to 0 upon DI instruction execution or interrupt acknowledgment and is set to 1 upon EI instruction execution.

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

e) In-service priority flags (ISP1, ISP0)
These flags manage the priority of acknowledgeable maskable vectored interrupts. Vectored interrupt requests specified lower than the value of ISP[1:0] flags by the priority specification flag registers (PRn0L, PRn0H, PRn1L, PRn1H, PRn2L, PRn2H, PRn3L, and PRn3H; see **29.3.3 Priority specification flag registers**) cannot be acknowledged. Actual request acknowledgment is controlled by the interrupt enable flag (IE).

Remark n = 0, 1

f) Carry flag (CY)
This flag holds overflow and underflow upon add/subtract instruction execution. It holds 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 - 7 Format of Stack Pointer

Symbol	15	14	13	12	11	10	9	8
SP	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8

7	6	5	4	3	2	1	0
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 the contents of the SP become undefined following a reset, be sure to initialize the SP before using the stack.

Caution 2. The space to which the general-purpose registers are allocated (FFEE0H to FFEFFH) cannot be used for instruction fetching or the stack.

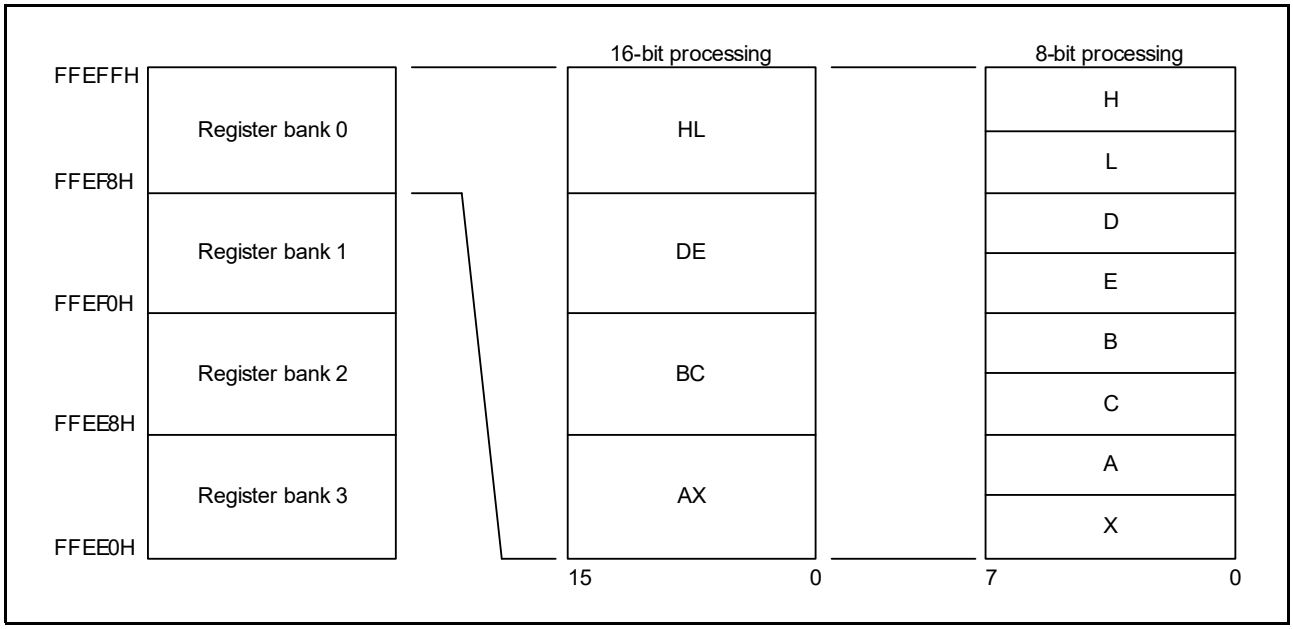
Caution 3. The internal RAM area in the following products cannot be used as stack memory when using the on-chip debugging trace function.
R7F101GxE (x = 6, 7, 8, A, B, E, F, G, J, L): FD300H to FD6FFH
R7F101GxG (x = 6, 7, 8, A, B, E, F, G, J, L): FD300H to FD6FFH

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 consist 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 interrupt processing for each bank.

Caution The space to which the general-purpose registers are allocated (FFEE0H to FFEFFH) cannot be used for instruction fetching or the stack.

Figure 3 - 8 Configuration of General-purpose Registers (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 indirect addressing), respectively. The default value of the ES register after reset is 0FH, and that of the CS register is 00H.

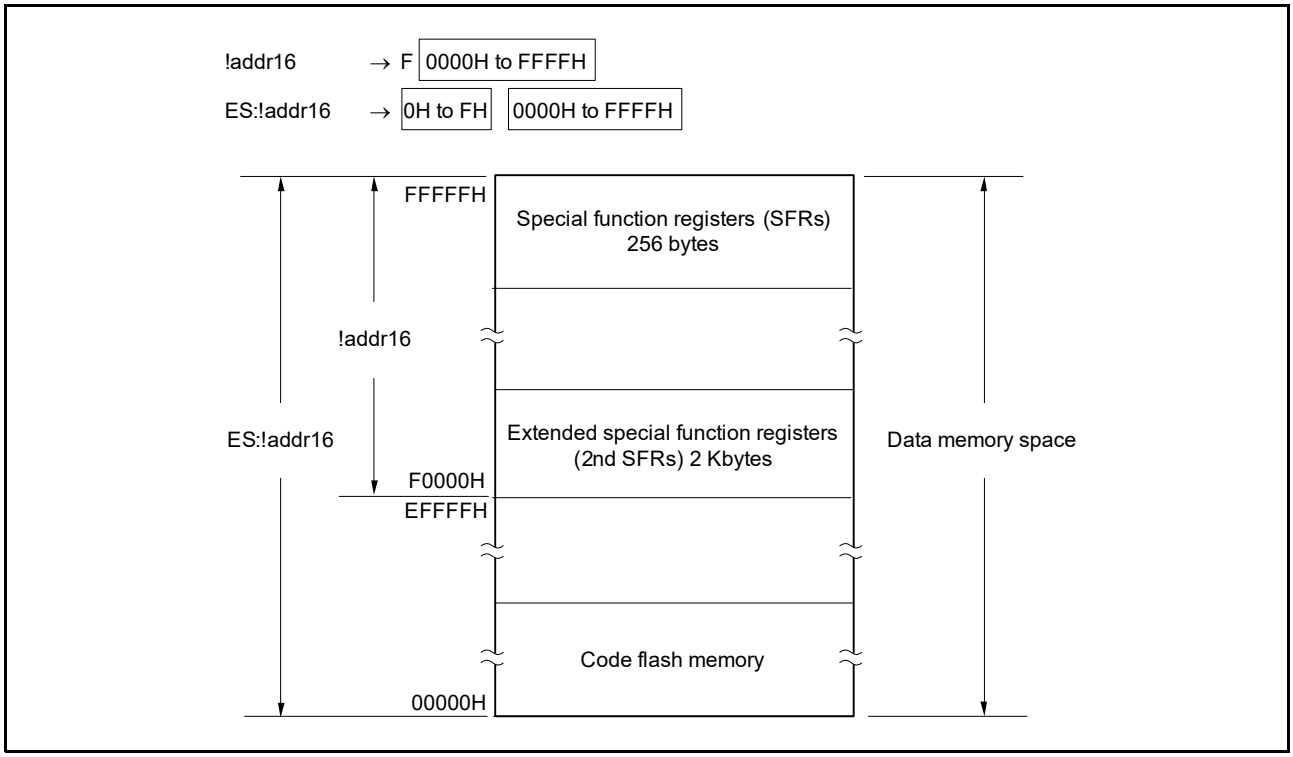
Figure 3 - 9 Configuration of ES and CS Registers

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

Symbol	7	6	5	4	3	2	1	0
CS	0	0	0	0	CS3	CS2	ES1	ES0

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 - 10 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.

Table 3 - 5 gives 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
  Items in this column indicate the states (values) of each of the registers after generation of a reset signal.

Caution **Only access the addresses to which SFRs are assigned.**

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

Table 3 - 5 List of Special Function Registers (SFRs) (1/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1 bit	8 bits	16 bits	
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	12-bit/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
FFF34H	Key return control register	KRCTL		R/W	✓	✓	—	00H
FFF35H	Key return flag register	KRF		R/W	—	✓	—	00H
FFF37H	Key return mode register 0	KRM0		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
FFF3AH	External interrupt rising edge enable register 1	EGP1		R/W	✓	✓	—	00H
FFF3BH	External interrupt falling edge enable register 1	EGN1		R/W	✓	✓	—	00H
FFF44H	Serial data register 02	TXD1/ SIO10	SDR02	R/W	—	✓	✓	0000H
FFF45H		—			—	—		
FFF46H	Serial data register 03	RXD1/ SIO11	SDR03	R/W	—	✓	✓	0000H
FFF47H		—			—	—		

Table 3 - 5 List of Special Function Registers (SFRs) (2/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1 bit	8 bits	16 bits	
FFF48H	Serial data register 10	TXD2/ SIO20	SDR10	R/W	—	✓	✓	0000H
FFF49H		—			—	—		
FFF4AH	Serial data register 11	RXD2/ SIO21	SDR11	R/W	—	✓	✓	0000H
FFF4BH		—			—	—		
FFF50H	IICA shift register 0	IICA0		R/W	—	✓	—	00H
FFF51H	IICA status register 0	IICS0		R	✓	✓	—	00H
FFF52H	IICA flag register 0	IICF0		R/W	✓	✓	—	00H
FFF60H	Timer RG general register C	TRGGRC		R/W	—	—	✓	FFFFH
FFF61H								
FFF62H	Timer RG general register D	TRGGRD		R/W	—	—	✓	FFFFH
FFF63H								
FFF64H	Timer data register 02	TDR02		R/W	—	—	✓	0000H
FFF65H								
FFF66H	Timer data register 03	TDR03L	TDR03	R/W	—	✓	✓	00H
FFF67H		TDR03H			—	✓		00H
FFF6CH	Timer RD general register C0	TRDGRC0		R/W	—	—	✓	FFFFH
FFF6DH								
FFF6EH	Timer RD general register D0	TRDGRD0		R/W	—	—	✓	FFFFH
FFF6FH								
FFF70H	Timer RD general register C1	TRDGRC1		R/W	—	—	✓	FFFFH
FFF71H								
FFF72H	Timer RD general register D1	TRDGRD1		R/W	—	—	✓	FFFFH
FFF73H								
FFF74H	Timer RD extended compare register D0	TRDCMPD0		R/W	—	—	✓	FFFFH
FFF75H								
FFF76H	Timer RD extended compare register C1	TRDCMPC1		R/W	—	—	✓	FFFFH
FFF77H								
FFF78H	Timer RD extended compare register D1	TRDCMPD1		R/W	—	—	✓	FFFFH
FFF79H								
FFF7AH	Timer RD A/D conversion trigger buffer register 0/ timer-KB PWM output gating mode buffer register	TRDADTB0/ TRDCMPF1		R/W	—	—	✓	FFFFH
FFF7BH								
FFF7CH	Timer RD A/D conversion trigger buffer register 1	TRDADTB1		R/W	—	—	✓	FFFFH
FFF7DH								
FFF7EH	Timer RD simultaneous update trigger register 0	TRDRDT0	TRDRDT	R/W	—	✓	✓	00H
FFF7FH	Timer RD simultaneous update trigger register 1	TRDRDT1			—	✓		00H
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
FFFA7H	Subsystem clock select register	CKSEL		R/W	✓	✓	—	00H
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	✓	✓	—	19H

Table 3 - 5 List of Special Function Registers (SFRs) (3/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1 bit	8 bits	16 bits	
FFFABH	Watchdog timer enable register	WDTE		R/W	—	✓	—	9AH/1AH **Note 3**
FFFACH	CRC input register	CRCIN		R/W	—	✓	—	00H
FFFD0H	Interrupt request flag register 2	IF2L	IF2	R/W	✓	✓	✓	00H
FFFD1H		IF2H		R/W	✓	✓		00H
FFFD2H	Interrupt request flag register 3	IF3L	IF3	R/W	✓	✓	✓	00H
FFFD3H		IF3H		R/W	✓	✓		00H
FFFD4H	Interrupt mask flag register 2	MK2L	MK2	R/W	✓	✓	✓	FFH
FFFD5H		MK2H		R/W	✓	✓		FFH
FFFD6H	Interrupt mask flag register 3	MK3L	MK3	R/W	✓	✓	✓	FFH
FFFD7H		MK3H		R/W	✓	✓		FFH
FFFD8H	Priority specification flag register 02	PR02L	PR02	R/W	✓	✓	✓	FFH
FFFD9H		PR02H		R/W	✓	✓		FFH
FFFDAH	Priority specification flag register 03	PR03L	PR03	R/W	✓	✓	✓	FFH
FFFDBH		PR03H		R/W	✓	✓		FFH
FFFDCH	Priority specification flag register 12	PR12L	PR12	R/W	✓	✓	✓	FFH
FFFDDH		PR12H		R/W	✓	✓		FFH
FFFDEH	Priority specification flag register 13	PR13L	PR13	R/W	✓	✓	✓	FFH
FFFDFH		PR13H		R/W	✓	✓		FFH
FFFE0H	Interrupt request flag register 0	IF0L	IF0	R/W	✓	✓	✓	00H
FFFE1H		IF0H		R/W	✓	✓		00H
FFFE2H	Interrupt request flag register 1	IF1L	IF1	R/W	✓	✓	✓	00H
FFFE3H		IF1H		R/W	✓	✓		00H
FFFE4H	Interrupt mask flag register 0	MK0L	MK0	R/W	✓	✓	✓	FFH
FFFE5H		MK0H		R/W	✓	✓		FFH
FFFE6H	Interrupt mask flag register 1	MK1L	MK1	R/W	✓	✓	✓	FFH
FFFE7H		MK1H		R/W	✓	✓		FFH
FFFE8H	Priority specification flag register 00	PR00L	PR00	R/W	✓	✓	✓	FFH
FFFE9H		PR00H		R/W	✓	✓		FFH
FFFEAH	Priority specification flag register 01	PR01L	PR01	R/W	✓	✓	✓	FFH
FFFEBH		PR01H		R/W	✓	✓		FFH
FFFECH	Priority specification flag register 10	PR10L	PR10	R/W	✓	✓	✓	FFH
FFFEDH		PR10H		R/W	✓	✓		FFH
FFFEEH	Priority specification flag register 11	PR11L	PR11	R/W	✓	✓	✓	FFH
FFFEFH		PR11H		R/W	✓	✓		FFH
FFFF0H	Multiplication and accumulation register (L)	MACRL		R/W	—	—	✓	0000H
FFFF1H								
FFFF2H	Multiplication and accumulation register (H)	MACRH		R/W	—	—	✓	0000H
FFFF3H								
FFFFEH	Processor mode control register	PMC		R/W	✓	✓	—	00H

Note 1. For the reset resources, see **Section 32 Reset Function**.

Note 2. The initial value depends on the source of the reset. See **34.3.1 Voltage detection register (LVIM)**.

Note 3. The reset value of the WDTE register is determined by the setting of the option byte.

Remark For extended SFRs (2nd SFRs), see **Table 3 - 6 List of Extended Special Function Registers (2nd SFRs)**.

3.2.5 Extended special function registers (2nd SFRs)

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.

Table 3 - 6 gives 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
  Items in this column indicate the states (values) of each of the registers after generation of a reset signal.

Caution **Only access the addresses to which extended SFRs (2nd SFRs) are assigned.**

Remark For SFRs in the SFR area, see **3.1.4 Special function register (SFR) area**.

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (1/19)

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
F0014H	A/D converter mode register 3	ADM3		R/W	✓	✓	—	00H
F0015H	Analog input channel specification register 0	ADS0		R/W	—	✓	—	00H
F0016H	Analog input channel specification register 1	ADS1		R/W	—	✓	—	00H
F0017H	Analog input channel specification register 2	ADS2		R/W	—	✓	—	00H
F0018H	Analog input channel specification register 3	ADS3		R/W	—	✓	—	00H
F0019H	Conversion setting register	ADSCTL		R/W	✓	✓	—	00H
F001AH	Conversion trigger specification register 0	ADTR0		R/W	—	✓	—	00H
F001BH	Conversion trigger specification register 1	ADTR1		R/W	—	✓	—	00H
F001CH	Conversion trigger specification register 2	ADTR2		R/W	—	✓	—	00H
F001DH	Conversion trigger specification register 3	ADTR3		R/W	—	✓	—	00H
F001FH	A/D conversion sampling mode specification register	ADSPMOD		R/W	—	✓	—	00H
F0020H	12-bit/10-bit A/D conversion result register 0	—	ADCR0	R	—	—	✓	0000H
F0021H	8-bit A/D conversion result register 0H	ADCR0H		R	—	✓	—	00H
F0022H	12-bit/10-bit A/D conversion result register 1	—	ADCR1	R	—	—	✓	0000H
F0023H	8-bit A/D conversion result register 1H	ADCR1H		R	—	✓	—	00H
F0024H	12-bit/10-bit A/D conversion result register 2	—	ADCR2	R	—	—	✓	0000H
F0025H	8-bit A/D conversion result register 2H	ADCR2H		R	—	✓	—	00H
F0026H	12-bit/10-bit A/D conversion result register 3	—	ADCR3	R	—	—	✓	0000H
F0027H	8-bit A/D conversion result register 3H	ADCR3H		R	—	✓	—	00H
F0028H	Conversion interrupt control register	ADINTCTL		R/W	✓	✓	—	00H
F0029H	Conversion interrupt status register	ADINTST		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
F0036H	Pull-up resistor option register 6	PU6		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
F0047H	Port input mode register 7	PIM7		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
F0057H	Port output mode register 7	POM7		R/W	✓	✓	—	00H
F0060H	Port mode control A register 0	PMCA0		R/W	✓	✓	—	FFH
F0061H	Port mode control A register 1	PMCA1		R/W	✓	✓	—	FFH

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (2/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0062H	Port mode control A register 2	PMCA2		R/W	✓	✓	—	FFH
F006CH	Port mode control A register 12	PMCA12		R/W	✓	✓	—	FFH
F006EH	Port mode control A register 14	PMCA14		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 I/O select register 0	TIOS0		R/W	—	✓	—	00H
F0077H	Peripheral I/O redirection register 0	PIOR0		R/W	—	✓	—	00H
F0078H	Invalid memory access detection control register	IAWCTL		R/W	—	✓	—	00H
F0079H	UART loopback select register	ULBS		R/W	✓	✓	—	00H
F007AH	IICA input mode selection register	IICM		R/W	—	✓	—	00H
F007BH	Port mode select register	PMS		R/W	✓	✓	—	00H
F007CH	Prefetch buffer enable register	PFBER		R/W	✓	✓	—	00H
F007DH	Global digital input disable register	GDIDIS		R/W	✓	✓	—	00H
F0090H	Data flash control register	DFLCTL		R/W	✓	✓	—	00H
F0098H	Peripheral clock control register	PCKC		R/W	✓	✓	—	00H
F00A0H	High-speed on-chip oscillator trimming register	HIOTRM		R/W	—	✓	—	Undefined **Note 1**
F00A8H	High-speed on-chip oscillator frequency select register	HOCODIV		R/W	—	✓	—	Undefined **Note 2**
F00AAH	Flash operating mode select register	FLMODE		R/W	✓	✓	—	00H/80H/C0H **Note 3**
F00ABH	Flash operating mode protect register	FLMWRP		R/W	✓	✓	—	00H
F00B0H	Flash security flag monitoring register	FLSEC		R	—	—	✓	Undefined
F00B2H	Flash FSW monitoring register S	FLFSWS		R	—	—	✓	Undefined
F00B4H	Flash FSW monitoring register E	FLFSWE		R	—	—	✓	Undefined
F00B6H	Flash memory sequencer initial setting register	FSSET		R/W	—	✓	—	00H
F00B7H	Flash extra area sequencer control register	FSSE		R/W	✓	✓	—	00H
F00C0H	Flash protect command register	PFCMD		W	—	✓	—	—
F00C1H	Flash status register	PFS		R	✓	✓	—	00H
F00F0H	Peripheral enable register 0	PER0		R/W	✓	✓	—	00H
F00F1H	Peripheral reset control register 0	PRR0		R/W	✓	✓	—	00H
F00F2H	Middle-speed on-chip oscillator frequency select register	MOCODIV		R/W	—	✓	—	00H
F00F3H	Subsystem clock supply mode control register	OSMC		R/W	✓	✓	—	Undefined **Note 7**
F00F5H	RAM parity error control register	RPECTL		R/W	✓	✓	—	00H
F00F9H	Power-on-reset status register	PORSR		R/W	✓	✓	—	00H
F00FAH	Peripheral enable register 1	PER1		R/W	✓	✓	—	00H
F00FBH	Peripheral reset control register 1	PRR1		R/W	✓	✓	—	00H
F00FCH	Peripheral enable register 2	PER2		R/W	✓	✓	—	00H
F00FDH	Peripheral reset control register 2	PRR2		R/W	✓	✓	—	00H
F00FEH	BCD correction result register	BCDADJ		R	—	✓	—	Undefined
F00FFH	Interrupt vector jump enable register	VECTCTRL		R/W	—	✓	—	00H
F0100H	Serial status register 00	SSR00L	SSR00	R	—	✓	✓	0000H
F0101H		—			—	—		
F0102H	Serial status register 01	SSR01L	SSR01	R	—	✓	✓	0000H
F0103H		—			—	—		

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (3/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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								
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		—			—	—		
F0140H	Serial status register 10	SSR10L	SSR10	R	—	✓	✓	0000H
F0141H		—			—	—		
F0142H	Serial status register 11	SSR11L	SSR11	R	—	✓	✓	0000H
F0143H		—			—	—		

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (4/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0148H	Serial flag clear trigger register 10	SIR10L	SIR10	R/W	—	✓	✓	0000H
F0149H		—			—	—		
F014AH	Serial flag clear trigger register 11	SIR11L	SIR11	R/W	—	✓	✓	0000H
F014BH		—			—	—		
F0150H	Serial mode register 10	SMR10		R/W	—	—	✓	0020H
F0151H								
F0152H	Serial mode register 11	SMR11		R/W	—	—	✓	0020H
F0153H								
F0158H	Serial communication operation setting register 10	SCR10		R/W	—	—	✓	0087H
F0159H								
F015AH	Serial communication operation setting register 11	SCR11		R/W	—	—	✓	0087H
F015BH								
F0160H	Serial channel enable status register 1	SE1L	SE1	R	✓	✓	✓	0000H
F0161H		—			—	—		
F0162H	Serial channel start register 1	SS1L	SS1	R/W	✓	✓	✓	0000H
F0163H		—			—	—		
F0164H	Serial channel stop register 1	ST1L	ST1	R/W	✓	✓	✓	0000H
F0165H		—			—	—		
F0166H	Serial clock select register 1	SPS1L	SPS1	R/W	—	✓	✓	0000H
F0167H		—			—	—		
F0168H	Serial output register 1	SO1		R/W	—	—	✓	0F0FH
F0169H								
F016AH	Serial output enable register 1	SOE1L	SOE1	R/W	✓	✓	✓	0000H
F016BH		—			—	—		
F0174H	Serial output level register 1	SOL1L	SOL1	R/W	—	✓	✓	0000H
F0175H		—			—	—		
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		—			—	—		
F01A4H	Timer status register 02	TSR02L	TSR02	R	—	✓	✓	0000H
F01A5H		—			—	—		

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (5/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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		—			—	—		
F0212H	Middle-speed on-chip oscillator trimming register	MIOTRM		R/W	—	✓	—	90H
F0213H	Low-speed on-chip oscillator trimming register	LIOTRM		R/W	—	✓	—	80H
F0214H	High-speed system clock division register	MOSCDIV		R/W	—	✓	—	00H
F0215H	Standby mode release setting register	WKUPMD		R/W	✓	✓	—	00H
F0218H	LVD detection flag clearing register	LVDFCLR		R/W	✓	✓	—	00H
F0219H	High-speed clock select register	HSCLKSEL		R/W	✓	✓	—	00H/02H **Note 6**
F0220H	Second count register	SEC		R/W	—	✓	—	Undefined
F0221H	Minute count register	MIN		R/W	—	✓	—	Undefined
F0222H	Hour count register	HOUR		R/W	—	✓	—	Undefined
F0223H	Day-of-week count register	WEEK		R/W	—	✓	—	Undefined
F0224H	Day count register	DAY		R/W	—	✓	—	Undefined
F0225H	Month count register	MONTH		R/W	—	✓	—	Undefined
F0226H	Year count register	YEAR		R/W	—	✓	—	Undefined
F0227H	Time error correction register	SUBCUD		R/W	—	✓	—	00H
F0228H	Alarm minute register	ALARMWM		R/W	—	✓	—	Undefined
F0229H	Alarm hour register	ALARMWH		R/W	—	✓	—	Undefined
F022AH	Alarm day-of-week register	ALARMWW		R/W	—	✓	—	Undefined
F022BH	Realtime clock control register 0	RTCC0		R/W	✓	✓	—	00H
F022CH	Realtime clock control register 1	RTCC1		R/W	✓	✓	—	00H
F0230H	IICA control register 00	IICCTL00		R/W	✓	✓	—	00H
F0231H	IICA control register 01	IICCTL01		R/W	✓	✓	—	00H
F0232H	IICA low-level width setting register 0	IICWL0		R/W	—	✓	—	FFH
F0233H	IICA high-level width setting register 0	IICWH0		R/W	—	✓	—	FFH
F0234H	Slave address register 0	SVA0		R/W	—	✓	—	00H
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
F02A0H	Output current select register 0	CCS0		R/W	—	✓	—	00H

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (6/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F02A4H	Output current select register 4	CCS4	R/W	—	✓	—	00H
F02A5H	Output current select register 5	CCS5	R/W	—	✓	—	00H
F02A6H	Output current select register 6	CCS6	R/W	—	✓	—	01H
F02A7H	Output current select register 7	CCS7	R/W	—	✓	—	00H
F02A8H	Output current control enable register	CCDE	R/W	✓	✓	—	00H
F02ACH	Timer RD output port mask enable register	TRDPOE	R/W	—	✓	—	00H
F02ADH	Peripheral I/O redirection register 1	PIOR1	R/W	—	✓	—	00H
F02AEH	Peripheral I/O redirection register 2	PIOR2	R/W	—	✓	—	00H
F02AFH	Peripheral I/O redirection register 3	PIOR3	R/W	—	✓	—	00H
F02B0H	Port digital input disable register 0	PDIDIS0	R/W	✓	✓	—	00H
F02B1H	Port digital input disable register 1	PDIDIS1	R/W	✓	✓	—	00H
F02B3H	Port digital input disable register 3	PDIDIS3	R/W	✓	✓	—	00H
F02B5H	Port digital input disable register 5	PDIDIS5	R/W	✓	✓	—	00H
F02B7H	Port digital input disable register 7	PDIDIS7	R/W	✓	✓	—	00H
F02BDH	Port digital input disable register 13	PDIDIS13	R/W	✓	✓	—	00H
F02C0H	Flash programming mode control register	FLPMC	R/W	—	✓	—	08H
F02C1H	Flash area selection register	FLARS	R/W	✓	✓	—	00H
F02C2H	Flash address pointer register L	FLAPL	R/W	—	—	✓	0000H
F02C4H	Flash address pointer register H	FLAPH	R/W	—	✓	—	00H
F02C5H	Flash memory sequencer control register	FSSQ	R/W	✓	✓	—	00H
F02C6H	Flash end address pointer register L	FLSEDL	R/W	—	—	✓	0000H
F02C8H	Flash end address pointer register H	FLSEDH	R/W	—	✓	—	00H
F02C9H	Flash registers initialization register	FLRST	R/W	✓	✓	—	00H
F02CAH	Flash memory sequencer status register L	FSASTL	R	✓	✓	—	00H
F02CBH	Flash memory sequencer status register H	FSASTH	R	✓	✓	—	00H/04H
F02CCH	Flash write buffer register L	FLWL	R/W	—	—	✓	0000H
F02CEH	Flash write buffer register H	FLWH	R/W	—	—	✓	0000H
F02E0H	DTC base address register	DTCBAR	R/W	—	✓	—	FDH
F02E5H	PLL control register	DSCCTL	R/W	✓	✓	—	56H
F02E6H	Main clock control register	MCKC	R/W	✓	✓	—	00H
F02E8H	DTC activation enable register 0	DTCEN0	R/W	✓	✓	—	00H
F02E9H	DTC activation enable register 1	DTCEN1	R/W	✓	✓	—	00H
F02EAH	DTC activation enable register 2	DTCEN2	R/W	✓	✓	—	00H
F02EBH	DTC activation enable register 3	DTCEN3	R/W	✓	✓	—	00H
F02ECH	DTC activation enable register 4	DTCEN4	R/W	✓	✓	—	00H
F02EDH	DTC activation enable register 5	DTCEN5	R/W	✓	✓	—	00H
F02EEH	DTC activation enable register 6	DTCEN6	R/W	✓	✓	—	00H
F02F0H	Flash memory CRC control register	CRC0CTL	R/W	✓	✓	—	00H
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

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (7/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
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
F030EH	Event output destination select register 14	ELSELR14	R/W	—	✓	—	00H
F030FH	Event output destination select register 15	ELSELR15	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
F0313H	Event output destination select register 19	ELSELR19	R/W	—	✓	—	00H
F0314H	Event output destination select register 20	ELSELR20	R/W	—	✓	—	00H
F0315H	Event output destination select register 21	ELSELR21	R/W	—	✓	—	00H
F0316H	Event output destination select register 22	ELSELR22	R/W	—	✓	—	00H
F0317H	Event output destination select register 23	ELSELR23	R/W	—	✓	—	00H
F0318H	Event output destination select register 24	ELSELR24	R/W	—	✓	—	00H
F0319H	Event output destination select register 25	ELSELR25	R/W	—	✓	—	00H
F031AH	Event output destination select register 26	ELSELR26	R/W	—	✓	—	00H
F031BH	Event output destination select register 27	ELSELR27	R/W	—	✓	—	00H
F031CH	Event output destination select register 28	ELSELR28	R/W	—	✓	—	00H
F031DH	Event output destination select register 29	ELSELR29	R/W	—	✓	—	00H
F031EH	Event output destination select register 30	ELSELR30	R/W	—	✓	—	00H
F031FH	Event output destination select register 31	ELSELR31	R/W	—	✓	—	00H
F0320H	Event output destination select register 32	ELSELR32	R/W	—	✓	—	00H
F0321H	Event output destination select register 33	ELSELR33	R/W	—	✓	—	00H
F0330H	D/A converter mode register 0	DAM0	R/W	✓	✓	—	00H
F0331H	D/A converter mode register 1	DAM1	R/W	✓	✓	—	00H
F0332H	D/A converter mode register 2	DAM2	R/W	—	✓	—	00H
F0333H	D/A conversion value setting register 2	DACS2	R/W	—	✓	—	00H
F0334H	D/A conversion value setting register 0	DACS0	R/W	—	—	✓	0000H
F0335H							
F0336H	D/A conversion value setting register 1	DACS1L (DACS1)	R/W	—	✓	✓	0000H
F0337H		—		—	—		
F0340H	Comparator mode setting register 0	COMPMDR0	R/W	✓	✓	—	00H
F0341H	Comparator filter control register 0	COMPFIR0	R/W	—	✓	—	00H
F0342H	Comparator output control register 0	COMPOCR0	R/W	✓	✓	—	00H
F0344H	Comparator mode setting register 1	COMPMDR1	R/W	✓	✓	—	00H
F0345H	Comparator filter control register 1	COMPFIR1	R/W	—	✓	—	00H
F0346H	Comparator output control register 1	COMPOCR1	R/W	✓	✓	—	00H
F0347H	PGA control register	PGACTL	R/W	✓	✓	—	00H
F0348H	PGA input channel select register	PGAINS	R/W	✓	✓	—	00H
F034AH	Comparator 0 input signal selection control register	CMP0SEL	R/W	—	✓	—	00H
F034BH	Comparator 1 input signal selection control register	CMP1SEL	R/W	—	✓	—	00H
F034CH	Comparator 2 input signal selection control register	CMP2SEL	R/W	—	✓	—	00H

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (8/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F034DH	Comparator 3 input signal selection control register	CMP3SEL		R/W	—	✓	—	00H
F034EH	Comparator output control register 2	COMPOCR2		R/W	—	✓	—	00H
F0350H	Timer RX counter	TRX		R/W	—	—	✓	0000H
F0351H								
F0352H	Timer RX count buffer counter	TRXBUF		R	—	—	✓	0000H
F0353H								
F0354H	Timer RX function control register 1	TRXCR1		R/W	—	✓	—	00H
F0355H	Timer RX function control register 2	TRXCR2		R/W	✓	✓	—	00H
F0356H	Timer RX status register	TRXSR		R/W	✓	✓	—	00H
F0358H	PWMOPA control register 0	OPCTL0		R/W	—	✓	—	00H
F0359H	PWMOPA cutoff control register 0	OPDF0		R/W	—	✓	—	00H
F035AH	PWMOPA cutoff control register 1	OPDF1		R/W	—	✓	—	00H
F035BH	PWMOPA edge selection register	OPEDGE		R/W	—	✓	—	00H
F035CH	PWMOPA status register	OPSR		R	—	✓	—	00H
F0360H	Interval timer compare register 00	ITLCMP000	ITLCMP00	R/W Note 4	—	✓	✓	FFFFH
F0361H		ITLCMP001			—	✓		
F0362H	Interval timer compare register 01	ITLCMP012	ITLCMP01	R/W Note 4	—	✓	✓	FFFFH
F0363H		ITLCMP013			—	✓		
F0364H	Interval timer capture register 00	ITLCAP00		R	—	—	✓	0000H
F0365H								
F0366H	Interval timer control register	ITLCTL0		R/W	✓	✓	—	00H
F0367H	Interval timer clock select register 0	ITLCSEL0		R/W	—	✓	—	00H
F0368H	Interval timer frequency division register 0	ITLFDIV00		R/W	—	✓	—	00H
F0369H	Interval timer frequency division register 1	ITLFDIV01		R/W	—	✓	—	00H
F036AH	Interval timer capture control register 0	ITLCC0		R/W	✓	✓	—	00H
F036BH	Interval timer status register	ITLS0		R/W Note 5	—	✓	—	00H
F036CH	Interval timer match detection mask register	ITLMKF0		R/W	—	✓	—	00H
F0370H	External interrupt control register 0	INTPCTL0		R/W	—	✓	—	00H
F0371H	External interrupt control register 1	INTPCTL1		R/W	—	✓	—	00H
F0372H	External interrupt control register 2	INTPCTL2		R/W	—	✓	—	00H
F0373H	Timer clock select register 2	TPS2		R/W	—	✓	—	00H
F0390H	Timer RD ELC register	TRDELC		R/W	—	✓	—	00H
F0391H	Timer RD timer-KB PWM output gating mode control register	TRDBCR		R/W	—	✓	—	00H
F0392H	Timer RD timer KB PWM output monitor register	TRDBOF		R	—	✓	—	00H
F0393H	Timer RD start register	TRDSTR		R/W	—	✓	—	0CH
F0394H	Timer RD mode register	TRDMR		R/W	—	✓	—	00H
F0395H	Timer RD PWM function select register	TRDPMR		R/W	✓	✓	—	00H
F0396H	Timer RD function control register	TRDFCR		R/W	✓	✓	—	80H
F0397H	Timer RD output master enable register 1	TRDOER1		R/W	—	✓	—	FFH
F0398H	Timer RD output master enable register 2	TRDOER2		R/W	—	✓	—	00H
F0399H	Timer RD output control register	TRDOCR		R/W	✓	✓	—	00H
F039AH	Timer RD digital filter function select register 0	TRDDF0		R/W	—	✓	—	00H
F039BH	Timer RD digital filter function select register 1	TRDDF1		R/W	—	✓	—	00H
F03A0H	Timer RD control register 0	TRDCR0		R/W	—	✓	—	00H
F03A1H	Timer RD I/O control register A0	TRDIORA0		R/W	—	✓	—	00H

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (9/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F03A2H	Timer RD I/O control register C0	TRDIORC0		R/W	—	✓	—	88H
F03A3H	Timer RD status register 0	TRDSR0		R/W	—	✓	—	00H
F03A4H	Timer RD interrupt enable register 0	TRDIER0		R/W	—	✓	—	00H
F03A5H	Timer RD PWM output level control register 0	TRDPOCR0		R/W	—	✓	—	00H
F03A6H	Timer RD counter 0	TRD0		R/W	—	—	✓	0000H
F03A7H								
F03A8H	Timer RD general register A0	TRDGRA0		R/W	—	—	✓	FFFFH
F03A9H								
F03AAH	Timer RD general register B0	TRDGRB0		R/W	—	—	✓	FFFFH
F03ABH								
F03B0H	Timer RD control register 1	TRDCR1		R/W	—	✓	—	00H
F03B1H	Timer RD I/O control register A1	TRDIORA1		R/W	—	✓	—	00H
F03B2H	Timer RD I/O control register C1	TRDIORC1		R/W	—	✓	—	88H
F03B3H	Timer RD status register 1	TRDSR1		R/W	—	✓	—	40H
F03B4H	Timer RD interrupt enable register 1	TRDIER1		R/W	—	✓	—	00H
F03B5H	Timer RD PWM output level control register 1	TRDPOCR1		R/W	—	✓	—	00H
F03B6H	Timer RD counter 1	TRD1		R/W	—	—	✓	0000H
F03B7H								
F03B8H	Timer RD general register A1	TRDGRA1		R/W	—	—	✓	FFFFH
F03B9H								
F03BAH	Timer RD general register B1	TRDGRB1		R/W	—	—	✓	FFFFH
F03BBH								
F03C0H	Timer RD extended compare register B0	TRDCMPB0		R/W	—	—	✓	FFFFH
F03C1H								
F03C4H	Timer RD extended compare register A1	TRDCMPA1		R/W	—	—	✓	FFFFH
F03C5H								
F03C8H	Timer RD extended compare register B1	TRDCMPB1		R/W	—	—	✓	FFFFH
F03C9H								
F03CCH	Timer RD A/D conversion trigger compare register 0/timer-KB PWM output gating mode compare register	TRDADTC0/TRDCMPE1		R/W	—	—	✓	FFFFH
F03CDH								
F03D0H	Timer RD A/D conversion trigger compare register 1	TRDADTC1		R/W	—	—	✓	FFFFH
F03D1H								
F03D6H	Timer RD simultaneous update flag register 0	TRDRSF0	TRDRSF	R	—	✓	✓	0000H
F03D7H	Timer RD simultaneous update flag register 1	TRDRSF1		R	—	✓		
F03D8H	Timer RD A/D conversion trigger control register	TRDADCR		R/W	—	✓	—	00H
F03E0H	Timer RG mode register 0	TRGMR0		R/W	✓	✓	—	00H
F03E1H	Timer RG count control register	TRGCNTC		R/W	—	✓	—	00H
F03E2H	Timer RG control register	TRGCR		R/W	✓	✓	—	00H
F03E3H	Timer RG interrupt enable register 0	TRGIER0		R/W	✓	✓	—	00H
F03E4H	Timer RG status register 0	TRGSR0		R/W	✓	✓	—	00H
F03E5H	Timer RG I/O control register	TRGIOR		R/W	✓	✓	—	00H
F03E6H	Timer RG counter	TRG		R/W	—	—	✓	FFFFH
F03E7H								
F03E8H	Timer RG general register A	TRGGRA		R/W	—	—	✓	FFFFH
F03E9H								
F03EAH	Timer RG general register B	TRGGRB		R/W	—	—	✓	FFFFH
F03EBH								

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (10/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F03F0H	Timer RG mode register 1	TRGMR1	R/W	✓	✓	—	00H
F03F1H	Timer RG output enable register	TRGOER	R/W	✓	✓	—	00H
F03F2H	Timer RG output control register	TRGOCR	R/W	✓	✓	—	00H
F03F3H	Timer RG interrupt enable register 1	TRGIER1	R/W	✓	✓	—	00H
F03F4H	Timer RG status register 1	TRGSR1	R/W	✓	✓	—	00H
F03F5H	Timer RG start register	TRGSTR	R/W	✓	✓	—	02H
F03F6H	Timer RG phase counting control register 0	TRGCTL0	R/W	✓	✓	—	00H
F03F7H	Timer RG phase counting control register 1	TRGCTL1	R/W	✓	✓	—	00H
F03F8H	Timer RG phase change time measurement counter	TRGPMC	R/W	—	—	✓	0000H
F03F9H							
F03FAH	Timer RG phase change time capture register 0	TRGCAP0	R	—	—	✓	FFFFH
F03FBH							
F03FCH	Timer RG phase change time capture register 1	TRGCAP1	R	—	—	✓	FFFFH
F03FDH							
F0400H	16-bit timer KB compare register 20	TKBCR20	R/W	—	—	✓	0000H
F0401H							
F0402H	16-bit timer KB compare register 21	TKBCR21	R/W	—	—	✓	0000H
F0403H							
F0404H	16-bit timer KB compare register 22	TKBCR22	R/W	—	—	✓	0000H
F0405H							
F0406H	16-bit timer KB compare register 23	TKBCR23	R/W	—	—	✓	0000H
F0407H							
F0408H	16-bit timer KB trigger compare register 2	TKBTGCR2	R/W	—	—	✓	0000H
F0409H							
F040AH	16-bit timer KB smooth start initial duty register 20	TKBSIR20	R/W	—	—	✓	0000H
F040BH							
F040CH	16-bit timer KB smooth start initial duty register 21	TKBSIR21	R/W	—	—	✓	0000H
F040DH							
F040EH	16-bit timer KB dithering count register 20	TKBDNR20	R/W	—	✓	—	00H
F040FH	16-bit timer KB smooth start step width register 20	TKBSSR20	R/W	—	✓	—	00H
F0410H	16-bit timer KB dithering count register 21	TKBDNR21	R/W	—	✓	—	00H
F0411H	16-bit timer KB smooth start step width register 21	TKBSSR21	R/W	—	✓	—	00H
F0412H	16-bit timer KB trigger register 2	TKBTRG2	W	✓	✓	—	00H
F0413H	16-bit timer KB flag register 2	TKBFLG2	R	✓	✓	—	00H
F0414H	16-bit timer KB compare 1L & dithering count register 20	TKBCRLD20	R/W	—	—	✓	0000H
F0415H							
F0416H	16-bit timer KB compare 3L & dithering count register 21	TKBCRLD21	R/W	—	—	✓	0000H
F0417H							
F0420H	16-bit timer counter KB2	TKBCNT2	R	—	—	✓	FFFFH
F0421H							
F0422H	16-bit timer KB operation control register 20	TKBCTL20	R/W	—	—	✓	0000H
F0423H							
F0424H	16-bit timer KB maximum frequency limit setting register 2	TKBMFR2	R/W	—	—	✓	0000H
F0425H							
F0426H	16-bit timer KB output control register 20	TKBIOC20	R/W	✓	✓	—	00H
F0427H	16-bit timer KB flag clear trigger register 2	TKBCLR2	W	✓	✓	—	00H
F0428H	16-bit timer KB output control register 21	TKBIOC21	R/W	✓	✓	—	00H

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (11/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0429H	16-bit timer KB operation control register 21	TKBCTL21	R/W	✓	✓	—	00H
F042AH	16-bit timer KB operation control register 22	TKBCTL22	R/W	—	—	✓	0000H
F042BH							
F0430H	Forced output stop function control register 20	TKBPACTL20	R/W	—	—	✓	0000H
F0431H							
F0432H	Forced output stop function control register 21	TKBPACTL21	R/W	—	—	✓	0000H
F0433H							
F0434H	Forced output stop function 1 start trigger register 2	TKBPAHFS2	R/W	✓	✓	—	00H
F0435H	Forced output stop function 1 cancel trigger register 2	TKBPAHFT2	R/W	✓	✓	—	00H
F0436H	Forced output stop function flag register 2	TKBPAFLG2	R	✓	✓	—	00H
F0437H	Forced output stop function control register 22	TKBPACTL22	R/W	✓	✓	—	00H
F0438H	Forced output stop function control register 23	TKBPACTL23	R/W	—	✓	—	00H
F0439H	Forced output stop function control register 24	TKBPACTL24	R/W	—	✓	—	00H
F043AH	Pulse characteristics measurement capture register 20	TKBPAPLS20	R	—	—	✓	0000H
F043BH							
F043CH	Pulse characteristics measurement capture register 21	TKBPAPLS21	R	—	—	✓	0000H
F043DH							
F043EH	Pulse characteristics measurement capture register 20L	TKBPAPLS20L	R	—	✓	—	00H
F043FH	Pulse characteristics measurement capture register 21L	TKBPAPLS21L	R	—	✓	—	00H
F0440H	CWDW register file 0L	CWDW0L	R/W	—	—	✓	0000H
F0441H							
F0442H	CWDW register file 0H	CWDW0H	R/W	—	—	✓	0000H
F0443H							
F0444H	CWDW register file 1L	CWDW1L	R/W	—	—	✓	0000H
F0445H							
F0446H	CWDW register file 1H	CWDW1H	R/W	—	—	✓	0000H
F0447H							
F0448H	CWDW register file 2L	CWDW2L	R/W	—	—	✓	0000H
F0449H							
F044AH	CWDW register file 2H	CWDW2H	R/W	—	—	✓	0000H
F044BH							
F044CH	CWDW register file 3L	CWDW3L	R/W	—	—	✓	0000H
F044DH							
F044EH	CWDW register file 3H	CWDW3H	R/W	—	—	✓	0000H
F044FH							
F0450H	CWDW register file 4L	CWDW4L	R/W	—	—	✓	0000H
F0451H							
F0452H	CWDW register file 4H	CWDW4H	R/W	—	—	✓	0000H
F0453H							
F0454H	CWDW register file 5L	CWDW5L	R/W	—	—	✓	0000H
F0455H							
F0456H	CWDW register file 5H	CWDW5H	R/W	—	—	✓	0000H
F0457H							
F0458H	CWDW register file 6L	CWDW6L	R/W	—	—	✓	0000H
F0459H							

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (12/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F045AH	CWDW register file 6H	CWDW6H	R/W	—	—	✓	0000H
F045BH							
F045CH	CWDW register file 7L	CWDW7L	R/W	—	—	✓	0000H
F045DH							
F045EH	CWDW register file 7H	CWDW7H	R/W	—	—	✓	0000H
F045FH							
F0480H	Interrupt vector change register 0	FLSIVC0	R/W	—	—	✓	0000H
F0481H							
F0482H	Interrupt vector change register 1	FLSIVC1	R/W	—	—	✓	000FH
F0483H							
F0488H	Code flash memory guard register	GFLASH0	R/W	—	—	✓	0000H
F0489H							
F048AH	Data flash memory guard register	GFLASH1	R/W	—	—	✓	0000H
F048BH							
F048CH	Flash security area guard register	GFLASH2	R/W	—	—	✓	0000H
F048DH							
F048EH	Guard register of IAWCTL register	GIAWCTL	R/W	—	—	✓	0000H
F048FH							
F0490H	16-bit timer KB skipping control register 0	TKBTCTL0	R/W	—	✓	—	00H
F0491H	16-bit timer KB times-of-skipping setting register 0	TKBTCMP0	R/W	—	✓	—	00H
F0492H	16-bit timer KB skipping control register 1	TKBTCTL1	R/W	—	✓	—	00H
F0493H	16-bit timer KB times-of-skipping setting register 1	TKBTCMP1	R/W	—	✓	—	00H
F0494H	16-bit timer KB skipping control register 2	TKBTCTL2	R/W	—	✓	—	00H
F0495H	16-bit timer KB times-of-skipping setting register 2	TKBTCMP2	R/W	—	✓	—	00H
F0498H	Timer RD skipping control register	TRDTCTL	R/W	—	✓	—	00H
F0499H	Timer RD skipping count setting register	TRDTCMP	R/W	—	✓	—	00H
F04A0H	Window register	WIND	R/W	—	—	✓	0000H
F04A1H							
F04B0H	Address bus select register	ADBSEL	R/W	—	—	✓	0000H
F04B1H							
F04B2H	RAM parity error control register 2	RPECTL2	R/W	✓	✓	—	00H
F04B3H	Division control register	FAADUC	R	—	✓	—	00H
F04B4H	Division data register CL	FAADCL	R	—	—	✓	0000H
F04B5H							
F04B6H	Division data register CH	FAADCH	R	—	—	✓	0000H
F04B7H							
F04B8H	Division data register AL	FAADAL	R	—	—	✓	0000H
F04B9H							
F04BAH	Division data register AH	FAADAH	R	—	—	✓	0000H
F04BBH							
F04BCH	Division data register BL	FAADBL	R	—	—	✓	0000H
F04BDH							
F04BEH	Division data register BH	FAADBH	R	—	—	✓	0000H
F04BFH							
F04C0H	DALI bit timing violation threshold register 1	BTVTHR1	R/W	—	—	✓	4F00H
F04C1H							

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (13/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F04C2H	DALI bit timing violation threshold register 2	BTVTHR2	R/W	—	—	✓	654FH
F04C3H							
F04C4H	DALI bit timing violation threshold register 3	BTVTHR3	R/W	—	—	✓	009DH
F04C5H							
F04C6H	DALI bit timing violation threshold register 4	BTVTHR4	R/W	—	—	✓	00DBH
F04C7H							
F04C8H	DALI collision threshold register 1	COLTHR1	R/W	—	—	✓	380FH
F04C9H							
F04CAH	DALI collision threshold register 2	COLTHR2	R/W	—	—	✓	443CH
F04CBH							
F04CCH	DALI collision threshold register 3	COLTHR3	R/W	—	—	✓	7148H
F04CDH							
F04CEH	DALI collision threshold register 4	COLTHR4	R/W	—	—	✓	8879H
F04CFH							
F04D0H	DALI collision threshold register 5	COLTHR5	R/W	—	—	✓	008EH
F04D1H							
F04D2H	DALI configuration register 1	CNFR1	R/W	—	—	✓	00FFH
F04D3H							
F04D4H	DALI configuration register 2	CNFR2	R/W	—	—	✓	0000H
F04D5H							
F04D6H	DALITxD0 waveform adjustment register 1	TXWR1	R/W	—	—	✓	003FH
F04D7H							
F04D8H	DALIRxD0 waveform adjustment register 1	RXWR1	R/W	—	—	✓	3F00H
F04D9H							
F04DEH	DALI transmit data register 1H	TDR1H	R/W	—	—	✓	0000H
F04DFH							
F04E0H	DALI transmit data register 1L	TDR1L	R/W	—	—	✓	0000H
F04E1H							
F04E2H	DALI transmit control register 1	TRSTR1	W	—	—	✓	0000H
F04E3H							
F04E4H	DALI reception timing adjustment register 0	FTDC0	R/W	—	—	✓	0000H
F04E5H							
F04E6H	DALI control register 1	CTR1	R/W	—	—	✓	0000H
F04E7H							
F04E8H	DALITxD0 control register 1	TXDCTR1	R/W	—	—	✓	0000H
F04E9H							
F04EEH	DALI reception data register 1H	RDR1H	R	—	—	✓	0000H
F04EFH							
F04F0H	DALI reception data register 1L	RDR1L	R	—	—	✓	0000H
F04F1H							
F04F2H	DALI status register 1	STR1	R	—	—	✓	0000H
F04F3H							
F04F4H	DALI status register 2	STR2	R	—	—	✓	0000H
F04F5H							
F04F6H	DALI collision register 1	COLR1	R	—	—	✓	0800H
F04F7H							
F04FAH	DALI flag error clearing register 1	FECR1	W	—	—	✓	0800H
F04FBH							

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (14/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F04FCH	DALI software reset register 1	SWRR1	W	—	—	✓	0800H
F04FDH							
F0500H	Timer RJ counter register 0	TRJ0	R/W	—	—	✓	FFFFH
F0501H							
F0540H	Random number seed data register	TRNGSDR	R	✓	✓	—	00H
F0542H	Random number seed command register 0	TRNGSCR0	R/W	✓	✓	—	00H
F0600H	Accumulator register L	A0L	R/W	—	—	✓	0000H
F0601H							
F0602H	Accumulator register H	A0H	R/W	—	—	✓	0000H
F0603H							
F0604H	Multiplier register L/timing comparison register 0L	M0L/TMCMP0L	R/W	—	—	✓	0000H
F0605H							
F0606H	Multiplier register H/timing comparison register 0H	M0H/TMCMP0H	R/W	—	—	✓	0000H
F0607H							
F0608H	Shift count register L/timing comparison register 1L	M1L/TMCMP1L	R/W	—	—	✓	0000H
F0609H							
F060AH	Shift count register H/timing comparison register 1H	M1H/TMCMP1H	R/W	—	—	✓	0000H
F060BH							
F060CH	Upper limit register L/timing comparison register 2L	L0L/TMCMP2L	R/W	—	—	✓	0000H
F060DH							
F060EH	Upper limit register H/timing comparison register 2H	L0H/TMCMP2H	R/W	—	—	✓	0000H
F060FH							
F0610H	Lower limit register L/timing comparison register 3L	L1L/TMCMP3L	R/W	—	—	✓	0000H
F0611H							
F0612H	Lower limit register H/timing comparison register 3H	L1H/TMCMP3H	R/W	—	—	✓	0000H
F0613H							
F0614H	Addend register L/timing comparison register 4L	R0L/TMCMP4L	R/W	—	—	✓	0000H
F0615H							
F0616H	Addend register H/timing comparison register 4H	R0H/TMCMP4H	R/W	—	—	✓	0000H
F0617H							
F0618H	Timing comparison register 5L	TMCMP5L	R/W	—	—	✓	0000H
F0619H							
F061AH	Address pointer for accumulator/timing comparison register 5H	DP0/TMCMP5H	R/W	—	—	✓	0000H
F061BH							
F061CH	Timing comparison mask register 0L	TMMSK0L	R/W	—	—	✓	0000H
F061DH							
F061EH	Address pointer for operation parameters/ timing comparison mask register 0H	DP1/TMMSK0H	R/W	—	—	✓	0000H
F061FH							
F0620H	Timing comparison mask register 1L	TMMSK1L	R/W	—	—	✓	0000H
F0621H							
F0622H	Address pointer for storing operation results/ timing comparison mask register 1H	DRP0/TMMSK1H	R/W	—	—	✓	0000H
F0623H							
F0624H	Timing comparison mask register 2L	TMMSK2L	R/W	—	—	✓	0000H
F0625H							
F0626H	Timing comparison mask register 2H	TMMSK2H	R/W	—	—	✓	0000H
F0627H							
F0628H	Timing comparison mask register 3L	TMMSK3L	R/W	—	—	✓	0000H
F0629H							

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (15/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F062AH	Processor control register/timing comparison mask register 3H	FAACNT/TMMSK3H	R/W	—	—	✓	0000H
F062BH							
F062CH	Timing comparison mask register 4L	TMMSK4L	R/W	—	—	✓	0000H
F062DH							
F062EH	Program pointer/timing comparison mask register 4H	PG0/TMMSK4H	R/W	—	—	✓	0000H
F062FH							
F0630H	Timing comparison mask register 5L	TMMSK5L	R/W	—	—	✓	0000H
F0631H							
F0632H	Timing comparison mask register 5H	TMMSK5H	R/W	—	—	✓	0000H
F0633H							
F0634H	Free-running counter register L	FCNTL	R/W	—	—	✓	0000H
F0635H							
F0636H	Flag bit register/free-running counter register H	FAAFLG/FCNTH	R/W	—	—	✓	0000H
F0637H							
F0638H	Free-running counter control register	FCCNT	R/W	—	—	✓	0000H
F0639H							
F0648H	Interrupt vector register 0L	IV0L	R/W	—	—	✓	0000H
F0649H							
F064AH	Interrupt vector register 0H	IV0H	R/W	—	—	✓	0000H
F064BH							
F064CH	Interrupt vector register 1L	IV1L	R/W	—	—	✓	0000H
F064DH							
F064EH	Interrupt vector register 1H	IV1H	R/W	—	—	✓	0000H
F064FH							
F0650H	Interrupt vector register 2L	IV2L	R/W	—	—	✓	0000H
F0651H							
F0652H	Interrupt vector register 2H	IV2H	R/W	—	—	✓	0000H
F0653H							
F0654H	Interrupt vector register 3L	IV3L	R/W	—	—	✓	0000H
F0655H							
F0656H	Interrupt vector register 3H	IV3H	R/W	—	—	✓	0000H
F0657H							
F0658H	Interrupt vector register 4L	IV4L	R/W	—	—	✓	0000H
F0659H							
F065AH	Interrupt vector register 4H	IV4H	R/W	—	—	✓	0000H
F065BH							
F065CH	Interrupt vector register 5L	IV5L	R/W	—	—	✓	0000H
F065DH							
F065EH	Interrupt vector register 5H	IV5H	R/W	—	—	✓	0000H
F065FH							
F0660H	Interrupt vector register 6L	IV6L	R/W	—	—	✓	0000H
F0661H							
F0662H	Interrupt vector register 6H	IV6H	R/W	—	—	✓	0000H
F0663H							
F0664H	Interrupt vector register 7L	IV7L	R/W	—	—	✓	0000H
F0665H							
F0666H	Interrupt vector register 7H	IV7H	R/W	—	—	✓	0000H
F0667H							

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (16/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0668H	Interrupt vector register 8L	IV8L	R/W	—	—	✓	0000H
F0669H							
F066AH	Interrupt vector register 8H	IV8H	R/W	—	—	✓	0000H
F066BH							
F066CH	Interrupt vector register 9L	IV9L	R/W	—	—	✓	0000H
F066DH							
F066EH	System control register/interrupt vector register 9H	DSYSC/IV9H	R/W	—	—	✓	0000H
F066FH							
F0670H	Interrupt vector register 10L	IV10L	R/W	—	—	✓	0000H
F0671H							
F0672H	Interrupt vector register 10H	IV10H	R/W	—	—	✓	0000H
F0673H							
F0674H	Interrupt vector register 11L	IV11L	R/W	—	—	✓	0000H
F0675H							
F0676H	Interrupt vector register 11H	IV11H	R/W	—	—	✓	0000H
F0677H							
F0678H	Interrupt vector register 12L	IV12L	R/W	—	—	✓	0000H
F0679H							
F067AH	Interrupt vector register 12H	IV12H	R/W	—	—	✓	0000H
F067BH							
F067CH	Interrupt vector register 13L	IV13L	R/W	—	—	✓	0000H
F067DH							
F067EH	Interrupt vector register 13H	IV13H	R/W	—	—	✓	0000H
F067FH							
F0680H	Interrupt vector register 14L	IV14L	R/W	—	—	✓	0000H
F0681H							
F0682H	Stack pointer/interrupt vector register 14H	SP0/IV14H	R/W	—	—	✓	0000H
F0683H							
F0684H	Interrupt vector register 15L	IV15L	R/W	—	—	✓	0000H
F0685H							
F0686H	Interrupt vector register 15H	IV15H	R/W	—	—	✓	0000H
F0687H							
F06A0H	Sense control register 0L	IEVSC0L	R/W	—	—	✓	0000H
F06A1H							
F06A2H	Sense control register 0H	IEVSC0H	R/W	—	—	✓	0000H
F06A3H							
F06A4H	Sense control register 1	IEVSC1	R/W	—	—	✓	0000H
F06A5H							
F0740H	16-bit timer KB compare register 00	TKBCR00	R/W	—	—	✓	0000H
F0741H							
F0742H	16-bit timer KB compare register 01	TKBCR01	R/W	—	—	✓	0000H
F0743H							
F0744H	16-bit timer KB compare register 02	TKBCR02	R/W	—	—	✓	0000H
F0745H							
F0746H	16-bit timer KB compare register 03	TKBCR03	R/W	—	—	✓	0000H
F0747H							
F0748H	16-bit timer KB trigger compare register 0	TKBTGCR0	R/W	—	—	✓	0000H
F0749H							

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (17/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F074AH	16-bit timer KB smooth start initial duty register 00	TKBSIR00	R/W	—	—	✓	0000H
F074BH							
F074CH	16-bit timer KB smooth start initial duty register 01	TKBSIR01	R/W	—	—	✓	0000H
F074DH							
F074EH	16-bit timer KB dithering count register 00	TKBDNR00	R/W	—	✓	—	00H
F074FH	16-bit timer KB smooth start step width register 00	TKBSSR00	R/W	—	✓	—	00H
F0750H	16-bit timer KB dithering count register 01	TKBDNR01	R/W	—	✓	—	00H
F0751H	16-bit timer KB smooth start step width register 01	TKBSSR01	R/W	—	✓	—	00H
F0752H	16-bit timer KB trigger register 0	TKBTRG0	W	✓	✓	—	00H
F0753H	16-bit timer KB flag register 0	TKBFLG0	R	✓	✓	—	00H
F0754H	16-bit timer KB compare 1L & dithering count register 00	TKBCRLD00	R/W	—	—	✓	0000H
F0755H							
F0756H	16-bit timer KB compare 3L & dithering count register 01	TKBCRLD01	R/W	—	—	✓	0000H
F0757H							
F0760H	16-bit timer counter KB0	TKBCNT0	R	—	—	✓	FFFFH
F0761H							
F0762H	16-bit timer KB operation control register 00	TKBCTL00	R/W	—	—	✓	0000H
F0763H							
F0764H	16-bit timer KB maximum frequency limit setting register 0	TKBMFR0	R/W	—	—	✓	0000H
F0765H							
F0766H	16-bit timer KB output control register 00	TKBIOC00	R/W	✓	✓	—	00H
F0767H	16-bit timer KB flag clear trigger register 0	TKBCLR0	W	✓	✓	—	00H
F0768H	16-bit timer KB output control register 01	TKBIOC01	R/W	✓	✓	—	00H
F0769H	16-bit timer KB operation control register 01	TKBCTL01	R/W	✓	✓	—	00H
F076AH	16-bit timer KB operation control register 02	TKBCTL02	R/W	—	—	✓	0000H
F076BH							
F0770H	Forced output stop function control register 00	TKBPACTL00	R/W	—	—	✓	0000H
F0771H							
F0772H	Forced output stop function control register 01	TKBPACTL01	R/W	—	—	✓	0000H
F0773H							
F0774H	Forced output stop function 1 start trigger register 0	TKBPAHFS0	R/W	✓	✓	—	00H
F0775H	Forced output stop function 1 cancel trigger register 0	TKBPAHFT0	R/W	✓	✓	—	00H
F0776H	Forced output stop function flag register 0	TKBPAFLG0	R	✓	✓	—	00H
F0777H	Forced output stop function control register 02	TKBPACTL02	R/W	✓	✓	—	00H
F0778H	Forced output stop function control register 03	TKBPACTL03	R/W	—	✓	—	00H
F0779H	Forced output stop function control register 04	TKBPACTL04	R/W	—	✓	—	00H
F077AH	Pulse characteristics measurement capture register 00	TKBPAPLS00	R	—	—	✓	0000H
F077BH							
F077CH	Pulse characteristics measurement capture register 01	TKBPAPLS01	R	—	—	✓	0000H
F077DH							
F077EH	Pulse characteristics measurement capture register 00L	TKBPAPLS00L	R	—	✓	—	00H
F077FH	Pulse characteristics measurement capture register 01L	TKBPAPLS01L	R	—	✓	—	00H
F0780H	16-bit timer KB compare register 10	TKBCR10	R/W	—	—	✓	0000H
F0781H							

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (18/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0782H	16-bit timer KB compare register 11	TKBCR11	R/W	—	—	✓	0000H
F0783H							
F0784H	16-bit timer KB compare register 12	TKBCR12	R/W	—	—	✓	0000H
F0785H							
F0786H	16-bit timer KB compare register 13	TKBCR13	R/W	—	—	✓	0000H
F0787H							
F0788H	16-bit timer KB trigger compare register 1	TKBTGCR1	R/W	—	—	✓	0000H
F0789H							
F078AH	16-bit timer KB smooth start initial duty register 10	TKBSIR10	R/W	—	—	✓	0000H
F078BH							
F078CH	16-bit timer KB smooth start initial duty register 11	TKBSIR11	R/W	—	—	✓	0000H
F078DH							
F078EH	16-bit timer KB dithering count register 10	TKBDNR10	R/W	—	✓	—	00H
F078FH	16-bit timer KB smooth start step width register 10	TKBSSR10	R/W	—	✓	—	00H
F0790H	16-bit timer KB dithering count register 11	TKBDNR11	R/W	—	✓	—	00H
F0791H	16-bit timer KB smooth start step width register 11	TKBSSR11	R/W	—	✓	—	00H
F0792H	16-bit timer KB trigger register 1	TKBTRG1	W	✓	✓	—	00H
F0793H	16-bit timer KB flag register 1	TKBFLG1	R	✓	✓	—	00H
F0794H	16-bit timer KB compare 1L & dithering count register 10	TKBCRLD10	R/W	—	—	✓	0000H
F0795H							
F0796H	16-bit timer KB compare 3L & dithering count register 11	TKBCRLD11	R/W	—	—	✓	0000H
F0797H							
F07A0H	16-bit timer counter KB1	TKBCNT1	R	—	—	✓	FFFFH
F07A1H							
F07A2H	16-bit timer KB operation control register 10	TKBCTL10	R/W	—	—	✓	0000H
F07A3H							
F07A4H	16-bit timer KB maximum frequency limit setting register 1	TKBMFR1	R/W	—	—	✓	0000H
F07A5H							
F07A6H	16-bit timer KB output control register 10	TKBIOC10	R/W	✓	✓	—	00H
F07A7H	16-bit timer KB flag clear trigger register 1	TKBCLR1	W	✓	✓	—	00H
F07A8H	16-bit timer KB output control register 11	TKBIOC11	R/W	✓	✓	—	00H
F07A9H	16-bit timer KB operation control register 11	TKBCTL11	R/W	✓	✓	—	00H
F07AAH	16-bit timer KB operation control register 12	TKBCTL12	R/W	—	—	✓	0000H
F07ABH							
F07B0H	Forced output stop function control register 10	TKBPACTL10	R/W	—	—	✓	0000H
F07B1H							
F07B2H	Forced output stop function control register 11	TKBPACTL11	R/W	—	—	✓	0000H
F07B3H							
F07B4H	Forced output stop function 1 start trigger register 1	TKBPAHFS1	R/W	✓	✓	—	00H
F07B5H	Forced output stop function 1 cancel trigger register 1	TKBPAHFT1	R/W	✓	✓	—	00H
F07B6H	Forced output stop function flag register 1	TKBPAFLG1	R	✓	✓	—	00H
F07B7H	Forced output stop function control register 12	TKBPACTL12	R/W	✓	✓	—	00H
F07B8H	Forced output stop function control register 13	TKBPACTL13	R/W	—	✓	—	00H
F07B9H	Forced output stop function control register 14	TKBPACTL14	R/W	—	✓	—	00H

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (19/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F07BAH F07BBH	Pulse characteristics measurement capture register 10	TKBPAPLS10	R	—	—	✓	0000H
F07BCH F07BDH	Pulse characteristics measurement capture register 11	TKBPAPLS11	R	—	—	✓	0000H
F07BEH	Pulse characteristics measurement capture register 10L	TKBPAPLS10L	R	—	✓	—	00H
F07BFH	Pulse characteristics measurement capture register 11L	TKBPAPLS11L	R	—	✓	—	00H

Note 1. The value after a reset is adjusted at the time of shipment.

Note 2. The value after a reset is the value set in the FRQSEL[2:0] bits of the user option byte (000C2H).

Note 3. The initial value of the FLMODE register is set to the value of the MODE[1:0] bits updated with the set value of the CMODE[1:0] bits in the option byte at address 000C2H.

Note 4. This register is only accessible when the values of the mode selection bits ITLMD00 and ITLMD01 in the interval timer control register (ITLCTL0) are 1 and 0.

Note 5. Writing 1 to any bit of this register is ignored. To clear the ITF0C and ITF0i flags, use an 8-bit memory manipulation instruction to write 0 to the corresponding bit and 1 to the other bits.

Note 6. The value of bit 1 after a reset is the value set in the FRQSEL4 bit of the user option byte (000C2H).

Note 7. The RTCLPC and WUTMMCK bits have the value 0 and the HIPREC bit has the value 1 following a reset.

Remark For SFRs in the SFR area, see **Table 3 - 5 List of Special Function Registers (SFRs)**.

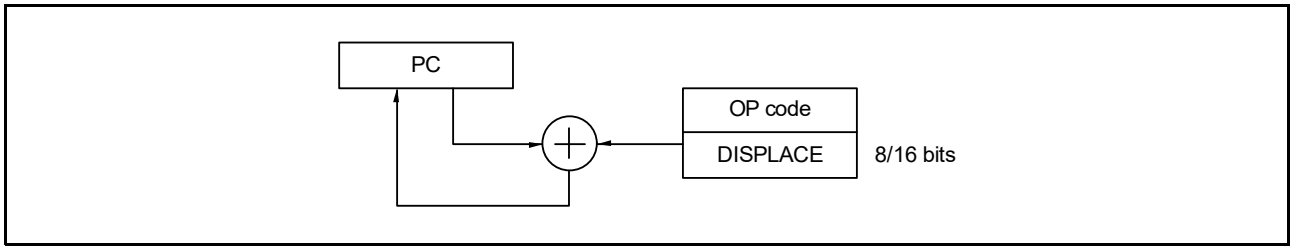
3.3 Instruction Address Addressing

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 - 11 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 - 12 Example of CALL !!addr20/BR !!addr20

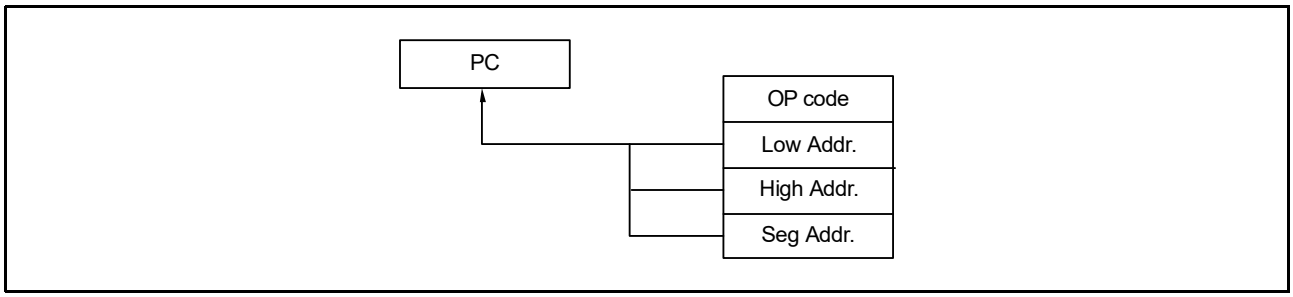
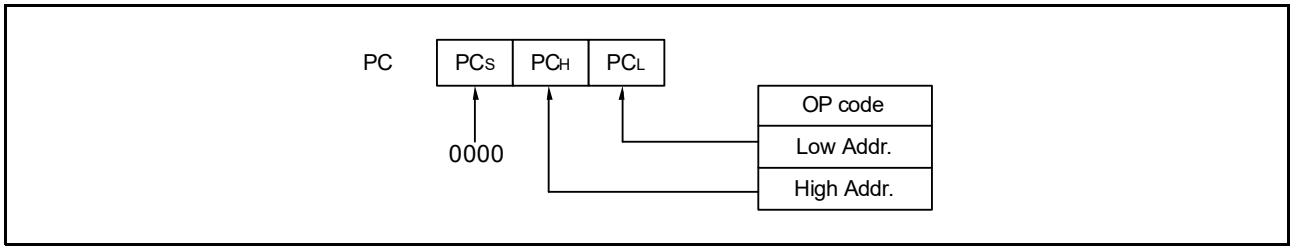


Figure 3 - 13 Example of CALL !addr16/BR !addr16



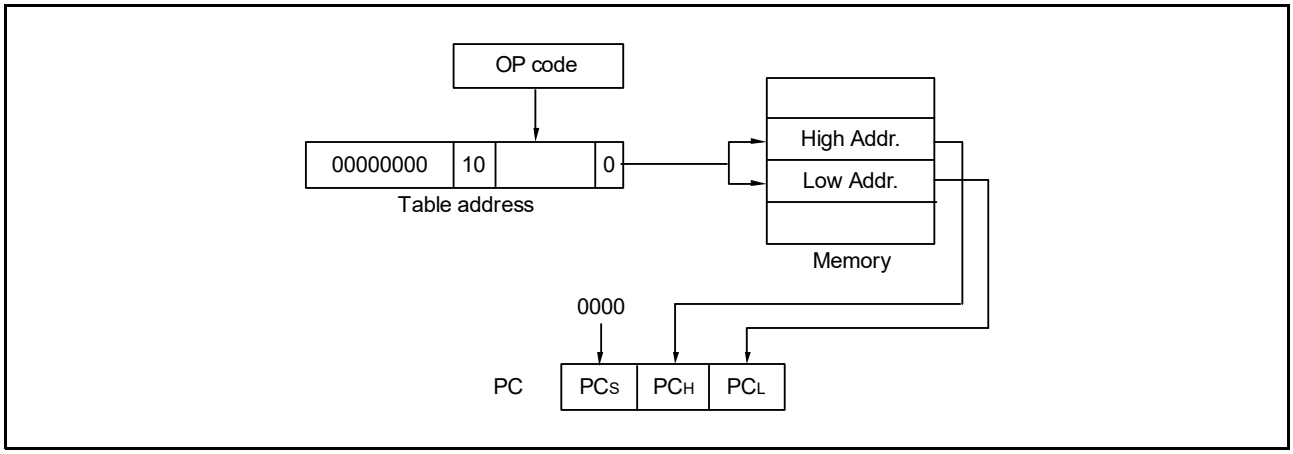
3.3.3 Table indirect addressing

[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-Kbyte space from 00000H to 0FFFFH.

Figure 3 - 14 Outline of Table Indirect Addressing

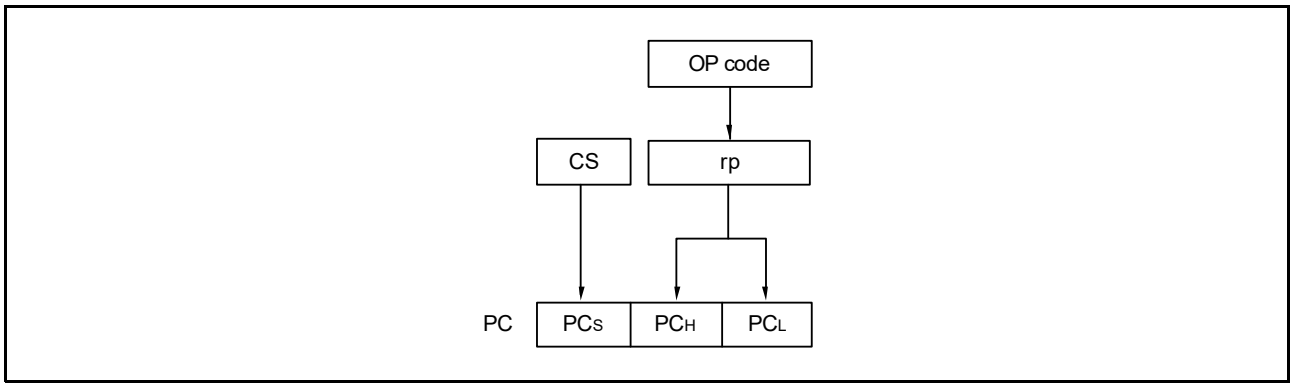


3.3.4 Register indirect addressing

[Function]

Register indirect 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 indirect addressing can be applied only to the CALL AX, BC, DE, HL, and BR AX instructions.

Figure 3 - 15 Outline of Register Indirect Addressing



3.4 Addressing for Processing Data Addresses

3.4.1 Implied addressing

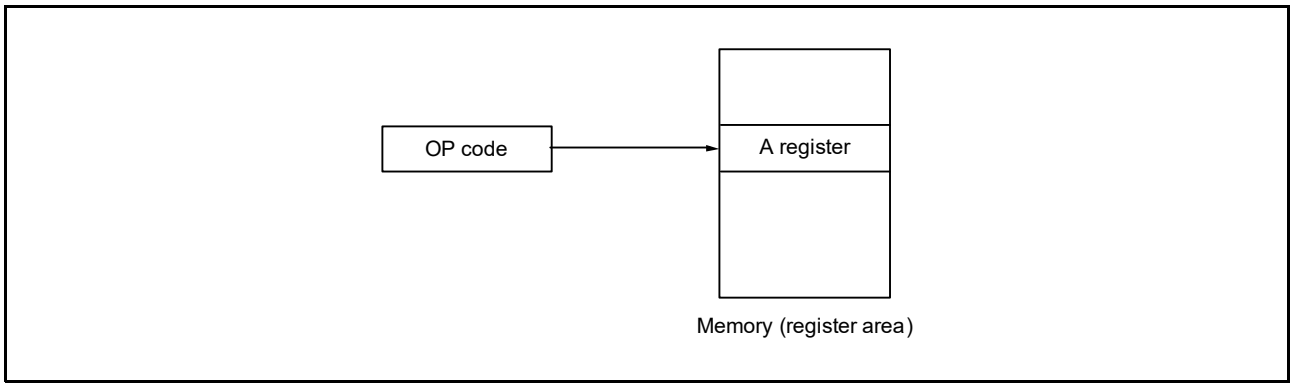
[Function]

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]

Implied addressing can be applied only to MULU X.

Figure 3 - 16 Outline of Implied Addressing



3.4.2 Register addressing

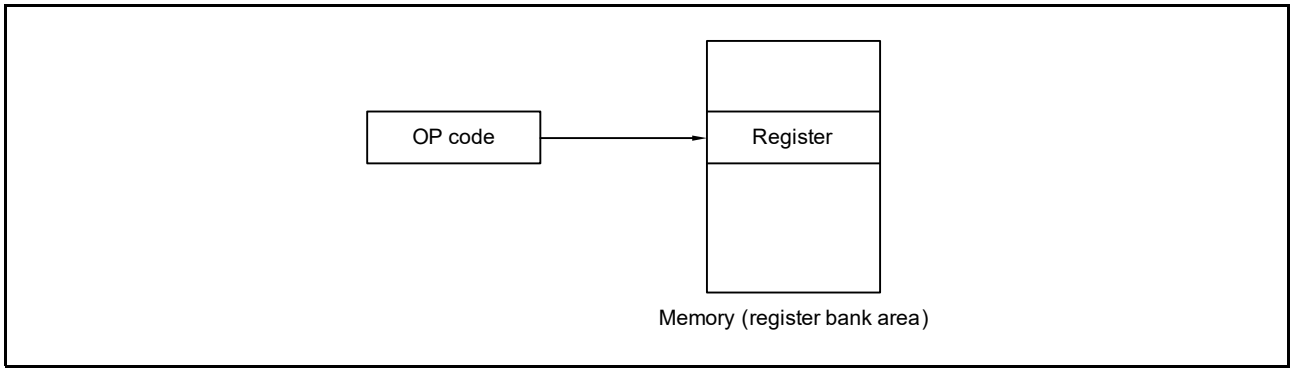
[Function]

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 - 17 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 - 18 Example of !addr16

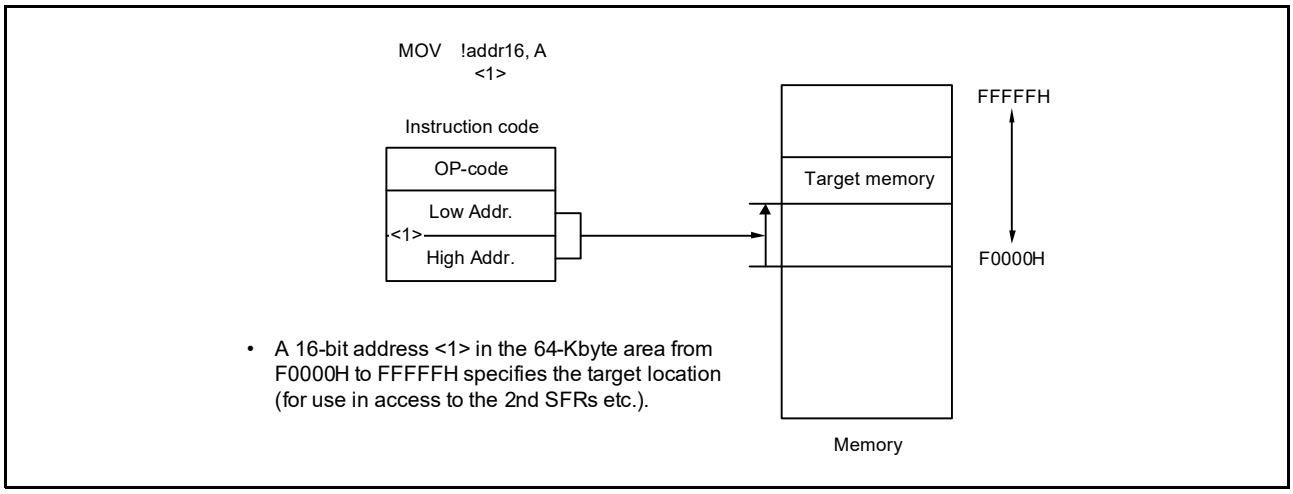
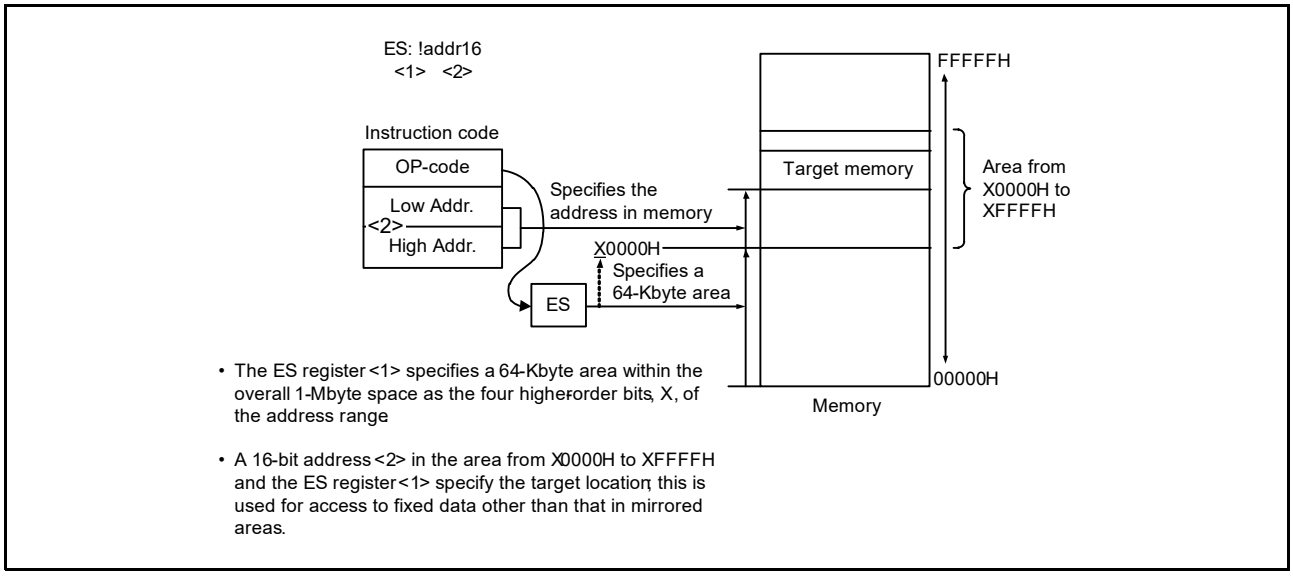


Figure 3 - 19 Example of ES:!addr16



3.4.4 Short direct addressing

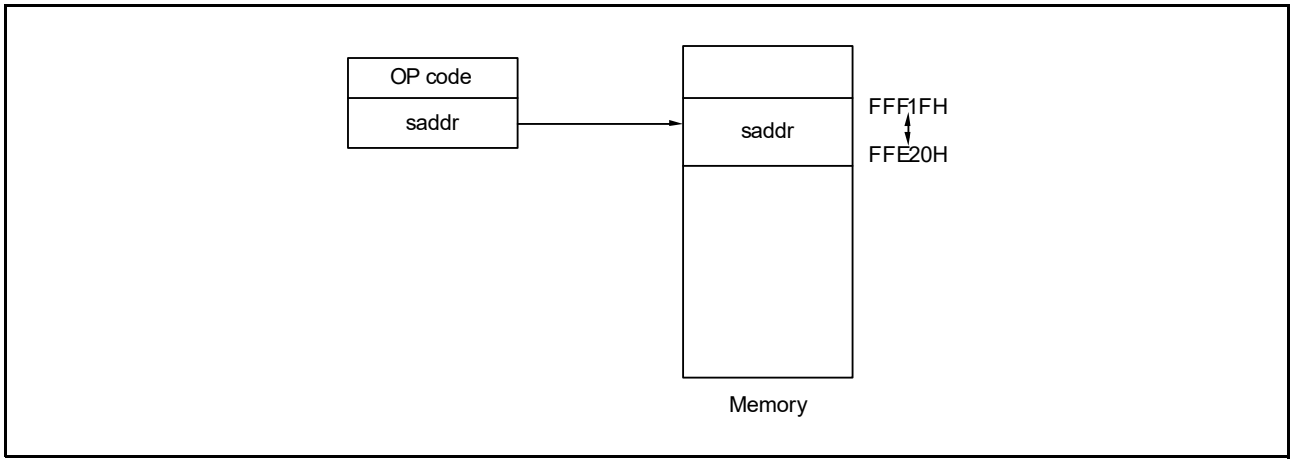
[Function]

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 - 20 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

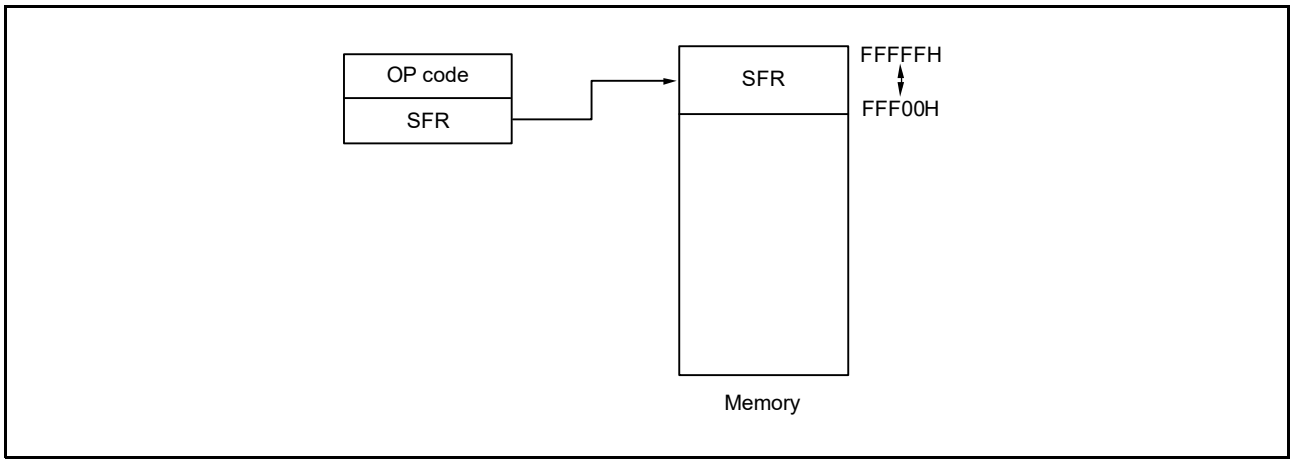
[Function]

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)

Figure 3 - 21 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 - 22 Example of [DE], [HL]

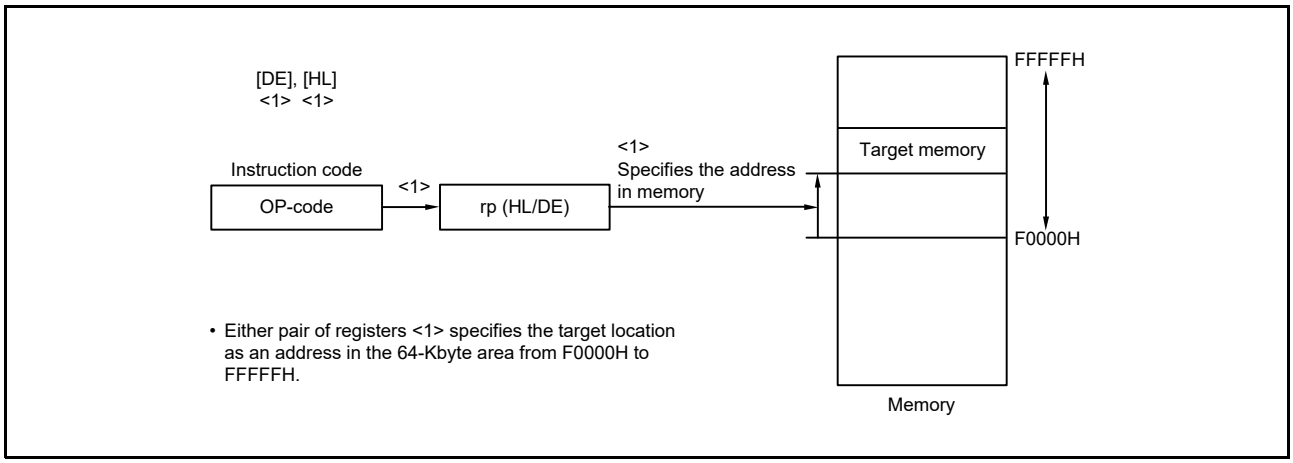
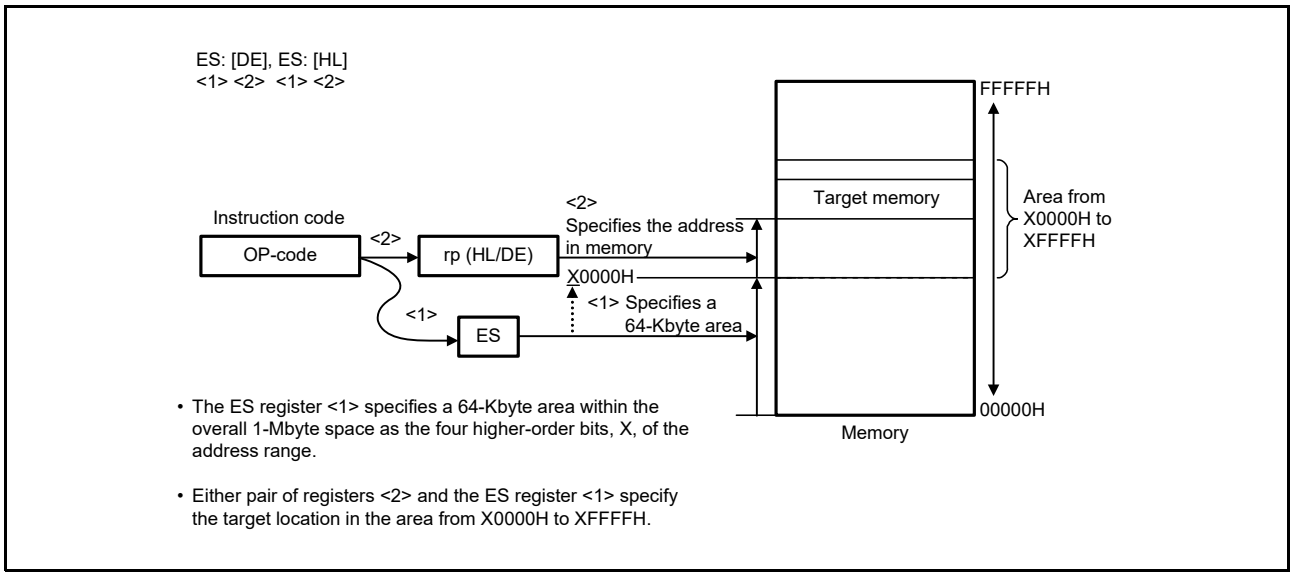


Figure 3 - 23 Example of ES:[DE], ES:[HL]



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 - 24 Example of [SP + byte]

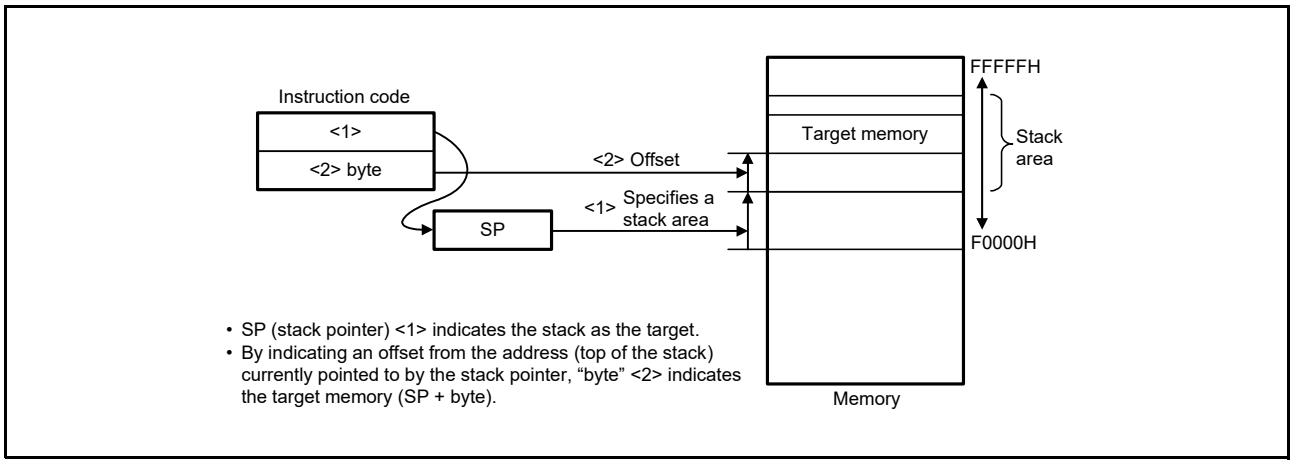


Figure 3 - 25 Example of [HL + byte], [DE + byte]

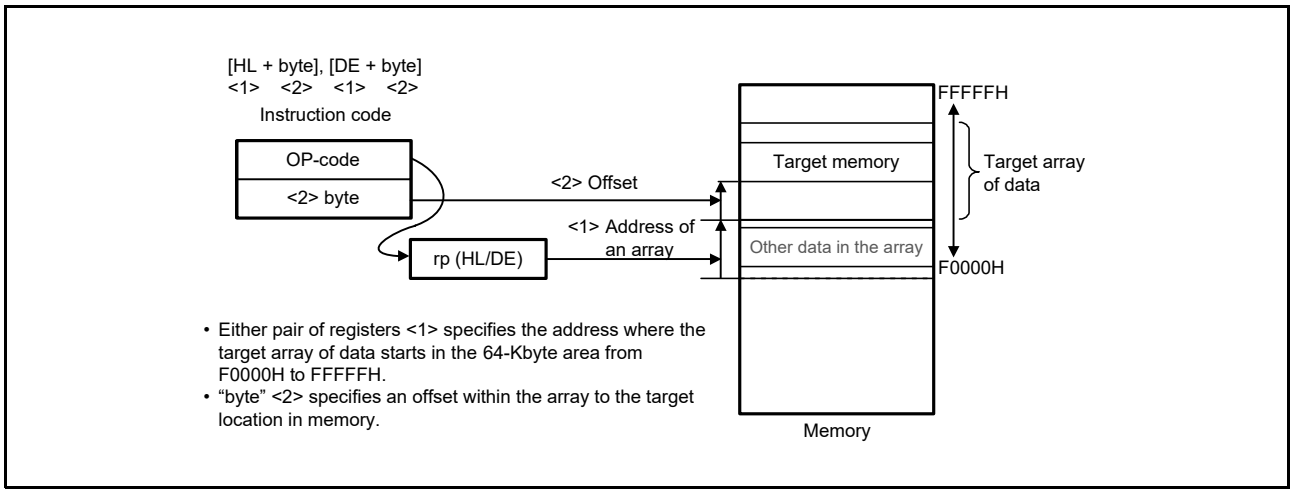


Figure 3 - 26 Example of word[B], word[C]

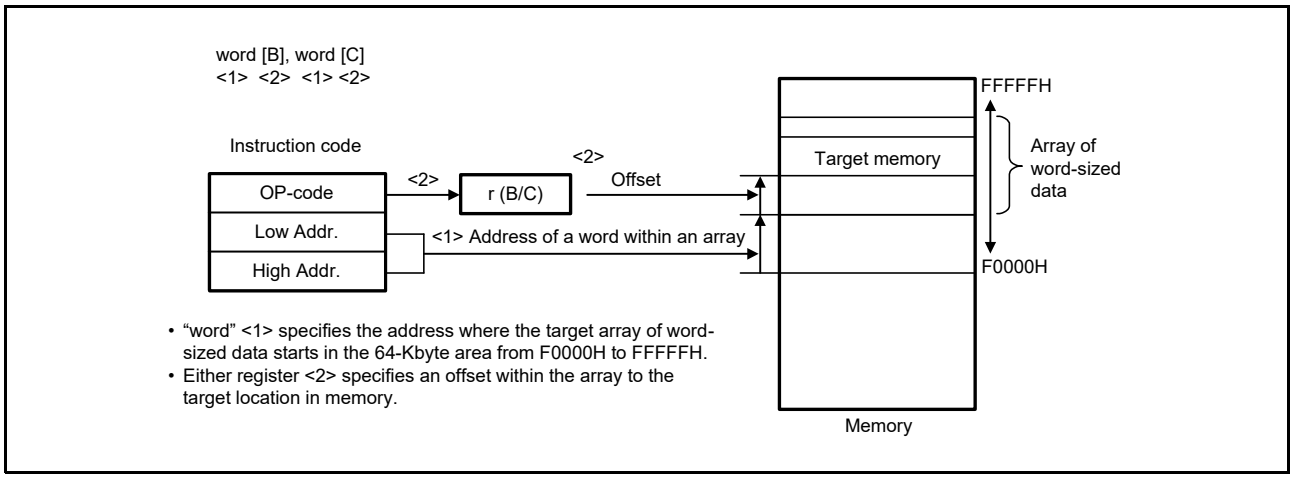


Figure 3 - 27 Example of word[BC]

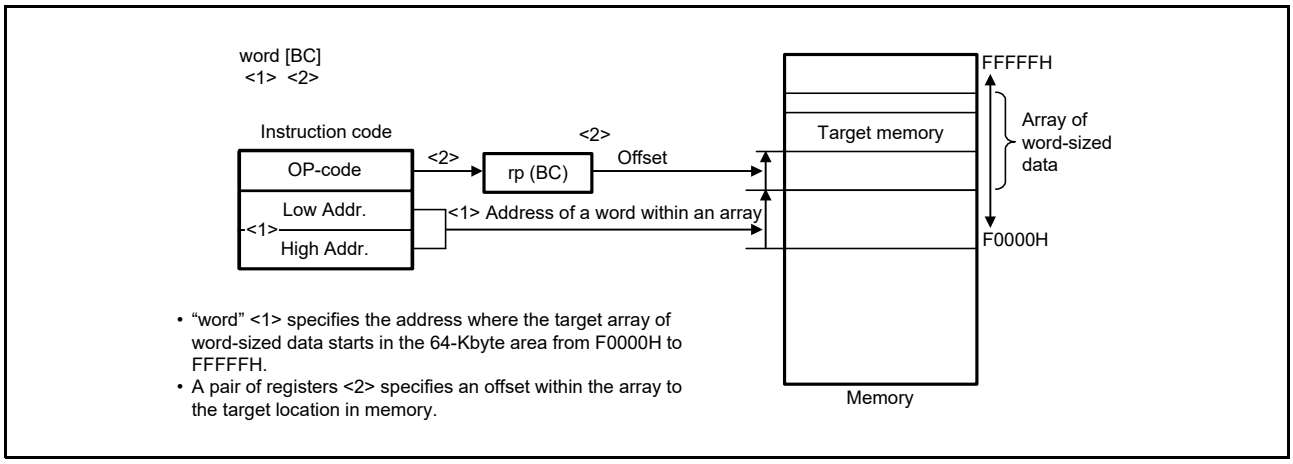


Figure 3 - 28 Example of ES:[HL + byte], ES:[DE + byte]

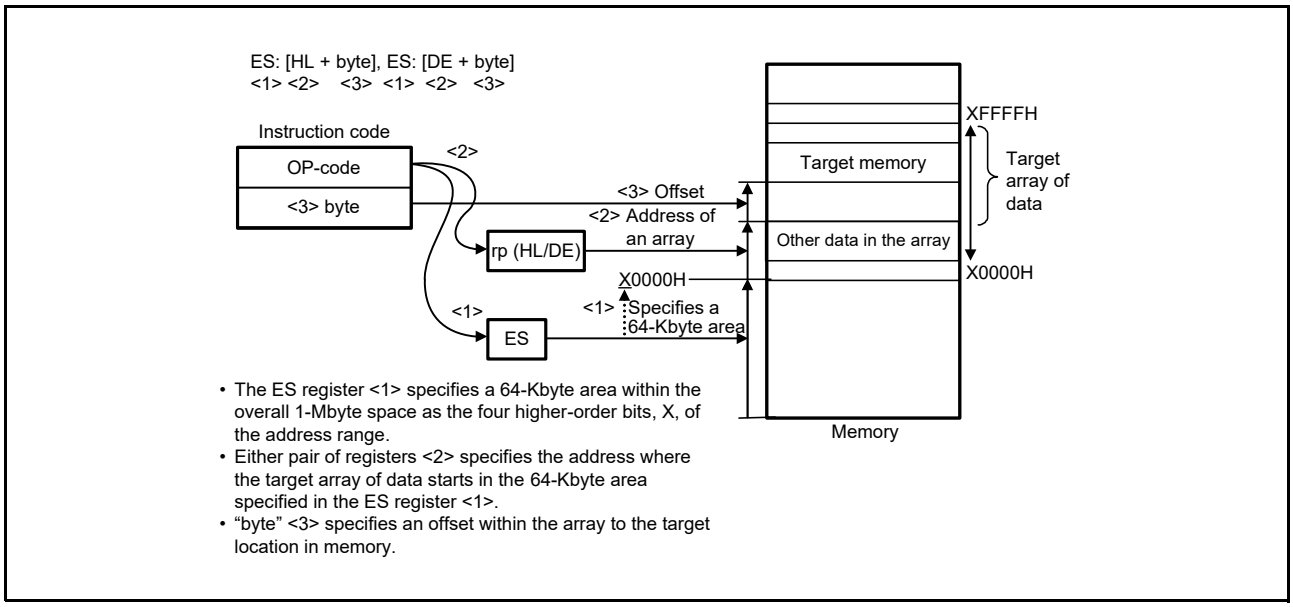
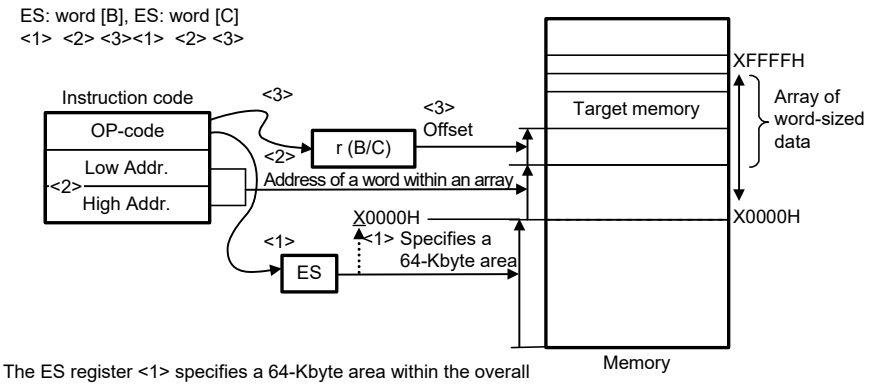
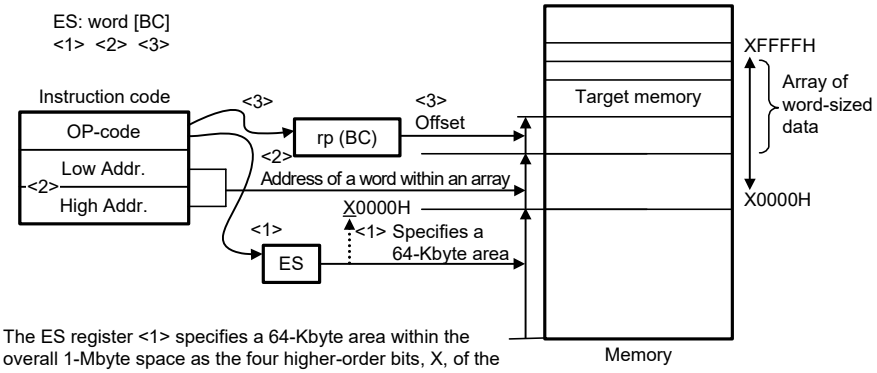


Figure 3 - 29 Example of ES:word[B], ES:word[C]



- 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 - 30 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 - 31 Example of [HL + B], [HL + C]

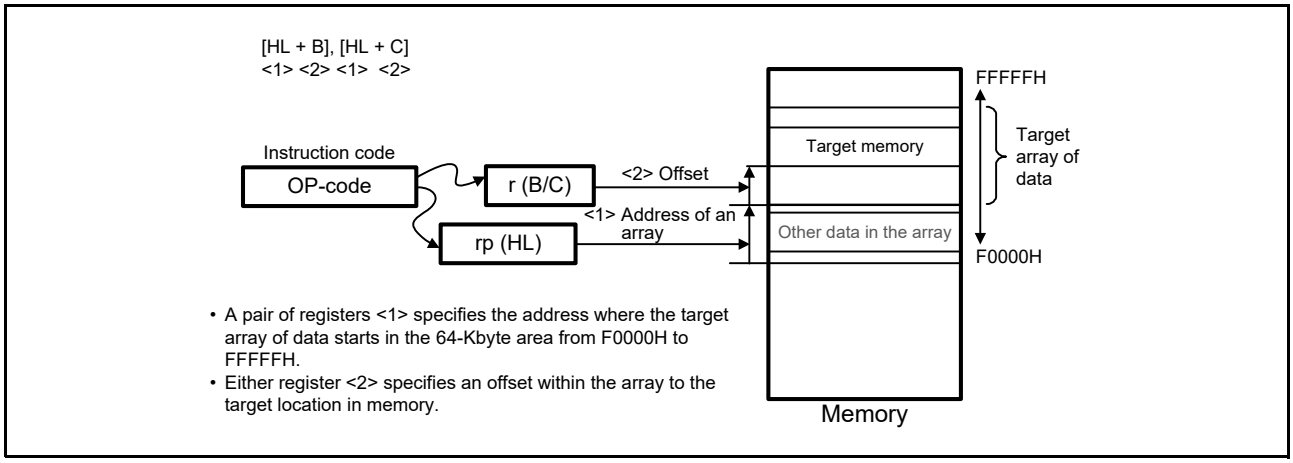
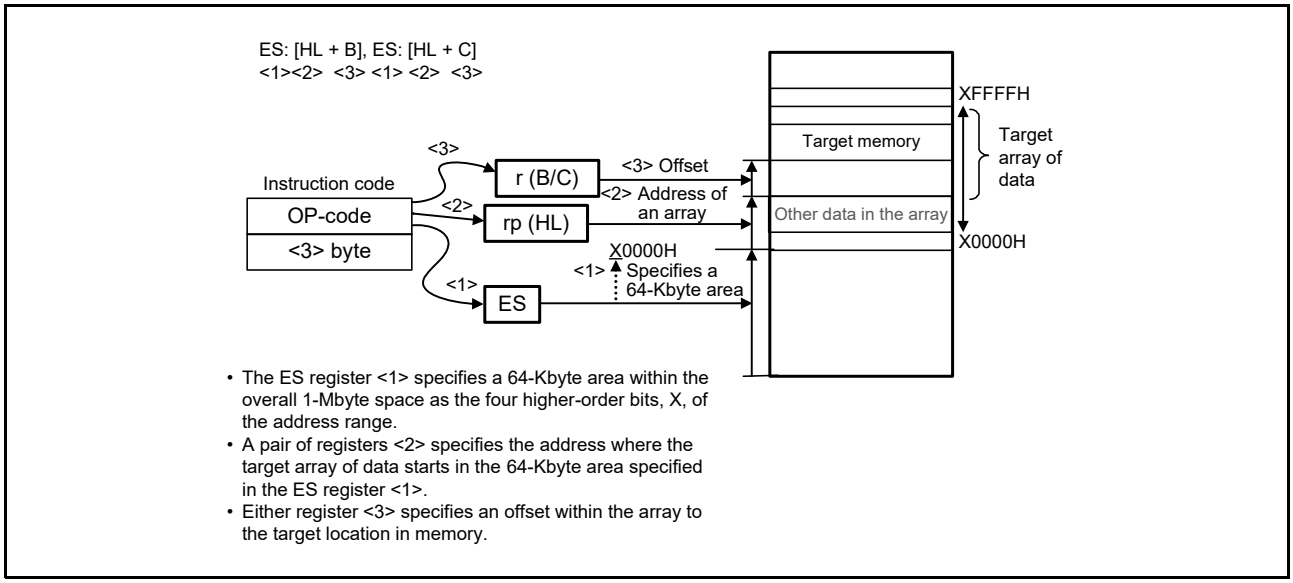


Figure 3 - 32 Example of ES:[HL + B], ES:[HL + C]



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 PSW AX/BC/DE/HL POP PSW 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 - 33** to **3 - 38**.

Figure 3 - 33 Example of PUSH rp

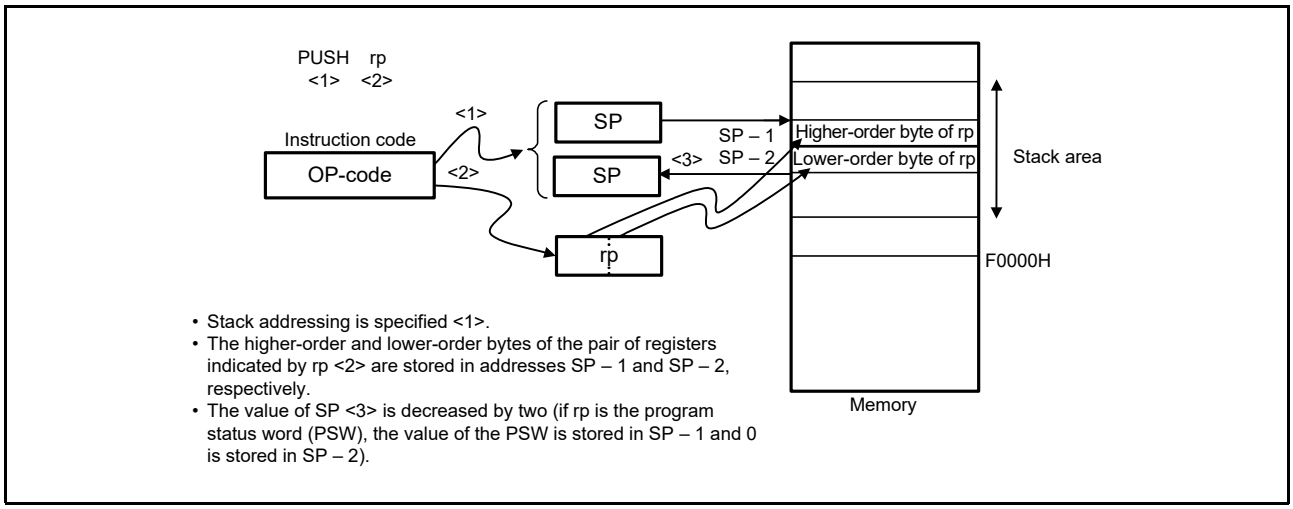
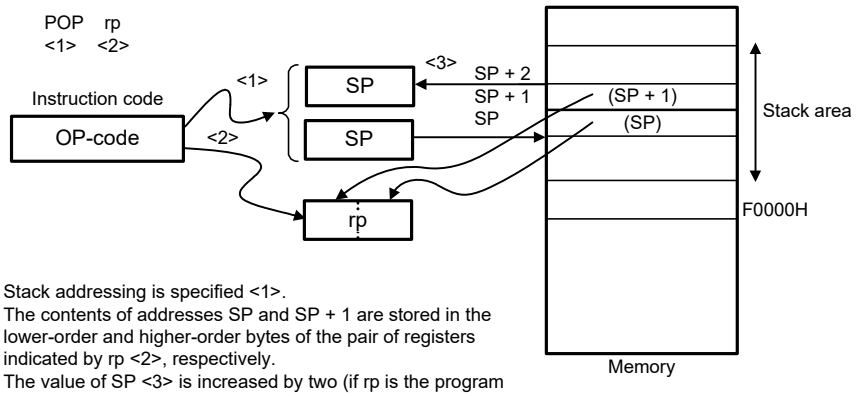
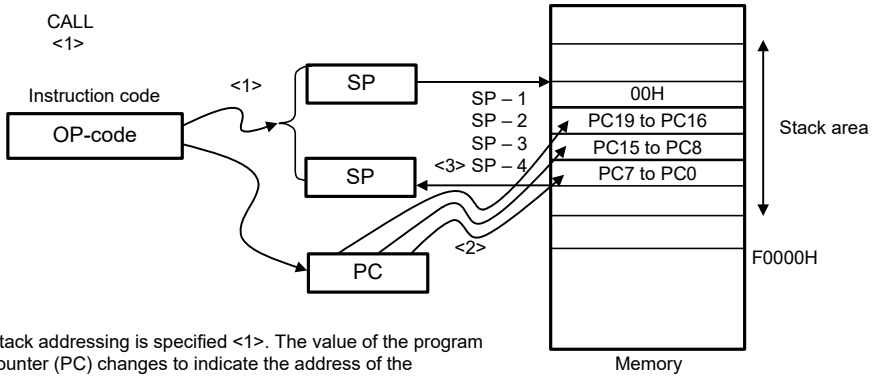


Figure 3 - 34 Example of POP



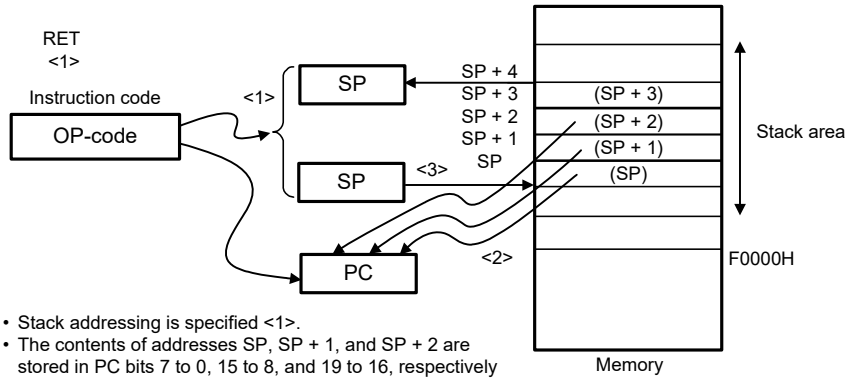
- Stack addressing is specified <1>.
- The contents of addresses SP and SP + 1 are stored in the lower-order and higher-order bytes of the pair of registers indicated by rp <2>, respectively.
- The value of SP <3> is increased by two (if rp is the program status word (PSW), the content of address SP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

Figure 3 - 35 Example of CALL, CALLT



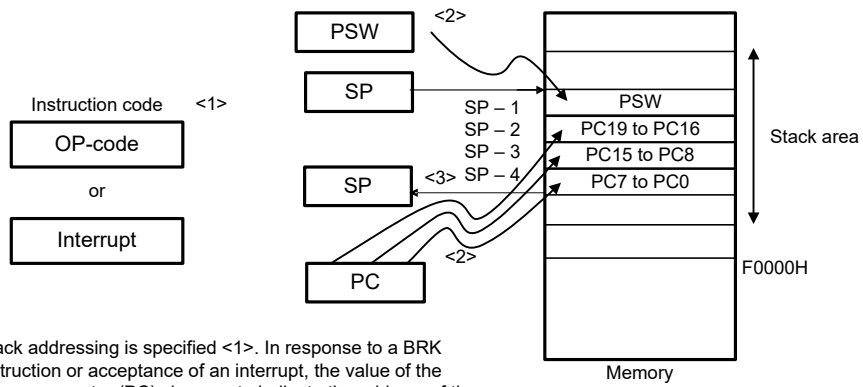
- Stack addressing is specified <1>. The value of the program counter (PC) changes to indicate the address of the instruction following the CALL instruction.
- 00H, the values of PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP – 1, SP – 2, SP – 3, and SP – 4, respectively <2>.
- The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

Figure 3 - 36 Example of RET



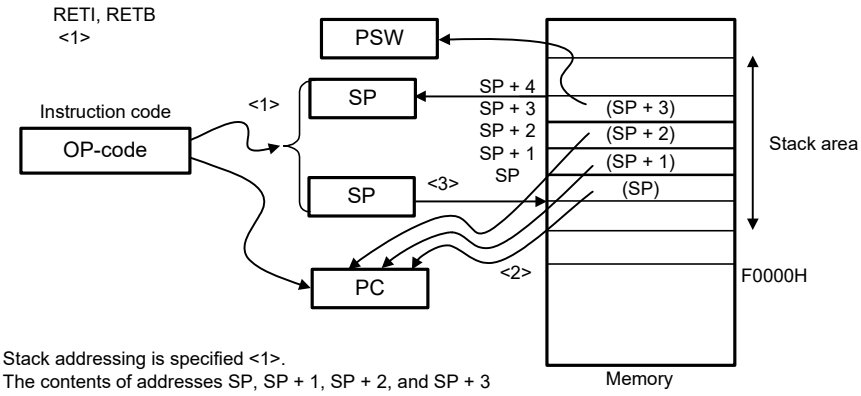
- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, and SP + 2 are stored in PC bits 7 to 0, 15 to 8, and 19 to 16, respectively <2>.
- The value of SP <3> is increased by four.

Figure 3 - 37 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 - 38 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.

Section 4 Flexible Application Accelerator (FAA)

4.1 Functions of the FAA

The flexible application accelerator (FAA) is a Renesas original application accelerator with a Harvard architecture. It can execute 32-bit multiplication, addition, and subtraction in a single cycle. It consists of the processor (GRNFAA), instruction code memory, data memory, an FAA bus, a window register (WIND), a system controller (GRNSYSC), an interrupt controller (GRNINTC), an input event controller (GRNINPUTC), and a reference timing controller (GRNTIMEC).

Table 4 - 1 shows the functions of the FAA.

Table 4 - 1 Functions of the FAA

Item	Function
Processor	A single operation instruction is executed in a single cycle. (When multiple operation instructions are executed consecutively, the second and the subsequent instructions are executed in two cycles each.) • Multiplication: 32-bit signed × 32-bit signed → 32-bit signed Results of multiplication (64-bit) can be right-shifted by a desired number of bits. • Addition: 32-bit signed + 32-bit signed → 32-bit signed Internally calculated with 33-bit precision • Subtraction: 32-bit signed – 32-bit signed → 32-bit signed Internally calculated with 33-bit precision • Limit operation: Operation parameter registers (33 bits × 4) in which upper and lower limits can be set. Processor internal registers • Operation parameter registers (32 bits × 6) • Address pointer registers (12 bits × 6) • Flag bit register (16 bits × 1 (including 4 valid flag bits))
Memory	• Instruction code memory: 4 Kbytes • Data memory: 2 Kbytes • Data shared memory (SHDMEM): 32 bytes
Interrupt	• Multiple interrupts available • Interrupt sources Input event detection interrupts: 10 Timing compare-match interrupts: 6
Input event processing	• Input channels: 10 • Detected edges: Rise, fall, or both
Timing processing	• Reference timing counter bits: 24 • Compare-match channels: 6

In this section, the following terms are used.

CPU addresses: The register addresses of the peripheral functions of the MCU and register addresses of the data shared memory (SHDMEM) in the case of access by the CPU.

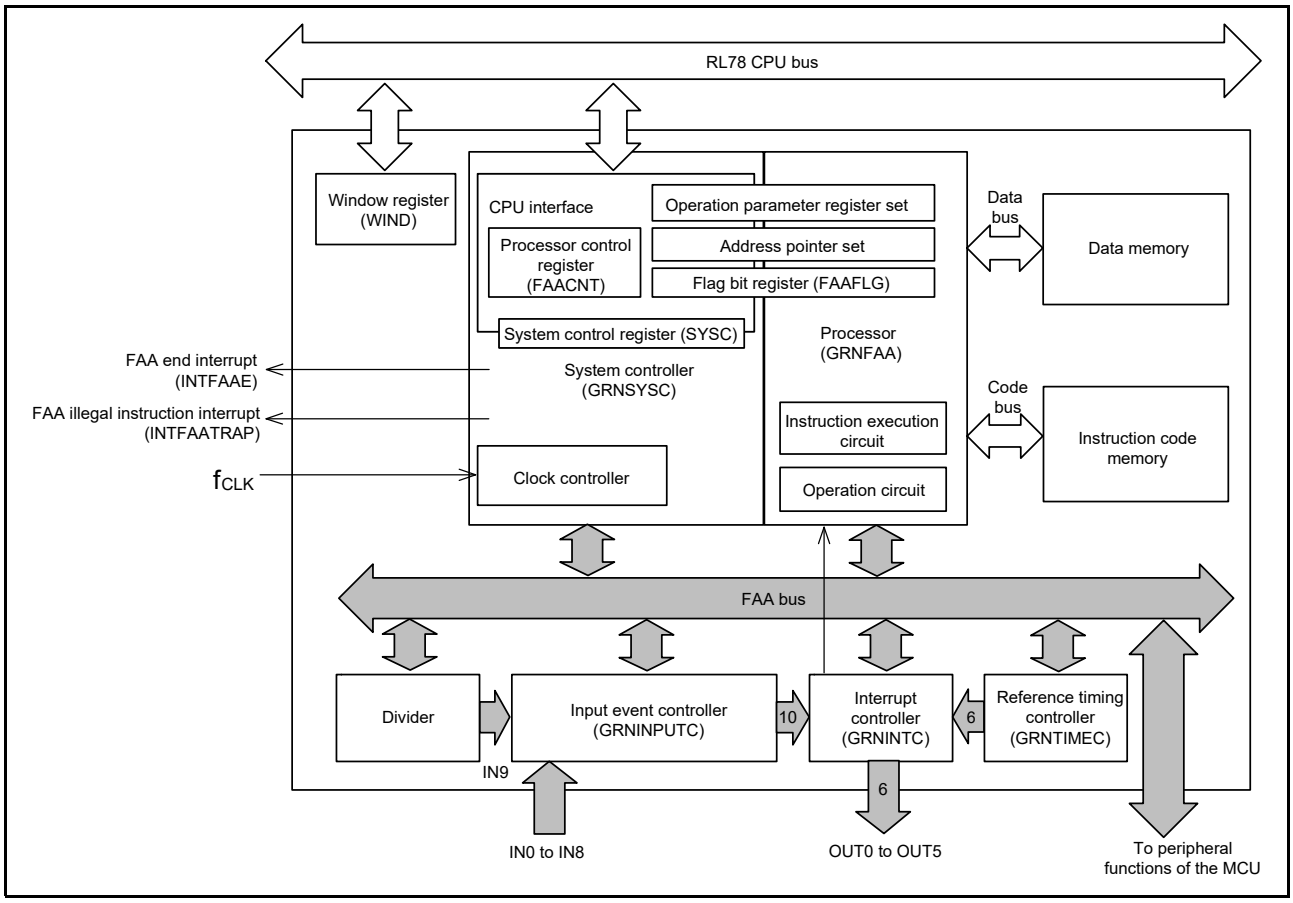
FAA addresses: The register addresses of the peripheral functions of the MCU, register addresses of the data shared memory (SHDMEM), and register addresses of the divider in the case of access by the FAA.

4.2 Configuration of the FAA

The FAA consists of the following hardware.

1. Processor (GRNFAA)
   The processor executes various operation instructions including multiplication, addition, subtraction, limit operation, and shift instruction of signed 32-bit data in a single clock cycle (execution cycle).
2. Instruction code memory
   The instruction code memory contains the program executed by the FAA.
3. Data memory
   The data memory contains the data necessary for the FAA to execute the program.
4. FAA bus
   The FAA bus is used for access to the registers of the input event controller (GRNINPUTC), interrupt controller (GRNINTC), and reference timing controller (GRNTIMEC), and those registers of the address bus select function, data shared memory (SHDMEM), divider, and internal peripheral functions of the MCU which are accessible by the FAA.
5. Window register (WIND)
   The window register is set when the CPU accesses each of the FAA hardware mapped into the 2nd SFR area of the CPU.
6. System controller (GRNSYSC)
   The system controller has a CPU interface for control over the operation of the FAA. It can also place the FAA in low power consumption mode.
7. Interrupt controller (GRNINTC)
   The interrupt controller controls the interrupts in the FAA.
8. Input event controller (GRNINPUTC)
   The input event controller detects edges of external trigger input signals (interrupt request signals from the internal peripheral functions of the MCU).
9. Reference timing controller (GRNTIMEC)
   The reference timing controller includes one 24-bit free-running counter and six timing comparison registers.
   It generates the periodic timing for operations by the FAA.

Figure 4 - 1 Block Diagram of the FAA



Remark IN0: External interrupt edge detection 0 (INTP0)
IN1: ELC trigger signal 0
IN2: ELC trigger signal 1
IN3: ELC trigger signal 2
IN4: ELC trigger signal 3
IN5: ELC trigger signal 4
IN6: ELC trigger signal 5
IN7: ELC trigger signal 6
IN8: ELC trigger signal 7
IN9: End of division operation

OUT0: FAA timing compare-match interrupt 0 (INTTIMEC0)
OUT1: FAA timing compare-match interrupt 1 (INTTIMEC1)
OUT2: FAA timing compare-match interrupt 2 (INTTIMEC2)
OUT3: FAA timing compare-match interrupt 3 (INTTIMEC3)
OUT4: FAA timing compare-match interrupt 4 (INTTIMEC4)
OUT5: FAA timing compare-match interrupt 5 (INTTIMEC5)

4.3 Peripheral Enable Register 2 (PER2)

The PER2 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. If the FAA is to be used, be sure to set bit 7 (FAAEN) of this register to 1. The PER2 register should be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H. The PER2 register is not accessible from the FAA.

Figure 4 - 2 Format of Peripheral Enable Register 2 (PER2)

Address: F00FCH
FAA address: -
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER2	FAAEN	MEMEN	TKBEN	TRGEN	TRD0EN	PWMOPEN	TRXEN	TRJ0EN

FAAEN	Control of supply of an input clock to the FAA
0	Stops supply of an input clock. • With the following exceptions, the SFRs used by the FAA can be read but not written. Note that only writing to the WIND register, ENB bit in the SYSC register, and ADBSEL register is possible.
1	Enables supply of an input clock. • The SFRs used by the FAA can be read and written.

4.4 Peripheral Reset Control Register 2 (PRR2)

The PRR2 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. To place the FAA in the reset state, be sure to set bit 7 (FAARES) of this register to 1. The PRR2 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H. The PRR2 register is not accessible from the FAA.

Figure 4 - 3 Format of Peripheral Reset Control Register 2 (PRR2)

Address: F00FDH
FAA address: -
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PRR2	FAARES	MEMRES	TKBRES	TRGRES	TRD0RES	PWMOPRES	TRXRES	TRJ0RES

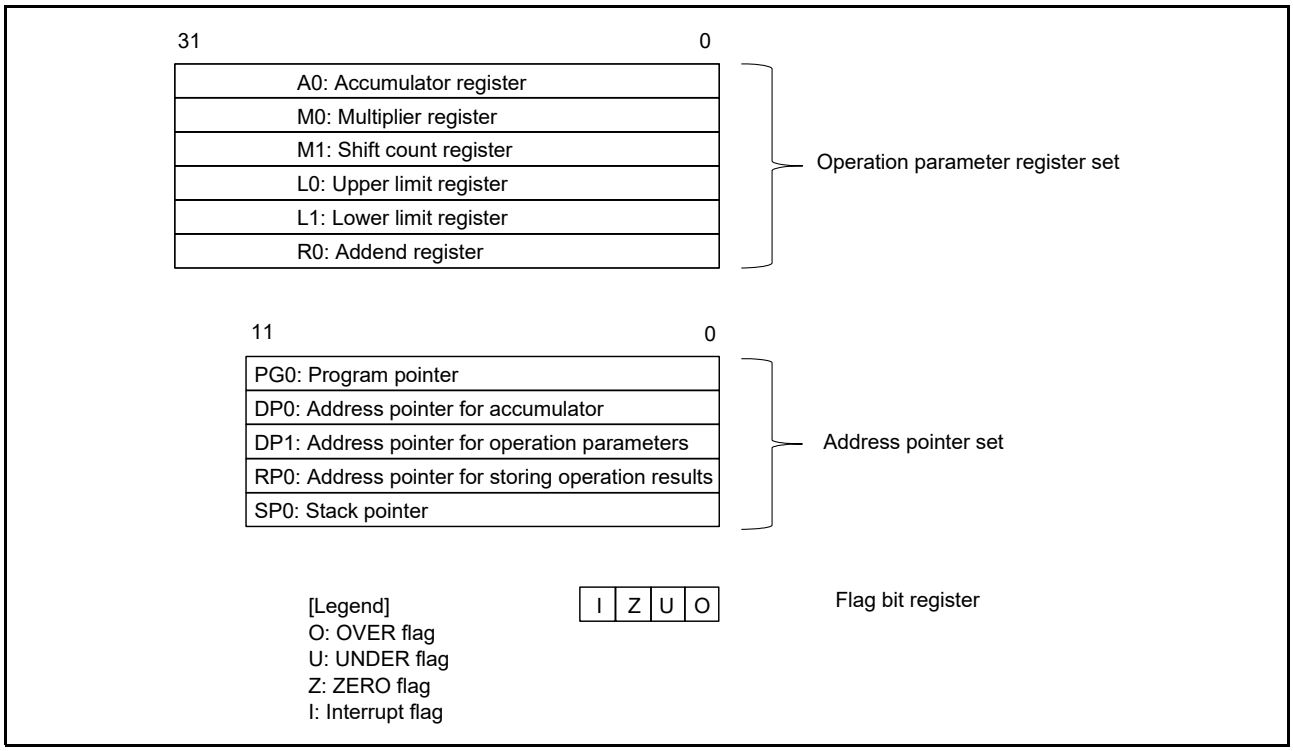
FAARES	Control resetting of the FAA
0	The FAA is released from the reset state.
1	The FAA is in the reset state.

4.5 Processor (GRNFAA)

4.5.1 Register configuration of the processor (GRNFAA)

Figure 4 - 4 shows the register configuration of the processor (GRNFAA).
Registers of the FAA include the operation parameter register set, address pointer set, and flag bit register.
Registers of the operation parameter register set and address pointer set are 32 and 12 bits wide, respectively.
For details of each of the registers, see **4.9 System Controller (GRNSYSC)**.

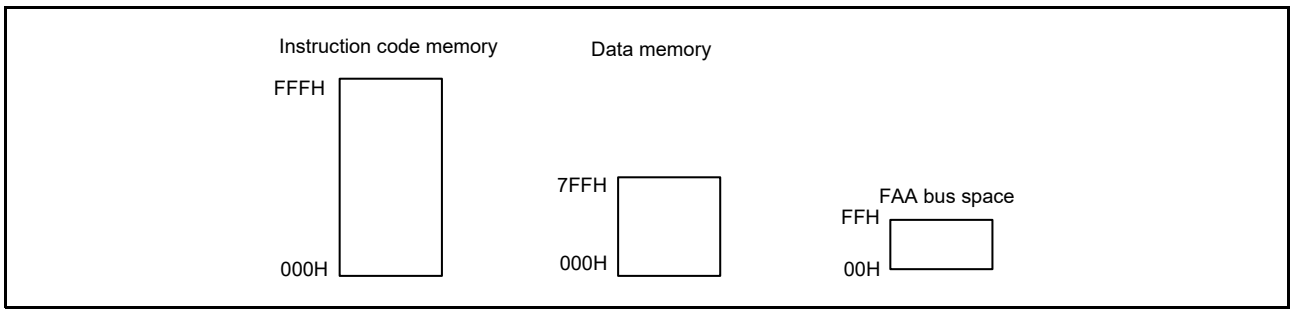
Figure 4 - 4 Register Configuration of the Processor (GRNFAA)



4.6 Memory Space of the FAA

Figure 4 - 5 shows the address map of the FAA. The memory space of the FAA is divided into the instruction code memory space, data memory space, and FAA bus space. The instruction code memory is accessed in 8-bit units or 16-bit units and the instructions are then decoded. The data memory can be accessed in 32-bit units, with the addresses specified by registers of the address pointer set. The FAA bus can be accessed by using the IN and OUT instructions, with the addresses specified in the instruction code.

Figure 4 - 5 Address Map of the FAA



4.7 FAA Bus

Table 4 - 2 shows the address map of the registers that are accessible by the FAA via the FAA bus.
All registers of the input event controller (GRNINPUTC), interrupt controller (GRNINTC), and reference timing controller (GRNTIMEC), and those registers of the address bus select function, data shared memory (SHDMEM), and internal peripheral functions of the MCU which are accessible by the FAA are in the address map.

Table 4 - 2 Address Map of the Registers that Are Accessible by the FAA via the FAA Bus (1/2)

Address	Register Name
00H	System control register (SYSC)
01H	Timing comparison register 0 (TMCMP0)
02H	Timing comparison register 1 (TMCMP1)
03H	Timing comparison register 2 (TMCMP2)
04H	Timing comparison register 3 (TMCMP3)
05H	Timing comparison register 4 (TMCMP4)
06H	Timing comparison register 5 (TMCMP5)
07H	Timing comparison mask register 0 (TMMSK0)
08H	Timing comparison mask register 1 (TMMSK1)
09H	Timing comparison mask register 2 (TMMSK2)
0AH	Timing comparison mask register 3 (TMMSK3)
0BH	Timing comparison mask register 4 (TMMSK4)
0CH	Timing comparison mask register 5 (TMMSK5)
0DH	Free-running counter register (FCNT)
0EH	Free-running counter control register (FCCNT)
0FH to 11H	Reserved for system (access prohibited)
12H	Interrupt vector register 0 (IV0)
13H	Interrupt vector register 1 (IV1)
14H	Interrupt vector register 2 (IV2)
15H	Interrupt vector register 3 (IV3)
16H	Interrupt vector register 4 (IV4)
17H	Interrupt vector register 5 (IV5)
18H	Interrupt vector register 6 (IV6)
19H	Interrupt vector register 7 (IV7)
1AH	Interrupt vector register 8 (IV8)
1BH	Interrupt vector register 9 (IV9)
1CH	Interrupt vector register 10 (IV10)
1DH	Interrupt vector register 11 (IV11)
1EH	Interrupt vector register 12 (IV12)
1FH	Interrupt vector register 13 (IV13)
20H	Interrupt vector register 14 (IV14)
21H	Interrupt vector register 15 (IV15)
22H to 27H	Reserved for system (access prohibited)
28H	Sense control register 0 (IEVSC0)

Table 4 - 2 Address Map of the Registers that Are Accessible by the FAA via the FAA Bus (2/2)

Address	Register Name
29H	Sense control register 1 (IEVSC1)
2AH to 3FH	Reserved for system (access prohibited)
40H to BFH	Refer to **Table 4 - 14 List of FAA Addresses and Access Size for Registers of Peripheral Functions** and **Table 4 - 15 List of FAA Addresses and Access Size for Registers (2nd SFRs) of Peripheral Functions**.
C0H to DFH	Reserved for system (access prohibited)
E0H	CWDW register file 0
E1H	CWDW register file 1
E2H	CWDW register file 2
E3H	CWDW register file 3
E4H	CWDW register file 4
E5H	CWDW register file 5
E6H	CWDW register file 6
E7H	CWDW register file 7
E8H to FFH	Reserved for system (access prohibited)

4.8 Window Register (WIND)

The internal memory space of the FAA is mapped to the target resource allocation area in the 2nd SFR area of the CPU. The target resource allocation area is from F0600H to F06FFH (256 bytes). Since the target resource allocation area is smaller than the memory area of the FAA, the value of the window register (WIND) is used as page information to expand the range by shifting the mapping. During access to the target resource allocation area, supply the input clock to the FAA (FAAEN = 1).
For access to the instruction code memory space (window register (WIND) = 0020H to 002FH), data memory space (window register (WIND) = 10H to 17H), or FAA bus space (window register (WIND) = 30H) in the FAA from the CPU or DTC, follow the procedure below.

<1> Change the value of the window register (WIND) to 0000H or confirm that it is 00H.
<2> Set the SYSC.SLP bit (sleep bit) to 0 to place the processor (GRNFAA) in the normal state.
<3> Set the SYSC.ENB bit (FAA operation enable bit) to 0 to disable the operation of the FAA.
<4> Confirm that the SYSC.EXE bit (processor operating bit) is cleared to 0 (processor is halted).

After the above steps, change the value of the window register (WIND) to the value corresponding to the resource of the FAA to be accessed before accessing the target resource.
The WIND register should be set by a 16-bit memory manipulation instruction.
The value of this register following a reset is 0000H.
Figure 4 - 6 shows the relationship between the window register (WIND), 2nd SFR area of the CPU, and internal memory space of the FAA.

Figure 4 - 6 Relationship between the Window Register (WIND), 2nd SFR Area of the CPU, and Internal Memory Space of the FAA

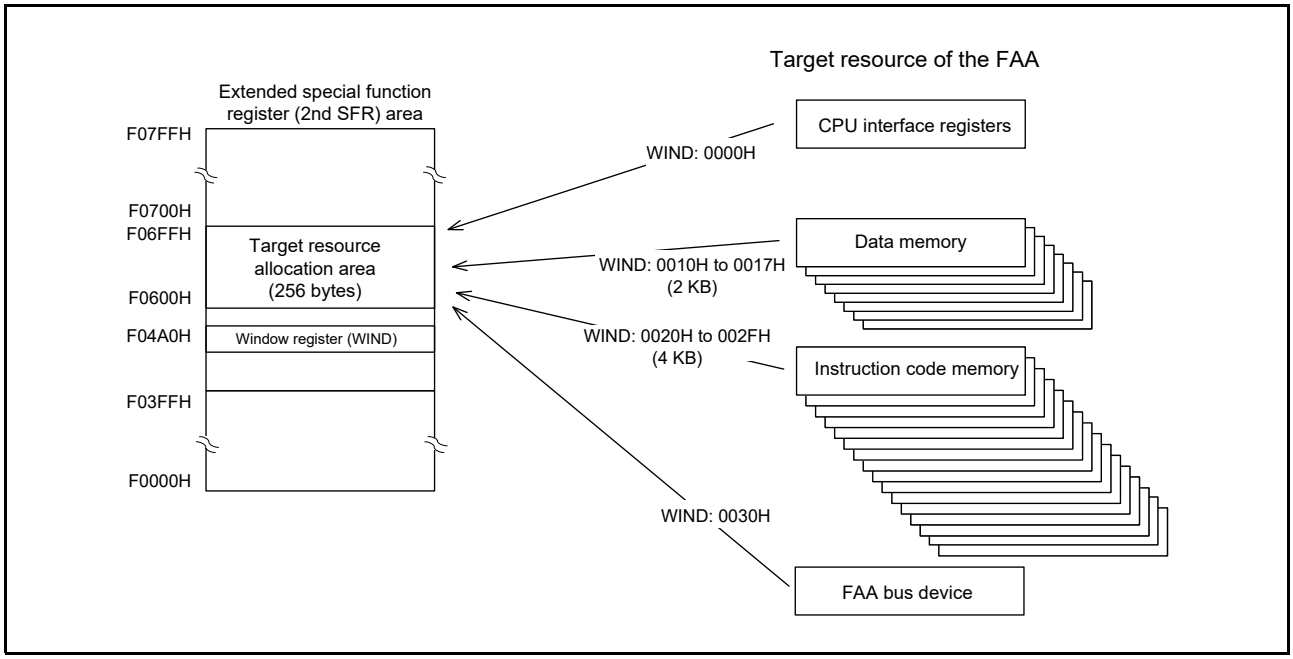


Figure 4 - 7 Format of Window Register (WIND)

Address: F04A0H
FAA address: -
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
WIND	0							

7	6	5	4	3	2	1	0

Bits 14 to 0	Target Resource of the FAA
0000H	CPU interface registers
0010H to 0017H	Data memory (2 KB)
0020H to 002FH	Instruction code memory (4 KB)
0030H	Registers of the following FAA bus devices. • Interrupt controller (GRNINTC) • Input event controller (GRNINPUTC) • Reference timing controller (GRNTIMEC)
Other than above	Setting prohibited
Specify the target resource of the FAA to be mapped into the F0600H to F06FFH area.	

4.9 System Controller (GRNSYSC)

The system controller (GRNSYSC) has a CPU interface for control over the operation of the FAA. It can also place the FAA in low power consumption mode.

Table 4 - 3 shows the functional overview of the system controller.

Table 4 - 3 Functional Overview of the System Controller

Item	Function
Low power consumption function	Stops supply of the clock signal to the processor (GRNFAA) by using the system control register (SYSC (DSYSC[Note])) and places the FAA in the low power consumption mode.
CPU interface	Enables access by the CPU while the FAA is halted.

Note DSYSC is a symbol for the extended special function register (2nd SFR).

4.9.1 List of registers of the system controller

Table 4 - 4 lists the registers of the system controller.

The instruction code memory, data memory, and FAA bus devices of the FAA are also accessible from the CPU. However, in the case of attempted access by the CPU at the same time as access by the processor (GRNFAA), the access by the CPU is invalid. In such cases, writing by the CPU is ineffective and values read are undefined.

Only 16-bit access is possible.

To access the CPU interface, follow the procedure described in **4.8 Window Register (WIND)**.

Table 4 - 4 List of Registers of the System Controller

Extended Special Function Register (2nd SFR)						FAA					
CPU Address	WIND Register Value	Register Name	Symbol	R/W (CPU)	Bits	FAA Address	Register Name		Symbol	R/W (FAA)	Bits
F0600H	0000H	Accumulator register L	A0L	R/W	16	—	Operation parameter register set	Accumulator register	A0	R/W	32
F0602H		Accumulator register H	A0H	R/W	16						
F0604H		Multiplier register L	M0L	R/W	16	—		Multiplier register	M0	R/W	32
F0606H		Multiplier register H	M0H	R/W	16						
F0608H		Shift count register L	M1L	R/W	16	—		Shift count register	M1	R/W	32
F060AH		Shift count register H	M1H	R/W	16						
F060CH		Upper limit register L	L0L	R/W	16	—		Upper limit register	L0	R/W	32
F060EH		Upper limit register H	L0H	R/W	16						
F0610H		Lower limit register L	L1L	R/W	16	—		Lower limit register	L1	R/W	32
F0612H		Lower limit register H	L1H	R/W	16						
F0614H		Addend register L	R0L	R/W	16	—		Addend register	R0	R/W	32
F0616H		Addend register H	R0H	R/W	16						
F061AH		Address pointer for accumulator	DP0	R/W	16	—	Address pointer set	Address pointer for accumulator	DP0	R/W	16
F061EH		Address pointer for operation parameters	DP1	R/W	16	—		Address pointer for operation parameters	DP1	R/W	16
F0622H		Address pointer for storing operation results	DRP0	R/W	16	—		Address pointer for storing operation results	RP0	R/W	16
F062AH		Processor control register	FAACNT	R/W	16	—	Processor control register		FAACNT	R/W	16
F062EH		Program pointer	PG0	R/W	16	—	Address pointer set	Program pointer	PG0	R/W	16
F0636H		Flag bit register	FAAFLG	R/W	16	—	Flag bit register		FAAFLG	R/W	16
F066EH		System control register	DSYSC	R/W	16	00H	System control register		SYSC	R/W	16
F0682H		Stack pointer	SP0	R/W	16	—	Address pointer set	Stack pointer	SP0	R/W	16

4.9.1.1 System control register (SYSC (DSYSC))

The SYSC (DSYSC) register**Note** has the following functions.

- Reading the operating state of the processor (GRNFAA)
- Enabling operation of the FAA
- Generating a software reset
- Placing the FAA in low power consumption mode

This register is allocated to both the CPU address and FAA address. When the CPU or DTC accesses this register, the window register (WIND) should be set to 00H; it does not need to be set to 30H, which is used for access to the FAA bus space. The SYSC (DSYSC) register should be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Note DSYSC is a symbol for the extended special function register (2nd SFR).

Figure 4 - 8 Format of System Control Register (SYSC (DSYSC)) (1/2)

Address: F066EH
FAA address: 00H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SYSC (DSYSC)	0	0	0	0	0	0	EXE	ENB

7	6	5	4	3	2	1	0
0	0	0	0	0	0	SRST	SLP

EXE	Processor (GRNFAA) operating bit
0	The processor (GRNFAA) is halted.
1	The processor (GRNFAA) is operating.
This bit indicates the operating state of the processor (GRNFAA). This bit indicates whether the processor (GRNFAA) is operating or halted. This bit cannot be accessed by the FAA. Values cannot be written to this bit.	

ENB	FAA operation enable bit
0	Disables operation.
1	Enables operation.
This bit enables operation of the FAA. Before accessing the memory area (instruction code/data) and FAA bus area of the FAA by the CPU or DTC, set the SLP bit to place the processor (GRNFAA) in the normal state; set the ENB bit to disable the operation of the FAA; and then confirm that the EXE bit is cleared to 0 (processor (GRNFAA) is halted). This bit cannot be accessed by the FAA.	

SRST	Software reset bit
0	Invalid
1	Software reset
Setting this bit to 1 initializes the FAA by a software reset. After a software reset, wait at least four clock cycles before accessing the FAA (register access or interrupt input).	

Figure 4 - 8 Format of System Control Register (SYSC (DSYSC)) (2/2)

SLP	Sleep bit
0	Normal state
1	Low power consumption mode
Setting this bit to 1 while the processor (GRNFAA) is stopped causes a transition to low power consumption mode, in which supply of the clock signal to the processor (GRNFAA), instruction code memory, data memory, and the FAA bus is stopped. Set this bit while the processor (GRNFAA) is halted.	

4.9.1.2 Operation parameter register set

This is a set of operation parameter registers for the processor (GRNFAA).
When accessing (writing or reading) any of these registers by the CPU or DTC, be sure to make an access while the processor (GRNFAA) is halted.
The registers of the processor (GRNFAA) are 32 bits wide. In access to this register set from the CPU or DTC, the order of access is the 16 higher-order bits and then the 16 lower-order bits.

The operation parameter registers as a processor (GRNFAA) are listed below.

- Accumulator register (A0)
- Multiplier register (M0)
- Shift count register (M1)
- Upper limit register (L0)
- Lower limit register (L1)
- Addend register (R0)

The operation parameter registers for access by the CPU or DTC are listed below.

- Accumulator registers H and L (A0H and A0L)
- Multiplier registers H and L (M0H and M0L)
- Shift count registers H and L (M1H and M1L)
- Upper limit registers H and L (L0H and L0L)
- Lower limit registers H and L (L1H and L1L)
- Addend registers H and L (R0H and R0L)

Figure 4 - 9 Format of Accumulator Register and Accumulator Registers H and L (A0, A0H, and A0L)

Address: F0602H (A0H), F0600H (A0L)
FAA address: -
After reset: 0000 0000H (A0), 0000H (A0H, A0L)
R/W: R/W (WIND = 00H)

Symbol A0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0	Value of the accumulator register (A0)

Symbol A0H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 15 to 0	16 higher-order bits of the accumulator register (A0)

Symbol A0L

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 15 to 0	16 lower-order bits of the accumulator register (A0)

Figure 4 - 10 Format of Multiplier Register and Multiplier Registers H and L (M0, M0H, and M0L)

Address: F0606H (M0H), F0604H (M0L)
FAA address: -
After reset: 0000 0000H (M0), 0000H (M0H, M0L)
R/W: R/W

Symbol M0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0	Value of the multiplier register (M0)

Symbol M0H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 15 to 0	16 higher-order bits of the multiplier register (M0)

Symbol M0L

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 15 to 0	16 lower-order bits of the multiplier register (M0)

Figure 4 - 11 Format of Shift Count Register and Shift Count Registers H and L (M1, M1H, and M1L)

Address: F060AH (M1H), F0608H (M1L)
FAA address: -
After reset: 0000 0000H (M1), 0000H (M1H, M1L)
R/W: R/W

Symbol M1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0	Value of the shift count register (M1)

Symbol M1H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 15 to 0	16 higher-order bits of the shift count register (M1)

Symbol M1L

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 15 to 0	16 lower-order bits of the shift count register (M1)

Figure 4 - 12 Format of Upper Limit Register and Upper Limit Registers H and L (L0, L0H, and L0L)

Address: F060EH (L0H), F060CH (L0L)
FAA address: -
After reset: 0000 0000H (L0), 0000H (L0H, L0L)
R/W: R/W

Symbol L0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0	Value of the upper limit register (L0)

Symbol L0H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 15 to 0	16 higher-order bits of the upper limit register (L0)

Symbol L0L

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 15 to 0	16 lower-order bits of the upper limit register (L0)

Figure 4 - 13 Format of Lower Limit Register and Lower Limit Registers H and L (L1, L1H, and L1L)

Address: F0612H (L1H), F0610H (L1L)
FAA address: -
After reset: 0000 0000H (L1), 0000H (L1H, L1L)
R/W: R/W

Symbol L1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0	Value of the lower limit register (L1)

Symbol L1H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 15 to 0	16 higher-order bits of the lower limit register (L1)

Symbol L1L

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 15 to 0	16 lower-order bits of the lower limit register (L1)

Figure 4 - 14 Format of Addend Register and Addend Registers H and L (R0, R0H, and R0L)

Address: F0616H (R0H), F0614H (R0L)
FAA address: -
After reset: 0000 0000H (R0), 0000H (R0H, R0L)
R/W: R/W

Symbol R0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0	Value of the addend register (R0)

Symbol R0H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 15 to 0	16 higher-order bits of the addend register (R0)

Symbol R0L

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 15 to 0	16 low-order bits of the addend register (R0)

4.9.1.3 Address pointer set

This is a set of 12-bit address pointers for the processor (GRNFAA).
When accessing (writing or reading) any of these pointers by the CPU or DTC, be sure to make an access while the processor (GRNFAA) is halted.

The address pointer set includes the following.

- Address pointer for accumulator (DP0)
- Address pointer for operation parameters (DP1)
- Address pointer for storing operation results (RP0 (DRP0[Note]))
- Program pointer (PG0)
- Stack pointer (SP0)

Note DRP0 is a symbol for the extended special function register (2nd SFR).

Figure 4 - 15 Format of Address Pointer for Accumulator (DP0)

Address: F061AH
FAA address: -
After reset: 0000H
R/W: R/W (WIND = 00H)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DP0	0	0	0	0												

Bits 11 to 0	Address pointer for accumulator (DP0) of the FAA

Figure 4 - 16 Format of Address Pointer for Operation Parameters (DP1)

Address: F061EH
FAA address: -
After reset: 0000H
R/W: R/W (WIND = 00H)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DP1	0	0	0	0												

Bits 11 to 0	Address pointer for operation parameters (DP1) of the FAA

Figure 4 - 17 Format of Address Pointer for Storing Operation Results (RP0 (DRP0Note))

Address: F0622H
FAA address: -
After reset: 0000H
R/W: R/W (WIND = 00H)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RP0 (DRP0Note)	0	0	0	0												

Bits 11 to 0	Address pointer for storing operation results (RP0 (DRP0Note)) of the FAA

Note DRP0 is a symbol for the extended special function register (2nd SFR).

Figure 4 - 18 Format of Program Pointer (PG0)

Address: F062EH
FAA address: -
After reset: 0000H
R/W: R/W (WIND = 00H)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PG0	0	0	0	0												

Bits 11 to 0	Program pointer (PG0) of the FAA

Figure 4 - 19 Format of Stack Pointer (SP0)

Address: F0682H
FAA address: -
After reset: 0000H
R/W: R/W (WIND = 00H)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SP0	0	0	0	0												

Bits 11 to 0	Stack pointer (SP0) of the FAA

4.9.1.4 Flag bit register (FAAFLG)

This is a 16-bit flag register for the processor (GRNFAA). When accessing (writing or reading) this register by the CPU or DTC, be sure to make an access while the processor (GRNFAA) is halted.

Figure 4 - 20 Format of Flag Bit Register (FAAFLG)

Address: F0636H
FAA address: -
After reset: 0000H
R/W: R/W (WIND = 00H)

Symbol	15	14	13	12	11	10	9	8
FAAFLG	0	0	0	I	0	0	0	ZERO

7	6	5	4	3	2	1	0
0	0	0	0	0	0	UNDER	OVER

I	Accesses the I flag of the processor (GRNFAA).
ZERO	Accesses the ZERO flag of the processor (GRNFAA).
UNDER	Accesses the UNDER flag of the processor (GRNFAA).
OVER	Accesses the OVER flag of the processor (GRNFAA).

4.9.1.5 Processor control register (FAACNT)

This is a 16-bit register to control the operation of the processor (GRNFAA). This register is for the setting to start or stop execution of programs by the processor (GRNFAA).

Figure 4 - 21 Format of Processor Control Register (FAACNT)

Address: F062AH
FAA address: -
After reset: 0000H
R/W: R/W (WIND = 00H)

Symbol	15	14	13	12	11	10	9	8
FAACNT	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	START

START	Program execution bit
0	Stops program execution.
1	Starts program execution.
Setting this bit to 1 starts execution of programs from the address held by PG0 (the program pointer) at the time. Writing 0 to this bit immediately stops execution of programs. When the FAA operation enable bit is 0 (disabling operation), writing to this bit is ignored.	

4.9.2 Operation of the system controller

4.9.2.1 Low power consumption mode

Supply of the clock signal to the processor (GRNFAA) can be stopped while the FAA is not operating, thus causing a transition to the low power consumption mode. The clock controller is used to stop supply of the clock signal.
To stop supply of the clock signal by using the FAA, follow the procedure below.

<1> Set the SYSC.SLP bit to 1.
<2> Execute the STOP instruction or other means[Note] to stop the FAA.
<3> The EXE bit, which indicates whether the processor (GRNFAA) is operating or halted, is cleared to 0 and supply of the clock signal stops.

Note For the conditions under which the FAA stops, refer to **4.14.2 Controlling execution of programs by the FAA**.

Supply of the clock signal is resumed according to the procedure below.

<1> The interrupt controller (GRNINTC) receives the interrupt request signal from the reference timing controller (GRNTIMEC) or input event controller (GRNINPUTC), and outputs the sleep cancellation signal to the clock controller.
<2> The clock controller cancels the sleep state to resume supply of the clock signal. The interrupt controller (GRNINTC) issues the interrupt request to the processor (GRNFAA).
<3> The FAA starts operating.

Figure 4 - 22 shows the transition timing to low power consumption mode and **Table 4 - 5** shows the block stop function.

Figure 4 - 22 Transition Timing to Low Power Consumption Mode

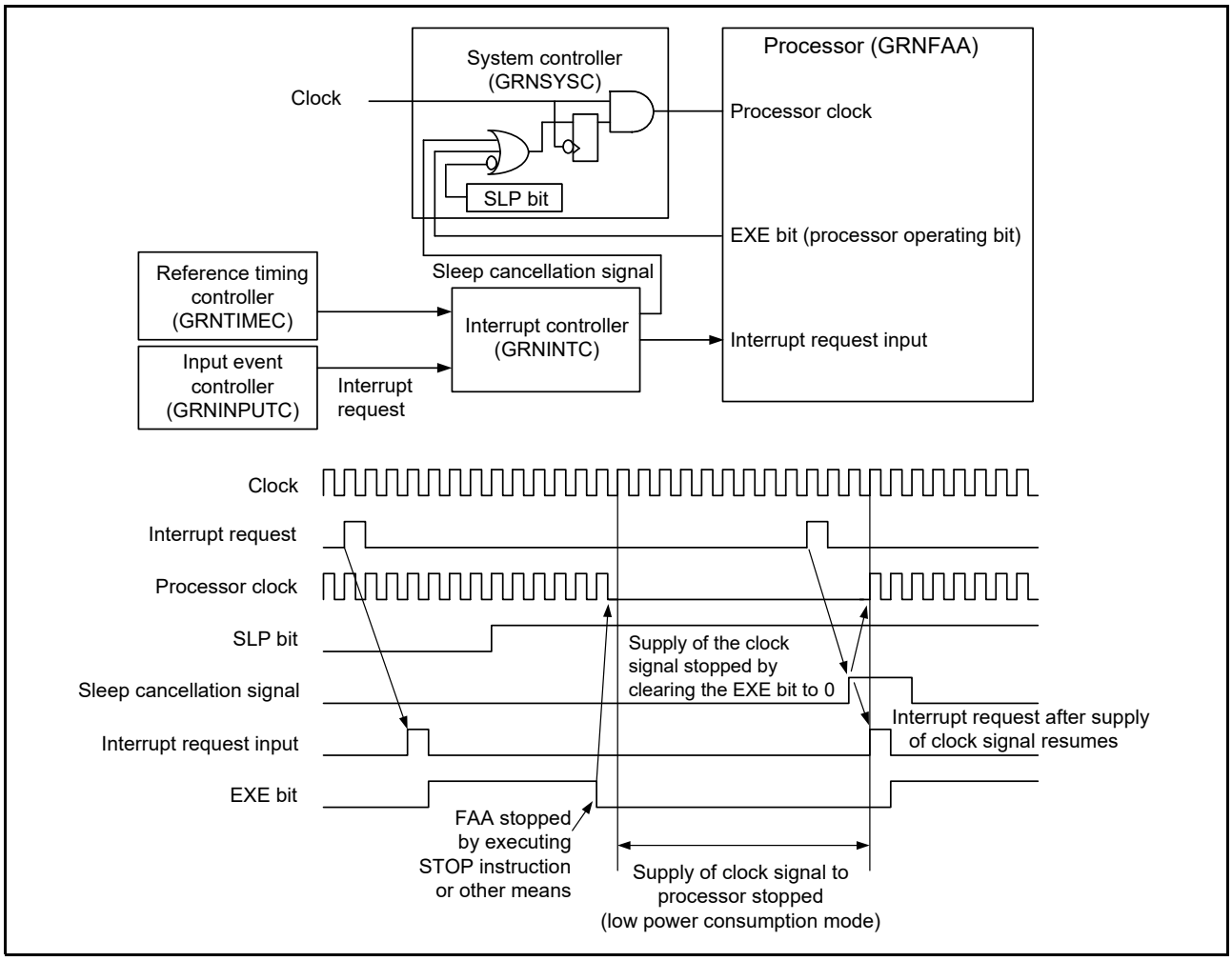


Table 4 - 5 Block Stop Function

State	Operating Blocks	Halted Blocks
Normal operation	Processor (GRNFAA) Instruction code memory Data memory Input event controller (GRNINPUTC) Reference timing controller (GRNTIMEC) Interrupt controller (GRNINTC) Clock controller FAA bus	None
Low power consumption mode	Input event controller (GRNINPUTC) Reference timing controller (GRNTIMEC) Interrupt controller (GRNINTC) Clock controller	Processor (GRNFAA) Instruction code memory Data memory FAA bus

4.10 Interrupt Controller (GRNINTC)

When input event detection interrupt requests (0 to 9) are input from the peripheral functions to the input event controller and when timing compare-match interrupt requests (0 to 5) are generated by the reference timing controller (GRNTIMEC) in the FAA, the FAA arbitrates between them and causes execution to branch to the interrupt processing with the highest priority. This is handled by the interrupt controller (GRNINTC). When the interrupt processing with the highest priority is complete, execution returns to the program that had been suspended.

4.10.1 Overview of the interrupt controller

Table 4 - 6 shows the functional overview of the interrupt controller.

Table 4 - 6 Functional Overview of the Interrupt Controller

Item	Function
Maskable interrupt	Input event detection interrupts: 10 Timing compare-match interrupts: 6
Interrupt method	I flag (interrupt flag) control 0: Disables interrupts. 1: Enables interrupts. Once the I flag in the flag bit register is set to 1, when the interrupt request is accepted and the interrupt is generated, the I flag is automatically cleared to 0. Setting the I flag to 1 again in the interrupt processing routine enables multiple interrupts.

4.10.2 List of registers of the interrupt controller and interrupt sources

Table 4 - 7 lists the registers of the interrupt controller. **Table 4 - 8** lists the interrupt sources.
For the method to access the registers of the interrupt controller from the CPU, refer to **4.8 Window Register (WIND)**.

Table 4 - 7 List of Registers of the Interrupt Controller (1/2)

Extended Special Function Register (2nd SFR)						FAA				
WIND Register Value	CPU Address	Register Name	Symbol	R/W (CPU)	Bits	FAA Address	Register Name	Symbol	R/W (FAA)	Bits
30H	F0648H	Interrupt vector register 0L	IV0L	R/W	16	12H	Interrupt vector register 0	IV0	R/W	32
30H	F064AH	Interrupt vector register 0H	IV0H	R/W	16					
30H	F064CH	Interrupt vector register 1L	IV1L	R/W	16	13H	Interrupt vector register 1	IV1	R/W	32
30H	F064EH	Interrupt vector register 1H	IV1H	R/W	16					
30H	F0650H	Interrupt vector register 2L	IV2L	R/W	16	14H	Interrupt vector register 2	IV2	R/W	32
30H	F0652H	Interrupt vector register 2H	IV2H	R/W	16					
30H	F0654H	Interrupt vector register 3L	IV3L	R/W	16	15H	Interrupt vector register 3	IV3	R/W	32
30H	F0656H	Interrupt vector register 3H	IV3H	R/W	16					
30H	F0658H	Interrupt vector register 4L	IV4L	R/W	16	16H	Interrupt vector register 4	IV4	R/W	32
30H	F065AH	Interrupt vector register 4H	IV4H	R/W	16					
30H	F065CH	Interrupt vector register 5L	IV5L	R/W	16	17H	Interrupt vector register 5	IV5	R/W	32
30H	F065EH	Interrupt vector register 5H	IV5H	R/W	16					
30H	F0660H	Interrupt vector register 6L	IV6L	R/W	16	18H	Interrupt vector register 6	IV6	R/W	32
30H	F0662H	Interrupt vector register 6H	IV6H	R/W	16					
30H	F0664H	Interrupt vector register 7L	IV7L	R/W	16	19H	Interrupt vector register 7	IV7	R/W	32
30H	F0666H	Interrupt vector register 7H	IV7H	R/W	16					
30H	F0668H	Interrupt vector register 8L	IV8L	R/W	16	1AH	Interrupt vector register 8	IV8	R/W	32
30H	F066AH	Interrupt vector register 8H	IV8H	R/W	16					
30H	F066CH	Interrupt vector register 9L	IV9L	R/W	16	1BH	Interrupt vector register 9	IV9	R/W	32
30H	F066EH	Interrupt vector register 9H	IV9H	R/W	16					
30H	F0670H	Interrupt vector register 10L	IV10L	R/W	16	1CH	Interrupt vector register 10	IV10	R/W	32
30H	F0672H	Interrupt vector register 10H	IV10H	R/W	16					
30H	F0674H	Interrupt vector register 11L	IV11L	R/W	16	1DH	Interrupt vector register 11	IV11	R/W	32
30H	F0676H	Interrupt vector register 11H	IV11H	R/W	16					

Table 4 - 7 List of Registers of the Interrupt Controller (2/2)

Extended Special Function Register (2nd SFR)						FAA				
WIND Register Value	CPU Address	Register Name	Symbol	R/W (CPU)	Bits	FAA Address	Register Name	Symbol	R/W (FAA)	Bits
30H	F0678H	Interrupt vector register 12L	IV12L	R/W	16	1EH	Interrupt vector register 12	IV12	R/W	32
30H	F067AH	Interrupt vector register 12H	IV12H	R/W	16					
30H	F067CH	Interrupt vector register 13L	IV13L	R/W	16	1FH	Interrupt vector register 13	IV13	R/W	32
30H	F067EH	Interrupt vector register 13H	IV13H	R/W	16					
30H	F0680H	Interrupt vector register 14L	IV14L	R/W	16	20H	Interrupt vector register 14	IV14	R/W	32
30H	F0682H	Interrupt vector register 14H	IV14H	R/W	16					
30H	F0684H	Interrupt vector register 15L	IV15L	R/W	16	21H	Interrupt vector register 15	IV15	R/W	32
30H	F0686H	Interrupt vector register 15H	IV15H	R/W	16					

Table 4 - 8 Interrupt Sources

Source Number (Priority)	Register Name (Symbol)	Internal Interrupt Source of the FAA	Interrupt Source Inputs from Peripheral Functions
0 (highest)	Interrupt vector register 0 (IV0)	Timing compare-match interrupt 0	—
1	Interrupt vector register 1 (IV1)	Timing compare-match interrupt 1	—
2	Interrupt vector register 2 (IV2)	Timing compare-match interrupt 2	—
3	Interrupt vector register 3 (IV3)	Input event detection interrupt 0	External interrupt edge detection 0 (INTP0)
4	Interrupt vector register 4 (IV4)	Input event detection interrupt 1	ELC trigger signal 0
5	Interrupt vector register 5 (IV5)	Input event detection interrupt 2	ELC trigger signal 1
6	Interrupt vector register 6 (IV6)	Input event detection interrupt 3	ELC trigger signal 2
7	Interrupt vector register 7 (IV7)	Input event detection interrupt 4	ELC trigger signal 3
8	Interrupt vector register 8 (IV8)	Timing compare-match interrupt 3	—
9	Interrupt vector register 9 (IV9)	Timing compare-match interrupt 4	—
10	Interrupt vector register 10 (IV10)	Timing compare-match interrupt 5	—
11	Interrupt vector register 11 (IV11)	Input event detection interrupt 5	ELC trigger signal 4
12	Interrupt vector register 12 (IV12)	Input event detection interrupt 6	ELC trigger signal 5
13	Interrupt vector register 13 (IV13)	Input event detection interrupt 7	ELC trigger signal 6
14	Interrupt vector register 14 (IV14)	Input event detection interrupt 8	ELC trigger signal 7
15 (lowest)	Interrupt vector register 15 (IV15)	Input event detection interrupt 9	End of division operation

4.10.2.1 Interrupt vector registers 0 to 15, 0H to 15H, and 0L to 15L (IV0 to IV15, IV0H to IV15H, and IV0L to IV15L)

The IVn, IVnH, and IVnL (n = 0 to 15) registers[Note] control the interrupt operations. They have the following functions.

- Enabling or disabling interrupt processing
- Indicating whether or not an interrupt request has been generated
- Setting the interrupt vector addresses

Note IVnH and IVnL are symbols for extended special function registers (2nd SFRs).

Figure 4 - 23 Format of Interrupt Vector Registers n, nH, and nL (IVn, IVnH, and IVnL)

Address: (See **Table 4 - 7 List of Registers of the Interrupt Controller**.)
FAA address: 12H (IV0) to 21H (IV15)
After reset: 0000 0000H (IVn), 0000H (IVnH, IVnL)
R/W: R/W

Symbol

IVn	31	30	29	28	27	26	25	24
	IVnEN	0	IVnST	0	0	0	0	0
	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
	0	0	0	0	IVnVEC[11:8]			
	7	6	5	4	3	2	1	0
	IVnVEC[7:0]							

IVnH	15	14	13	12	11	10	9	8
	IVnEN	0	IVnST	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0

IVnL	15	14	13	12	11	10	9	8
	0	0	0	0	IVnVEC[11:8]			
	7	6	5	4	3	2	1	0
	IVnVEC[7:0]							

IVnEN	Interrupt processing enable bit
0	Disables interrupt processing.
1	Enables interrupt processing.
Enables or disables interrupt processing.	

IVnST	Interrupt vector status bit
0	An interrupt request has not been generated.
1	An interrupt request has been generated.
Indicates whether or not an interrupt request has been generated. Values cannot be written to this bit.	

IVnVEC [11:0]	Interrupt vector bit
—	Specify the code memory address at which interrupt processing begins.

4.10.3 Operation of the interrupt controller

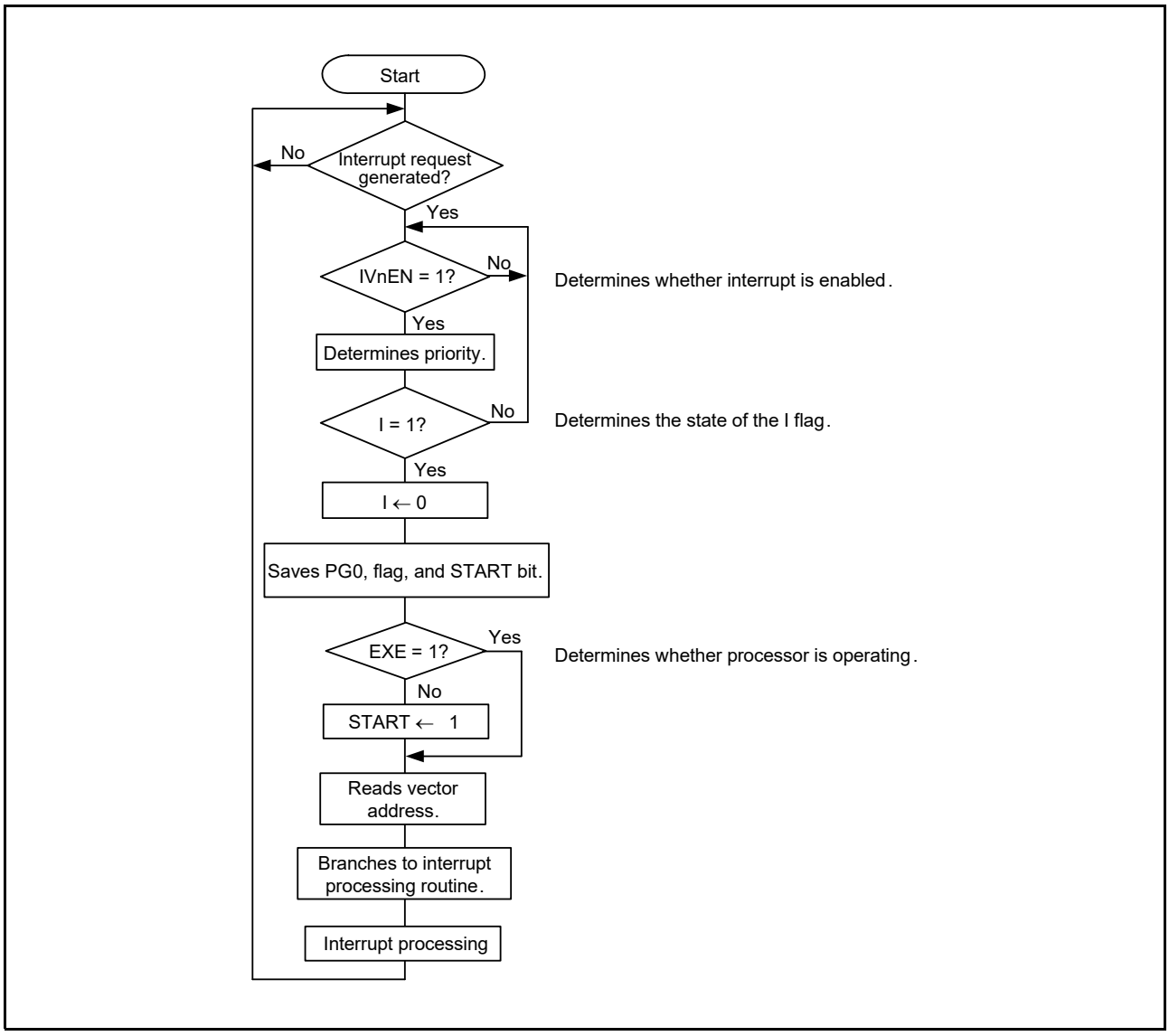
Interrupts allow the execution of programs in response to triggers externally input to the processor (GRNFAA). The start address of the program to be executed should be set in the corresponding interrupt vector register n (IVn) (n = 0 to 15).
Interrupts are disabled while the program is stopped by a break when a debugger is in use. If an interrupt request is generated during break, the request is discarded.
Similarly, if an interrupt request is generated when the FAA operation enable bit is 0, the request is discarded.

The following describes the sequence of interrupt operation.

1. When an interrupt request is generated, it is checked whether the corresponding interrupt enable bit (IVnEN (n = 0 to 15)) is set to 1.
   If IVnEN = 0, the interrupt request is discarded.
2. The interrupt controller (GRNINTC) selects the interrupt request with the highest priority and suspends processing for any other requests.
   The order of interrupt priority is determined by the vector numbers, with 0 having the highest and 15 having the lowest.
   Vector number 0 > vector number 1 > vector number 2 > ..... > vector number 14 > vector number 15
3. While the I flag is 0, the interrupt controller (GRNINTC) suspends processing for all interrupt requests.
   If the I flag is 1, the interrupt controller (GRNINTC) accepts the interrupt requests. Once the interrupt controller (GRNINTC) accepts the interrupt request, the I flag is automatically cleared to 0. To control multiple interrupts, set the I flag to 1 in the interrupt processing routine.
4. The interrupt controller (GRNINTC) saves the program pointer (PG0), flags, and START bit to the data memory address pointed to by SP0 (stack pointer).
5. The interrupt controller (GRNINTC) determines whether the processor (GRNFAA) is operating. If the processor is not operating, set the START bit to 1 to start the processor (GRNFAA).
6. The processor (GRNFAA) reads the vector address corresponding to the interrupt request accepted, and starts the execution of the interrupt processing routine at that address.
7. For the method to exit the interrupt processing routine, refer to **4.14.2 Controlling execution of programs by the FAA**.

Figure 4 - 24 shows the flowchart of the interrupt operation.

Figure 4 - 24 Flowchart of Interrupt Operation



4.10.3.1 Setting when timing compare-match interrupts 0 to 2 (INTTIMEC0 to INTTIMEC2) are used as an interrupt to the CPU

When timing compare-match interrupts 0 to 2 (INTTIMEC0 to INTTIMEC2) are used as an interrupt to the CPU, set the IVnEN (n = 0 to 2) bit (interrupt processing enable bit) in interrupt vector register n (IVn (n = 0 to 2) to 1.
When INTTIMEC0 to INTTIMEC2 are only used as an interrupt to the CPU (when INTTIMEC0 to INTTIMEC2 are not used as FAA timing compare-match interrupt n), set the I flag of the processor (GRNFAA) to 0.

4.10.3.2 Setting when timing compare-match interrupts 3 to 5 (INTTIMEC3 to INTTIMEC5) are output to the event link controller as an event signal

When timing compare-match interrupts 3 to 5 (INTTIMEC3 to INTTIMEC5) are output to the event link controller as an event signal, set the IVnEN (n = 8 to 10) bit (interrupt processing enable bit) in interrupt vector register n (IVn (n = 8 to 10) to 1.
When INTTIMEC3 to INTTIMEC5 are only output to the event link controller as an event signal (when INTTIMEC3 to INTTIMEC5 are not used as FAA timing compare-match interrupt n), set the I flag of the processor (GRNFAA) to 0.

4.10.3.3 Setting when timing compare-match interrupt 0 (INTTIMEC0) is used as a DTC activation source

When timing compare-match interrupt 0 (INTTIMEC0) is used as a DTC activation source, set the IV0EN bit (interrupt processing enable bit) in interrupt vector register 0 (IV0) to 1.
When INTTIMEC0 is only used as a DTC activation source (when INTTIMEC0 is not used as FAA timing compare-match interrupt n), set the I flag of the processor (GRNFAA) to 0.

4.11 Input Event Controller (GRNINPUTC)

4.11.1 Overview of the input event controller

The input event controller (GRNINPUTC) detects the edge of the event input signal externally input to the FAA.
Table 4 - 9 shows the function of the input event controller.

Table 4 - 9 Function of the Input Event Controller

Item	Function
Detected edge of input event signal	Rising edge, falling edge, or both edges

For the relationship between the input event signals and internal interrupts of the FAA, refer to **Table 4 - 8 Interrupt Sources**.

4.11.2 List of registers of the input event controller

Table 4 - 10 lists the registers of the input event controller.
For the method to access the registers of the input event controller from the CPU, refer to **4.8 Window Register (WIND)**.
All registers of this module are accessible in 16-bit units.

Table 4 - 10 List of Registers of Input Event Controller

Extended Special Function Register (2nd SFR)						FAA				
WIND Register Value	CPU Address	Register Name	Symbol	R/W (CPU)	Bits	FAA Address	Register Name	Symbol	R/W (FAA)	Bits
30H	F06A0H	Sense control register 0L	IEVSC0L	R/W	16	28H	Sense control register 0	IEVSC0	R/W	32
30H	F06A2H	Sense control register 0H	IEVSC0H	R/W	16					
30H	F06A4H	Sense control register 1	IEVSC1	R/W	16	29H	Sense control register 1	IEVSC1	R/W	32
—	—	—	—	—	—					

4.11.2.1 Sense control registers 0, 0H, and 0L (IEVSC0, IEVSC0H, and IEVSC0L)

The IEVSC0, IEVSC0H, and IEVSC0L registers[Note] specify the edge to be detected for the event input signals externally input to the FAA.

Note IEVSC0H and IEVSC0L are symbols for extended special function registers (2nd SFRs).

Figure 4 - 25 Format of Sense Control Registers 0, 0H, and 0L (IEVSC0, IEVSC0H, and IEVSC0L) (1/2)

Address: F06A2H (IEVSC0H), F06A0H (IEVSC0L)
FAA address: 28H (IEVSC0)
After reset: 0000 0000H (IEVSC0), 0000H (IEVSC0H, IEVSC0L)
R/W: R/W

Symbol	31	30	29	28	27	26	25	24
IEVSC0	0	0	IVSC7[1:0]		0	0	IVSC6[1:0]	

23	22	21	20	19	18	17	16
0	0	IVSC5[1:0]		0	0	IVSC4[1:0]	

15	14	13	12	11	10	9	8
0	0	IVSC3[1:0]		0	0	IVSC2[1:0]	

7	6	5	4	3	2	1	0
0	0	IVSC1[1:0]		0	0	IVSC0[1:0]	

Symbol	15	14	13	12	11	10	9	8
IEVSC0H	0	0	IVSC7[1:0]		0	0	IVSC6[1:0]	

7	6	5	4	3	2	1	0
0	0	IVSC5[1:0]		0	0	IVSC4[1:0]	

Symbol	15	14	13	12	11	10	9	8
IEVSC0L	0	0	IVSC3[1:0]		0	0	IVSC2[1:0]	

7	6	5	4	3	2	1	0
0	0	IVSC1[1:0]		0	0	IVSC0[1:0]	

IVSC7[1:0]		ELC trigger signal 6
0	0	Detects falling edge.
0	1	Detects rising edge.
1	0	Detects both edges (input level should not change in less than 3 cycles).
1	1	Setting prohibited

IVSC6[1:0]		ELC trigger signal 5
0	0	Detects falling edge.
0	1	Detects rising edge.
1	0	Detects both edges (input level should not change in less than 3 cycles).
1	1	Setting prohibited

Figure 4 - 25 Format of Sense Control Registers 0, 0H, and 0L (IEVSC0, IEVSC0H, and IEVSC0L) (2/2)

IVSC5[1:0]		ELC trigger signal 4
0	0	Detects falling edge.
0	1	Detects rising edge.
1	0	Detects both edges (input level should not change in less than 3 cycles).
1	1	Setting prohibited

IVSC4[1:0]		ELC trigger signal 3
0	0	Detects falling edge.
0	1	Detects rising edge.
1	0	Detects both edges (input level should not change in less than 3 cycles).
1	1	Setting prohibited

IVSC3[1:0]		ELC trigger signal 2
0	0	Detects falling edge.
0	1	Detects rising edge.
1	0	Detects both edges (input level should not change in less than 3 cycles).
1	1	Setting prohibited

IVSC2[1:0]		ELC trigger signal 1
0	0	Detects falling edge.
0	1	Detects rising edge.
1	0	Detects both edges (input level should not change in less than 3 cycles).
1	1	Setting prohibited

IVSC1[1:0]		ELC trigger signal 0
0	0	Detects falling edge.
0	1	Detects rising edge.
1	0	Detects both edges (input level should not change in less than 3 cycles).
1	1	Setting prohibited

IVSC0[1:0]		External interrupt edge detection 0 (INTP0)
0	0	Detects falling edge.
0	1	Detects rising edge.
1	0	Detects both edges (input level should not change in less than 3 cycles).
1	1	Setting prohibited

4.11.2.2 Sense control register 1 (IEVSC1)

The IEVSC1 register specifies the edge to be detected for the event input signals externally input to the FAA.

Figure 4 - 26 Format of Sense Control Register 1 (IEVSC1)

Address: F06A4H
FAA address: 29H
After reset: 0000 0000H (FAA), 0000H (2nd SFR)
R/W: R/W

Symbol	31	30	29	28	27	26	25	24
IEVSC1 (FAA)	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	IVSC9[1:0]		0	0	IVSC8[1:0]	

Symbol	15	14	13	12	11	10	9	8
IEVSC1 (2nd SFR)	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	IVSC9[1:0]		0	0	IVSC8[1:0]	

IVSC9[1:0]Note		End of division operation
0	0	Detects falling edge.
0	1	Detects rising edge.
1	0	Detects both edges (input level should not change in less than 3 cycles).
1	1	Setting prohibited

IVSC8[1:0]		ELC trigger signal 7
0	0	Detects falling edge.
0	1	Detects rising edge.
1	0	Detects both edges (input level should not change in less than 3 cycles).
1	1	Setting prohibited

Note When the IVSC9[1:0] bits are used for the FAA interrupt sources, set the bits so that the rising edge should be detected.

4.11.3 Operation of the input event controller

To have a program start in response to detection of the selected change in any of input event signals 0 to 9, interrupt vector register n (IVn) (n = 0 to 15) corresponding to the pertinent input event must be set. Here, it is not necessary to set the START bit in the processor control register (FAACNT).
To prevent processing being started at an unintended time, clear the processing requests in response to the change in the input level.
The following describes the procedure for having a program start in response to detection of a change in the level, taking input event signal 0 as an example.

<1> Clear the I flag to 0 (disable interrupts).
<2> Clear the interrupt processing enable bit (IV3.IV3EN) of interrupt vector register n to 0 (disable interrupt processing).
<3> Set the start address of the program to be executed in the interrupt vector bits (IV3.IV3VEC[11:0]) of interrupt vector register n.
<4> Set the sense control bits (IEVSC0.IVSC0[1:0]) (select the edge to be detected).
<5> Set the interrupt processing enable bit (IV3.IV3EN) to 1 (enable interrupt processing).
<6> Set the I flag to 1 (enable interrupts).

For interrupt vector register n (IVn) (n = 0 to 15) assigned to input event detection interrupts 0 to 9, refer to **Table 4 - 7 List of Registers of the Interrupt Controller** and **Table 4 - 8 Interrupt Sources**.
When the input of input event detection 0 corresponding to the interrupt vector register changes as specified with the sense control bits (IEVSC0.IVSC0[1:0]), the interrupt requests are accepted according to the priority. Then, the program execution starts at the address set in the corresponding interrupt vector bits (IV3.IV3VEC[11:0]).
The I flag is automatically cleared to 0. To enable multiple interrupts, set the I flag again.
When an application requires measurement of the intervals between processing for interrupts for use as a control parameter, read out the values of the free-running counter when interrupt processing is executed in response to changes in the input level.

4.12 Reference Timing Controller (GRNTIMEC)

4.12.1 Overview of the reference timing controller

The reference timing controller is a timer of the FAA. It has a 24-bit free-running counter and handles the detection of compare-matches with up to six values. **Table 4 - 11** lists the functions of the reference timing controller.

Table 4 - 11 Functions of the Reference Timing Controller

Item	Function
Number of counter bits	24
Counter start/stop function	Provided
Compare-match function	6 types
Mask function	Provided

4.12.2 List of registers of the reference timing controller

Table 4 - 12 lists the registers of the reference timing controller.
For the method to access the registers of the reference timing controller from the CPU, refer to **4.8 Window Register (WIND)**.
All registers of this module are accessible in 16-bit units.

Table 4 - 12 List of Registers of the Reference Timing Controller (1/2)

Extended Special Function Register (2nd SFR)						FAA				
WIND Register Value	CPU Address	Register Name	Symbol	R/W (CPU)	Bits	FAA Address	Register Name	Symbol	R/W (FAA)	Bits
30H	F0604H	Timing comparison register 0L	TMCMP0L	R/W	16	01H	Timing comparison register 0	TMCMP0	R/W	32
30H	F0606H	Timing comparison register 0H	TMCMP0H	R/W	16					
30H	F0608H	Timing comparison register 1L	TMCMP1L	R/W	16	02H	Timing comparison register 1	TMCMP1	R/W	32
30H	F060AH	Timing comparison register 1H	TMCMP1H	R/W	16					
30H	F060CH	Timing comparison register 2L	TMCMP2L	R/W	16	03H	Timing comparison register 2	TMCMP2	R/W	32
30H	F060EH	Timing comparison register 2H	TMCMP2H	R/W	16					
30H	F0610H	Timing comparison register 3L	TMCMP3L	R/W	16	04H	Timing comparison register 3	TMCMP3	R/W	32
30H	F0612H	Timing comparison register 3H	TMCMP3H	R/W	16					
30H	F0614H	Timing comparison register 4L	TMCMP4L	R/W	16	05H	Timing comparison register 4	TMCMP4	R/W	32
30H	F0616H	Timing comparison register 4H	TMCMP4H	R/W	16					
30H	F0618H	Timing comparison register 5L	TMCMP5L	R/W	16	06H	Timing comparison register 5	TMCMP5	R/W	32
30H	F061AH	Timing comparison register 5H	TMCMP5H	R/W	16					
30H	F061CH	Timing comparison mask register 0L	TMMSK0L	R/W	16	07H	Timing comparison mask register 0	TMMSK0	R/W	32
30H	F061EH	Timing comparison mask register 0H	TMMSK0H	R/W	16					
30H	F0620H	Timing comparison mask register 1L	TMMSK1L	R/W	16	08H	Timing comparison mask register 1	TMMSK1	R/W	32
30H	F0622H	Timing comparison mask register 1H	TMMSK1H	R/W	16					
30H	F0624H	Timing comparison mask register 2L	TMMSK2L	R/W	16	09H	Timing comparison mask register 2	TMMSK2	R/W	32
30H	F0626H	Timing comparison mask register 2H	TMMSK2H	R/W	16					
30H	F0628H	Timing comparison mask register 3L	TMMSK3L	R/W	16	0AH	Timing comparison mask register 3	TMMSK3	R/W	32
30H	F062AH	Timing comparison mask register 3H	TMMSK3H	R/W	16					
30H	F062CH	Timing comparison mask register 4L	TMMSK4L	R/W	16	0BH	Timing comparison mask register 4	TMMSK4	R/W	32
30H	F062EH	Timing comparison mask register 4H	TMMSK4H	R/W	16					
30H	F0630H	Timing comparison mask register 5L	TMMSK5L	R/W	16	0CH	Timing comparison mask register 5	TMMSK5	R/W	32
30H	F0632H	Timing comparison mask register 5H	TMMSK5H	R/W	16					

Table 4 - 12 List of Registers of the Reference Timing Controller (2/2)

Extended Special Function Register (2nd SFR)						FAA				
WIND Register Value	CPU Address	Register Name	Symbol	R/W (CPU)	Bits	FAA Address	Register Name	Symbol	R/W (FAA)	Bits
30H	F0634H	Free-running counter register L	FCNTL	R/W	16	0DH	Free-running counter register	FCNT	R/W	32
30H	F0636H	Free-running counter register H	FCNTH	R/W	16					
30H	F0638H	Free-running counter control register	FCCNT	R/W	16	0EH	Free-running counter control register	FCCNT	R/W	32
—	—	—	—	—	—					

4.12.2.1 Timing comparison registers 0 to 5, 0H to 5H, and 0L to 5L (TMCMP0 to TMCMP5, TMCMP0H to TMCMP5H, and TMCMP0L to TMCMP5L)

The TMCMPn, TMCMPnH, and TMCMPnL (n = 0 to 5) registers[Note] specify timing values for comparison with the value of the free-running counter. When using the CPU to set the value to be compared, set it while the free-running counter is halted. The CPU accesses the register in 16-bit units.

Note TMCMPnH and TMCMPnL are symbols for extended special function registers (2nd SFRs).

Figure 4 - 27 Format of Timing Comparison Registers n, nH, and nL (TMCMPn, TMCMPnH, and TMCMPnL)

Address: F0604H to F061AH (See **Table 4 - 12 List of Registers of the Reference Timing Controller**.)
FAA address: 01H to 06H (See **Table 4 - 12 List of Registers of the Reference Timing Controller**.)
After reset: 0000 0000H (TMCMPn), 0000H (TMCMPnH, TMCMPnL)
R/W: R/W

Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMCMPn	0	0	0	0	0	0	0	0	TMCMP[23:16]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMCMP[15:0]															

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMCMPnH	0	0	0	0	0	0	0	0	TMCMP[23:16]							

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMCMPnL	TMCMP[15:0]															

TMCMP [23:0]	Timing value for comparison

4.12.2.2 Timing comparison mask registers 0 to 5, 0H to 5H, and 0L to 5L (TMMSK0 to TMMSK5, TMMSK0H to TMMSK5H, and TMMSK0L to TMMSK5L)

The TMMSKn, TMMSKnH, and TMMSKnL (n = 0 to 5) registers[Note] mask timing comparison. When using the CPU to mask the timing comparison, mask it while the free-running counter is halted. The CPU accesses the register in 16-bit units.

Note TMMSKnH and TMMSKnL are symbols for extended special function registers (2nd SFRs).

Figure 4 - 28 Format of Timing Comparison Mask Registers n, nH, and nL (TMMSKn, TMMSKnH, and TMMSKnL)

Address: F061CH to F0632H (See **Table 4 - 12 List of Registers of the Reference Timing Controller**.)
FAA address: 07H to 0CH (See **Table 4 - 12 List of Registers of the Reference Timing Controller**.)
After reset: 0000 0000H (TMMSKn), 0000H (TMMSKnH, TMMSKnL)
R/W: R/W

Symbol	31	30	29	28	27	26	25	24
TMMSKn	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
	TMMSK23	TMMSK22	TMMSK21	TMMSK20	TMMSK19	TMMSK18	TMMSK17	TMMSK16
	15	14	13	12	11	10	9	8
	TMMSK15	TMMSK14	TMMSK13	TMMSK12	TMMSK11	TMMSK10	TMMSK9	TMMSK8
	7	6	5	4	3	2	1	0
	TMMSK7	TMMSK6	TMMSK5	TMMSK4	TMMSK3	TMMSK2	TMMSK1	TMMSK0

Symbol	15	14	13	12	11	10	9	8
TMMSKnH	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	TMMSK23	TMMSK22	TMMSK21	TMMSK20	TMMSK19	TMMSK18	TMMSK17	TMMSK16

Symbol	15	14	13	12	11	10	9	8
TMMSKnL	TMMSK15	TMMSK14	TMMSK13	TMMSK12	TMMSK11	TMMSK10	TMMSK9	TMMSK8
	7	6	5	4	3	2	1	0
	TMMSK7	TMMSK6	TMMSK5	TMMSK4	TMMSK3	TMMSK2	TMMSK1	TMMSK0

TMMSKm [Note]	Timing comparison mask
0	Bit m in the timing comparison register is not masked.
1	Bit m in the timing comparison register is masked to prevent comparison between the register value and free-running counter value at bit m.

Note m = 0 to 23

4.12.2.3 Free-running counter register and free-running counter registers H and L (FCNT, FCNTH, and FCNTL)

The FCNT, FCNTH, and FCNTL registers[Note] specify the value of the free-running counter. When using the CPU to read the register, be sure to read the 16 lower-order bits before the 16 higher-order bits. Only change the value while the free-running counter is halted.

Note FCNTH and FCNTL are symbols for extended special function registers (2nd SFRs).

Figure 4 - 29 Format of Free-running Counter Register and Free-running Counter Registers H and L (FCNT, FCNTH, and FCNTL)

Address: F0636H (FCNTH), F0634H (FCNTL)
FAA address: 0DH
After reset: 0000 0000H (FCNT), 0000H (FCNTH, FCNTL)
R/W: R/W

Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FCNT	0	0	0	0	0	0	0	0								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 23 to 0	Value of the free-running counter

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCNTH	0	0	0	0	0	0	0	0								

Bits 7 to 0	Value of the free-running counter

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCNTL																

Bits 15 to 0	Value of the free-running counter

4.12.2.4 Free-running counter control register (FCCNT)

The FCCNT register starts or stops the free-running counter.

Figure 4 - 30 Format of Free-running Counter Control Register (FCCNT)

Address: F0638H
FAA address: 0EH
After reset: 0000 0000H (FAA), 0000H (2nd SFR)
R/W: R/W

Symbol	31	30	29	28	27	26	25	24
FCCNT (FAA)	0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	FCEN

Symbol	15	14	13	12	11	10	9	8
FCCNT (2nd SFR)	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	FCEN

FCEN	Free-running counter enable bit
0	Stops the free-running counter.
1	Starts the free-running counter.
Starts or stops the free-running counter.	

4.12.3 Operation of the reference timing controller

To have a program start in response to timing compare-match interrupts (0 to 5) generated by the reference timing controller, set interrupt vector register n (IVn) (n = 0 to 15) corresponding to the pertinent compare-match.
Here, it is not necessary to set the START bit in the processor control register (FAACNT).
To prevent the process from being started at an unintended timing, once clear the processing requests generated in response to the timing compare-match interrupts.
The following describes the procedure to start the program by timing compare-match interrupt 0, as an example.

<1> Clear the I flag to 0 (disable interrupts).
<2> Clear the interrupt processing enable bit (IV0.IV0EN) to 0 (disable interrupt processing).
<3> Set the start address of the program to be executed in the interrupt vector bits (IV0.IV0VEC[11:0]).
<4> Clear the free-running counter enable bit (FCCNT.FCEN) to 0 (stop the free-running counter).
<5> Set the free-running counter register (FCNT) to its initial value (this step can be omitted if setting is not required).
<6> Set the value to be compared in timing comparison register 0 (TMCMP0).
<7> Set the comparison mask value in timing comparison mask register 0 (TMMSK0).
<8> Set the interrupt processing enable bit (IV0.IV0EN) to 1 (enable interrupt processing).
<9> Set the free-running counter enable bit (FCCNT.FCEN) to 1 (start the free-running counter).
<10> Set the I flag to 1 (enable interrupts).

Steps <5> to <7> can be done in any order.
For interrupt vector register n (IVn) (n = 0 to 15) assigned to timing compare-match interrupts 0 to 5, refer to **Table 4 - 7 List of Registers of the Interrupt Controller** and **Table 4 - 8 Interrupt Sources**.
To periodically generate timing compare-match interrupt 0, the following two methods are available.

1. Using timing comparison mask register 0 (TMMSK0)
   To generate timing compare-match interrupt 0 every 4096 cycles, for example, set the registers as follows.
   Timing comparison register 0 (TMCMP0): 0000 0FFFH
   Timing comparison mask register 0 (TMMSK0): 00FF F000H
   Here, after the free-running counter starts counting, the 12 higher-order bits of the free-running counter are masked. And when the 12 lower-order bits reach all 1s (FFFH) (every 4096 cycles), timing compare-match interrupt 0 is generated.
   This method can be used when unmasked bits in timing comparison register 0 (TMCMP0) are all 1s.
2. Not using timing comparison mask register 0 (TMMSK0)
   To generate timing compare-match interrupt 0 every 500 cycles, for example, set the registers as follows.
   Timing comparison register 0 (TMCMP0): 0000 01F4H
   Timing comparison mask register 0 (TMMSK0): 0000 0000H (not used)
   Here, when the value of the free-running counter matches the value set in the timing comparison register, timing compare-match interrupt 0 is generated. In the processing routine for the interrupt, set the timing comparison register to its current value + 0000 01F4H as the value for matching to serve as the trigger for the next round of processing.
   When using this method, the timing comparison register should be set again.

4.13 Address Bus Select Function

4.13.1 Overview of the address bus select function

This function allows direct access to the peripheral functions of this product by the processor.
By using the address bus select register (ADBSEL), either access from the CPU or access from the FAA can be selected.
This function enables direct access to the registers of the following peripheral functions by the processor.

- A/D converter
- D/A converter
- Timer RD2/PWM option unit A (PWMOPA)
- Timer RG2
- Timer RX
- 16-bit timers KB30, KB31, and KB32
- Comparator
- Programmable gain amplifier
- Serial array unit
- Timer array unit
- Port functions (port 1)
- Security functions

Setting the bus select bit (xxSEL) to 1 in the address bus select register (ADBSEL) corresponding to the peripheral function allows direct access to the control registers of the peripheral function by the processor.

4.13.2 List of registers of the address bus select function

Table 4 - 13 lists the registers of the address bus select function.

Table 4 - 13 List of Registers of the Address Bus Select Function

Extended Special Function Register (2nd SFR)					FAA				
CPU Address	Register Name	Symbol	R/W (CPU)	Bits	FAA Address	Register Name	Symbol	R/W (FAA)	Bits
F04B0H	Address bus select register	ADBSEL	R/W	16	—	—	—	—	—
—	—	—	—	—	BEH	FAA register access trigger register	FAAAC	R/W	16
—	—	—	—	—	BFH	FAA address pointer	FAAAP	R/W	16

4.13.2.1 Address bus select register (ADBSEL)

The ADBSEL register selects access to the peripheral functions. This register is accessible from the RL78 CPU and is set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H. The ADBSEL register is not accessible from the FAA.

Figure 4 - 31 Format of Address Bus Select Register (ADBSEL)

Address: F04B0H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
ADBSEL	FAADIVSEL	0	TRNGSEL	PORTSEL	TKB32SEL	TKB31SEL	TKB30SEL	TRGSEL

7	6	5	4	3	2	1	0
TRD0SEL	PWMOP SEL	TRXSEL	DACSEL	PGACMP SEL	ADCSEL	SAU0SEL	TAU0SEL

xxSEL	Bus select bit
0	Bus access by the CPU is enabled.
1	Bus access by the FAA is enabled.

Selects the source of bus access to the registers of the peripheral functions. The correspondences between the bits and peripheral functions are as follows.
FAADIVSEL: Divider (for the FAA)
TRNGSEL: Security functions
PORTSEL: Port functions (port 1)
TKB32SEL: 16-bit timer KB32
TKB31SEL: 16-bit timer KB31
TKB30SEL: 16-bit timer KB30
TRGSEL: Timer RG2
TRDSEL: Timer RD2
PWMOPSEL: Timer RD2/PWM option unit A
TRXSEL: Timer RX
DACSEL: D/A converter
PGACMPSEL: Programmable gain amplifier, comparator
ADCSEL: A/D converter
SAU0SEL: Serial array unit 0
TAU0SEL: Timer array unit

Remark Regardless of the setting of the ADCSEL bit, the ADCR, ADCRH, ADCR0, ADCR0H, ADCR1, ADCR1H, ADCR2, ADCR2H, ADCR3, and ADCR3H registers are accessible by both the CPU and FAA at the same time.

4.13.2.2 FAA address pointer (FAAAP)

The FAAAP register specifies the register addresses of peripheral functions. The processor can access a register of a peripheral function by specifying the register address of the peripheral functions in the FAAP bits and executing the I/O instructions through the FAA register access trigger register (FAAAC). The FAAAP register is not accessible from the CPU. For the registers of peripheral functions that are accessible by using the FAA address pointer, see **Table 4 - 15 List of FAA Addresses and Access Size for Registers (2nd SFRs) of Peripheral Functions**.

Figure 4 - 32 Format of FAA Address Pointer (FAAAP)

Address: -
FAA address: BFH
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
FAAAP	FAAWD	0	0	FAA1ST	FAAP[11:8]			

7	6	5	4	3	2	1	0
FAAP[7:0]							

FAAWD	Selects the unit of access.
0	Byte
1	Word

FAA1ST	Selects the register region containing the target register.
0	Extended special function register (2nd SFR)
1	Special function register (SFR)

FAAP[11:0]	Specify the access address.
See **Tables 4 - 14** and **4 - 15**.	Specify the 12 lower-order bits of the register address of the peripheral function for access. For example, when access is to be to 12-bit/10-bit A/D conversion result register n (ADCRn), FAAP[11:0] = F1EH.

4.13.2.3 FAA register access trigger register (FAAAC)

The FAAAC register acts as the trigger for access to a register of a peripheral function at the address specified by the FAA address pointer (FAAAP). Using the OUT instruction of the processor to write a value to the FAAAC register acts as the trigger for writing to the given register of the peripheral function. Using the IN instruction of the processor to read the value of the given register of the peripheral function from the FAAAC register acts as the trigger for reading. The FAAAC register is not accessible from the CPU.

Figure 4 - 33 Format of FAA Register Access Trigger Register (FAAAC)

Address: -
FAA address: BEH
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAAAC																

Bits 15 to 0	In the case of writing, the value written to this register is written to the SFR specified by the FAAAP register. In the case of reading, the value of the SFR specified by the FAAAP register is read.

4.13.3 Operation of the address bus select function

To allow direct access to the peripheral functions of this product by the processor, the bus select bit (xxSEL) in the address bus select register (ADBSEL) corresponding to the pertinent peripheral function should be set to 1.
There are two different types of register access to the peripheral functions as described below.

- Access to a peripheral function register through the FAA address map
- Access to a peripheral function register by using the FAA address pointer (FAAAP)

The following describes the access procedures.

<1> Access to a peripheral function register through the FAA address map

1. Byte access to CPU addresses (read)
   - Set the bus select bit (xxSEL) corresponding to the pertinent peripheral function to 1.
   - Execute the IN instruction to read.
     The value read from the register is loaded to the 8 lower-order bits (7 to 0) in the accumulator register (A0) of the processor. The higher-order bits, 31 to 8, are all 0.
2. Byte access to CPU addresses (write)
   - Set the bus select bit (xxSEL) corresponding to the pertinent peripheral function to 1.
   - Execute the OUT instruction to write.
     The value of the 8 lower-order bits (7 to 0) in the accumulator register (A0) of the processor is written to the register. The higher-order bits, 31 to 8, are invalid.
3. Word access (only to CPU even addresses)
   - Set the bus select bit (xxSEL) corresponding to the pertinent peripheral function to 1.
   - Execute the IN instruction to read.
     The value read from the register is loaded to the 16 lower-order bits (15 to 0) in the accumulator register (A0) of the processor. The higher-order bits, 31 to 16, are all 0.
   - Execute the OUT instruction to write.
     The value of the 16 lower-order bits (15 to 0) in the accumulator register (A0) of the processor is written to the register. The higher-order bits, 31 to 16, are invalid.

<2> Access to a peripheral function register by using the FAA address pointer (FAAAP)

1. Byte access to CPU addresses (read)
   - Set the bus select bit (xxSEL) corresponding to the pertinent peripheral function to 1.
   - Specify the 12 lower-order bits of the address of the peripheral function register for access in the FAAP[11:0] bits.
   - Specify the SFR region (SFR or 2nd SFR) in the register type select bit (FAA1ST).
   - Set the access size select bit (FAAWD) to 0.
   - Execute the IN instruction to read from the FAA register access trigger register (FAAAC).
     The value read from the register is loaded to the 8 lower-order bits (7 to 0) in the accumulator register (A0) of the processor. The higher-order bits, 31 to 8, are all 0.
2. Byte access to CPU addresses (write)
   - Set the bus select bit (xxSEL) corresponding to the pertinent peripheral function to 1.
   - Specify the 12 lower-order bits of the address of the peripheral function register for access in the FAAP[11:0] bits.
   - Specify the SFR region (SFR or 2nd SFR) in the register type select bit (FAA1ST).
   - Set the access size select bit (FAAWD) to 0.
   - Execute the OUT instruction to write to the FAA register access trigger register (FAAAC).
     The value of the 8 lower-order bits (7 to 0) in the FAAAC register is written to the register. The higher-order bits, 15 to 8, are invalid.
3. Word access (only to CPU even addresses)
   - Set the bus select bit (xxSEL) corresponding to the pertinent peripheral function to 1.
   - Specify the 12 lower-order bits of the address of the peripheral function register for access in the FAAP[11:0] bits.
   - Specify the SFR region (SFR or 2nd SFR) in the register type select bit (FAA1ST).
   - Set the access size select bit (FAAWD) to 1.
   - Execute the IN instruction to read from the FAA register access trigger register (FAAAC).
     The value read from the register is loaded to the 16 lower-order bits (15 to 0) in the accumulator register (A0) of the processor. The higher-order bits, 31 to 16, are all 0.
   - Execute the OUT instruction to write to the FAA register access trigger register (FAAAC).
     The value of the 16 bits (15 to 0) in the FAAAC register is written to the register.

Tables 4 - 14 and **4 - 15** list the FAA addresses and access size for the registers of the peripheral functions. For details on the target registers, refer to the sections on the individual peripheral functions.

Table 4 - 14 List of FAA Addresses and Access Size for Registers of Peripheral Functions (1/2)

FAA Address	At the Time of Access through the FAA Address Pointer			Special Function Register (SFR) Name	Symbol[Note]		R/W	Byte-unit Access	Word-unit Access
	FAAWD	FAA1ST	FAAP [11:0]						
—	0	1	F01H	Port register 1	P1		R/W	✓	—
—	0/1	1	F10H	Serial data register 00	TXD0/ SIO00	SDR00	R/W	✓	✓
—	0/1	1	F12H	Serial data register 01	RXD0/ SIO01	SDR01	R/W	✓	✓
—	1	1	F18H	Timer data register 00	TDR00		R/W	—	✓
—	0/1	1	F1AH	Timer data register 01	TDR01L	TDR01	R/W	✓	✓
—	0	1	F1BH		TDR01H			✓	
5EH	1	1	F1EH	12-bit/10-bit A/D conversion result register	ADCR		R	—	✓
—	0	1	F1FH	8-bit A/D conversion result register H	ADCRH		R	✓	—
—	0	1	F21H	Port mode register 1	PM1		R/W	✓	—
5FH	0	1	F30H	A/D converter mode register 0	ADM0		R/W	✓	—
60H	0	1	F31H	Analog input channel specification register	ADS		R/W	✓	—
61H	0	1	F32H	A/D converter mode register 1	ADM1		R/W	✓	—
—	0/1	1	F44H	Serial data register 02	TXD1/ SIO10	SDR02	R/W	✓	✓
—	0/1	1	F46H	Serial data register 03	RXD1/ SIO11	SDR03	R/W	✓	✓
5CH	1	1	F60H	Timer RG general register C	TRGGRC		R/W	—	✓
5DH	1	1	F62H	Timer RG general register D	TRGGRD		R/W	—	✓
—	1	1	F64H	Timer data register 02	TDR02		R/W	—	✓
—	0/1	1	F66H	Timer data register 03	TDR03L	TDR03	R/W	✓	✓
—	0	1	F67H		TDR03H			✓	
40H	1	1	F6CH	Timer RD general register C0	TRDGRC0		R/W	—	✓
41H	1	1	F6EH	Timer RD general register D0	TRDGRD0		R/W	—	✓
42H	1	1	F70H	Timer RD general register C1	TRDGRC1		R/W	—	✓
43H	1	1	F72H	Timer RD general register D1	TRDGRD1		R/W	—	✓
44H	1	1	F74H	Timer RD extended compare register D0	TRDCMPD0		R/W	—	✓
45H	1	1	F76H	Timer RD extended compare register C1	TRDCMPC1		R/W	—	✓
46H	1	1	F78H	Timer RD extended compare register D1	TRDCMPD1		R/W	—	✓
47H	1	1	F7AH	Timer RD A/D conversion trigger buffer register 0/timer-KB PWM output gating mode buffer register	TRDADTB0/ TRDCMPF1		R/W	—	✓
48H	1	1	F7CH	Timer RD A/D conversion trigger buffer register 1	TRDADTB1		R/W	—	✓
—	1	1	F7EH	Timer RD simultaneous update trigger register	TRDRDT		R/W	—	✓
49H	0	1	F7EH	Timer RD simultaneous update trigger register 0	TRDRDT0		R/W	✓	—

Table 4 - 14 List of FAA Addresses and Access Size for Registers of Peripheral Functions (2/2)

FAA Address	At the Time of Access through the FAA Address Pointer			Special Function Register (SFR) Name	Symbol[Note]	R/W	Byte-unit Access	Word-unit Access
	FAAWD	FAA1ST	FAAP [11:0]					
4AH	0	1	F7FH	Timer RD simultaneous update trigger register 1	TRDRDT1	R/W	✓	—

Note For access through the FAA address pointer, "_PTR" is appended to the end of the symbol.

Table 4 - 15 List of FAA Addresses and Access Size for Registers (2nd SFRs) of Peripheral Functions (1/9)

FAA Address	At the Time of Access through the FAA Address Pointer			Extended Special Function Register (2nd SFR) Name	Symbol[Note 1]		R/W	Byte-unit Access	Word-unit Access
	FAAWD	FAA1ST	FAAP [11:0]						
62H	0	0	010H	A/D converter mode register 2	ADM2		R/W	✓	—
63H	0	0	011H	Conversion result comparison upper limit setting register	ADUL		R/W	✓	—
64H	0	0	012H	Conversion result comparison lower limit setting register	ADLL		R/W	✓	—
—	0	0	013H	A/D test register	ADTES		R/W	✓	—
—	0	0	014H	A/D converter mode register 3	ADM3		R/W	✓	—
—	0	0	015H	Analog input channel specification register 0	ADS0		R/W	✓	—
—	0	0	016H	Analog input channel specification register 1	ADS1		R/W	✓	—
—	0	0	017H	Analog input channel specification register 2	ADS2		R/W	✓	—
—	0	0	018H	Analog input channel specification register 3	ADS3		R/W	✓	—
—	0	0	019H	Conversion setting register	ADSCTL		R/W	✓	—
—	0	0	01AH	Conversion trigger specification register 0	ADTR0		R/W	✓	—
—	0	0	01BH	Conversion trigger specification register 1	ADTR1		R/W	✓	—
—	0	0	01CH	Conversion trigger specification register 2	ADTR2		R/W	✓	—
—	0	0	01DH	Conversion trigger specification register 3	ADTR3		R/W	✓	—
—	0	0	01FH	A/D conversion sampling mode specification register	ADSPMOD		R/W	✓	—
65H	1	0	020H	12-bit/10-bit A/D conversion result register 0	—	ADCR0	R	—	✓
—	0	0	021H	8-bit A/D conversion result register 0H	ADCR0H		R	✓	—
66H	1	0	022H	12-bit/10-bit A/D conversion result register 1	—	ADCR1	R	—	✓
—	0	0	023H	8-bit A/D conversion result register 1H	ADCR1H		R	✓	—
67H	1	0	024H	12-bit/10-bit A/D conversion result register 2	—	ADCR2	R	—	✓
—	0	0	025H	8-bit A/D conversion result register 2H	ADCR2H		R	✓	—
68H	1	0	026H	12-bit/10-bit A/D conversion result register 3	—	ADCR3	R	—	✓
—	0	0	027H	8-bit A/D conversion result register 3H	ADCR3H		R	✓	—
—	0	0	028H	Conversion interrupt control register	ADINTCTL		R/W	✓	—
—	0	0	029H	Conversion interrupt status register	ADINTST		R/W	✓	—
—	0	0	031H	Pull-up resistor option register 1	PU1		R/W	✓	—
—	0	0	041H	Port input mode register 1	PIM1		R/W	✓	—
—	0	0	051H	Port output mode register 1	POM1		R/W	✓	—
—	0	0	061H	Port mode control A register 1	PMCA1		R/W	✓	—
—	0/1	0	100H	Serial status register 00	SSR00L	SSR00	R	✓	✓
—	0/1	0	102H	Serial status register 01	SSR01L	SSR01	R	✓	✓
—	0/1	0	104H	Serial status register 02	SSR02L	SSR02	R	✓	✓
—	0/1	0	106H	Serial status register 03	SSR03L	SSR03	R	✓	✓
—	0/1	0	108H	Serial flag clear trigger register 00	SIR00L	SIR00	R/W	✓	✓
—	0/1	0	10AH	Serial flag clear trigger register 01	SIR01L	SIR01	R/W	✓	✓

Table 4 - 15 List of FAA Addresses and Access Size for Registers (2nd SFRs) of Peripheral Functions (2/9)

FAA Address	At the Time of Access through the FAA Address Pointer			Extended Special Function Register (2nd SFR) Name	Symbol[Note 1]		R/W	Byte-unit Access	Word-unit Access
	FAAWD	FAA1ST	FAAP [11:0]						
—	0/1	0	10CH	Serial flag clear trigger register 02	SIR02L	SIR02	R/W	✓	✓
—	0/1	0	10EH	Serial flag clear trigger register 03	SIR03L	SIR03	R/W	✓	✓
—	1	0	110H	Serial mode register 00	SMR00		R/W	—	✓
—	1	0	112H	Serial mode register 01	SMR01		R/W	—	✓
—	1	0	114H	Serial mode register 02	SMR02		R/W	—	✓
—	1	0	116H	Serial mode register 03	SMR03		R/W	—	✓
—	1	0	118H	Serial communication operation setting register 00	SCR00		R/W	—	✓
—	1	0	11AH	Serial communication operation setting register 01	SCR01		R/W	—	✓
—	1	0	11CH	Serial communication operation setting register 02	SCR02		R/W	—	✓
—	1	0	11EH	Serial communication operation setting register 03	SCR03		R/W	—	✓
—	0/1	0	120H	Serial channel enable status register 0	SE0L	SE0	R	✓	✓
—	0/1	0	122H	Serial channel start register 0	SS0L	SS0	R/W	✓	✓
—	0/1	0	124H	Serial channel stop register 0	ST0L	ST0	R/W	✓	✓
—	0/1	0	126H	Serial clock select register 0	SPS0L	SPS0	R/W	✓	✓
—	1	0	128H	Serial output register 0	SO0		R/W	—	✓
—	0/1	0	12AH	Serial output enable register 0	SOE0L	SOE0	R/W	✓	✓
—	0/1	0	134H	Serial output level register 0	SOL0L	SOL0	R/W	✓	✓
—	0/1	0	138H	Serial standby control register 0	SSC0L	SSC0	R/W	✓	✓
—	1	0	180H	Timer counter register 00	TCR00		R	—	✓
—	1	0	182H	Timer counter register 01	TCR01		R	—	✓
—	1	0	184H	Timer counter register 02	TCR02		R	—	✓
—	1	0	186H	Timer counter register 03	TCR03		R	—	✓
—	1	0	190H	Timer mode register 00	TMR00		R/W	—	✓
—	1	0	192H	Timer mode register 01	TMR01		R/W	—	✓
—	1	0	194H	Timer mode register 02	TMR02		R/W	—	✓
—	1	0	196H	Timer mode register 03	TMR03		R/W	—	✓
—	0/1	0	1A0H	Timer status register 00	TSR00L	TSR00	R	✓	✓
—	0/1	0	1A2H	Timer status register 01	TSR01L	TSR01	R	✓	✓
—	0/1	0	1A4H	Timer status register 02	TSR02L	TSR02	R	✓	✓
—	0/1	0	1A6H	Timer status register 03	TSR03L	TSR03	R	✓	✓
—	0/1	0	1B0H	Timer channel enable status register 0	TE0L	TE0	R	✓	✓
—	0/1	0	1B2H	Timer channel start register 0	TS0L	TS0	R/W	✓	✓
—	0/1	0	1B4H	Timer channel stop register 0	TT0L	TT0	R/W	✓	✓
—	1	0	1B6H	Timer clock select register 0	TPS0		R/W	—	✓
—	0/1	0	1B8H	Timer output register 0	TO0L	TO0	R/W	✓	✓

Table 4 - 15 List of FAA Addresses and Access Size for Registers (2nd SFRs) of Peripheral Functions (3/9)

FAA Address	At the Time of Access through the FAA Address Pointer			Extended Special Function Register (2nd SFR) Name	Symbol[Note 1]		R/W	Byte-unit Access	Word-unit Access
	FAAWD	FAA1ST	FAAP [11:0]						
—	0/1	0	1BAH	Timer output enable register 0	TOE0L	TOE0	R/W	✓	✓
—	0/1	0	1BCH	Timer output level register 0	TOL0L	TOL0	R/W	✓	✓
—	0/1	0	1BEH	Timer output mode register 0	TOM0L	TOM0	R/W	✓	✓
—	0	0	330H	D/A converter mode register 0	DAM0		R/W	✓	—
—	0	0	331H	D/A converter mode register 1	DAM1		R/W	✓	—
—	0	0	332H	D/A converter mode register 2	DAM2		R/W	✓	—
69H	0	0	333H	D/A conversion value setting register 2	DACS2		R/W	✓	—
6AH	1	0	334H	D/A conversion value setting register 0	DACS0		R/W	—	✓
6BH	0/1	0	336H	D/A conversion value setting register 1	DACS1L	DACS1	R/W	✓	✓
—	0	0	340H	Comparator mode setting register 0	COMPMDR0		R/W	✓	—
—	0	0	341H	Comparator filter control register 0	COMPFIR0		R/W	✓	—
—	0	0	342H	Comparator output control register 0	COMPOCR0		R/W	✓	—
—	0	0	344H	Comparator mode setting register 1	COMPMDR1		R/W	✓	—
—	0	0	345H	Comparator filter control register 1	COMPFIR1		R/W	✓	—
—	0	0	346H	Comparator output control register 1	COMPOCR1		R/W	✓	—
—	0	0	347H	PGA control register	PGACTL		R/W	✓	—
6CH	0	0	348H	PGA input channel select register	PGAINS		R/W	✓	—
—	0	0	34AH	Comparator 0 input signal selection control register	CMP0SEL		R/W	✓	—
—	0	0	34BH	Comparator 1 input signal selection control register	CMP1SEL		R/W	✓	—
—	0	0	34CH	Comparator 2 input signal selection control register	CMP2SEL		R/W	✓	—
—	0	0	34DH	Comparator 3 input signal selection control register	CMP3SEL		R/W	✓	—
—	0	0	34EH	Comparator output control register 2	COMPOCR2		R/W	✓	—
6DH	1	0	350H	Timer RX counter	TRX		R/W	—	✓
—	1	0	352H	Timer RX count buffer counter	TRXBUF		R	—	✓
—	0	0	354H	Timer RX function control register 1	TRXCR1		R/W	✓	—
—	0	0	355H	Timer RX function control register 2	TRXCR2		R/W	✓	—
—	0	0	356H	Timer RX status register	TRXSR		R/W	✓	—
—	0	0	358H	PWMOPA control register 0	OPCTL0		R/W	✓	—
—	0	0	359H	PWMOPA cutoff control register 0	OPDF0		R/W	✓	—
—	0	0	35AH	PWMOPA cutoff control register 1	OPDF1		R/W	✓	—
—	0	0	35BH	PWMOPA edge selection register	OPEDGE		R/W	✓	—
—	0	0	35CH	PWMOPA status register	OPSR		R	✓	—
—	0	0	390H	Timer RD ELC register	TRDELC		R/W	✓	—
—	0	0	391H	Timer RD timer-KB PWM output gating mode control register	TRDBCR		R/W	✓	—

Table 4 - 15 List of FAA Addresses and Access Size for Registers (2nd SFRs) of Peripheral Functions (4/9)

FAA Address	At the Time of Access through the FAA Address Pointer			Extended Special Function Register (2nd SFR) Name	Symbol[Note 1]	R/W	Byte-unit Access	Word-unit Access
	FAAWD	FAA1ST	FAAP [11:0]					
—	0	0	392H	Timer RD timer KB PWM output monitor register	TRDBOF	R	✓	—
—	0	0	393H	Timer RD start register	TRDSTR	R/W	✓	—
—	0	0	394H	Timer RD mode register	TRDMR	R/W	✓	—
—	0	0	395H	Timer RD PWM function select register	TRDPMR	R/W	✓	—
—	0	0	396H	Timer RD function control register	TRDFCR	R/W	✓	—
56H	0	0	397H	Timer RD output master enable register 1	TRDOER1	R/W	✓	—
—	0	0	398H	Timer RD output master enable register 2	TRDOER2	R/W	✓	—
—	0	0	399H	Timer RD output control register	TRDOCR	R/W	✓	—
—	0	0	39AH	Timer RD digital filter function select register 0	TRDDF0	R/W	✓	—
—	0	0	39BH	Timer RD digital filter function select register 1	TRDDF1	R/W	✓	—
—	0	0	3A0H	Timer RD control register 0	TRDCR0	R/W	✓	—
—	0	0	3A1H	Timer RD I/O control register A0	TRDIORA0	R/W	✓	—
—	0	0	3A2H	Timer RD I/O control register C0	TRDIORC0	R/W	✓	—
—	0	0	3A3H	Timer RD status register 0	TRDSR0	R/W	✓	—
—	0	0	3A4H	Timer RD interrupt enable register 0	TRDIER0	R/W	✓	—
—	0	0	3A5H	Timer RD PWM output level control register 0	TRDPOCR0	R/W	✓	—
4BH	1	0	3A6H	Timer RD counter 0	TRD0	R/W	—	✓
4CH	1	0	3A8H	Timer RD general register A0	TRDGRA0	R/W	—	✓
4DH	1	0	3AAH	Timer RD general register B0	TRDGRB0	R/W	—	✓
—	0	0	3B0H	Timer RD control register 1	TRDCR1	R/W	✓	—
—	0	0	3B1H	Timer RD I/O control register A1	TRDIORA1	R/W	✓	—
—	0	0	3B2H	Timer RD I/O control register C1	TRDIORC1	R/W	✓	—
—	0	0	3B3H	Timer RD status register 1	TRDSR1	R/W	✓	—
—	0	0	3B4H	Timer RD interrupt enable register 1	TRDIER1	R/W	✓	—
—	0	0	3B5H	Timer RD PWM output level control register 1	TRDPOCR1	R/W	✓	—
4EH	1	0	3B6H	Timer RD counter 1	TRD1	R/W	—	✓
4FH	1	0	3B8H	Timer RD general register A1	TRDGRA1	R/W	—	✓
50H	1	0	3BAH	Timer RD general register B1	TRDGRB1	R/W	—	✓
51H	1	0	3C0H	Timer RD extended compare register B0	TRDCMPB0	R/W	—	✓
52H	1	0	3C4H	Timer RD extended compare register A1	TRDCMPA1	R/W	—	✓
53H	1	0	3C8H	Timer RD extended compare register B1	TRDCMPB1	R/W	—	✓
54H	1	0	3CCH	Timer RD A/D conversion trigger compare register 0/timer-KB PWM output gating mode compare register	TRDADTC0/ TRDCMPE1	R/W	—	✓
55H	1	0	3D0H	Timer RD A/D conversion trigger compare register 1	TRDADTC1	R/W	—	✓

Table 4 - 15 List of FAA Addresses and Access Size for Registers (2nd SFRs) of Peripheral Functions (5/9)

FAA Address	At the Time of Access through the FAA Address Pointer			Extended Special Function Register (2nd SFR) Name	Symbol[Note 1]	R/W	Byte-unit Access	Word-unit Access
	FAAWD	FAA1ST	FAAP [11:0]					
—	1	0	3D6H	Timer RD simultaneous update flag register	TRDRSF	R	—	✓
57H	0	0	3D6H	Timer RD simultaneous update flag register 0	TRDRSF0	R	✓	—
58H	0	0	3D7H	Timer RD simultaneous update flag register 1	TRDRSF1	R	✓	—
—	0	0	3D8H	Timer RD A/D conversion trigger control register	TRDADCR	R/W	✓	—
—	0	0	3E0H	Timer RG mode register 0	TRGMR0	R/W	✓	—
—	0	0	3E1H	Timer RG count control register	TRGCNTC	R/W	✓	—
—	0	0	3E2H	Timer RG control register	TRGCR	R/W	✓	—
—	0	0	3E3H	Timer RG interrupt enable register 0	TRGIER0	R/W	✓	—
—	0	0	3E4H	Timer RG status register 0	TRGSR0	R/W	✓	—
—	0	0	3E5H	Timer RG I/O control register	TRGIOR	R/W	✓	—
59H	1	0	3E6H	Timer RG counter	TRG	R/W	—	✓
5AH	1	0	3E8H	Timer RG general register A	TRGGRA	R/W	—	✓
5BH	1	0	3EAH	Timer RG general register B	TRGGRB	R/W	—	✓
—	0	0	3F0H	Timer RG mode register 1	TRGMR1	R/W	✓	—
—	0	0	3F1H	Timer RG output enable register	TRGOER	R/W	✓	—
—	0	0	3F2H	Timer RG output control register	TRGOCR	R/W	✓	—
—	0	0	3F3H	Timer RG interrupt enable register 1	TRGIER1	R/W	✓	—
—	0	0	3F4H	Timer RG status register 1	TRGSR1	R/W	✓	—
—	0	0	3F5H	Timer RG start register	TRGSTR	R/W	✓	—
—	0	0	3F6H	Timer RG phase counting control register 0	TRGCTL0	R/W	✓	—
—	0	0	3F7H	Timer RG phase counting control register 1	TRGCTL1	R/W	✓	—
—	1	0	3F8H	Timer RG phase change time measurement counter	TRGPMC	R/W	—	✓
—	1	0	3FAH	Timer RG phase change time capture register 0	TRGCAP0	R	—	✓
—	1	0	3FCH	Timer RG phase change time capture register 1	TRGCAP1	R	—	✓
96H	1	0	400H	16-bit timer KB compare register 20	TKBCR20	R/W	—	✓
97H	1	0	402H	16-bit timer KB compare register 21	TKBCR21	R/W	—	✓
98H	1	0	404H	16-bit timer KB compare register 22	TKBCR22	R/W	—	✓
99H	1	0	406H	16-bit timer KB compare register 23	TKBCR23	R/W	—	✓
9AH	1	0	408H	16-bit timer KB trigger compare register 2	TKBTGCR2	R/W	—	✓
9BH	1	0	40AH	16-bit timer KB smooth start initial duty register 20	TKBSIR20	R/W	—	✓
9CH	1	0	40CH	16-bit timer KB smooth start initial duty register 21	TKBSIR21	R/W	—	✓
—	0	0	40EH	16-bit timer KB dithering count register 20	TKBDNR20	R/W	✓	—

Table 4 - 15 List of FAA Addresses and Access Size for Registers (2nd SFRs) of Peripheral Functions (6/9)

FAA Address	At the Time of Access through the FAA Address Pointer			Extended Special Function Register (2nd SFR) Name	Symbol[Note 1]	R/W	Byte-unit Access	Word-unit Access
	FAAWD	FAA1ST	FAAP [11:0]					
—	0	0	40FH	16-bit timer KB smooth start step width register 20	TKBSSR20	R/W	✓	—
—	0	0	410H	16-bit timer KB dithering count register 21	TKBDNR21	R/W	✓	—
—	0	0	411H	16-bit timer KB smooth start step width register 21	TKBSSR21	R/W	✓	—
9DH	0	0	412H	16-bit timer KB trigger register 2	TKBTRG2	W	✓	—
9EH	0	0	413H	16-bit timer KB flag register 2	TKBFLG2	R	✓	—
9FH	1	0	414H	16-bit timer KB compare 1L & dithering count register 20	TKBCRLD20	R/W	—	✓
A0H	1	0	416H	16-bit timer KB compare 3L & dithering count register 21	TKBCRLD21	R/W	—	✓
—	1	0	420H	16-bit timer counter KB2	TKBCNT2	R	—	✓
—	1	0	422H	16-bit timer KB operation control register 20	TKBCTL20	R/W	—	✓
—	1	0	424H	16-bit timer KB maximum frequency limit setting register 2	TKBMFR2	R/W	—	✓
—	0	0	426H	16-bit timer KB output control register 20	TKBIOC20	R/W	✓	—
—	0	0	427H	16-bit timer KB flag clear trigger register 2	TKBCLR2	W	✓	—
—	0	0	428H	16-bit timer KB output control register 21	TKBIOC21	R/W	✓	—
—	0	0	429H	16-bit timer KB operation control register 21	TKBCTL21	R/W	✓	—
—	1	0	42AH	16-bit timer KB operation control register 22	TKBCTL22	R/W	—	✓
—	1	0	430H	Forced output stop function control register 20	TKBPACTL20	R/W	—	✓
—	1	0	432H	Forced output stop function control register 21	TKBPACTL21	R/W	—	✓
A5H	0	0	434H	Forced output stop function 1 start trigger register 2	TKBPAHFS2	W	✓	—
A6H	0	0	435H	Forced output stop function 1 cancel trigger register 2	TKBPAHFT2	W	✓	—
A7H	0	0	436H	Forced output stop function flag register 2	TKBPAFLG2	R	✓	—
—	0	0	437H	Forced output stop function control register 22	TKBPACTL22	R/W	✓	—
—	0	0	438H	Forced output stop function control register 23	TKBPACTL23	R/W	✓	—
—	0	0	439H	Forced output stop function control register 24	TKBPACTL24	R/W	✓	—
A8H	1	0	43AH	Pulse characteristics measurement capture register 20	TKBPAPLS20	R	—	✓
A9H	1	0	43CH	Pulse characteristics measurement capture register 21	TKBPAPLS21	R	—	✓
—	1	0	43EH	Pulse characteristics measurement capture register 20L	TKBPAPLS20L	R/W	✓	—
—	1	0	43FH	Pulse characteristics measurement capture register 21L	TKBPAPLS21L	R/W	✓	—

Table 4 - 15 List of FAA Addresses and Access Size for Registers (2nd SFRs) of Peripheral Functions (7/9)

FAA Address	At the Time of Access through the FAA Address Pointer			Extended Special Function Register (2nd SFR) Name	Symbol[Note 1]	R/W	Byte-unit Access	Word-unit Access
	FAAWD	FAA1ST	FAAP [11:0]					
—	0	0	490H	16-bit timer KB skipping control register 0	TKBTCTL0	R/W	✓	—
—	0	0	491H	16-bit timer KB times-of-skipping setting register 0	TKBTCMP0	R/W	✓	—
—	0	0	492H	16-bit timer KB skipping control register 1	TKBTCTL1	R/W	✓	—
—	0	0	493H	16-bit timer KB times-of-skipping setting register 1	TKBTCMP1	R/W	✓	—
—	0	0	494H	16-bit timer KB skipping control register 2	TKBTCTL2	R/W	✓	—
—	0	0	495H	16-bit timer KB times-of-skipping setting register 2	TKBTCMP2	R/W	✓	—
—	0	0	540H	Random number seed data register	TRNGSDR[Note 2]	R	✓	—
—	0	0	542H	Random number seed command register 0	TRNGSCR0[Note 3]	R/W	✓	—
6EH	1	0	740H	16-bit timer KB compare register 00	TKBCR00	R/W	—	✓
6FH	1	0	742H	16-bit timer KB compare register 01	TKBCR01	R/W	—	✓
70H	1	0	744H	16-bit timer KB compare register 02	TKBCR02	R/W	—	✓
71H	1	0	746H	16-bit timer KB compare register 03	TKBCR03	R/W	—	✓
72H	1	0	748H	16-bit timer KB trigger compare register 0	TKBTGCR0	R/W	—	✓
73H	1	0	74AH	16-bit timer KB smooth start initial duty register 00	TKBSIR00	R/W	—	✓
74H	1	0	74CH	16-bit timer KB smooth start initial duty register 01	TKBSIR01	R/W	—	✓
—	0	0	74EH	16-bit timer KB dithering count register 00	TKBDNR00	R/W	✓	—
—	0	0	74FH	16-bit timer KB smooth start step width register 00	TKBSSR00	R/W	✓	—
—	0	0	750H	16-bit timer KB dithering count register 01	TKBDNR01	R/W	✓	—
—	0	0	751H	16-bit timer KB smooth start step width register 01	TKBSSR01	R/W	✓	—
75H	0	0	752H	16-bit timer KB trigger register 0	TKBTRG0	W	✓	—
76H	0	0	753H	16-bit timer KB flag register 0	TKBFLG0	R	✓	—
77H	1	0	754H	16-bit timer KB compare 1L & dithering count register 00	TKBCRLD00	R/W	—	✓
78H	1	0	756H	16-bit timer KB compare 3L & dithering count register 01	TKBCRLD01	R/W	—	✓
—	1	0	760H	16-bit timer counter KB0	TKBCNT0	R	—	✓
—	1	0	762H	16-bit timer KB operation control register 00	TKBCTL00	R/W	—	✓
—	1	0	764H	16-bit timer KB maximum frequency limit setting register 0	TKBMFR0	R/W	—	✓
—	0	0	766H	16-bit timer KB output control register 00	TKBIOC00	R/W	✓	—
—	0	0	767H	16-bit timer KB flag clear trigger register 0	TKBCLR0	W	✓	—
—	0	0	768H	16-bit timer KB output control register 01	TKBIOC01	R/W	✓	—
—	0	0	769H	16-bit timer KB operation control register 01	TKBCTL01	R/W	✓	—
—	1	0	76AH	16-bit timer KB operation control register 02	TKBCTL02	R/W	—	✓

Table 4 - 15 List of FAA Addresses and Access Size for Registers (2nd SFRs) of Peripheral Functions (8/9)

FAA Address	At the Time of Access through the FAA Address Pointer			Extended Special Function Register (2nd SFR) Name	Symbol[Note 1]	R/W	Byte-unit Access	Word-unit Access
	FAAWD	FAA1ST	FAAP [11:0]					
—	1	0	770H	Forced output stop function control register 00	TKBPACTL00	R/W	—	✓
—	1	0	772H	Forced output stop function control register 01	TKBPACTL01	R/W	—	✓
7DH	0	0	774H	Forced output stop function 1 start trigger register 0	TKBPAHFS0	W	✓	—
7EH	0	0	775H	Forced output stop function 1 cancel trigger register 0	TKBPAHFT0	W	✓	—
7FH	0	0	776H	Forced output stop function flag register 0	TKBPAFLG0	R	✓	—
—	0	0	777H	Forced output stop function control register 02	TKBPACTL02	R/W	✓	—
—	0	0	778H	Forced output stop function control register 03	TKBPACTL03	R/W	✓	—
—	0	0	779H	Forced output stop function control register 04	TKBPACTL04	R/W	✓	—
80H	1	0	77AH	Pulse characteristics measurement capture register 00	TKBPAPLS00	R	—	✓
81H	1	0	77CH	Pulse characteristics measurement capture register 01	TKBPAPLS01	R	—	✓
—	0	0	77EH	Pulse characteristics measurement capture register 00L	TKBPAPLS00L	R/W	✓	—
—	0	0	77FH	Pulse characteristics measurement capture register 01L	TKBPAPLS01L	R/W	✓	—
82H	1	0	780H	16-bit timer KB compare register 10	TKBCR10	R/W	—	✓
83H	1	0	782H	16-bit timer KB compare register 11	TKBCR11	R/W	—	✓
84H	1	0	784H	16-bit timer KB compare register 12	TKBCR12	R/W	—	✓
85H	1	0	786H	16-bit timer KB compare register 13	TKBCR13	R/W	—	✓
86H	1	0	788H	16-bit timer KB trigger compare register 1	TKBTGCR1	R/W	—	✓
87H	1	0	78AH	16-bit timer KB smooth start initial duty register 10	TKBSIR10	R/W	—	✓
88H	1	0	78CH	16-bit timer KB smooth start initial duty register 11	TKBSIR11	R/W	—	✓
—	0	0	78EH	16-bit timer KB dithering count register 10	TKBDNR10	R/W	✓	—
—	0	0	78FH	16-bit timer KB smooth start step width register 10	TKBSSR10	R/W	✓	—
—	0	0	790H	16-bit timer KB dithering count register 11	TKBDNR11	R/W	✓	—
—	0	0	791H	16-bit timer KB smooth start step width register 11	TKBSSR11	R/W	✓	—
89H	0	0	792H	16-bit timer KB trigger register 1	TKBTRG1	W	✓	—
8AH	0	0	793H	16-bit timer KB flag register 1	TKBFLG1	R	✓	—
8BH	1	0	794H	6-bit timer KB compare 1L & dithering count register 10	TKBCRLD10	R/W	—	✓
8CH	1	0	796H	16-bit timer KB compare 3L & dithering count register 11	TKBCRLD11	R/W	—	✓

Table 4 - 15 List of FAA Addresses and Access Size for Registers (2nd SFRs) of Peripheral Functions (9/9)

FAA Address	At the Time of Access through the FAA Address Pointer			Extended Special Function Register (2nd SFR) Name	Symbol[Note 1]	R/W	Byte-unit Access	Word-unit Access
	FAAWD	FAA1ST	FAAP [11:0]					
—	1	0	7A0H	16-bit timer counter KB1	TKBCNT1	R	—	✓
—	1	0	7A2H	16-bit timer KB operation control register 10	TKBCTL10	R/W	—	✓
—	1	0	7A4H	16-bit timer KB maximum frequency limit setting register 1	TKBMFR1	R/W	—	✓
—	0	0	7A6H	16-bit timer KB output control register 10	TKBIOC10	R/W	✓	—
—	0	0	7A7H	16-bit timer KB flag clear trigger register 1	TKBCLR1	W	✓	—
—	0	0	7A8H	16-bit timer KB output control register 11	TKBIOC11	R/W	✓	—
—	0	0	7A9H	16-bit timer KB operation control register 11	TKBCTL11	R/W	✓	—
—	1	0	7AAH	16-bit timer KB operation control register 12	TKBCTL12	R/W	—	✓
—	1	0	7B0H	Forced output stop function control register 10	TKBPACTL10	R/W	—	✓
—	1	0	7B2H	Forced output stop function control register 11	TKBPACTL11	R/W	—	✓
91H	0	0	7B4H	Forced output stop function 1 start trigger register 1	TKBPAHFS1	W	✓	—
92H	0	0	7B5H	Forced output stop function 1 cancel trigger register 1	TKBPAHFT1	W	✓	—
93H	0	0	7B6H	Forced output stop function flag register 1	TKBPAFLG1	R	✓	—
—	0	0	7B7H	Forced output stop function control register 12	TKBPACTL12	R/W	✓	—
—	0	0	7B8H	Forced output stop function control register 13	TKBPACTL13	R/W	✓	—
—	0	0	7B9H	Forced output stop function control register 14	TKBPACTL14	R/W	✓	—
94H	1	0	7BAH	Pulse characteristics measurement capture register 10	TKBPAPLS10	R	—	✓
95H	1	0	7BCH	Pulse characteristics measurement capture register 11	TKBPAPLS11	R	—	✓
—	0	0	7BEH	Pulse characteristics measurement capture register 10L	TKBPAPLS10L	R/W	✓	—
—	0	0	7BFH	Pulse characteristics measurement capture register 11L	TKBPAPLS11L	R/W	✓	—

Note 1. For access through the FAA address pointer, "_PTR" is appended to the end of the symbol.

Note 2. Execute the IN instruction twice in succession to read from the FAA register access trigger register (FAAAC). The value read from the register is loaded to the 8 lower-order bits (7 to 0) in the accumulator register (A0) of the processor.

Note 3. Execute the IN instruction twice in succession to read from the FAA register access trigger register (FAAAC). The value read from the register is loaded to the 8 lower-order bits (7 to 0) in the accumulator register (A0) of the processor.
Execute the OUT instruction twice in succession to write to the FAA register access trigger register (FAAAC). The value of the 8 lower-order bits (7 to 0) in the FAAAC register is written to the register.

4.14 FAA Operation

4.14.1 Combined operation of the CPU and FAA

Operations by the CPU and FAA can be combined to suit the application. **Table 4 - 16** lists the typical cases of usage.

Table 4 - 16 Typical Cases of Usage for Combined Operations by the CPU and FAA

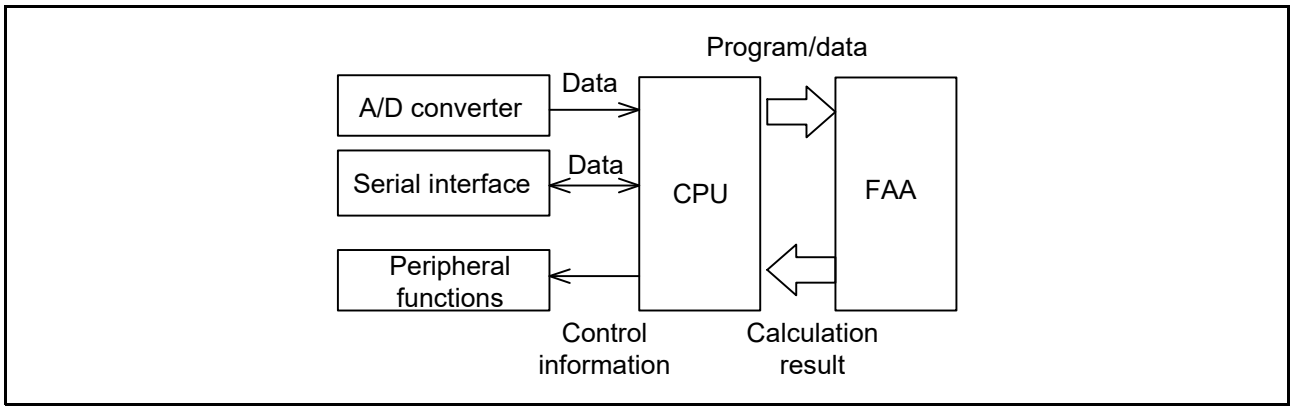
Example of Operation	FAA Operation	Typical Usage
Subprocessor operation 1	Calculation only	Code operation, pattern recognition, etc.
Subprocessor operation 2	Data acquisition from peripheral functions and calculation	Digital filter, signal analysis, etc.
Subprocessor operation 3	Calculation and control of peripheral functions	ADPCM, PWM output control, etc.
Independent operation of the FAA	Data acquisition from peripheral functions, calculation, and control of peripheral functions	Inverter control, noise cancellation, etc.

In addition to the above four examples of usage, the input event controller (GRNINPUTC), interrupt controller (GRNINTC), and reference timing controller (GRNTIMEC) can be used in various ways for various applications.

4.14.1.1 Subprocessor operation 1

The CPU only allows calculations by the FAA when this is necessary.
Before the FAA starts execution, the CPU transfers the program for calculation, data, and control information for the FAA to the instruction code memory, data memory, and CPU interface registers. The CPU only allows calculations by the FAA when this is necessary.
The results of calculations can be read from the data memory or data shared memory (SHDMEM).
Figure 4 - 34 shows the image of the subprocessor operation 1.

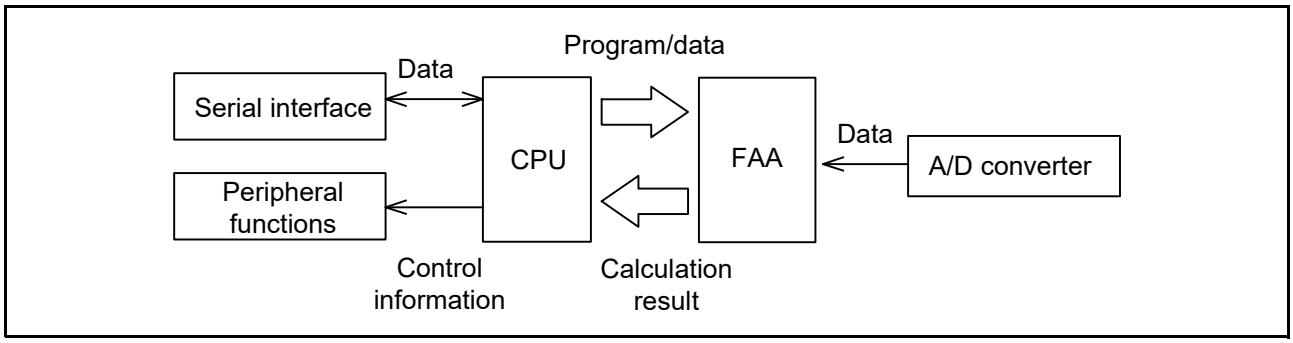
Figure 4 - 34 Image of Subprocessor Operation 1



4.14.1.2 Subprocessor operation 2

The CPU only allows acquisition of data from the peripheral functions and calculations by the FAA when this is necessary.
Before the FAA starts execution, the CPU transfers the program for calculation, data, and control information for the FAA to the instruction code memory, data memory, and CPU interface registers. The CPU only allows calculations by the FAA when this is necessary. The FAA directly acquires the data necessary for calculations from the peripheral functions. The results of calculations can be read from the data memory or data shared memory (SHDMEM).
Figure 4 - 35 shows the image of the subprocessor operation 2.

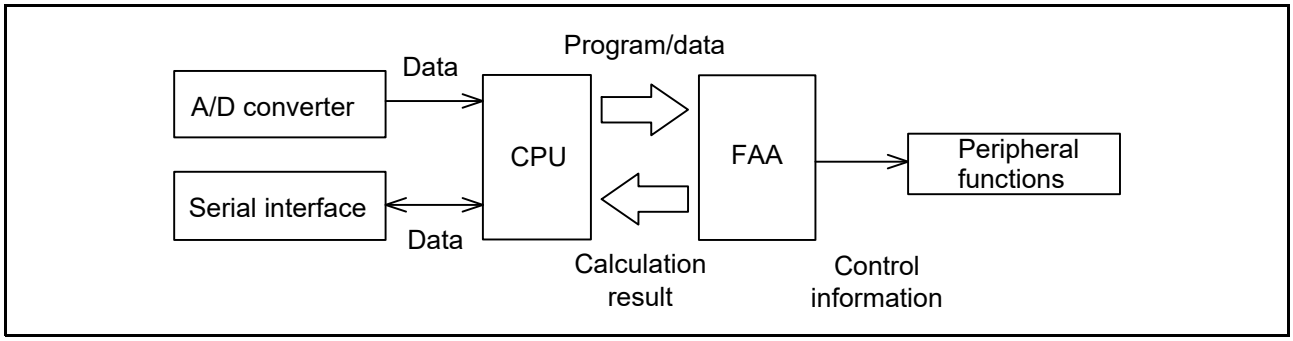
Figure 4 - 35 Image of Subprocessor Operation 2



4.14.1.3 Subprocessor operation 3

The CPU only allows calculations and control of the peripheral functions by the FAA when this is necessary.
Before the FAA starts execution, the CPU transfers the program for calculation, data, and control information for the FAA to the instruction code memory, data memory, and CPU interface registers. The CPU only allows calculations by the FAA when this is necessary. The FAA directly controls the peripheral functions based on the result of calculations.
The results of calculations can be read from the data memory or data shared memory (SHDMEM).
Figure 4 - 36 shows the image of the subprocessor operation 3.

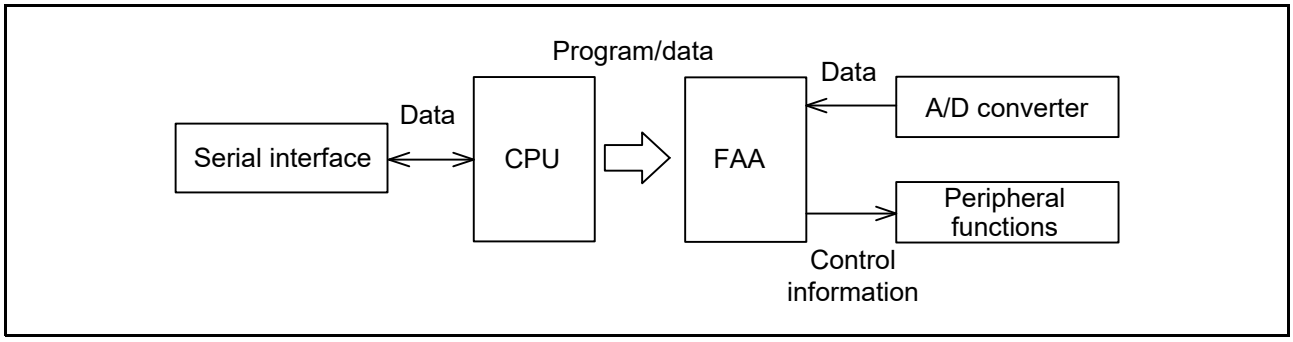
Figure 4 - 36 Image of Subprocessor Operation 3



4.14.1.4 Independent operation of the FAA

The FAA acquires data from the peripheral functions, executes calculation, and controls the peripheral functions. Before the FAA starts execution, the CPU transfers the program for calculation, data, and control information for the FAA to the instruction code memory, data memory, and CPU interface registers. The CPU then gives the instruction to the FAA to start execution. The FAA directly acquires some data necessary for calculation from the peripheral functions and executes calculation. The FAA directly controls the peripheral functions based on the results of calculations.
Figure 4 - 37 shows the image of the independent operation of the FAA.

Figure 4 - 37 Image of Independent Operation of the FAA



4.14.2 Controlling execution of programs by the FAA

Setting the FAA operation enable bit in the system control register to 1 enables the operation of the FAA. Once enabled, the FAA starts the execution of programs in response to either of the following events.

- The program execution bit (START) in the processor control register is set to 1.
- An interrupt request is accepted (refer to **4.10 Interrupt Controller (GRNINTC)**).

The FAA stops the execution of programs in response to any of the following events.

- The STOP instruction is executed.
- The START bit is cleared to 0.
- The undefined instruction is executed.
- The RETI instruction being executed after an interrupt request was generated to restart the program while it had been halted.

After program execution is stopped, clear the FAA operation enable bit to 0.
When program execution is stopped, the START bit is cleared to 0. In addition, when program execution is stopped by execution of the STOP instruction or when the undefined instruction is executed, the FAA outputs the interrupt request signal to the CPU. The FAA does not output the interrupt request signal to the CPU if the CPU clears the START bit to 0 or program execution is stopped by the RETI instruction.
If the undefined instruction is executed, the contents of the registers or data memory are not guaranteed.

4.14.3 Interrupt request output to the CPU

The FAA outputs an interrupt request in response to any of the following events.

- The STOP instruction is executed.: FAA end interrupt (INTFAAE)
- The undefined instruction is executed.: FAA illegal instruction interrupt (INTFAATRAP)
- A compare-match due to the free-running counter of the reference timing controller (GRNTIMEC): FAA timing compare-match interrupts n (INTTIMEC0, INTTIMEC1, INTTIMEC2)

For the default priority levels and vector table addresses of the interrupts, refer to **Table 29 - 1 Interrupt Source List**.

4.15 Instruction Set

The FAA supports the following instruction sets.

4.15.1 Transfer instructions, arithmetic operation instructions, compare instruction, branch instructions, I/O instructions, and control instructions

Table 4 - 17 Instruction Set 1 (1/2)

Classification	Instruction	Code Size	Number of Execution Cycles
Transfer instructions	MOV A0, M0	1	1
	MOV A0, M1	1	1
	MOV A0, R0	1	1
	MOV A0, L0	1	1
	MOV A0, L1	1	1
	MOV #XX, DP0	2	1
	MOV #XX, DP1	2	1
	MOV #XX, RP0	2	1
	MOV #XX, SP0	2	1
	MOV (DP0+), A0	1	1
	MOV (DP0–), A0	1	1
	MOV (DP1+), M0	1	1
	MOV (DP1+), M1	1	1
	MOV (DP1+), R0	1	1
	MOV (DP1+), L0	1	1
	MOV (DP1+), L1	1	1
	MOV A0, (RP0+)	1	1
	MOV A0, (RP0–)	1	1
Arithmetic operation instructions	MUL	1	1 Note
	ADD	1	1 Note
	SUB	1	1 Note
	MUL_ADD	1	1 Note
	MUL_SUB	1	1 Note
	LIMIT	1	1 Note
Compare instruction	CMP (DP0+)	1	1
Branch instructions	JMP #XX	2	1
	JMP OVER, #XX	2	1
	JMP UNDER, #XX	2	1
	JMP ZERO, #XX	2	1
	JMP NOT_ZERO, #XX	2	1
I/O instructions	OUT A0, (XX)	2	1
	IN (XX), A0	2	1

Table 4 - 17 Instruction Set 1 (2/2)

Classification	Instruction	Code Size	Number of Execution Cycles
Control instructions	NOP	1	1
	STOP	1	1

Note When executing instructions consecutively, the number of instruction cycles for the second and later instructions is 2.

4.15.2 Logic operation instructions, stack manipulation instructions, subroutine-related instructions, and interrupt-related instructions

Table 4 - 18 Instruction Set 2

Classification	Instruction	Code Size	Number of Execution Cycles
Logic operation instructions	OR A0, R0	1	1
	AND A0, R0	1	1
	XOR A0, R0	1	1
	NOT A0	1	1
	ABS A0	1	1
	ABS_S A0	1	1
	SFT_RL	1	1
	SFT_RA	1	1
	SFT_LL	1	1
	SFT_LA	1	1
Stack manipulation instructions and subroutine-related instructions	PUSH A0	1	1
	PUSH M0	1	1
	PUSH M1	1	1
	PUSH R0	1	1
	PUSH L0	1	1
	PUSH L1	1	1
	PUSH DP0	1	1
	PUSH DP1	1	1
	PUSH RP0	1	1
	POP A0	1	1
	POP M0	1	1
	POP M1	1	1
	POP R0	1	1
	POP L0	1	1
	POP L1	1	1
	POP DP0	1	1
	POP DP1	1	1
	POP RP0	1	1
	JSR #XX	2	1
	RET	1	2
Interrupt-related instructions	CLI	1	1
	STI	1	1
	RETI	1	2

4.15.3 Extended transfer instructions and desaturating arithmetic operation instructions

Table 4 - 19 Instruction Set 3

Classification	Instruction	Code Size	Number of Execution Cycles
Extended transfer instructions	MOV (XX, DP0), A0	2	1
	MOV (XX, DP1), M0	2	1
	MOV (XX, DP1), M1	2	1
	MOV (XX, DP1), R0	2	1
	MOV (XX, DP1), L0	2	1
	MOV (XX, DP1), L1	2	1
	MOV A0, (XX, RP0)	2	1
	MOV (#XX), A0	2	1
	MOV (#XX), M0	2	1
	MOV (#XX), M1	2	1
	MOV (#XX), R0	2	1
	MOV SP0, RP0	1	1
	MOV RP0, SP0	1	1
Desaturating arithmetic operation instructions	MUL_R	1	1 Note
	ADD_R	1	1 Note
	SUB_R	1	1 Note
	MUL_ADD_R	1	1 Note
	MUL_SUB_R	1	1 Note

Note When executing instructions consecutively, the number of instruction cycles for the second and later instructions is 2.

4.15.4 Executing two instructions simultaneously

The FAA accelerates the operation by executing two instructions simultaneously.
However, when the following conditions are applicable, instructions are not executed simultaneously but executed one by one.

1. Instruction with memory access
2. Operation instruction
3. JMP instruction, stack manipulation instruction, subroutine-related instruction, I/O instruction, STOP instruction

The following shows whether simultaneous execution is possible or not.

		2nd Instruction				
		MOV RG	MOV MR	MOV MW	CAL	JMP
1st Instruction	MOV RG	✓	✓	✓	×	×
	MOV MR	✓	×	×	×	×
	MOV MW	✓	×	×	×	×
	CAL	×	×	×	×	×
	JMP	×	×	×	×	×

✓: Simultaneous execution possible ×: Simultaneous execution not possible
MOV RG: Inter-register transfer instruction
MOV MR: Memory-reading instruction
MOV MW: Data memory-writing instruction
CAL: Operation instruction
JMP: JMP instruction, stack manipulation instruction, subroutine-related instruction, I/O instruction, STOP instruction

Classification of each instruction is shown below.

MOV RG	MOV MR	MOV MW	CAL	JMP
MOV A0, M0	MOV (DP0+), A0	MOV A0, (RP0+)	MUL	JMP #XX
MOV A0, M1	MOV (DP1+), M0	MOV A0, (RP0–)	ADD	JMP OVER, #XX
MOV A0, R0	MOV (DP1+), M1	MOV A0, (XX, RP0)	SUB	JMP UNDER, #XX
MOV A0, L0	MOV (DP1+), R0		MUL_ADD	JMP ZERO, #XX
MOV A0, L1	MOV (DP1+), L0		MUL_SUB	JMP NOT_ZERO, #XX
MOV #XX, DP0	MOV (DP1+), L1		LIMIT	POP A0
MOV #XX, DP1	MOV (DP0–), A0		MUL_R	POP M0
MOV #XX, RP0	MOV (XX, DP0), A0		ADD_R	POP M1
MOV #XX, SP0	MOV (XX, DP1), M0		SUB_R	POP R0
CLI	MOV (XX, DP1), M1		MUL_ADD_R	POP L0
STI	MOV (XX, DP1), R0		MUL_SUB_R	POP L1
NOP	MOV (XX, DP1), L0		CMP (DP0+)	POP DP0
	MOV (XX, DP1), L1		OR A0, R0	POP DP1
	MOV (#XX), A0		AND A0, R0	POP RP0
	MOV (#XX), M0		XOR A0, R0	PUSH A0
	MOV (#XX), M1		NOT A0	PUSH M0
	MOV (#XX), R0		ABS A0	PUSH M1
			SFT_RL	PUSH R0
			SFT_RA	PUSH L0
			SFT_LL	PUSH L1
			SFT_LA	PUSH DP0
			ABS_S A0	PUSH DP1
				PUSH RP0
				JSR #XX
				RET
				RETI
				MOV SP0, RP0
				MOV RP0, SP0
				OUT A0, (XX)
				IN (XX), A0
				STOP

4.16 Explanation on Instructions

4.16.1 Transfer instructions

4.16.1.1 MOV A0, M0

Assembler Format	Operation	Code	Number of Execution Cycles
MOV A0, M0	A0 → M0	01H	1

This instruction transfers the contents of register A0 (the accumulator register) to register M0 (the multiplier register).

4.16.1.2 MOV A0, M1

Assembler Format	Operation	Code	Number of Execution Cycles
MOV A0, M1	A0 → M1	02H	1

This instruction transfers the contents of register A0 (the accumulator register) to register M1 (the shift count register).

4.16.1.3 MOV A0, R0

Assembler Format	Operation	Code	Number of Execution Cycles
MOV A0, R0	A0 → R0	03H	1

This instruction transfers the contents of register A0 (the accumulator register) to register R0 (the addend register).

4.16.1.4 MOV A0, L0

Assembler Format	Operation	Code	Number of Execution Cycles
MOV A0, L0	A0 → L0	05H	1

This instruction transfers the contents of register A0 (the accumulator register) to register L0 (the upper limit register).

4.16.1.5 MOV A0, L1

Assembler Format	Operation	Code	Number of Execution Cycles
MOV A0, L1	A0 → L1	06H	1

This instruction transfers the contents of register A0 (the accumulator register) to register L1 (the lower limit register).

4.16.1.6 MOV #XX, DP0

Assembler Format	Operation	Code	Number of Execution Cycles
MOV #XX, DP0	#XX → DP0	80H, XXH (b1 and b0 in the first byte correspond to b11 and b10 in #XX, and the second byte to b9 to b2)	1

This instruction sets the immediate value XX in register DP0 (the address pointer for the accumulator).
Because the data memory is accessed in 32-bit units, the lower two bits in DP0 are fixed to 0.

4.16.1.7 MOV #XX, DP1

Assembler Format	Operation	Code	Number of Execution Cycles
MOV #XX, DP1	#XX → DP1	84H, XXH (b1 and b0 in the first byte correspond to b11 and b10 in #XX, and the second byte to b9 to b2)	1

This instruction sets the immediate value XX in register DP1 (the address pointer for operation parameters).
Because the data memory is accessed in 32-bit units, the lower two bits in DP1 are fixed to 0.

4.16.1.8 MOV #XX, RP0

Assembler Format	Operation	Code	Number of Execution Cycles
MOV #XX, RP0	#XX → RP0	88H, XXH (b1 and b0 in the first byte correspond to b11 and b10 in #XX, and the second byte to b9 to b2)	1

This instruction sets the immediate value XX in register RP0 (the address pointer for storing operation results).
Because the data memory is accessed in 32-bit units, the lower two bits in RP0 are fixed to 0.

4.16.1.9 MOV #XX, SP0

Assembler Format	Operation	Code	Number of Execution Cycles
MOV #XX, SP0	#XX → SP0	90H + mm, nn mm: b11 and b10 in immediate value XX nn: b9 to b2 in immediate value XX	1

This instruction sets the immediate value XX in register SP0 (the stack pointer).
Because the data memory is accessed in 32-bit units, the lower two bits in SP0 are fixed to 0.

4.16.1.10 MOV (DP0+), A0

Assembler Format	Operation	Code	Number of Execution Cycles
MOV (DP0+), A0	Data Memory (DP0) → A0 DP0 + 4 → DP0	07H	1

This instruction transfers the contents of the data memory at the address indicated by register DP0 (the address pointer for the accumulator) to register A0 (the accumulator register). After this, this instruction increments the value in register DP0.

4.16.1.11 MOV (DP0–), A0

Assembler Format	Operation	Code	Number of Execution Cycles
MOV (DP0–), A0	Data Memory (DP0) → A0 DP0 – 4 → DP0	21H	1

This instruction transfers the contents of the data memory at the address indicated by register DP0 (the address pointer for the accumulator) to register A0 (the accumulator register). After this, this instruction decrements the value in register DP0.

4.16.1.12 MOV (DP1+), M0

Assembler Format	Operation	Code	Number of Execution Cycles
MOV (DP1+), M0	Data Memory (DP1) → M0 DP1 + 4 → DP1	08H	1

This instruction transfers the contents of the data memory at the address indicated by register DP1 (the address pointer for operation parameters) to register M0 (the multiplier register). After this, this instruction increments the value in register DP1.

4.16.1.13 MOV (DP1+), M1

Assembler Format	Operation	Code	Number of Execution Cycles
MOV (DP1+), M1	Data Memory (DP1) → M1 DP1 + 4 → DP1	09H	1

This instruction transfers the contents of the data memory at the address indicated by register DP1 (the address pointer for operation parameters) to register M1 (the shift count register). After this, this instruction increments the value in register DP1.

4.16.1.14 MOV (DP1+), R0

Assembler Format	Operation	Code	Number of Execution Cycles
MOV (DP1+), R0	Data Memory (DP1) → R0 DP1 + 4 → DP1	0AH	1

This instruction transfers the contents of the data memory at the address indicated by register DP1 (the address pointer for operation parameters) to register R0 (the addend register). After this, this instruction increments the value in register DP1.

4.16.1.15 MOV (DP1+), L0

Assembler Format	Operation	Code	Number of Execution Cycles
MOV (DP1+), L0	Data Memory (DP1) → L0 DP1 + 4 → DP1	0BH	1

This instruction transfers the contents of the data memory at the address indicated by register DP1 (the address pointer for operation parameters) to register L0 (the upper limit register). After this, this instruction increments the value in register DP1.

4.16.1.16 MOV (DP1+), L1

Assembler Format	Operation	Code	Number of Execution Cycles
MOV (DP1+), L1	Data Memory (DP1) → L1 DP1 + 4 → DP1	0CH	1

This instruction transfers the contents of the data memory at the address indicated by register DP1 (the address pointer for operation parameters) to register L1 (the lower limit register). After this, this instruction increments the value in register DP1.

4.16.1.17 MOV A0, (RP0+)

Assembler Format	Operation	Code	Number of Execution Cycles
MOV A0, (RP0+)	A0 → Data Memory (RP0) RP0 + 4 → RP0	0DH	1

This instruction transfers the contents of register A0 (the accumulator register) to the data memory at the address indicated by register RP0 (the address pointer for storing operation results). After this, this instruction increments the value in register RP0.

4.16.1.18 MOV A0, (RP0–)

Assembler Format	Operation	Code	Number of Execution Cycles
MOV A0, (RP0–)	A0 → Data Memory (RP0) RP0 – 4 → RP0	22H	1

This instruction transfers the contents of register A0 (the accumulator register) to the data memory at the address indicated by register RP0 (the address pointer for storing operation results). After this, this instruction decrements the value in register RP0.

4.16.2 Arithmetic operation instructions

4.16.2.1 MUL

Assembler Format	Operation	Code	Number of Execution Cycles
MUL	A0 × M0 → Shift (M1) → A0	0EH	1

This instruction performs signed 32-bit multiplication of the values in registers A0 (the accumulator register) and M0 (the multiplier register), right-shifts the resulting value by the number of bits indicated by register M1 (the shift count register), and stores the result in register A0. Register M1 should be set to a value in the range from 0 to 64. Multiplication and shifting of the value are handled with a precision of 64 bits after internally converting the values into absolute values, and the absolute value of the result is returned to a signed 32-bit value after shifting. If the value is out of the range of signed 32-bit values, it is truncated to the upper or lower limit of signed 32-bit values.
If the result of calculation is 0, the ZERO flag is set. Otherwise, it is cleared.

4.16.2.2 ADD

Assembler Format	Operation	Code	Number of Execution Cycles
ADD	A0 + R0 → A0	0FH	1

This instruction performs signed 32-bit addition of the values in registers A0 (the accumulator register) and R0 (the addend register), and stores the result in register A0. If the result of addition overflows, the result is truncated to the upper or lower limit of signed 32-bit values.
If the result of calculation is 0, the ZERO flag is set. Otherwise, it is cleared.

4.16.2.3 SUB

Assembler Format	Operation	Code	Number of Execution Cycles
SUB	A0 – R0 → A0	11H	1

This instruction subtracts the signed 32-bit value in R0 (the addend register) from that in register A0 (the accumulator register), and stores the result in register A0. If the result of subtraction overflows, the result is truncated to the upper or lower limit of signed 32-bit values.
If the result of calculation is 0, the ZERO flag is set. Otherwise, it is cleared.

4.16.2.4 MUL_ADD

Assembler Format	Operation	Code	Number of Execution Cycles
MUL_ADD	A0 × M0 → Shift (M1) → A0 + R0 → A0	12H	1

This instruction performs signed 32-bit multiplication of the values in registers A0 (the accumulator register) and M0 (the multiplier register), and right-shifts the resulting value by the number of bits indicated by register M1 (the shift count register). It then adds the result to the value of R0 (the addend register) as signed 32-bit values, and stores the result in register A0. Multiplication and shifting of the value are handled with a precision of 64 bits after internally converting the values into absolute values, and the absolute value of the result is returned to a signed 32-bit value after shifting. If the value is out of the range of signed 32-bit values, it is truncated to the upper or lower limit of signed 32-bit values.
If the addition following the multiplication produces an overflow, the result is truncated to the upper or lower limit of signed 32-bit values. Register M1 should be set to a value in the range from 0 to 64.
If the result of calculation is 0, the ZERO flag is set. Otherwise, it is cleared.

4.16.2.5 MUL_SUB

Assembler Format	Operation	Code	Number of Execution Cycles
MUL_SUB	A0 × M0 → Shift (M1) → A0 – R0 → A0	14H	1

This instruction performs signed 32-bit multiplication of the values in registers A0 (the accumulator register) and M0 (the multiplier register), and right-shifts the resulting value by the number of bits indicated by register M1 (the shift count register). It then subtracts the signed 32-bit value in R0 (the addend register) from the signed 32-bit result, and stores the value in register A0. Multiplication and shifting of the value are handled with a precision of 64 bits after internally converting the values into absolute values, and the absolute value of the result is returned to a signed 32-bit value after shifting. If the value is out of the range of signed 32-bit values, it is truncated to the upper or lower limit of signed 32-bit values.
If the subtraction following the multiplication produces an overflow, the result is truncated to the upper or lower limit of signed 32-bit values. Register M1 should be set to a value in the range from 0 to 64.
If the result of calculation is 0, the ZERO flag is set. Otherwise, it is cleared.

4.16.2.6 LIMIT

Assembler Format	Operation	Code	Number of Execution Cycles
LIMIT	LIMIT (L0, L1) → A0	1CH	1

This instruction compares a value in register A0 (the accumulator register) with values in registers L0 (the upper limit register) and L1 (the lower limit register). If a value in A0 is greater than a value in L0, this instruction sets the value in A0 to a value in L0. If a value in A0 is smaller than a value in L1, this instruction changes the value in A0 to a value in L1. Then, this instruction stores the result in A0.
If a value in A0 is 0, the ZERO flag is set. Otherwise, the flag is cleared. If a value in A0 is greater than a value in L0, the OVER flag is set. If the value in A0 is equal to or smaller than a value in L0, the flag is cleared. If a value in A0 is smaller than a value in L1, the UNDER flag is set. If the value in A0 is equal to or greater than a value in L1, the flag is cleared.

4.16.3 Compare instruction

4.16.3.1 CMP (DP0+)

Assembler Format	Operation	Code	Number of Execution Cycles
CMP (DP0+)	Compare Data Memory (DP0) DP0 + 4 → DP0	1DH	1

This instruction compares the contents of the data memory at the address indicated by register DP0 (the address pointer for the accumulator) with a value in register A0 (the accumulator register), and sets or clears the ZERO flag. After this, this instruction increments the value in register DP0.

4.16.4 Branch instructions

4.16.4.1 JMP #XX

Assembler Format	Operation	Code	Number of Execution Cycles
JMP #XX	#XX → PG0	BXH, XXH	1

This instruction causes execution to jump to the address indicated by the immediate value XX.

4.16.4.2 JMP OVER, #XX

Assembler Format	Operation	Code	Number of Execution Cycles
JMP OVER, #XX	If OVER = “1” #XX → PG0	CXH, XXH	1

If the OVER flag is 1, this instruction causes execution to jump to the address indicated by the immediate value XX.

4.16.4.3 JMP UNDER, #XX

Assembler Format	Operation	Code	Number of Execution Cycles
JMP UNDER, #XX	If UNDER = “1” #XX → PG0	DXH, XXH	1

If the UNDER flag is 1, this instruction causes execution to jump to the address indicated by the immediate value XX.

4.16.4.4 JMP ZERO, #XX

Assembler Format	Operation	Code	Number of Execution Cycles
JMP ZERO, #XX	If ZERO = “1” #XX → PG0	EXH, XXH	1

If the ZERO flag is 1, this instruction causes execution to jump to the address indicated by the immediate value XX.

4.16.4.5 JMP NOT_ZERO, #XX

Assembler Format	Operation	Code	Number of Execution Cycles
JMP NOT_ZERO, #XX	If ZERO = “0” #XX → PG0	FXH, XXH	1

If the ZERO flag is 0, this instruction causes execution to jump to the address indicated by the immediate value XX.

4.16.5 I/O instructions

4.16.5.1 OUT A0, (XX)

Assembler Format	Operation	Code	Number of Execution Cycles
OUT A0, (XX)	A0 → I/O Port (XX)	1EH, XXH	1

This instruction outputs a value in register A0 (the accumulator register) to the external port at the address XX.

4.16.5.2 IN (XX), A0

Assembler Format	Operation	Code	Number of Execution Cycles
IN (XX), A0	I/O Port (XX) → A0	1FH, XXH	1

This instruction inputs data from the external port at the address XX, and stores the data in register A0 (the accumulator register).

4.16.6 Control instructions

4.16.6.1 NOP

Assembler Format	Operation	Code	Number of Execution Cycles
NOP	No operation	00H	1

This instruction only increments the program counter, causing the next instruction to be executed.

4.16.6.2 STOP

Assembler Format	Operation	Code	Number of Execution Cycles
STOP	STOP	20H	1

This instruction stops the FAA, and at the same time outputs an interrupt to the main CPU.

4.16.7 Logic operation instructions

4.16.7.1 OR A0, R0

Assembler Format	Operation	Code	Number of Execution Cycles
OR A0, R0	A0 or R0 → A0	23H	1

This instruction performs logical OR between a value in register A0 (the accumulator register) and a value in register R0, and stores the result in register A0. If a value in register A0 is 0, the ZERO flag is set. Otherwise, the flag is cleared.

4.16.7.2 AND A0, R0

Assembler Format	Operation	Code	Number of Execution Cycles
AND A0, R0	A0 and R0 → A0	24H	1

This instruction performs logical AND between a value in register A0 (the accumulator register) and a value in register R0, and stores the result in register A0. If a value in register A0 is 0, the ZERO flag is set. Otherwise, the flag is cleared.

4.16.7.3 XOR A0, R0

Assembler Format	Operation	Code	Number of Execution Cycles
XOR A0, R0	A0 xor R0 → A0	25H	1

This instruction performs logical exclusive-OR between a value in register A0 (the accumulator register) and a value in register R0, and stores the result in register A0. If a value in register A0 is 0, the ZERO flag is set. Otherwise, the flag is cleared.

4.16.7.4 NOT A0

Assembler Format	Operation	Code	Number of Execution Cycles
NOT A0	not A0 → A0	26H	1

This instruction inverts all bits in register A0 (the accumulator register), and stores the result in register A0. If a value in register A0 is 0, the ZERO flag is set. Otherwise, the flag is cleared.

4.16.7.5 ABS A0

Assembler Format	Operation	Code	Number of Execution Cycles
ABS A0	abs A0 → A0	27H	1

This instruction obtains the absolute value of the value in register A0 (the accumulator register), and stores the result in register A0. If a value in register A0 is 0, the ZERO flag is set. Otherwise, the flag is cleared.

4.16.7.6 ABS_S A0

Assembler Format	Operation	Code	Number of Execution Cycles
ABS_S A0	abs A0 → A0	4FH	1

This instruction obtains the absolute value of the value in register A0 (the accumulator register), and stores the result in register A0.
If a value in register A0 is 80000000H, 7FFFFFFFH is stored as the result in register A0. If a value in register A0 is 0, the ZERO flag is set. Otherwise, the flag is cleared.

4.16.7.7 SFT_RL

Assembler Format	Operation	Code	Number of Execution Cycles
SFT_RL	Arithmetic right shift A0 → A0	28H	1

This instruction right-shifts a value in register A0 (the accumulator register), and stores the result in register A0. The value of the shifted out bit is stored in the UNDER flag. If the result is 0, the ZERO flag is set. Otherwise, the flag is cleared.

4.16.7.8 SFT_RA

Assembler Format	Operation	Code	Number of Execution Cycles
SFT_RA	Arithmetic right shift A0 → A0	29H	1

This instruction arithmetically right-shifts a value in register A0 (the accumulator register), and stores the result in register A0. The value of the shifted out bit is stored in the UNDER flag. If the result is 0, the ZERO flag is set. Otherwise, the flag is cleared.

4.16.7.9 SFT_LL

Assembler Format	Operation	Code	Number of Execution Cycles
SFT_LL	Arithmetic left shift A0 → A0	2AH	1

This instruction left-shifts a value in register A0 (the accumulator register), and stores the result in register A0. The value of the shifted out bit is stored in the OVER flag. If the result is 0, the ZERO flag is set. Otherwise, the flag is cleared.

4.16.7.10 SFT_LA

Assembler Format	Operation	Code	Number of Execution Cycles
SFT_LA	Arithmetic left shift A0 → A0	2BH	1

This instruction arithmetically left-shifts a value in register A0 (the accumulator register), and stores the result in register A0. If the result of left-shifting is outside the range of 32-bit maximum or minimum values, the value is limited to the 32-bit maximum or minimum value, and the OVER flag is set. If the result of left-shifting is within the range of signed 32-bit values, the OVER flag is cleared. If the result is 0, the ZERO flag is set. Otherwise, the flag is cleared.

4.16.8 Stack manipulation and subroutine-related instructions

4.16.8.1 PUSH A0/M0/M1/R0/L0/L1/DP0/DP1/RP0

Assembler Format	Operation	Code	Number of Execution Cycles
PUSH A0	SP0 – 4 → SP0 A0 → Data Memory (SP0)	2CH	1
PUSH M0	SP0 – 4 → SP0 M0 → Data Memory (SP0)	2DH	1
PUSH M1	SP0 – 4 → SP0 M1 → Data Memory (SP0)	2EH	1
PUSH R0	SP0 – 4 → SP0 R0 → Data Memory (SP0)	2FH	1
PUSH L0	SP0 – 4 → SP0 L0 → Data Memory (SP0)	30H	1
PUSH L1	SP0 – 4 → SP0 L1 → Data Memory (SP0)	31H	1
PUSH DP0	SP0 – 4 → SP0 DP0 → Data Memory (SP0)	32H	1
PUSH DP1	SP0 – 4 → SP0 DP1 → Data Memory (SP0)	33H	1
PUSH RP0	SP0 – 4 → SP0 RP0 → Data Memory (SP0)	34H	1

These instructions decrement the value in register SP0 (the stack pointer) by four, and then transfer the contents of a register to the data memory at the address indicated by register SP0.

4.16.8.2 POP A0/M0/M1/R0/L0/L1/DP0/DP1/RP0

Assembler Format	Operation	Code	Number of Execution Cycles
POP A0	Data Memory (SP0) → A0 SP0 + 4 → SP0	35H	1
POP M0	Data Memory (SP0) → M0 SP0 + 4 → SP0	36H	1
POP M1	Data Memory (SP0) → M1 SP0 + 4 → SP0	37H	1
POP R0	Data Memory (SP0) → R0 SP0 + 4 → SP0	38H	1
POP L0	Data Memory (SP0) → L0 SP0 + 4 → SP0	39H	1
POP L1	Data Memory (SP0) → L1 SP0 + 4 → SP0	3AH	1
POP DP0	Data Memory (SP0) → DP0 SP0 + 4 → SP0	3BH	1
POP DP1	Data Memory (SP0) → DP1 SP0 + 4 → SP0	3CH	1
POP RP0	Data Memory (SP0) → RP0 SP0 + 4 → SP0	3DH	1

These instructions transfer the contents of the data memory at the address indicated by SP0 (the stack pointer) to a register, and then increment the value in register SP0 by four.

4.16.8.3 JSR, #XX

Assembler Format	Operation	Code	Number of Execution Cycles
JSR, #XX	SP0 – 4 → SP0 PG0 + 2 → Data Memory (SP0) #XX → PG0	AXH, XXH	1

This instruction decrements the value in register SP0 (the stack pointer) by four, and then saves the value of “current program counter + 2” to the data memory at the address indicated by register SP0. After this, this instruction causes execution to jump to the address indicated by the immediate value XX.

4.16.8.4 RET

Assembler Format	Operation	Code	Number of Execution Cycles
RET	Data Memory (SP0) → PG0 SP0 + 4 → SP0	3EH	2

This instruction sets the contents of the data memory at the address indicated by SP0 (the stack pointer) in the program counter (PG0), and then increments the value in register SP0 by four.

4.16.9 Interrupt-related instructions

4.16.9.1 CLI

Assembler Format	Operation	Code	Number of Execution Cycles
CLI	“0” → I flag	51H	1

This instruction clears the I flag (interrupt flag) to 0.

4.16.9.2 STI

Assembler Format	Operation	Code	Number of Execution Cycles
STI	“1” → I flag	52H	1

This instruction sets the I flag (interrupt flag) to 1.

4.16.9.3 RETI

Assembler Format	Operation	Code	Number of Execution Cycles
RETI	Data Memory (SP0) → PG0, I/Z/U/O flag, START bit SP0 + 4 → SP0	54H	2

This instruction sets the contents of the data memory at the address indicated by SP0 (the stack pointer) in the program counter (PG0), the flag, and the START bit, and increments the value in register SP0 by four.

4.16.10 Extended transfer instructions

4.16.10.1 MOV (XX, DP0), A0

Assembler Format	Operation	Code	Number of Execution Cycles
MOV (XX, DP0), A0	Data Memory (XX × 4 + DP0) → A0	41H, XXH	1

This instruction transfers the contents of the location in data memory whose address is the result of adding the value “XX × 4” to the value of register DP0 (address pointer for the accumulator) to register A0 (the accumulator register). The range of the displacement, XX, is from –128 to +127.

4.16.10.2 MOV (XX, DP1), M0

Assembler Format	Operation	Code	Number of Execution Cycles
MOV (XX, DP1), M0	Data Memory (XX × 4 + DP1) → M0	42H, XXH	1

This instruction transfers the contents of the location in data memory whose address is the result of adding the value "XX × 4" to the value of register DP1 (address pointer for operation parameters) to register M0 (the multiplier register). The range of the displacement, XX, is from –128 to +127.

4.16.10.3 MOV (XX, DP1), M1

Assembler Format	Operation	Code	Number of Execution Cycles
MOV (XX, DP1), M1	Data Memory (XX × 4 + DP1) → M1	43H, XXH	1

This instruction transfers the contents of the location in data memory whose address is the result of adding the value "XX × 4" to the value of register DP1 (address pointer for operation parameters) to register M1 (the shift count register). The range of the displacement, XX, is from –128 to +127.

4.16.10.4 MOV (XX, DP1), R0

Assembler Format	Operation	Code	Number of Execution Cycles
MOV (XX, DP1), R0	Data Memory (XX × 4 + DP1) → R0	44H, XXH	1

This instruction transfers the contents of the location in data memory whose address is the result of adding the value "XX × 4" to the value of register DP1 (address pointer for operation parameters) to register R0 (the addend register). The range of the displacement, XX, is from –128 to +127.

4.16.10.5 MOV (XX, DP1), L0

Assembler Format	Operation	Code	Number of Execution Cycles
MOV (XX, DP1), L0	Data Memory (XX × 4 + DP1) → L0	45H, XXH	1

This instruction transfers the contents of the location in data memory whose address is the result of adding the value "XX × 4" to the value of register DP1 (address pointer for operation parameters) to register L0 (the upper limit register). The range of the displacement, XX, is from –128 to +127.

4.16.10.6 MOV (XX, DP1), L1

Assembler Format	Operation	Code	Number of Execution Cycles
MOV (XX, DP1), L1	Data Memory (XX × 4 + DP1) → L1	46H, XXH	1

This instruction transfers the contents of the location in data memory whose address is the result of adding the value "XX × 4" to the value of register DP1 (address pointer for operation parameters) to register L1 (the lower limit register). The range of the displacement, XX, is from –128 to +127.

4.16.10.7 MOV A0, (XX, RP0)

Assembler Format	Operation	Code	Number of Execution Cycles
MOV A0, (XX, RP0)	A0 → Data Memory (XX × 4 + RP0)	47H, XXH	1

This instruction transfers the contents of register A0 (the accumulator register) to the data memory whose address is the result of adding the value "XX × 4" to the value of register RP0 (the address pointer for storing operation results). The range of the displacement, XX, is from –128 to +127.

4.16.10.8 MOV (#XX), A0

Assembler Format	Operation	Code	Number of Execution Cycles
MOV (#XX), A0	Data Memory (XX) → A0	8CH + mm, nn mm: b11 and b10 in immediate value XX nn: b9 to b2 in immediate value XX	1

This instruction transfers the contents of the data memory at the address XX to register A0 (the accumulator register).

4.16.10.9 MOV (#XX), M0

Assembler Format	Operation	Code	Number of Execution Cycles
MOV (#XX), M0	Data Memory (XX) → M0	94H + mm, nn mm: b11 and b10 in immediate value XX nn: b9 to b2 in immediate value XX	1

This instruction transfers the contents of the data memory at the address XX to register M0 (the multiplier register).

4.16.10.10 MOV (#XX), M1

Assembler Format	Operation	Code	Number of Execution Cycles
MOV (#XX), M1	Data Memory (XX) → M1	98H + mm, nn mm: b11 and b10 in immediate value XX nn: b9 to b2 in immediate value XX	1

This instruction transfers the contents of the data memory at the address XX to register M1 (the shift count register).

4.16.10.11 MOV (#XX), R0

Assembler Format	Operation	Code	Number of Execution Cycles
MOV (#XX), R0	Data Memory (XX) → R0	9CH + mm, nn mm: b11 and b10 in immediate value XX nn: b9 to b2 in immediate value XX	1

This instruction transfers the contents of the data memory at the address XX to register R0 (the addend register).

4.16.10.12 MOV SP0, RP0

Assembler Format	Operation	Code	Number of Execution Cycles
MOV SP0, RP0	SP0 → RP0	48H	1

This instruction transfers the contents of register SP0 (stack pointer) to register RP0 (the address pointer for storing operation results).

4.16.10.13 MOV RP0, SP0

Assembler Format	Operation	Code	Number of Execution Cycles
MOV RP0, SP0	RP0 → SP0	49H	1

This instruction transfers the contents of register RP0 (the address pointer for storing operation results) to register SP0 (the stack pointer).

4.16.11 Desaturating arithmetic operation instructions

Results of desaturating arithmetic operation instructions are not rounded even if they are outside the signed 32-bit range.
All operations are performed with unsigned 32-bit values. Results of operations that are outside the unsigned 32-bit range are truncated, and only those bits within the unsigned 32-bit range are handled as the result.

4.16.11.1 MUL_R

Assembler Format	Operation	Code	Number of Execution Cycles
MUL_R	A0 × M0 → Shift (M1) → A0	4AH	1

This instruction performs unsigned 32-bit multiplication of the values in registers A0 (the accumulator register) and M0 (the multiplier register), right-shifts the resulting value by the number of bits indicated by register M1 (the shift count register), and stores the lower 32 bits in register A0.
Register M1 should be set to a value in the range from 0 to 64.
The OVER flag or UNDER flag is updated depending on the operation result. If the operation result exceeds the unsigned 32-bit upper limit, the OVER flag is set. Because this is an unsigned operation, the UNDER flag is always cleared in this instruction. If the OVER flag is not set and the operation result is 0, the ZERO flag is set. If the result is not 0, the ZERO flag is cleared.

4.16.11.2 ADD_R

Assembler Format	Operation	Code	Number of Execution Cycles
ADD_R	A0 + R0 → A0	4BH	1

This instruction performs unsigned 32-bit addition of the values in registers A0 (the accumulator register) and R0 (the addend register), and stores the result in register A0.
The OVER flag or UNDER flag is updated depending on the operation result. If the operation result exceeds the unsigned 32-bit upper limit, the OVER flag is set. Because this is an unsigned operation, the UNDER flag is always cleared in this instruction. If the OVER flag is not set and the operation result is 0, the ZERO flag is set. If the result is not 0, the ZERO flag is cleared.

4.16.11.3 SUB_R

Assembler Format	Operation	Code	Number of Execution Cycles
SUB_R	A0 – R0 → A0	4CH	1

This instruction subtracts the unsigned 32-bit value in R0 (the addend register) from that of register A0 (the accumulator register), and stores the result in register A0.
The OVER flag or UNDER flag is updated depending on the operation result. If the operation result exceeds the unsigned 32-bit lower limit, the UNDER flag is set. Because this is an unsigned operation, the OVER flag is always cleared in this instruction. If the UNDER flag is not set and the operation result is 0, the ZERO flag is set. If the result is not 0, the ZERO flag is cleared.

4.16.11.4 MUL_ADD_R

Assembler Format	Operation	Code	Number of Execution Cycles
MUL_ADD_R	A0 × M0 → Shift (M1) → A0 + R0 → A0	4DH	1

This instruction performs unsigned 32-bit multiplication of the values in registers A0 (the accumulator register) and M0 (the multiplier register), and right-shifts the resulting value by the number of bits indicated by register M1 (the shift count register). It then adds the result to the value of R0 (the addend register) as unsigned 32-bit values, and stores the result in register A0.
Register M1 should be set to a value in the range from 0 to 64.
The OVER flag or UNDER flag is updated depending on the operation result. If the operation result exceeds the unsigned 32-bit upper limit, the OVER flag is set. The UNDER flag is always cleared in this instruction. If neither the OVER flag nor the UNDER flag are set and the operation result is 0, the ZERO flag is set. If the result is not 0, the ZERO flag is cleared.

4.16.11.5 MUL_SUB_R

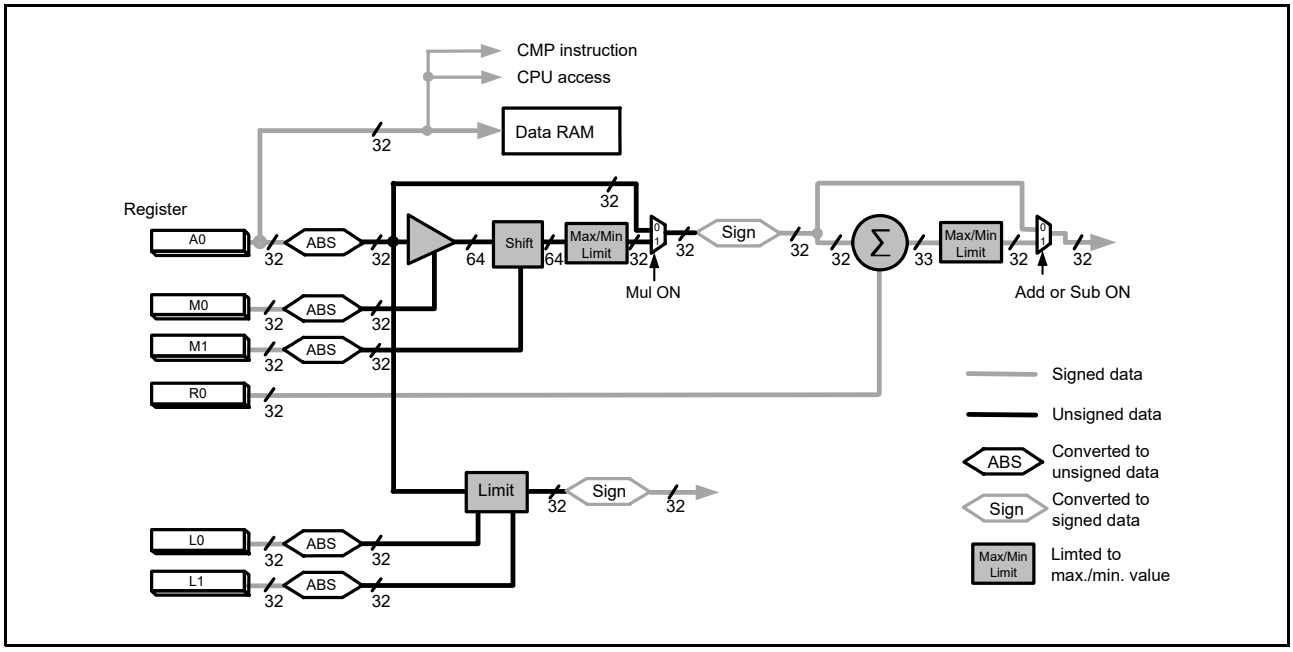
Assembler Format	Operation	Code	Number of Execution Cycles
MUL_SUB_R	A0 × M0 → Shift (M1) → A0 – R0 → A0	4EH	1

This instruction performs unsigned 32-bit multiplication of the values in registers A0 (the accumulator register) and M0 (the multiplier register), and right-shifts the resulting value by the number of bits indicated by register M1 (the shift count register). It then subtracts the unsigned 32-bit value in R0 (the addend register) from the unsigned 32-bit result, and stores the value in register A0.
Register M1 should be set to a value in the range from 0 to 64.
If the operation result exceeds the unsigned 32-bit upper limit, the OVER flag is set. If the result exceeds the lower limit, the UNDER flag is set. If neither the OVER flag nor the UNDER flag are set and the operation result is 0, the ZERO flag is set. If the result is not 0, the ZERO flag is cleared.

4.16.12 Operation bit count

Figure 4 - 38 shows the bit count in the FAA operation.
The FAA handles signed 32-bit multiplication, addition, and subtraction. In the case of multiplication, however, conversion to unsigned data proceeds before the 32-bit × 32-bit multiplication, and the resulting 64 bits are shifted. If the result exceeds the 32-bit maximum or minimum value, the result is limited to the 32-bit maximum or minimum value.
In case of addition and subtraction, if the result of a signed 32-bit operation exceeds the 32-bit maximum or minimum value, the result is limited to the 32-bit maximum or minimum value.

Figure 4 - 38 Operation Bit Count



4.17 Points for Caution

4.17.1 Writing of the program and data for the FAA to the code flash memory

Link the program and data for the FAA to the RL78 CPU program by using the optimizing linkage editor included in the C compiler package for the RL78 family (CC-RL), and write them to the code flash memory.

<R>

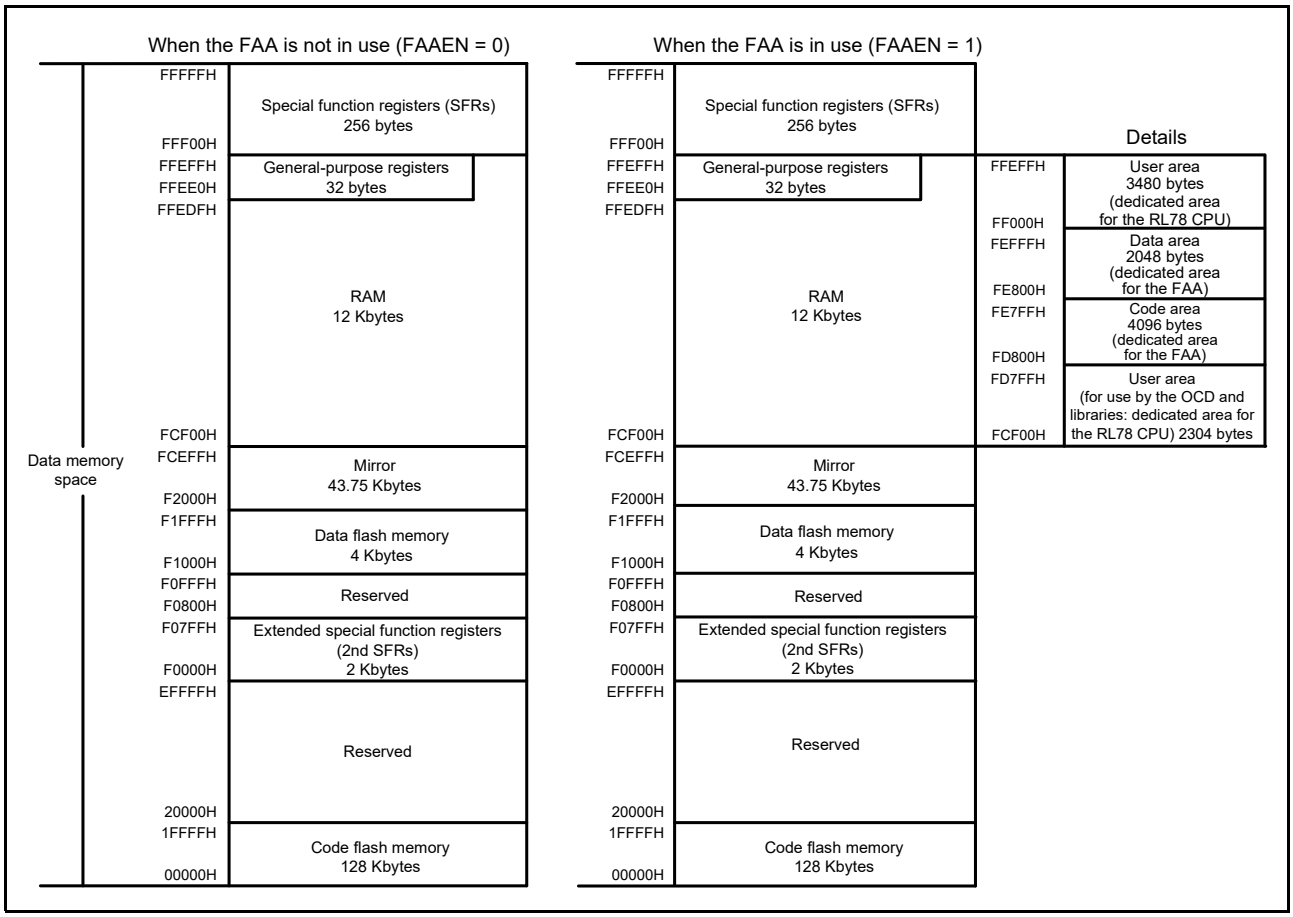
4.17.2 Transfer of the program and data for the FAA, which are stored in the code flash memory, respectively to the instruction code memory and data memory

The program and data for the FAA are stored in the code flash memory. To be used by the FAA, they must be transferred to the FAA's instruction code memory and data memory. To transfer the program and data, in each case start by changing the value of the window register (WIND) to the value corresponding to the resource of the FAA to be accessed. After that, write the program and then the data for the FAA that have been stored in the code flash memory to the target resource allocation area (F0600H to F06FFH). For details on the window register (WIND), see **4.8 Window Register (WIND)**.

When supply of an input clock to the FAA is stopped (FAAEN = 0), the CPU can access the instruction code memory and data memory. Note that, after having transferred the program for the FAA, the expected data will not be returned in response to reading of the instruction code memory by the CPU with CPU addresses, because the allocation of data in the instruction code memory is not the same as that in the code flash memory. Once the FAA program is in place, do not use the CPU for directly writing to the instruction code memory, as doing so might destroy the program for the FAA. Reading the data memory with CPU addresses does return the expected data, because the allocation of data is the same in both the code flash memory and data memory. Writing to the data memory with CPU addresses is also possible.

Figure 4 - 39 shows the memory map of the instruction code memory and data memory.

Figure 4 - 39 Memory Map of the Instruction Code Memory and Data Memory



Caution **To use the FAA with the FAA parity error reset enabled (RPECTL2.FAARPEREN = 1), be sure to initialize all locations in both the data and code areas and clear the parity error flag of the FAA (RPECTL2.FAARPEF = 0) beforehand. After that, enable the FAA parity error reset (RPECTL2.FAARPEREN = 1). The FAA parity error reset is disabled (RPECTL2.FAARPEREN = 0) following a reset. For details, see 35.3.4 RAM parity error detection.**

Section 5 Data Shared Memory (SHDMEM)

5.1 Overview of the Data Shared Memory (SHDMEM)

The SHDMEM is a memory that can be shared by the CPU and the FAA while the FAA is executing a program.

5.2 Peripheral Enable Register 2 (PER2)

The PER2 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. If the SHDMEM is to be used, be sure to set bit 6 (MEMEN) of this register to 1. The PER2 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H. The PER2 register is not accessible from the FAA.

Figure 5 - 1 Format of Peripheral Enable Register 2 (PER2)

Address: F00FCH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER2	FAAEN	MEMEN	TKBEN	TRGEN	TRD0EN	PWMOPEN	TRXEN	TRJ0EN

MEMEN	Control of supply of an input clock to the SHDMEM
0	Stops supply of an input clock. • The SFRs used by the SHDMEM can be read but not written.
1	Enables supply of an input clock. • The SFRs used by the SHDMEM can be read and written.

5.3 Peripheral Reset Control Register 2 (PRR2)

The PRR2 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. To place the SHDMEM in the reset state, be sure to set bit 6 (MEMRES) of this register to 1. The PRR2 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H. The PRR2 register is not accessible from the FAA.

Figure 5 - 2 Format of Peripheral Reset Control Register 2 (PRR2)

Address: F00FDH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PRR2	FAARES	MEMRES	TKBRES	TRGRES	TRD0RES	PWMOPRES	TRXRES	TRJ0RES

MEMRES	Control resetting of the SHDMEM
0	The SHDMEM is released from the reset state.
1	The SHDMEM is in the reset state.

5.4 List of Registers of the Data Shared Memory (SHDMEM)

Table 5 - 1 shows a list of registers of the data shared memory (SHDMEM).

Table 5 - 1 List of Registers of Data Shared Memory (SHDMEM)

Extended Special Function Register (2nd SFR)						FAA					
CPU Address	Register Name		Symbol	R/W (CPU)	Bits	FAA Address	Register Name		Symbol	R/W (FAA)	Bits
F0440H	Data shared memory (SHDMEM)	CDWD register file 0L	CWDW0L	R/W	16	E0H	Data shared memory (SHDMEM)	CDWD register file 0	CWDW0	R/W	32
F0442H		CDWD register file 0H	CWDW0H	R/W	16						
F0444H		CDWD register file 1L	CWDW1L	R/W	16	E1H		CDWD register file 1	CWDW1	R/W	32
F0446H		CDWD register file 1H	CWDW1H	R/W	16						
F0448H		CDWD register file 2L	CWDW2L	R/W	16	E2H		CDWD register file 2	CWDW2	R/W	32
F044AH		CDWD register file 2H	CWDW2H	R/W	16						
F044CH		CDWD register file 3L	CWDW3L	R/W	16	E3H		CDWD register file 3	CWDW3	R/W	32
F044EH		CDWD register file 3H	CWDW3H	R/W	16						
F0450H		CDWD register file 4L	CWDW4L	R/W	16	E4H		CDWD register file 4	CWDW4	R/W	32
F0452H		CDWD register file 4H	CWDW4H	R/W	16						
F0454H		CDWD register file 5L	CWDW5L	R/W	16	E5H		CDWD register file 5	CWDW5	R/W	32
F0456H		CDWD register file 5H	CWDW5H	R/W	16						
F0458H		CDWD register file 6L	CWDW6L	R/W	16	E6H		CDWD register file 6	CWDW6	R/W	32
F045AH		CDWD register file 6H	CWDW6H	R/W	16						
F045CH		CDWD register file 7L	CWDW7L	R/W	16	E7H		CDWD register file 7	CWDW7	R/W	32
F045EH		CDWD register file 7H	CWDW7H	R/W	16						

5.5 CDWD Register Files n, nH, and nL (CWDWn, CWDWnH, and CWDWnL) (n = 0 to 7)

The FAA can share the data written by the CPU through CWDW register files n (CWDWn). The CPU can share the data written by the FAA through CWDW register files nH and nL (CWDWnH, CWDWnL). If the CPU and FAA attempt writing to a register at the same time, writing by the CPU takes priority.

Figure 5 - 3 Format of CWDW Register Files n, nH, and nL (CWDWn, CWDWnH, and CWDWnL) (n = 0 to 7)

FAA address: E0H (CWDW0) to E7H (CWDW7)

Address: F0442H (CWDW0H), F0446H (CWDW1H), F044AH (CWDW2H), F044EH (CWDW3H), F0452H (CWDW4H), F0456H (CWDW5H), F045AH (CWDW6H), F045EH (CWDW7H), F0440H (CWDW0L), F0444H (CWDW1L), F0448H (CWDW2L), F044CH (CWDW3L), F0450H (CWDW4L), F0454H (CWDW5L), F0458H (CWDW6L), F045CH (CWDW7L)

After reset: 0000 0000H (CWDWn), 0000H (CWDWnH, CWDWnL)

R/W: R/W

Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CWDWn																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0	Value of the data shared memory (CWDWn)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CWDWnH																

Bits 15 to 0	Value of the data shared memory (CWDWnH)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CWDWnL																

Bits 15 to 0	Value of the data shared memory (CWDWnL)

Section 6 Divider

6.1 Functions of the Divider

The divider is an ALU dedicated to the flexible application accelerator (FAA). The following shows the function of the divider.

- 32 bits ÷ 32 bits = Unsigned 32-bit quotient, 32-bit remainder

6.2 Configuration of the Divider

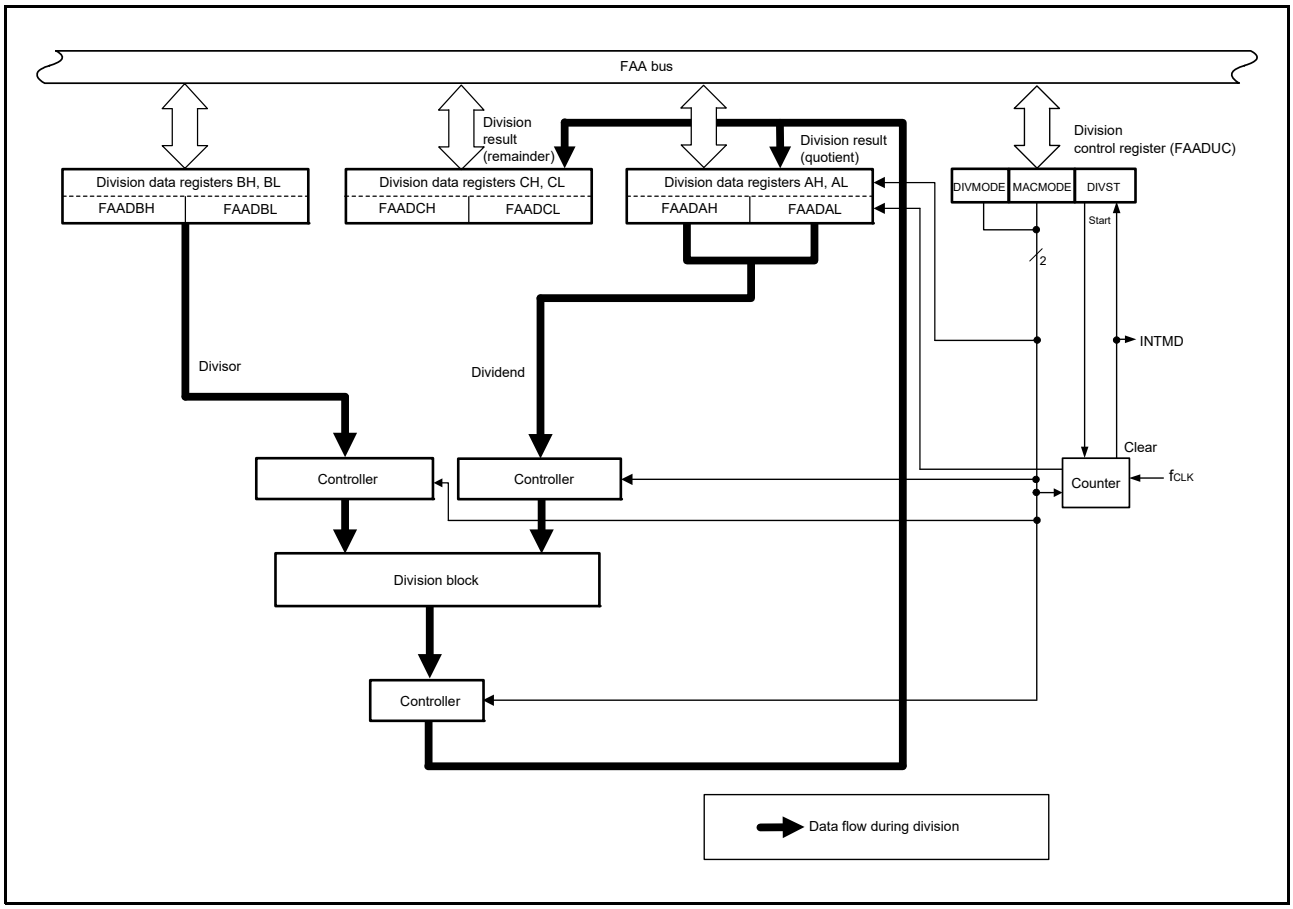
The divider consists of the following hardware.

Table 6 - 1 Configuration of the Divider

Item	Configuration
Registers	• Division data registers AH, AL (FAADAH, FAADAL) • Division data registers BH, BL (FAADBH, FAADBL) • Division data registers CH, CL (FAADCH, FAADCL)
Control registers	• Peripheral enable register 2 (PER2) • Peripheral reset control register 2 (PRR2) • Division control register (FAADUC)

Figure 6 - 1 shows a block diagram of the divider and **Table 6 - 2** lists the registers of the divider.

Figure 6 - 1 Block Diagram of Divider



Remark fCLK: CPU/peripheral hardware clock frequency

Table 6 - 2 List of Registers of the Divider

Extended Special Function Register (2nd SFR)					FAA				
CPU Address	Register Name	Symbol	R/W (CPU)	Bits	FAA Address	Register Name	Symbol	R/W (FAA)	Bits
F04B4H	Division data register CL	FAADCL	R	16	AEH	Division data register CL	FAADCL	R/W	16
F04B6H	Division data register CH	FAADCH	R	16	AFH	Division data register CH	FAADCH	R/W	16
F04B8H	Division data register AL	FAADAL	R	16	AAH	Division data register AL	FAADAL	R/W	16
F04BAH	Division data register AH	FAADAH	R	16	ABH	Division data register AH	FAADAH	R/W	16
F04BCH	Division data register BL	FAADBL	R	16	ACH	Division data register BL	FAADBL	R/W	16
F04BEH	Division data register BH	FAADBH	R	16	ADH	Division data register BH	FAADBH	R/W	16

6.2.1 Division data registers AH, AL (FAADAH, FAADAL)

The FAADAH and FAADAL registers are used to set dividend values for division and hold quotients after division. The processor writes the values of the 16 lower-order bits (bits 15 to 0) and the 16 higher-order bits (bits 31 to 16) in the accumulator register (A0) to the FAADAL and FAADAH registers, respectively, by using an OUT instruction. The processor also loads the values read from the FAADAL and FAADAH registers to the 16 lower-order bits (bits 15 to 0) and the 16 higher-order bits (bits 31 to 16) in the accumulator register (A0), respectively, by using an IN instruction. Note that simultaneous access to the FAADAH and FAADAL registers is not possible. The value of each register following a reset is 0000H.

Figure 6 - 2 Format of Division Data Registers AH, AL (FAADAH, FAADAL)

FAA address: ABH (FAADAH), AAH (FAADAL)
After reset: 0000H (FAADAH, FAADAL)
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAADAH																

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAADAL																

Caution 1. Do not rewrite the FAADAH and FAADAL registers values during division operation processing (when the division control register (FAADUC) value is 81H or C1H). The operation will be executed if this is the case, but the value of the result will be undefined.

Caution 2. The FAADAH and FAADAL registers values read during division operation processing (when the FAADUC register value is 81H or C1H) will not be guaranteed.

Table 6 - 3 shows the functions of the FAADAH and FAADAL registers during operation execution.

Table 6 - 3 Functions of FAADAH and FAADAL Registers During Operation Execution

Setting	Result of Operation
FAADAH: 16 higher-order bits of the unsigned dividend FAADAL: 16 lower-order bits of the unsigned dividend	FAADAH: 16 higher-order bits of the unsigned quotient FAADAL: 16 lower-order bits of the unsigned quotient

6.2.2 Division data registers BH, BL (FAADBH, FAADBL)

The FAADBH and FAADBL registers are used to set divisor values for division. The processor writes the values of the 16 lower-order bits (bits 15 to 0) and the 16 higher-order bits (bits 31 to 16) in the accumulator register (A0) to the FAADBL and FAADBH registers, respectively, by using an OUT instruction. The processor also loads the values read from the FAADBL and FAADBH registers to the 16 lower-order bits (bits 15 to 0) and the 16 higher-order bits (bits 31 to 16) in the accumulator register (A0), respectively, by using an IN instruction. Note that simultaneous access to the FAADBH and FAADBL registers is not possible. The value of each register following a reset is 0000H.

Figure 6 - 3 Format of Division Data Registers BH, BL (FAADBH, FAADBL)

FAA address: ADH (FAADBH), ACH (FAADBL)
After reset: 0000H (FAADBH, FAADBL)
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAADBH																

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAADBL																

Caution 1. Do not rewrite the FAADBH and FAADBL registers values during division operation processing (when the division control register (FAADUC) value is 81H or C1H). The value of the result of the operation will be undefined.

Caution 2. Do not set the FAADBH and FAADBL registers to 0000H. If this setting is made, the value of the result of the operation will be undefined.

Table 6 - 4 shows the functions of the FAADBH and FAADBL registers during operation execution.

Table 6 - 4 Functions of FAADBH and FAADBL Registers During Operation Execution

Setting	Result of Operation
FAADBH: 16 higher-order bits of the unsigned divisor FAADBL: 16 lower-order bits of the unsigned divisor	—

6.2.3 Division data registers CH, CL (FAADCH, FAADCL)

The FAADCH and FAADCL registers hold remainders. The processor writes the values of the 16 lower-order bits (bits 15 to 0) and the 16 higher-order bits (bits 31 to 16) in the accumulator register (A0) to the FAADCL and FAADCH registers, respectively, by using an OUT instruction. The processor also loads the values read from the FAADCL and FAADCH registers to the 16 lower-order bits (bits 15 to 0) and the 16 higher-order bits (bits 31 to 16) in the accumulator register (A0), respectively, by using an IN instruction. Note that simultaneous access to the FAADCH and FAADCL registers is not possible. The value of each register following a reset is 0000H.

Figure 6 - 4 Format of Division Data Registers CH, CL (FAADCH, FAADCL)

FAA address: AFH (FAADCH), AEH (FAADCL)
After reset: 0000H (FAADCH, FAADCL)
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAADCH																

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAADCL																

Caution The FAADCH and FAADCL registers values read during division operation processing (when the division control register (FAADUC) value is 81H or C1H) will not be guaranteed.

Table 6 - 5 shows the functions of the FAADCH and FAADCL registers during operation execution.

Table 6 - 5 Functions of FAADCH and FAADCL Registers During Operation Execution

Setting	Result of Operation
—	FAADCH: 16 higher-order bits of the unsigned remainder FAADCL: 16 lower-order bits of the unsigned remainder

The register configuration when division is executed is as follows.

<Dividend>		<Divisor>		<Quotient>		<Remainder>
{FAADAH, FAADAL}	÷	{FAADBH, FAADBL}	=	{FAADAH, FAADAL}	…	{FAADCH, FAADCL}

6.3 Registers Controlling the Divider

The divider is controlled by using the following registers.

- Peripheral enable register 2 (PER2)
- Peripheral reset control register 2 (PRR2)
- Division control register (FAADUC)

Table 6 - 6 List of Registers Controlling the Divider

Extended Special Function Register (2nd SFR)					FAA				
CPU Address	Register Name	Symbol	R/W (CPU)	Bits	FAA Address	Register Name	Symbol	R/W (FAA)	Bits
F04B3H	Division control register	FAADUC	R	8	B0H	Division control register	FAADUC	R/W	8

6.3.1 Peripheral enable register 2 (PER2)

The PER2 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. To use the divider, be sure to set bit 7 (FAAEN) of this register to 1. The PER2 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H. The PER2 register is not accessible from the FAA.

Figure 6 - 5 Format of Peripheral Enable Register 2 (PER2)

Address: F00FCH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER2	FAAEN	MEMEN	TKBEN	TRGEN	TRD0EN	PWMOPEN	TRXEN	TRJ0EN

FAAEN	Control of supply of an input clock to the divider
0	Stops supply of an input clock. • The SFRs used by the divider cannot be written. • Writing to the FAADUC.DIVMODE and FAADUC.MACMODE bits is possible even while the setting of the FAAEN bit is 0.
1	Enables supply of an input clock. • The SFRs used by the divider can be read and written.

6.3.2 Peripheral reset control register 2 (PRR2)

The PRR2 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. To place the divider in the reset state, be sure to set bit 7 (FAARES) of this register to 1. The PRR2 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H. The PRR2 register is not accessible from the FAA.

Figure 6 - 6 Format of Peripheral Reset Control Register 2 (PRR2)

Address: F00FDH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PRR2	FAARES	MEMRES	TKBRES	TRGRES	TRD0RES	PWMOPRES	TRXRES	TRJ0RES

FAARES	Control resetting of the divider
0	The divider is released from the reset state.
1	The divider is in the reset state.

6.3.3 Division control register (FAADUC)

The FAADUC register is an 8-bit register that controls the operation of the divider. The FAADUC register can be set by a 1-bit or 8-bit memory manipulation instruction. The processor writes the value of the 8 lower-order bits (bits 7 to 0) in the accumulator register (A0) to the FAADUC register by using an OUT instruction. The processor also loads the value read from the FAADUC register to the 8 lower-order bits (bits 7 to 0) in the accumulator register (A0) by using an IN instruction. The value of this register following a reset is 00H.

Figure 6 - 7 Format of Division Control Register (FAADUC)

FAA address: B0H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	5	4	3	2	1	<0>
FAADUC	DIVMODE	MACMODE	0	0	0	0	0	DIVST

DIVMODE	MACMODE	Interrupt generation selection
1	0	Generation of a division completion interrupt (INTMD)
1	1	Not generation of a division completion interrupt (INTMD)
Other than above		Setting prohibited

DIVST[Note]	Division operation start/stop
0	Division operation processing complete
1	Starts division operation/division operation processing in progress

Note Division operation is started by setting the DIVST bit to 1. The DIVST bit is automatically cleared to 0 when the operation is completed.

Caution 1. Be sure to set bits 5 to 1 to 0.
Caution 2. Do not rewrite the DIVMODE bit during operation processing (while the DIVST bit is 1). If this bit is rewritten, the value of the result of the operation will be undefined.
Caution 3. The DIVST bit cannot be cleared to 0 by using software during division operation processing (while the DIVST bit is 1).

6.4 Operation of the Divider

The following describes the operation of the divider.

- Initial setting
  - <1> Set the division control register (FAADUC) to 80H.
  - <2> Set the 16 higher-order bits of the dividend in division data register AH (FAADAH).
  - <3> Set the 16 lower-order bits of the dividend in division data register AL (FAADAL).
  - <4> Set the 16 higher-order bits of the divisor in division data register BH (FAADBH).
  - <5> Set the 16 lower-order bits of the divisor in division data register BL (FAADBL).
  - <6> Set bit 0 (DIVST) of the FAADUC register to 1.

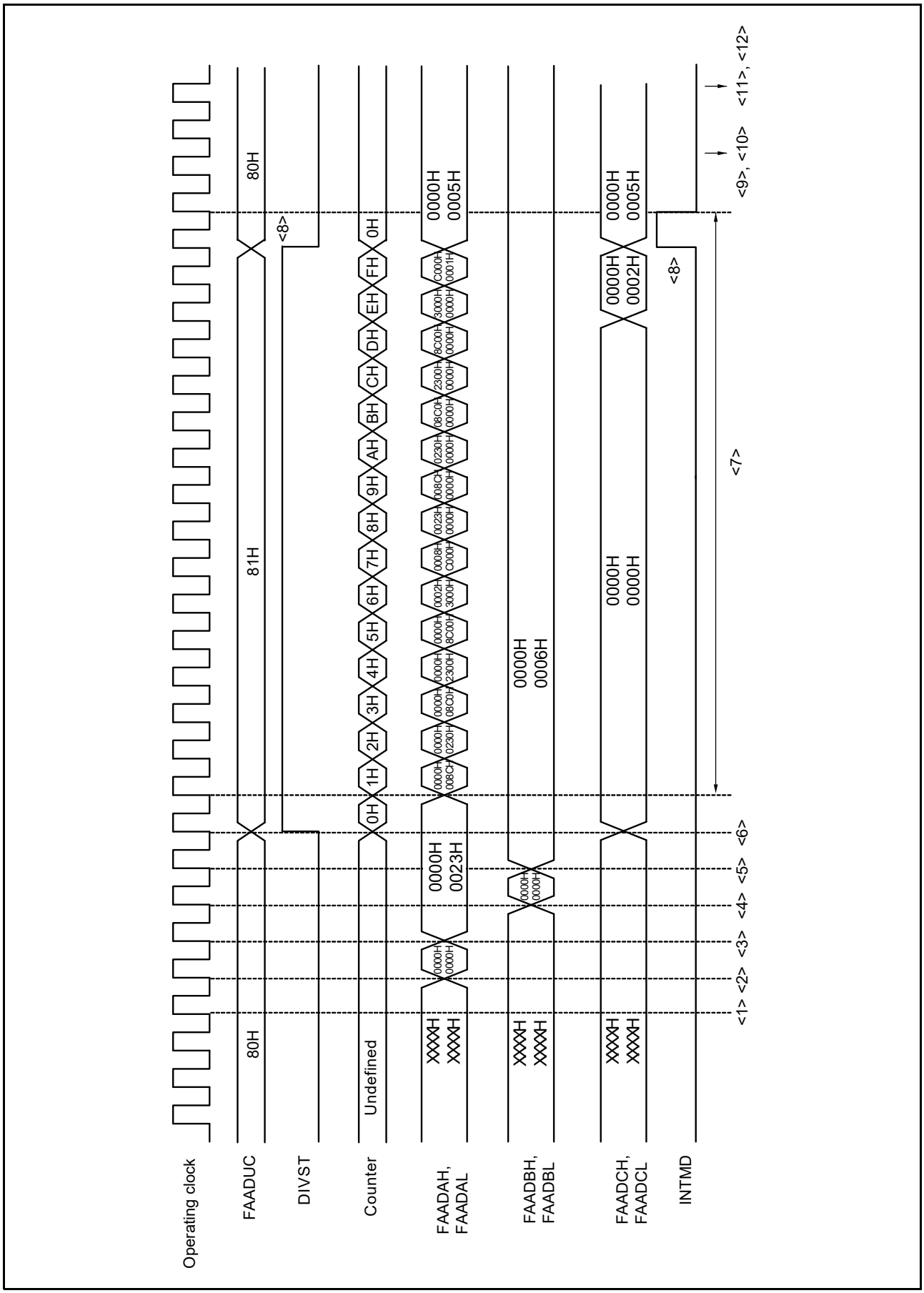
  There is no preferred order for the execution of steps <2> to <5>.
- During operation processing
  - <7> Use either of the following types of processing to detect completion of the operation.
    - A wait of at least 16 clock cycles: The operation will have been completed when 16 clock cycles have elapsed.
    - A check whether the DIVST bit has been cleared

  The values read from the FAADBL, FAADBH, FAADCL, and FAADCH registers during operation processing are not guaranteed.
- Operation completion
  - <8> The DIVST bit being cleared to 0 indicates completion of the operation. At this time, an interrupt request signal (INTMD) is generated if the operation was performed with MACMODE = 0.
  - <9> Read the 16 lower-order bits of the quotient from the FAADAL register.
  - <10> Read the 16 higher-order bits of the quotient from the FAADAH register.
  - <11> Read the 16 lower-order bits of the remainder from division data register CL (FAADCL).
  - <12> Read the 16 higher-order bits of the remainder from division data register CH (FAADCH).

  There is no preferred order for the execution of steps <9> to <12>.
- Next operation
  - <13> Start with the initial settings of each step to change the operation mode. When the same operation mode is used sequentially, settings <1> to <5> can be omitted.

Remark Steps <1> to <12> correspond to <1> to <12> in **Figure 6 - 8 Timing Diagram of Division Operation (Example: 35 ÷ 6 = 5, Remainder 5)**.

Figure 6 - 8 Timing Diagram of Division Operation (Example: 35 ÷ 6 = 5, Remainder 5)



Section 7 Port Functions

7.1 Port Functions

The RL78/G24 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 **Section 2 Pin Functions**.

7.2 Port Configuration

Ports include the following hardware.

Table 7 - 1 Port Configuration (1/2)

Item	Configuration
Control registers	• Port mode registers xx (PMxx) (xx = 0 to 7, 12, 14) • Port registers xx (Pxx) (xx = 0 to 7, 12 to14) • Pull-up resistor option registers xx (PUxx) (xx = 0, 1, 3 to 7, 12, 14) • Port input mode registers xx (PIMxx) (xx = 0, 1, 3, 5, 7) • Port output mode registers xx (POMxx) (xx = 0, 1, 3, 5, 7) • Port digital input disable registers xx (PDIDISxx) (xx = 0, 1, 3, 5, 7, 13) • Port mode control A registers xx (PMCAxx) (xx = 0 to 2, 12, 14) • Peripheral I/O redirection registers x (PIORx) (x = 0 to 3) • Global digital input disable register (GDIDIS) • Output current control enable register (CCDE) • Output current select registers x (CCSx) (x = 0, 4 to 7) • Port mode select register (PMS)
Port	• 20-pin products Total: 16 (CMOS I/O: 15 (N-ch open drain I/O [withstand voltage of V_{DD}]: 7, controlled current drive port: 2), CMOS input: 1) • 24-pin products Total: 20 (CMOS I/O: 19 (N-ch open drain I/O [withstand voltage of V_{DD}]: 8, controlled current drive port: 4), CMOS input: 1) • 25-pin products Total: 21 (CMOS I/O: 19 (N-ch open drain I/O [withstand voltage of V_{DD}]: 8, controlled current drive port: 4), CMOS input: 1, CMOS output: 1) • 30-pin products Total: 26 (CMOS I/O: 23 (N-ch open drain I/O [withstand voltage of V_{DD}]: 11, controlled current drive port: 6), CMOS input: 1, N-ch open drain I/O [withstand voltage of 6 V]: 2) • 32-pin products Total: 28 (CMOS I/O: 25 (N-ch open drain I/O [withstand voltage of V_{DD}]: 11, controlled current drive port: 7), CMOS input: 1, N-ch open drain I/O [withstand voltage of 6 V]: 2) • 40-pin products Total: 36 (CMOS I/O: 31 (N-ch open drain I/O [withstand voltage of V_{DD}]: 14, controlled current drive port: 7), CMOS input: 3, N-ch open drain I/O [withstand voltage of 6 V]: 2) • 44-pin products Total: 40 (CMOS I/O: 35 (N-ch open drain I/O [withstand voltage of V_{DD}]: 14, controlled current drive port: 8), CMOS input: 3, N-ch open drain I/O [withstand voltage of 6 V]: 2) • 48-pin products Total: 44 (CMOS I/O: 38 (N-ch open drain I/O [withstand voltage of V_{DD}]: 15, controlled current drive port: 8), CMOS input: 3, CMOS output: 1, N-ch open drain I/O [withstand voltage of 6 V]: 2) • 52-pin products Total: 48 (CMOS I/O: 42 (N-ch open drain I/O [withstand voltage of V_{DD}]: 17, controlled current drive port: 8), CMOS input: 3, CMOS output: 1, N-ch open drain I/O [withstand voltage of 6 V]: 2) • 64-pin products Total: 58 (CMOS I/O: 52 (N-ch open drain I/O [withstand voltage of EV_{DD}]: 19, controlled current drive port: 8), CMOS input: 3, CMOS output: 1, N-ch open drain I/O [withstand voltage of 6 V]: 2)

Table 7 - 1 Port Configuration (2/2)

Item	Configuration
Pull-up resistor	• 20-pin products Total: 11 • 24-pin products Total: 15 • 25-pin products Total: 15 • 30-pin products Total: 19 • 32-pin products Total: 21 • 40-pin products Total: 24 • 44-pin products Total: 27 • 48-pin products Total: 30 • 52-pin products Total: 34 • 64-pin products Total: 44

7.2.1 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 by port mode register 0 (PM0). When the P00 to P06 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, P03 and P04 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units by port input mode register 0 (PIM0).
Output from the P00 and P02 to P04 pins can be specified as N-ch open-drain output (withstand voltage of V_{DD}[Note 1]/ withstand voltage of EV_{DD}[Note 2]) in 1-bit units by port output mode register 0 (POM0).
This port can also be used for timer I/O, A/D converter analog input, and serial interface data I/O and clock I/O. Use the registers shown in **7.3 Registers to Control the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 7 - 4 Correspondence between Register Settings and Port Pin State**.
The settings of the port pins following a reset are as follows.

- P00 and P01 pins of the 20- to 64-pin products: Analog input mode
- P04 to P06 pins of the 36- to 64-pin products: Input mode
- P02 and P03 pins of the 52- to 64-pin products: Analog input mode

Note 1. This applies to the 20- to 52-pin products.
Note 2. This applies to the 64-pin products.

7.2.2 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 by 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, P11, and P14 to P17 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units by port input mode register 1 (PIM1).
Output from the P10 to P15 and P17 pins can be specified as N-ch open-drain output (withstand voltage of V_{DD}[Note 1]/ withstand voltage of EV_{DD}[Note 2]) in 1-bit units by port output mode register 1 (POM1).
Output from the P10, P11, P16, and P17 pins can be specified as controlled current drive port pins in 1-bit units by the output current control enable register (CCDE).
This port can also be used for serial interface data I/O and clock I/O, UART data transmission and reception for external device connection when programming flash memory, clock/buzzer output, timer I/O, external interrupt request input, comparator reference voltage input, and comparator output. Use the registers shown in **7.3 Registers to Control the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 7 - 4 Correspondence between Register Settings and Port Pin State**.
P10 to P17 are set to analog input mode following a reset.

Note 1. This applies to the 20- to 52-pin products.
Note 2. This applies to the 64-pin products.

7.2.3 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 by port mode register 2 (PM2).
This port can also be used for A/D converter analog input, A/D converter reference voltage input (+ side and - side), D/A converter output, and comparator reference voltage input. Use the registers shown in **7.3 Registers to Control the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 7 - 4 Correspondence between Register Settings and Port Pin State**.
P20 to P27 are set to analog input mode following a reset.

7.2.4 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 by 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 pin can be specified through a normal input buffer or a TTL input buffer in 1-bit units by port input mode register 3 (PIM3).
Output from the P30 pin can be specified as N-ch open-drain output (withstand voltage of V_{DD}**Note 1**/withstand voltage of EV_{DD}**Note 2**) in 1-bit units by port output mode register 3 (POM3).
This port can also be used for external interrupt request input, realtime clock correction clock output, clock/buzzer output, timer I/O, serial interface data I/O and clock I/O, and comparator output. Use the registers shown in **7.3 Registers to Control the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 7 - 4 Correspondence between Register Settings and Port Pin State**.
P30 and P31 are set to input mode following a reset.

Note 1. This applies to the 30- to 52-pin products.
Note 2. This applies to the 64-pin products.

7.2.5 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 by port mode register 4 (PM4). When the P40 to P43 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, timer I/O, serial interface data I/O and clock I/O, and external interrupt request input. Use the registers shown in **7.3 Registers to Control the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 7 - 4 Correspondence between Register Settings and Port Pin State**.
P40 to P43 are set to input mode following a reset.

7.2.6 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 by port mode register 5 (PM5). When the P50 to P55 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 and P55 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units by port input mode register 5 (PIM5).
Output from the P50, P51, and P55 pins can be specified as N-ch open-drain output (withstand voltage of V_{DD}[Note 1]/ withstand voltage of EV_{DD}[Note 2]) in 1-bit units by port output mode register 5 (POM5).
This port can also be used for external interrupt request input, and serial interface data I/O and clock I/O. Use the registers shown in **7.3 Registers to Control the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 7 - 4 Correspondence between Register Settings and Port Pin State**.
P50 to P55 are set to input mode following a reset.

Note 1. This applies to the 30- to 52-pin products.
Note 2. This applies to the 64-pin products.

7.2.7 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 by port mode register 6 (PM6). When the P62 and P63 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 6 (PU6).
Output from the P60 and P61 pins is N-ch open-drain output (withstand voltage of 6 V). Output from the P60 to P63 pins can be specified as controlled current drive port pins in 1-bit units by the output current control enable register (CCDE).
This port can also be used for serial interface data I/O and clock I/O, and timer I/O. Use the registers shown in **7.3 Registers to Control the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 7 - 4 Correspondence between Register Settings and Port Pin State**.
P60 to P63 are set to input mode following a reset.

7.2.8 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 by port mode register 7 (PM7). When this port is 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). Input to the P73 pin can be specified through a normal input buffer or a TTL input buffer in 1-bit units by port input mode register 7 (PIM7).
Output from the P71 to P74 pins can be specified as N-ch open-drain output (withstand voltage of V_{DD}[Note 1]/withstand voltage of EV_{DD}[Note 2]) in 1-bit units by port output mode register 7 (POM7).
This port can also be used for key interrupt input, serial interface data I/O and clock I/O, and external interrupt request input. Use the registers shown in **7.3 Registers to Control the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 7 - 4 Correspondence between Register Settings and Port Pin State**.
P70 to P77 are set to input mode following a reset.

Note 1. This applies to the 32- to 52-pin products.
Note 2. This applies to the 64-pin products.

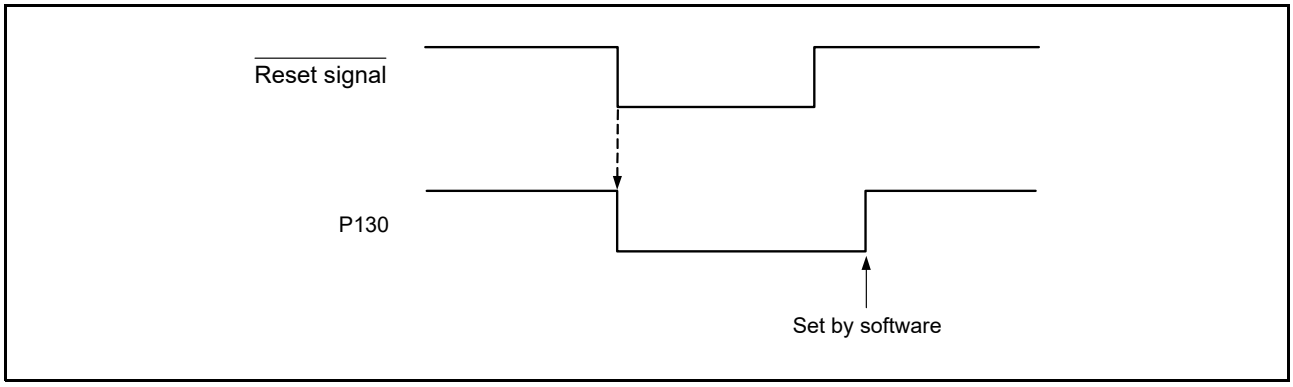
7.2.9 Port 12

P120 to P122 are 6-bit I/O ports with an output latch. Port 12 can be set to the input mode or output mode in 1-bit units by port mode register 12 (PM12). When this port is used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).
P123 and P124 are 2-bit input-only ports.
This port can also be used for A/D converter analog input, comparator analog input, connection of a resonator for the main system clock, connection of a resonator for the subsystem clock, external clock input for the main system clock, external clock input for the subsystem clock, and power supply for battery backup. Use the registers shown in **7.3 Registers to Control the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 7 - 4 Correspondence between Register Settings and Port Pin State**.
P120 is set to analog input mode and P121 to P124 are set to input mode following a reset.

7.2.10 Port 13

P130 is a 1-bit output-only port with an output latch. P137 is a 1-bit input-only port.
P130 is fixed to output mode, and P137 is fixed to input mode.
This port can also be used for external interrupt request input. Use the registers shown in **7.3 Registers to Control the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 7 - 4 Correspondence between Register Settings and Port Pin State**.

Remark When a reset signal is applied, P130 outputs a low-level signal. If P130 is set to a mode for outputting a high-level signal before a reset signal is applied, the P130 signal can be used to indicate a CPU reset.



7.2.11 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 by port mode register 14 (PM14). When the P140, P141, 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).
This port can also be used for clock/buzzer output, external interrupt request input, A/D converter analog input, serial interface data I/O and clock I/O, timer I/O, and comparator analog input. Use the registers shown in **7.3 Registers to Control the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 7 - 4 Correspondence between Register Settings and Port Pin State**.
P140 and P141 are set to input mode and P146 and P147 are set to analog input mode following a reset.

7.3 Registers to Control the Port Function

The following registers are used to control the port functions.

- Port mode registers xx (PMxx)
- Port registers xx (Pxx)
- Pull-up resistor option registers xx (PUxx)
- Port input mode registers xx (PIMxx)
- Port output mode registers xx (POMxx)
- Port digital input disable registers xx (PDIDISxx)
- Port mode control A registers xx (PMCAxx)
- Peripheral I/O redirection registers x (PIORx)
- Global digital input disable register (GDIDIS)
- Output current control enable register (CCDE)
- Output current select registers x (CCSx)
- Port mode select register (PMS)

Table 7 - 2 PMxx, Pxx, PUxx, PIMxx, POMxx, PDIDISxx, and PMCAxx Registers and the Bits Implemented in Each Product (1/3)

Port		Bit Name							64 Pins	52 Pins	48 Pins	44 Pins	40 Pins	32 Pins	30 Pins	25 Pins	24 Pins	20 Pins
		PMxx Register	Pxx Register	PUxx Register	PIMxx Register	POMxx Register	PDIDISxx Register	PMCAxx Register										
Port 0	0	PM00	P00	PU00	—	POM00	PDIDIS00	PMCA00	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	1	PM01	P01	PU01	PIM01	—	—	PMCA01	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	2	PM02	P02	PU02	—	POM02	PDIDIS02	PMCA02	✓	✓	—	—	—	—	—	—	—	—
	3	PM03	P03	PU03	PIM03	POM03	PDIDIS03	PMCA03	✓	✓	—	—	—	—	—	—	—	—
	4	PM04	P04	PU04	PIM04	POM04	PDIDIS04	—	✓	—	—	—	—	—	—	—	—	—
	5	PM05	P05	PU05	—	—	—	—	✓	—	—	—	—	—	—	—	—	—
	6	PM06	P06	PU06	—	—	—	—	✓	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 1	0	PM10	P10	PU10	PIM10	POM10	PDIDIS10	PMCA10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	1	PM11	P11	PU11	PIM11	POM11	PDIDIS11	PMCA11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	2	PM12	P12	PU12	—	POM12	PDIDIS12	PMCA12	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	3	PM13	P13	PU13	—	POM13	PDIDIS13	PMCA13	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	4	PM14	P14	PU14	PIM14	POM14	PDIDIS14	PMCA14	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	5	PM15	P15	PU15	PIM15	POM15	PDIDIS15	PMCA15	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	6	PM16	P16	PU16	PIM16	—	—	PMCA16	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
	7	PM17	P17	PU17	PIM17	POM17	PDIDIS17	PMCA17	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
Port 2	0	PM20	P20	—	—	—	—	PMCA20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	1	PM21	P21	—	—	—	—	PMCA21	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	2	PM22	P22	—	—	—	—	PMCA22	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	3	PM23	P23	—	—	—	—	PMCA23	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	4	PM24	P24	—	—	—	—	PMCA24	✓	✓	✓	✓	✓	—	—	—	—	—
	5	PM25	P25	—	—	—	—	PMCA25	✓	✓	✓	✓	✓	—	—	—	—	—
	6	PM26	P26	—	—	—	—	PMCA26	✓	✓	✓	✓	✓	—	—	—	—	—
	7	PM27	P27	—	—	—	—	PMCA27	✓	✓	✓	✓	—	—	—	—	—	—
Port 3	0	PM30	P30	PU30	PIM30	POM30	PDIDIS30	—	✓	✓	✓	✓	✓	✓	✓	—	—	—
	1	PM31	P31	PU31	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	—	—	—
	2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 4	0	PM40	P40	PU40	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	1	PM41	P41	PU41	—	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—
	2	PM42	P42	PU42	—	—	—	—	✓	—	—	—	—	—	—	—	—	—
	3	PM43	P43	PU43	—	—	—	—	✓	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 7 - 2 PMxx, Pxx, PUxx, PIMxx, POMxx, PDIDISxx, and PMCAxx Registers and the Bits Implemented in Each Product (2/3)

Port		Bit Name							64 Pins	52 Pins	48 Pins	44 Pins	40 Pins	32 Pins	30 Pins	25 Pins	24 Pins	20 Pins
		PMxx Register	Pxx Register	PUxx Register	PIMxx Register	POMxx Register	PDIDISxx Register	PMCAxx Register										
Port 5	0	PM50	P50	PU50	PIM50	POM50	PDIDIS50	—	✓	✓	✓	✓	✓	✓	✓	—	—	—
	1	PM51	P51	PU51	—	POM51	PDIDIS51	—	✓	✓	✓	✓	✓	✓	✓	—	—	—
	2	PM52	P52	PU52	—	—	—	—	✓	—	—	—	—	—	—	—	—	—
	3	PM53	P53	PU53	—	—	—	—	✓	—	—	—	—	—	—	—	—	—
	4	PM54	P54	PU54	—	—	—	—	✓	—	—	—	—	—	—	—	—	—
	5	PM55	P55	PU55	PIM55	POM55	PDIDIS55	—	✓	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 6	0	PM60	P60	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	—	—	—
	1	PM61	P61	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	—	—	—
	2	PM62	P62	PU62	—	—	—	—	✓	✓	✓	✓	✓	✓	—	—	—	—
	3	PM63	P63	PU63	—	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 7	0	PM70	P70	PU70	—	—	—	—	✓	✓	✓	✓	✓	✓	—	—	—	—
	1	PM71	P71	PU71	—	POM71	PDIDIS71	—	✓	✓	✓	✓	✓	—	—	—	—	—
	2	PM72	P72	PU72	—	POM72	PDIDIS72	—	✓	✓	✓	✓	✓	—	—	—	—	—
	3	PM73	P73	PU73	PIM73	POM73	PDIDIS73	—	✓	✓	✓	✓	✓	—	—	—	—	—
	4	PM74	P74	PU74	—	POM74	PDIDIS74	—	✓	✓	✓	—	—	—	—	—	—	—
	5	PM75	P75	PU75	—	—	—	—	✓	✓	✓	—	—	—	—	—	—	—
	6	PM76	P76	PU76	—	—	—	—	✓	✓	—	—	—	—	—	—	—	—
	7	PM77	P77	PU77	—	—	—	—	✓	✓	—	—	—	—	—	—	—	—
Port 12	0	PM120	P120	PU120	—	—	—	PMCA120	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	1	PM121	P121	PU121	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
	2	PM122	P122	PU122	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	3	—	P123	—	—	—	—	—	✓	✓	✓	✓	✓	—	—	—	—	—
	4	—	P124	—	—	—	—	—	✓	✓	✓	✓	✓	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 13	0	—	P130	—	—	—	—	—	✓	✓	✓	—	—	—	—	✓	—	—
	1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	P137	—	—	—	PDIDIS137	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 7 - 2 PMxx, Pxx, PUxx, PIMxx, POMxx, PDIDISxx, and PMCAxx Registers and the Bits Implemented in Each Product (3/3)

Port		Bit Name							64 Pins	52 Pins	48 Pins	44 Pins	40 Pins	32 Pins	30 Pins	25 Pins	24 Pins	20 Pins
		PMxx Register	Pxx Register	PUxx Register	PIMxx Register	POMxx Register	PDIDISxx Register	PMCAxx Register										
Port 14	0	PM140	P140	PU140	—	—	—	—	✓	✓	✓	—	—	—	—	—	—	—
	1	PM141	P141	PU141	—	—	—	—	✓	—	—	—	—	—	—	—	—	—
	2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	PM146	P146	PU146	—	—	—	PMCA146	✓	✓	✓	✓	—	—	—	—	—	—
	7	PM147	P147	PU147	—	—	—	PMCA147	✓	✓	✓	✓	✓	✓	✓	✓	✓	—

7.3.1 Port mode registers xx (PMxx)

The PMxx registers specify input or output mode for the ports in 1-bit units. These registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each PMxx register following a reset is FFH. To use an alternate function of a port pin, set the port mode register by referencing **7.5 Register Settings When Using Alternate Function**.

Figure 7 - 1 Format of Port Mode Registers xx (PMxx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	PM06	PM05	PM04	PM03	PM02	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	PM43	PM42	PM41	PM40	FFF24H	FFH	R/W
PM5	1	1	PM55	PM54	PM53	PM52	PM51	PM50	FFF25H	FFH	R/W
PM6	1	1	1	1	PM63	PM62	PM61	PM60	FFF26H	FFH	R/W
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
PM12	1	1	1	1	1	PM122	PM121	PM120	FFF2CH	FFH	R/W
PM14	PM147	PM146	1	1	1	1	PM141	PM140	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 that are not implemented to their initial values.

7.3.2 Port registers xx (Pxx)

The Pxx registers set the output latch values of ports. If data is read in the input mode, the pin level is read. If data 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. The value of each Pxx register following a reset is 00H.

Note If P00 to P03, P10 to P17, P20 to P27, P120, P146, or P147 is set up as an analog function port, when the port is read in the input mode, 0 is always returned instead of the pin level.

Figure 7 - 2 Format of Port Registers xx (Pxx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	P06	P05	P04	P03	P02	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	P43	P42	P41	P40	FFF04H	00H (output latch)	R/W
P5	0	0	P55	P54	P53	P52	P51	P50	FFF05H	00H (output latch)	R/W
P6	0	0	0	0	P63	P62	P61	P60	FFF06H	00H (output latch)	R/W
P7	P77	P76	P75	P74	P73	P72	P71	P70	FFF07H	00H (output latch)	R/W
P12	0	0	0	P124	P123	P122	P121	P120	FFF0CH	Undefined	R/W[Note 1]
P13	P137	0	0	0	0	0	0	P130	FFF0DH	**Note 2**	R/W[Note 1]
P14	P147	P146	0	0	0	0	P141	P140	FFF0EH	00H (output latch)	R/W

Pmn	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 1. P123, P124, and P137 are read-only.

Note 2. P137: Undefined
P130: 0 (output latch)

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

Remark m = 0 to 7, 12 to 14; n = 0 to 7

7.3.3 Pull-up resistor option registers xx (PUxx)

The PUxx registers specify whether to use the on-chip pull-up resistors. On-chip pull-up resistors can be used in 1-bit units only for the bits set to both normal output mode (POMmn = 0) and input mode (PMmn = 1) for the pins to which the use of an on-chip pull-up resistor has been specified in these registers. On-chip pull-up resistors are not 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. The value of the PU4 register following a reset is 01H, and that of the other PUxx is 00H.

Caution When data is input from a device operating at a different voltage to the TTL buffer for a port with the PIMn register, set PUmn = 0 and pull up to the power supply of the device operating at a different voltage via an external resistor.

Figure 7 - 3 Format of Pull-up Resistor Option Registers xx (PUxx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	PU06	PU05	PU04	PU03	PU02	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	PU43	PU42	PU41	PU40	F0034H	01H	R/W
PU5	0	0	PU55	PU54	PU53	PU52	PU51	PU50	F0035H	00H	R/W
PU6	0	0	0	0	PU63	PU62	0	0	F0036H	00H	R/W
PU7	PU77	PU76	PU75	PU74	PU73	PU72	PU71	PU70	F0037H	00H	R/W
PU12	0	0	0	0	0	PU122	PU121	PU120	F003CH	00H	R/W
PU14	PU147	PU146	0	0	0	0	PU141	PU140	F003EH	00H	R/W

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

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

7.3.4 Port input mode registers xx (PIMxx)

The PIMxx registers set the input buffer in 1-bit units. TTL input buffer can be selected during serial communication with an external device operating at a different voltage. The PIMxx registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each PIMxx register following a reset is 00H.

Figure 7 - 4 Format of Port Input Mode Registers xx (PIMxx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM0	0	0	0	PIM04	PIM03	0	PIM01	0	F0040H	00H	R/W
PIM1	PIM17	PIM16	PIM15	PIM14	0	0	PIM11	PIM10	F0041H	00H	R/W
PIM3	0	0	0	0	0	0	0	PIM30	F0043H	00H	R/W
PIM5	0	0	PIM55	0	0	0	0	PIM50	F0045H	00H	R/W
PIM7	0	0	0	0	PIM73	0	0	0	F0047H	00H	R/W

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

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

7.3.5 Port output mode registers xx (POMxx)

The POMxx registers set the output mode in 1-bit units. N-ch open drain output (withstand voltage of V_{DD}[Note 1]/ withstand voltage of EV_{DD}[Note 2]) mode can be selected during serial communication with an external device operating at a different voltage, and for the SDA00, SDA01, SDA10, SDA11, SDA20, and SDA21 pins during simplified I^2C communication with an external device operating at the same voltage. In addition, POMxx registers are used in combination with PUxx registers to specify whether to use on-chip pull-up resistors. The POMxx registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each POMxx register following a reset is 00H.

Caution An on-chip pull-up resistor is not connected to a bit for which N-ch open drain output (withstand voltage of V_{DD}[Note 1]/withstand voltage of EV_{DD}[Note 2]) mode (POMmn = 1) is set.

Note 1. This applies to the 20- to 52-pin products.
Note 2. This applies to the 64-pin products.

Figure 7 - 5 Format of Port Output Mode Registers xx (POMxx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM0	0	0	0	POM04	POM03	POM02	0	POM00	F0050H	00H	R/W
POM1	POM17	0	POM15	POM14	POM13	POM12	POM11	POM10	F0051H	00H	R/W
POM3	0	0	0	0	0	0	0	POM30	F0053H	00H	R/W
POM5	0	0	POM55	0	0	0	POM51	POM50	F0055H	00H	R/W
POM7	0	0	0	POM74	POM73	POM72	POM71	0	F0057H	00H	R/W

POMmn	Pmn pin output mode selection (m = 0, 1, 3, 5, 7; n = 0 to 7)
0	Normal output mode
1	N-ch open-drain output (withstand voltage of V_{DD}[Note 1]/withstand voltage of EV_{DD}[Note 2]) mode

Note 1. This applies to the 20- to 52-pin products.
Note 2. This applies to the 64-pin products.

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

7.3.6 Port digital input disable registers xx (PDIDISxx)

The PDIDISxx registers are used to prevent through-current flowing into input buffers. When N-ch open drain output is selected for serial communications with an external device operating at a different voltage or an input port is not used, low power consumption can be achieved by setting the corresponding bit in the given PDIDISxx register to 1. These registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each PDIDISxx register following a reset is 00H.

Figure 7 - 6 Format of Port Digital Input Disable Registers xx (PDIDISxx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PDIDIS0	0	0	0	PDIDIS04	PDIDIS03	PDIDIS02	0	PDIDIS00	F02B0H	00H	R/W
PDIDIS1	PDIDIS17	0	PDIDIS15	PDIDIS14	PDIDIS13	PDIDIS12	PDIDIS11	PDIDIS10	F02B1H	00H	R/W
PDIDIS3	0	0	0	0	0	0	0	PDIDIS30	F02B3H	00H	R/W
PDIDIS5	0	0	PDIDIS55	0	0	0	PDIDIS51	PDIDIS50	F02B5H	00H	R/W
PDIDIS7	0	0	0	PDIDIS74	PDIDIS73	PDIDIS72	PDIDIS71	0	F02B7H	00H	R/W
PDIDIS13	PDIDIS137	0	0	0	0	0	0	0	F02BDH	00H	R/W

PDIDISmn	Setting of input buffers (m = 0, 1, 3, 5, 7, 13; n = 0 to 7)
0	Input to the input buffer is enabled (default)
1	Input to the input buffer is disabled. Through-current flowing into the input buffer is prevented.

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

Remark For P123 and P124, low power consumption can be achieved by setting the EXCLKS bit to 0 and the OSCSELS bit to 1 in the clock operation mode control register (CMC) and setting the XTSTOP bit to 1 in the clock operation status control register (CSC).

7.3.7 Port mode control A registers xx (PMCAxx)

The PMCAxx registers specify the digital I/O or analog function in 1-bit units. These registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each PMCAxx register following a reset is FFH.

Figure 7 - 7 Format of Port Mode Control A Registers xx (PMCAxx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMCA0	1	1	1	1	PMCA03	PMCA02	PMCA01	PMCA00	F0060H	FFH	R/W
PMCA1	PMCA17	PMCA16	PMCA15	PMCA14	PMCA13	PMCA12	PMCA11	PMCA10	F0061H	FFH	R/W
PMCA2	PMCA27	PMCA26	PMCA25	PMCA24	PMCA23	PMCA22	PMCA21	PMCA20	F0062H	FFH	R/W
PMCA12	1	1	1	1	1	1	1	PMCA120	F006CH	FFH	R/W
PMCA14	PMCA147	PMCA146	1	1	1	1	1	1	F006EH	FFH	R/W

PMCAmn	Selection of digital I/O or analog function for Pmn pin (m = 0 to 2, 12, 14; n = 0 to 7)
0	Digital I/O
1	Analog function

Caution 1. Select input mode by using port mode register 0 to 2, 12, or 14 (PM0 to PM2, PM12, or PM14) for the port which is set to the analog function by the PMCAxx register.

Caution 2. Do not set the pin that is specified as digital I/O by the PMCAxx register to the analog function by the analog input channel specification register (ADS).

Caution 3. Be sure to set bits that are not implemented to their initial values.

7.3.8 Peripheral I/O redirection registers x (PIORx)

The PIORx registers are used to specify whether to enable or disable the peripheral I/O redirection. This function is used to switch the assignments of multiplexed functions to port pins. Enable a pin function that has been redirected after having used the PIORx register to assign the pin function to a port pin. Note that the settings for redirection can only be changed before the pin function is enabled. The PIORx registers can be set by an 8-bit memory manipulation instruction. The value of each PIORx register following a reset is 00H.

Figure 7 - 8 Format of Peripheral I/O Redirection Register 0 (PIOR0) (1/2)

Address: F0077H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PIOR0	PIOR07	PIOR06	PIOR05	PIOR04	PIOR03	PIOR02	PIOR01	PIOR00

Bit	Alternative function	64-pin		52-pin		48-pin		44- and 40-pin		32- and 30-pin		25- and 24-pin		20-pin	
		Setting value		Setting value		Setting value		Setting value		Setting value		Setting value		Setting value	
		0	1	0	1	0	1	0	1	0	1	0	1	0	1
PIOR07	TxD2	P13	P77	P13	P77	P13	P10	P13	P10	P13	P10	P13	P10	P13	P10
	RxD2	P14	P76	P14	P76	P14	P11	P14	P11	P14	P11	P14	P11	P14	P11
	SCL20	P15	—	P15	—	P15	—	P15	—	P15	—	P15	—	P15	—
	SDA20	P14	—	P14	—	P14	—	P14	—	P14	—	P14	—	P14	—
	SI20	P14	—	P14	—	P14	—	P14	—	P14	—	P14	—	P14	—
	SO20	P13	—	P13	—	P13	—	P13	—	P13	—	P13	—	P13	—
	SCK20	P15	—	P15	—	P15	—	P15	—	P15	—	P15	—	P15	—
PIOR06[Note]	INTP6	P140	P20	P140	P20	P140	P20	—	P20	—	P20	—	P20	—	P20
	INTP7	P141	P21	—	P21	—	P21	—	P21	—	P21	—	P21	—	P21
	TxD0_1[Note]	—	P12	—	P12	—	P12								
	RxD0_1[Note]	—	P11	—	P11	—	P11								
PIOR05	TxD1	P02	P72	P02	P72	P00	P72	P00	P72	P00	P30	These functions are not available for use. Set this bit to 0 (default value).			
	RxD1	P03	P73	P03	P73	P01	P73	P01	P73	P01	P31				
PIOR04	PCLBUZ1	P141	P55												
	INTP5	P16	P12												
PIOR03	PCLBUZ0	P140	P31	P140	P31	P140	P31								
PIOR02	SCLA0	P60	P14	P60	P14	P60	P14	P60	P14	P60	P14	—	P14	—	P14
	SDAA0	P61	P15	P61	P15	P61	P15	P61	P15	P61	P15	—	P15	—	P15

Figure 7 - 8 Format of Peripheral I/O Redirection Register 0 (PIOR0) (2/2)

Bit	Alternative function	64-pin		52-pin		48-pin		44- and 40-pin		32- and 30-pin		25- and 24-pin		20-pin	
		Setting value		Setting value		Setting value		Setting value		Setting value		Setting value		Setting value	
		0	1	0	1	0	1	0	1	0	1	0	1	0	1
PIOR01	INTP10	P76	P05	P76	—	—	—	—	—	—	—	—	—	—	—
	INTP11	P77	P06	P77	—	—	—	—	—	—	—	—	—	—	—
	TxD0	P51	P17	P51	P17	P51	P17	P51	P17	P51	P17	—	P17	—	—
	RxD0	P50	P16	P50	P16	P50	P16	P50	P16	P50	P16	—	P16	—	—
	SCL00	P30	—	P30	—	P30	—	P30	—	P30	—	—	—	—	—
	SDA00	P50	—	P50	—	P50	—	P50	—	P50	—	—	—	—	—
	SI00	P50	P16	P50	—	P50	—	P50	—	P50	—	—	—	—	—
	SO00	P51	P17	P51	—	P51	—	P51	—	P51	—	—	—	—	—
	SCK00	P30	P55	P30	—	P30	—	P30	—	P30	—	—	—	—	—
PIOR00	INTP1	P50	P52	These functions are not available for use. Set this bit to 0 (default value).											
	INTP2	P51	P53												
	INTP3	P30	P54												
	INTP4	P31	P55												
	INTP8	P74	P42												
	INTP9	P75	P43												

Note When the setting of bit 6 (PIOR06) of the PIOR0 register is 1, the TxD0_1 and RxD0_1 functions can be assigned to P12 and P11, respectively. At this time, TxD0 and RxD0 function assignments made by the PIOR01 bit become invalid, and the TxD0_1 and RxD0_1 functions become valid.

Caution 1. When the setting of bit 1 (PIOR01) of the PIOR0 register is 1, IIC20 and CSI20 cannot be used because TxD2 and RxD2 are redirected but SCL20, SDA20, SI20, SO20, and SCK20 are not. If UART2 is not to be used while PIOR01 is set to 1, CSI21 and IIC21 can be used.

Caution 2. When the setting of bit 1 (PIOR01) of the PIOR0 register is 1, CSI00 cannot be used in a product with 52 or fewer pins because SO00 and SI00 are redirected but SCK00 is not.

Remark —: These functions are not available for use.

Figure 7 - 9 Format of Peripheral I/O Redirection Register 1 (PIOR1)

Address: F02ADH
After reset: 00H
R/W: R/W

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

PIOR13	PIOR12	Selection of TRJO0 pin for timer RJ
0	0	Multiplexed with P30Note 1
0	1	Multiplexed with P50Note 1
1	0	Multiplexed with P00
1	1	Setting prohibited

PIOR11	PIOR10	Selection of TRJIO0 pin for timer RJ
0	0	Multiplexed with P01
0	1	Multiplexed with P31Note 1
1	0	Multiplexed with P41Note 2
1	1	Multiplexed with P06Note 3

Note 1. This setting is only applicable to the 64-, 52-, 48-, 44-, 40-, 32-, and 30-pin products.
Note 2. This setting is only applicable to the 64-, 52-, 48-, and 44-pin products.
Note 3. This setting is only applicable to the 64-pin products.

Figure 7 - 10 Format of Peripheral I/O Redirection Register 2 (PIOR2)

Address: F02AEH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PIOR2	PIOR27	PIOR26	PIOR25	PIOR24	PIOR23	PIOR22	PIOR21	PIOR20

Bit	Alternative function	64-, 52-, and 48-pin		44- and 40-pin		32-pin		30-pin		25- and 24-pin		20-pin	
		Setting value		Setting value		Setting value		Setting value		Setting value		Setting value	
		0	1	0	1	0	1	0	1	0	1	0	1
PIOR27	TRDIOC0	P16	P13	P16	P13	P16	P13	P16	P13	P16	P13	—	P13
PIOR26	TRDIOD0	P14	P71	P14	P71	P14	P17	P14	P17	P14	P17		
PIOR25	TRDIOD1	P10	P75	P10	P51	P10	P51	P10	P51	These functions are not available for use. Set this bit to 0 (default value).			
PIOR24	TRDIOC1	P11	P73	P11	P73	P11	P50	P11	P50				
PIOR23	TRDIOB1	P12	P74	P12	P30	P12	P30	P12	P30				
PIOR22	TRDIOA1	P13	P72	P13	P72	P13	P16	P13	P16	P13	P16		
PIOR21	TRDIOB0	P15	P70	P15	P70	P15	P70						
PIOR20	TRGIOA	P50	P17	P50	P17	P50	P17	P50	P17	—	P17		
	TRGIOB	P51	P16	P51	P16	P51	P16	P51	P16	—	P16		

Remark —: These functions are not available for use.

Figure 7 - 11 Format of Peripheral I/O Redirection Register 3 (PIOR3)

Address: F02AFH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PIOR3	PIOR37	PIOR36	PIOR35	PIOR34	PIOR33	PIOR32	PIOR31	PIOR30

Bit	Alternative function	64- and 52-pin		48-, 44-, and 40-pin		32- and 30-pin		25- and 24-pin		20-pin	
		Setting value		Setting value		Setting value		Setting value		Setting value	
		0	1	0	1	0	1	0	1	0	1
PIOR37	DALITxD0	P51	P02	P51	P00	P51	P00	—	P00	—	P00
	DALIRxD0	P50	P03	P50	P01	P50	P01	—	P01	—	P01
PIOR36	TI00	P00	P03	P00	P01	P00	P01	P00	P01	P00	P01
	TO00	P01	P122	P01	P122	P01	P122	P01	P122	P01	P122
	TI02/TO02	P17	P121	P17	P121	P17	P120	P17	P120	—	P120
PIOR35	TI01/TO01	P16	P71	P16	P71	P16	P15	P16	P15	—	P15
PIOR34	TI03	P31	P50	P31	P50	P31	P50	—	P11	—	P11
	TO03	P31	P11	P31	P11	P31	P11	—	P11	—	P11
PIOR33	VCOUT3	P11	P50	P11	P50	P11	P50	These functions are not available for use. Set this bit to 0 (default value).			
PIOR32	VCOUT2	P10	P51	P10	P51	P10	P51				
PIOR31	VCOUT1	P15	P30	P15	P30	P15	P30				
PIOR30	VCOUT0	P14	P31	P14	P31	P14	P31				

Remark —: These functions are not available for use.

7.3.9 Global digital input disable register (GDIDIS)

The GDIDIS register is used to prevent through-current flowing into the input buffers of input ports which use EVDD as the power supply when the EVDD power supply is turned off. When none of the I/O ports using EVDD as the power supply are used, low power consumption can be achieved by setting the GDIDIS0 bit to 1 to turn off the EVDD power supply. By setting the GDIDIS0 bit to 1, input to any input buffer using EVDD as the power supply is prohibited, preventing through-current from flowing when the EVDD power supply is turned off. The GDIDIS register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Remark The GDIDIS register is equipped with 64-pin products.

Figure 7 - 12 Format of Global Digital Input Disable Register (GDIDIS)

Address: F007DH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
GDIDIS	0	0	0	0	0	0	0	GDIDIS0

GDIDIS0	Setting of input buffers using EVDD power supply
0	Input to input buffers permitted (default)
1	Input to input buffers prohibited. No through-current flows to the input buffers.

Remark 1. The GDIDIS register is equipped with 64-pin products.
Remark 2. Even when input to input buffers is prohibited (GDIDIS0 = 1), peripheral functions which do not use port functions having EVDD as the power supply can be used.

Turn off the EVDD power supply with the following procedure.

<1> Prohibit input to input buffers (set GDIDIS0 = 1).
<2> Turn off the EVDD power supply.

Turn on again the EVDD power supply with the following procedure.

<1> Turn on the EVDD power supply.
<2> Permit input to input buffers (set GDIDIS0 = 0).

Caution 1. Do not input an input voltage equal to or greater than EVDD to an input port that uses EVDD as the power supply.
Caution 2. When input to input buffers is prohibited (GDIDIS0 = 1), the value read from port register xx (Pxx) of a port that uses EVDD as the power supply is 1. When 1 is set in port output mode register xx (POMxx) (N-ch open drain output (withstand voltage of EVDD) mode), the value read from port register xx (Pxx) is 0.

7.3.10 Output current control enable register (CCDE)

The CCDE register is used to specify the use of P10, P11, P16, P17, and P60 to P63 as controlled current drive port pins in 1-bit units. The setting of the corresponding output current select register x (CCSx) controls a pin for which this function is selected to be at the low level and produce the selected output current (low-level output current) or to be in the high-impedance state. The CCDE register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 7 - 13 Format of Output Current Control Enable Register (CCDE)

Address: F02A8H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
CCDE	CCDE07	CCDE06	CCDE05	CCDE04	CCDE03	CCDE02	CCDE01	CCDE00

CCDE07	Selection of digital I/O or output current control function for CCD07 (P11) pin
0	Digital I/O (alternate function other than current control function)
1	Current control function

CCDE06	Selection of digital I/O or output current control function for CCD06 (P10) pin
0	Digital I/O (alternate function other than current control function)
1	Current control function

CCDE05	Selection of digital I/O or output current control function for CCD05 (P61) pin
0	Digital I/O (alternate function other than current control function)
1	Current control function

CCDE04	Selection of digital I/O or output current control function for CCD04 (P60) pin
0	Digital I/O (alternate function other than current control function)
1	Current control function

CCDE03	Selection of digital I/O or output current control function for CCD03 (P63) pin
0	Digital I/O (alternate function other than current control function)
1	Current control function

CCDE02	Selection of digital I/O or output current control function for CCD02 (P62) pin
0	Digital I/O (alternate function other than current control function)
1	Current control function

CCDE01	Selection of digital I/O or output current control function for CCD01 (P17) pin
0	Digital I/O (alternate function other than current control function)
1	Current control function

CCDE00	Selection of digital I/O or output current control function for CCD00 (P16) pin
0	Digital I/O (alternate function other than current control function)
1	Current control function

(Cautions are listed on next page.)

Caution 1. When a port pin is to be used with output current control, make the setting for the output current control function and then set the corresponding bit in the PMxx register for output mode.

Caution 2. The state of a pin takes 10 µs to become stable after 1 having been written to the corresponding bit of the CCDE register.

7.3.11 Output current select registers x (CCSx)

The CCSx registers are used to set the current control for the port pins selected for output current control in the output current control enable register (CCDE) or to place them in the high-impedance state. For current control, the pins are at the low level and produce output currents controlled to be 2 mA, 5 mA, 10 mA, or 15 mA. The CCSx registers can be set by an 8-bit memory manipulation instruction. The value of each CCSx register following a reset is 00H.

Figure 7 - 14 Format of Output Current Select Registers x (CCSx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
CCS0	0	0	0	0	0	CCS02	CCS01	CCS00	F02A0H	00H	R/W
CCS4	0	0	0	0	0	CCS42	CCS41	CCS40	F02A4H	00H	R/W
CCS5	0	0	0	0	0	CCS52	CCS51	CCS50	F02A5H	00H	R/W
CCS6	0	0	0	0	0	CCS62	CCS61	CCS60	F02A6H	00H	R/W
CCS7	0	0	0	0	0	CCS72	CCS71	CCS70	F02A7H	00H	R/W

CCSn2	CCSn1	CCSn0	Setting for the low-level output current				
			n = 0 CCD00 to CCD03	n = 4 CCD04	n = 5 CCD05	n = 6 CCD06	n = 7 CCD07
0	0	0	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
0	0	1	2 mA	2 mA	2 mA	2 mA	2 mA
0	1	0	5 mA	5 mA	5 mA	5 mA	5 mA
0	1	1	10 mA	10 mA	10 mA	10 mA	10 mA
1	0	0	Setting prohibited	15 mA	15 mA	Setting prohibited	Setting prohibited
Others			Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited

7.3.12 Port mode select register (PMS)

The PMS register is used to specify whether the value in the output latch for a port is read or the output level on a port pin is read when the pin is in output mode (the PMmn bit of port mode register m (PMm) is 0). For details, see **35.3.11.1 Port mode select register (PMS)**.

7.4 Port Function Operations

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

7.4.1 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 state does not change. Therefore, byte data can be written to the ports used for both input and output.
   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.

7.4.2 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 state is read by a transfer instruction. The output latch contents do not change.

7.4.3 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 the read value. The result of the operation is written to the output latch, but since the output buffer is off, the pin state does not change. Therefore, byte data can be written to the ports used for both input and output.
   The data of the output latch is cleared when a reset signal is generated.

7.4.4 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching the EV_{DD} voltage

Communications with external devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) to this device is possible by switching EV_{DD}**Note** through the general-purpose I/O pins, in accord with the power supply of the device to be connected.

Note $EV_{DD} \leq V_{DD}$ must be satisfied.

7.4.5 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers

The port input mode registers xx (PIMxx) and port output mode registers xx (POMxx) can be used to switch the I/O buffers to enable communications with external devices that have different operating voltages (1.8 V, 2.5 V, or 3 V) to this device. To receive input from external devices operating at different voltages, set the relevant bits of port input mode registers 0, 1, 3, 5, and 7 (PIM0, PIM1, PIM3, PIM5, and PIM7) in 1-bit units to switch from normal input (CMOS) to the TTL input buffers. To output data to external devices operating at different voltages, set the relevant bits of port output mode registers 0, 1, 3, 5, and 7 (POM0, POM1, POM3, POM5, and POM7) in 1-bit units to switch from normal output (CMOS) and N-ch open drain output (withstand voltage of V_{DD}**Note 1**/withstand voltage of EV_{DD}**Note 2**). Setting port digital input disable registers 0, 1, 3, 5, 7, and 13 (PDIDIS0, PDIDIS1, PDIDIS3, PDIDIS5, PDIDIS7, and PDIDIS13) bit-by-bit can prevent the flow of through currents to the corresponding input buffers. The following describes connection through a serial interface to an external device operating at a different voltage.

Note 1. This applies to the 20- to 52-pin products.
Note 2. This applies to the 64-pin products.

1. Procedure for setting input pins of UART0 to UART2, CSI00, CSI10, and CSI20 for use with the TTL input buffers
   Use the following port pin or pins for each interface.
   P50 (P16) for UART0
   P03 (P01 and P73) for UART1
   P14 (P11) for UART2
   P30 and P50 (P55 and P16) for CSI00
   P03 and P04 for CSI10
   P14 and P15 for CSI20

   Remark Functions can be assigned to the pins in parentheses via settings in peripheral I/O redirection registers x (PIORx).

   <1> Using an external resistor, pull up externally the input 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 to PIM5, and PIM7 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/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. Procedure for setting output pins of UART0 to UART2, CSI00, CSI10, and CSI20 for use with the N-ch open-drain output mode
   P51 (P17) for UART0
   P02 (P72 and P00) for UART1
   P13 (P10) for UART2
   P30 and P51 (P55 and P17) for CSI00
   P02 and P04 for CSI10
   P13 and P15 for CSI20

   Remark Functions can be assigned to the pins in parentheses via settings in peripheral I/O redirection registers x (PIORx).

   <1> Using an external resistor, pull up externally the output 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 ports are in the input mode (Hi-Z).
   <3> Set the corresponding bit of the PDIDIS0, PDIDIS1, PDIDIS3, PDIDIS5, and PDIDIS7 registers to 1 to disable input to the input buffer.
   <4> Set the output latch of the corresponding port to 1.
   <5> Set the corresponding bit of the POM0, POM1, POM3, POM5, and POM7 registers to 1 to set the N-ch open drain output (withstand voltage of V_{DD}**Note 1**/withstand voltage of EV_{DD}**Note 2**) mode.
   <6> Enable the operation of the serial array unit and set the mode to the UART/simplified SPI (CSI**Note 3**) mode.
   <7> Set the corresponding bit of the PM0, PM1, PM3, PM5, and PM7 registers to the output mode. At this time, the output data is high level, so the pin is in the Hi-Z state.

   Note 1. This applies to the 20- to 52-pin products.
   Note 2. This applies to the 64-pin products.
   Note 3. 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.

3. Procedure for setting I/O pins of IIC00, IIC10, IIC11, and IIC20 for use in connection with a device operating at a different voltage (1.8 V, 2.5 V, or 3 V)
   P30 and P50 for IIC00
   P03 and P04 for IIC10
   P10 and P11 for IIC11
   P14 and P15 for IIC20

   <1> Using an external resistor, pull up externally the input 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 ports are in 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 (withstand voltage of V_{DD}**Note 1**/withstand voltage of EV_{DD}**Note 2**) mode.
   <5> 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.
   <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 PM0, PM1, 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.

   Note 1. This applies to the 20- to 52-pin products.
   Note 2. This applies to the 64-pin products.

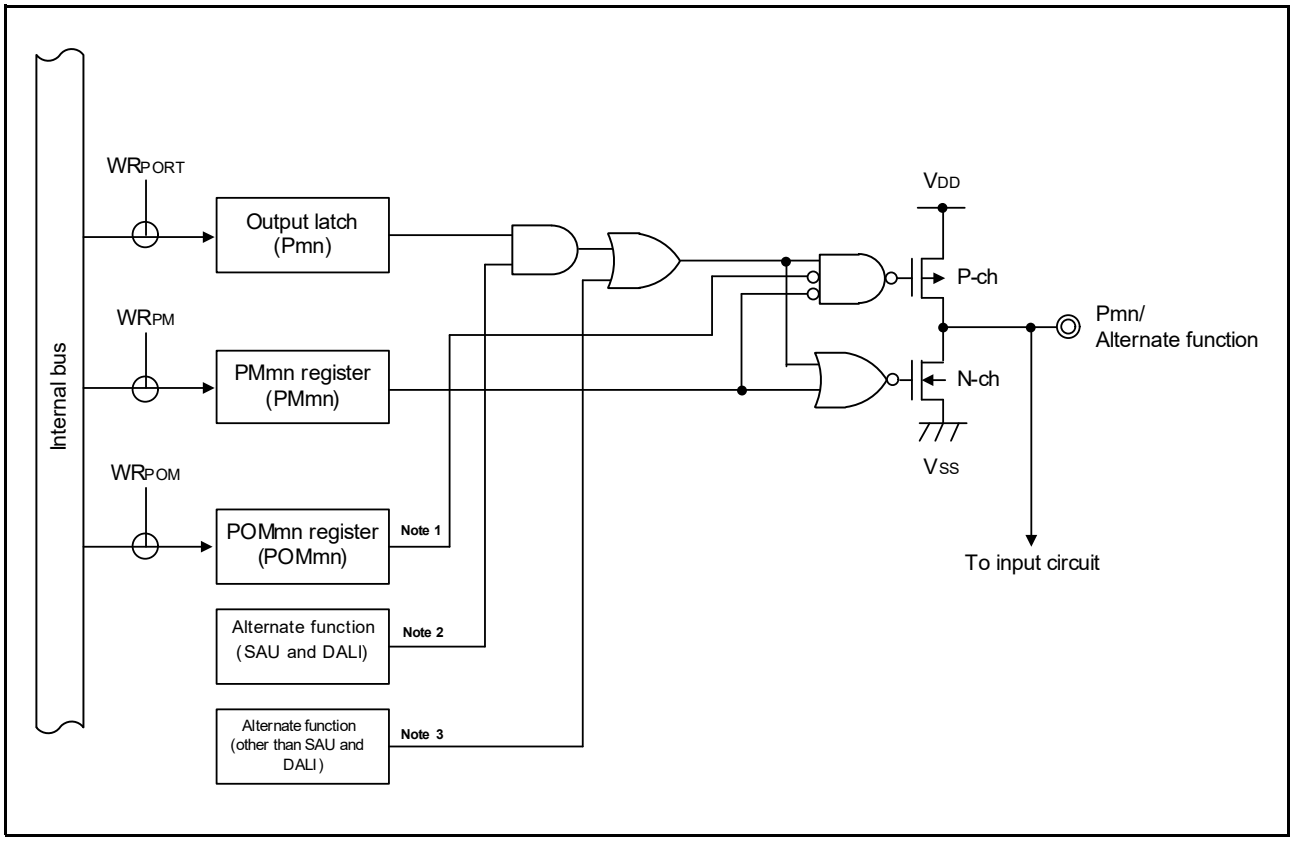
7.5 Register Settings When Using Alternate Function

7.5.1 Basic concept when using alternate function

In the beginning, for a pin that is also assigned to the analog function, use the corresponding port mode control A register xx (PMCAxx) to specify whether to use the pin for the analog function or digital input/output.

Figure 7 - 15 shows the basic configuration of the output circuit for a pin used for digital input/output. The outputs from multiplexed SAU and DALI functions and the outputs from the output latches for the port-pin functions are input to an AND gate. The output of the AND gate is input to an OR gate. The outputs of multiplexed functions other than the SAU and DALI (timer array unit, realtime clock, clock/buzzer output, IICA, etc.) are connected to other input pins of the OR gate. When such a pin is used for the port function or an alternate function, the unused alternate function must not hinder the output of the function to be used. A concept of basic settings for this case is shown in **Table 7 - 3**.

Figure 7 - 15 Basic Configuration of Output Circuit for a Pin



Note 1. For pins that have no related POMmn register, consider this signal to be 0.

Note 2. For pins that have no multiplexed functions, consider this signal to be 1.

Note 3. For pins that have no multiplexed functions, consider this signal to be 0.

Remark m: Port number (m = 0 to 7, 12 to 14); n: Bit number (n = 0 to 7)

Table 7 - 3 Concept of Basic Settings

Output Function of Used Pin	Output Settings of Unused Alternate Function		
	Port-pins Output Function	Output Functions of the SAU and DALI	Output Functions Other Than Those of the SAU and DALI
Port-pins output function	—	Output at the high level	Output at the low level
Output functions of the SAU and DALI	High level	Output at the high level	Output at the low level
Output functions other than those of the SAU and DALI	Low level	Don't care	Output at the low level[Note]

Note The output of the multiplexed functions which are not in use must be set to the low level because two or more output functions other than those of the SAU and DALI may be multiplexed on the same pin. The output of the multiplexed functions which are not in use must be set to the high level because two or more output functions of the SAU and DALI may be multiplexed on the same pin. For details on the setting method, see **7.5.2 Register settings for alternate function whose output function is not used**.

7.5.2 Register settings for alternate function whose output function is not used

When the output of an alternate function of the pin is not used, the following settings should be made. Note that when the peripheral I/O redirection function is applicable to an alternate function, the output can be switched to another pin by setting the corresponding peripheral I/O redirection register x (PIORx). This allows usage of the port function or other alternate function assigned to the target pin.

1. SOp = 1, TxDq = 1 (settings when the serial output (SOp/TxDq) of SAU is not used)
   When the serial output (SOp/TxDq) is not used, such as a case in which only the serial input of SAU is used, set the bit in serial output enable register m (SOEm) which corresponds to the unused output to 0 (output disabled) and set the SOmn bit in serial output register m (SOm) to 1 (high-level output). When SOp and TxDq pins are to be used for the port pin functions, set the PFOE1x bits corresponding to the SOp, TxDq, and SCKp pins to 1. If a pin is to be used for a multiplexed function other than the port pin function, the corresponding PFOE1x bit can be set to 0. These are the same settings as the initial state.

2. SCKp = 1, SDAr = 1, SCLr = 1 (settings when channel n in SAU is not used)
   When SAU is not used, set bit n (SEmn) in serial channel enable status register m (SEm) to 0 (operation stopped state), set the bit in serial output enable register m (SOEm) which corresponds to the unused output to 0 (output disabled), and set the SOmn and CKOmn bits in serial output register m (SOm) to 1 (high-level output). When SCKp, SDAr, and SCLr pins are to be used for the port pin functions, set the PFOE1x bits corresponding to the SOp, TxDq, and SCKp pins to 1. If a pin is to be used for a multiplexed function other than the port pin function, the corresponding PFOE1x bit can be set to 0. These are the same settings as the initial state.

3. TOmn = 0 (settings when the output of channel n in TAU is not used)
   When the TOmn output of TAU is not used, set the bit in timer output enable register 0 (TOE0) which corresponds to the unused output to 0 (output disabled) and set the corresponding bit in timer output register 0 (TO0) to 0 (low-level output). These are the same settings as the initial state. Regardless of the selection for timer output operation, a TOmn output can also be stopped by setting the PFOE0x bit corresponding to the TOmn pin to 0.

4. SDAAn = 0, SCLAn = 0 (setting when IICA is not used)
   When IICA is not used, set the IICEn bit in IICA control register n0 (IICCTLn0) to 0 (operation stopped). This is the same setting as the initial state.

5. PCLBUZn = 0 (setting when clock/buzzer output is not used)
   When the clock/buzzer output is not used, set the PCLOEn bit in clock output select register n (CKSn) to 0 (output disabled). This is the same setting as the initial state.

Remark p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), q: UART number (q = 0 to 3)
r: IIC number (r = 00, 01, 10, 11, 20, 21, 30, 31)

7.5.3 Register settings and port pin state

The correspondence between register settings and port pin state is shown in **Table 7 - 4**.

Table 7 - 4 Correspondence between Register Settings and Port Pin State

PMCAxx	PMxx	Pxx	PUxx	CCDE0x	CCSx	Pin State
1	×	×	×	×	×	Analog input/output
0	1	×	1	×	×	Pulled up
0	1	×	0	×	×	Hi-Z
0	0	1	×	×	×	Port output (high level)
0	0	0	×	1	000	Hi-Z
0	0	0	×	1	001 010 011 100	Controlled current drive port
0	0	0	×	0	×	Port output (low level)

Remark ×: Don't care

7.5.4 Examples of register settings for port and alternate functions

Examples of register settings for port and alternate functions are shown in **Table 7 - 5**. The registers used to control the port functions should be set as shown in **Table 7 - 5**. See the following remark for legends used in **Table 7 - 5**.

Caution **In examples of register settings, port digital input disable register xx (PDIDISxx) is set for input to the input buffer being enabled.**

Remark —: Not supported
×: Don’t care
PIORx: Peripheral I/O redirection registers x
POMxx: Port output mode registers xx
PMCAxx: Port mode control A registers xx
CCDE: Output current control enable register
CCSx: Output current select registers x
PMxx: Port mode registers xx
Pxx: Port output latch
Functions in parentheses can be assigned via settings in peripheral I/O redirection registers x (PIORx).

Table 7 - 5 Examples of Register and Output Latch Settings for Alternate Functions (1/22)

Pin Name	Function Used		PIORx	POMxx	PMCAxx	PMxx	Pxx	Alternate Function Output		20-pin	24-pin	25-pin	30-pin	32-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O						SAU and DALI	Other Than SAU and DALI										
P00	P00	Input	—	×	0	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	0/1	TxD1 = 1 Note 1, (DALITxD0) = 1 Note 1	(TRJO0) = 0										
		N-ch open drain output	—	1	0	0	0/1												
	ANI29	Analog input	—	×	1	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	IVCMP1	Analog input	—	×	1	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	PGAI1	Analog input	—	×	1	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TI00	Input	PIOR36 = 0	×	0	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TxD1	Output	PIOR05 = 0 Note 2	0/1	0	0	1	×	(TRJO0) = 0	✓	✓	✓	✓	✓	✓	✓	✓	—	—
	TRGCLKA	Input	—	×	0	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(TRJO0)	Output	PIOR13, PIOR12 = 10B	0	0	0	0	TxD1 = 1 Note 1, (DALITxD0) = 1 Note 1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(DALITxD0)	Output	PIOR37 = 1	0/1	0	0	1	×	×	✓	✓	✓	✓	✓	✓	✓	✓	—	—
P01	P01	Input	—	—	0	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0	0/1	—	TO00 = 0, TRJIO = 0	✓	✓	✓	✓	✓	✓	✓	✓		
	ANI30	Analog input	—	—	1	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	IVCMP2	Analog input	—	—	1	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	PGAI2	Analog input	—	—	1	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TO00	Output	PIOR36 = 0	—	0	0	0	—	TRJIO = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	RxD1	Input	PIOR05 = 0 Note 2	—	0	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	—	—
	TRGCLKB	Input	—	—	0	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TRJIO0	Input	PIOR11, PIOR10 = 00B	—	0	1	×	—	TO00 = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output		—	0	0	0	—	TO00 = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(DALIRxD0)	Input	PIOR37 = 1	—	0	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	—	—
	(TI00)	Input	PIOR36 = 1	—	0	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	—	—
P02	P02	Input	—	×	0	1	×	×	—	—	—	—	—	—	—	—	—	✓	✓
		Output	—	0	0	0	0/1	TxD1 = 1, SO10 = 1 Note 3, (DALITxD0) = 1	—										
		N-ch open drain output	—	1	0	0	0/1												
	ANI17	Analog input	—	×	1	1	×	×	—	—	—	—	—	—	—	—	—	✓	✓
	TxD1	Output	PIOR05 = 0	0/1	0	0	1	×	—	—	—	—	—	—	—	—	—	✓	✓
	SO10	Output	—	0/1	0	0	1	×	—	—	—	—	—	—	—	—	—	—	✓
	(DALITxD0)	Output	PIOR37 = 1	0/1	0	0	1	×	—	—	—	—	—	—	—	—	—	✓	✓
P03	P03	Input	—	×	0	1	×	×	—	—	—	—	—	—	—	—	—	✓	✓
		Output	—	0	0	0	0/1	SDA10 = 1 Note 3	—										
		N-ch open drain output	—	1	0	0	0/1												
	ANI16	Analog input	—	×	1	1	×	×	—	—	—	—	—	—	—	—	—	✓	✓
	SI10	Input	—	×	0	1	×	×	—	—	—	—	—	—	—	—	—	—	✓
	RxD1	Input	PIOR05 = 0	×	0	1	×	×	—	—	—	—	—	—	—	—	—	✓	✓
	SDA10	I/O	—	1	0	0	1	×	—	—	—	—	—	—	—	—	—	—	✓
	(DALIRxD0)	Input	PIOR37 = 1	×	0	1	×	×	—	—	—	—	—	—	—	—	—	✓	✓
	(TI00)	Input	PIOR36 = 1	×	0	1	×	×	—	—	—	—	—	—	—	—	—	✓	✓

Note 1. This setting is only applicable in the 20- to 48-pin products.

Note 2. This setting is only applicable in the 30- to 48-pin products.

Note 3. This setting is only applicable in the 64-pin products.

Table 7 - 5 Examples of Register and Output Latch Settings for Alternate Functions (2/22)

Pin Name	Function Used		PIORx	POMxx	PMCAxx	PMxx	Pxx	Alternate Function Output		20-pin	24-pin	25-pin	30-pin	32-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O						SAU and DALI	Other Than SAU and DALI										
P04	P04	Input	—	×	—	1	×	×	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	0	—	0	0/1	SCK10/SCL10 = 1	—	—	—	—	—	—	—	—	—	—	✓
		N-ch open drain output	—	1	—	0	0/1			—	—	—	—	—	—	—	—	—	✓
	SCK10	Input	—	×	—	1	×	×	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	0/1	—	0	1	×	—	—	—	—	—	—	—	—	—	—	✓
	SCL10	Output	—	0/1	—	0	1	×	—	—	—	—	—	—	—	—	—	—	✓
P05	P05	Input	—	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	—	0	0/1	—	—										
	(INTP10)	Input	PIOR01 = 1	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓
P06	P06	Input	—	—	—	1	×	—	×	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	—	0	0/1	—	(TRJIO) = 0	—	—	—	—	—	—	—	—	—	✓
	(TRJIO0)	Input	PIOR11, PIOR10 = 11B	—	—	1	×	—	×	—	—	—	—	—	—	—	—	—	✓
		Output		—	—	0	0	—	×	—	—	—	—	—	—	—	—	—	✓
	(INTP11)	Input	PIOR01 = 1	—	—	1	×	—	×	—	—	—	—	—	—	—	—	—	✓

Table 7 - 5 Examples of Register and Output Latch Settings for Alternate Functions (3/22)

Pin Name	Function Used		PIORx	POMxx	PMCAxx	CCDE	CCSx	PMxx	Pxx	Alternate Function Output		20-pin	24-pin	25-pin	30-pin	32-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O								SAU and DALI	Other Than SAU and DALI										
P10	P10	Input	—	×	0	CCDE06 = 0	CCS0[2:0] = xxxB	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	CCDE06 = 0	CCS0[2:0] = xxxB	0	0/1	SCK11/SCL11 = 1, (TxD2) = 1 Note 1	TRDIOD1 = 0, VCOUT2 = 0										
		N-ch open drain output	—	1	0	CCDE06 = 0	CCS0[2:0] = xxxB	0	0/1												
	ANI20	Analog input	—	×	1	CCDE06 = 0	CCS0[2:0] = xxxB	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	CCD06	Output	—	0/1	0	CCDE06 = 1	CCS0[2:0] = 001B to 011B	0	0	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SCK11	Input	—	×	0	CCDE06 = 0	CCS0[2:0] = xxxB	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0/1	0	CCDE06 = 0	CCS0[2:0] = xxxB	0	1	×	TRDIOD1 = 0, VCOUT2 = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SCL11	Output	—	0/1	0	CCDE06 = 0	CCS0[2:0] = xxxB	0	1	×	TRDIOD1 = 0, VCOUT2 = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TRDIOD1	Input	PIOR25 = 0 Note 2	×	0	CCDE06 = 0	CCS0[2:0] = xxxB	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output		0	0	CCDE06 = 0	CCS0[2:0] = xxxB	0	0	SCK11/SCL11 = 1, (TxD2) = 1 Note 1	VCOUT2 = 0										
	VCOUT2	Output	PIOR32 = 0 Note 2	0	0	CCDE06 = 0	CCS0[2:0] = xxxB	0	0	SCK11/SCL11 = 1, (TxD2) = 1 Note 1	TRDIOD1 = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(TxD2)	Output	PIOR07 = 1	0/1	0	CCDE06 = 0	CCS0[2:0] = xxxB	0	1	×	×	✓	✓	✓	✓	✓	✓	✓	✓	—	—
P11	P11	Input	—	×	0	CCDE07 = 0	CCS0[2:0] = xxxB	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	CCDE07 = 0	CCS0[2:0] = xxxB	0	0/1	SDA11 = 1	TRDIOC1 = 0, VCOUT3 = 0, (TO03) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		N-ch open drain output	—	1	0	CCDE07 = 0	CCS0[2:0] = xxxB	0	0/1												
	ANI21	Analog input	—	×	1	CCDE07 = 0	CCS0[2:0] = xxxB	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	PGAO	Analog output	—	×	1	CCDE07 = 0	CCS0[2:0] = xxxB	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	CCD07	Output	—	0/1	0	CCDE07 = 1	CCS0[2:0] = 001B to 011B	0	0	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SI11	Input	—	×	0	CCDE07 = 0	CCS0[2:0] = xxxB	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SDA11	I/O	—	1	0	CCDE07 = 0	CCS0[2:0] = xxxB	0	1	×	TRDIOC1 = 0, VCOUT3 = 0, (TO03) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TRDIOC1	Input	PIOR24 = 0 Note 2	×	0	CCDE07 = 0	CCS0[2:0] = xxxB	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output		0	0	CCDE07 = 0	CCS0[2:0] = xxxB	0	0	SDA11 = 1	VCOUT3 = 0, (TO03) = 0										
	VCOUT3	Output	PIOR33 = 0 Note 2	0	0	CCDE07 = 0	CCS0[2:0] = xxxB	0	0	SDA11 = 1	TRDIOC1 = 0, (TO03) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(RxD0_1)	Input	PIOR06 = 1	—	0	CCDE07 = 0	CCS0[2:0] = xxxB	1	×	×	×	—	—	—	—	—	—	—	✓	✓	✓
	(RxD2)	Input	PIOR07 = 1	×	0	CCDE07 = 0	CCS0[2:0] = xxxB	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	—	—
	(TI03)	Input	PIOR34 = 1	×	0	CCDE07 = 0	CCS0[2:0] = xxxB	1	×	×	×	✓	✓	✓	—	—	—	—	—	—	—
	(TO03)	Output	PIOR34 = 1	0	0	CCDE07 = 0	CCS0[2:0] = xxxB	0	0	SDA11 = 1	TRDIOC1 = 0, VCOUT3 = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note 1. This setting is only applicable in the 24- to 48-pin products.

Note 2. This setting is only applicable in the 30- to 64-pin products.

Table 7 - 5 Examples of Register and Output Latch Settings for Alternate Functions (4/22)

Pin Name	Function Used		PIORx	POMxx	PMCAxx	CCDE	CCSx	PMxx	Pxx	Alternate Function Output		20-pin	24-pin	25-pin	30-pin	32-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O								SAU and DALI	Other Than SAU and DALI										
P12	P12	Input	—	×	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	—	—	0	0/1	SO11 = 1	TRDIOB1 = 0, TKBO00 = 0										
		N-ch open drain output	—	1	0	—	—	0	0/1	SO11 = 1	TRDIOB1 = 0, TKBO00 = 0										
	ANI22	Analog input	—	×	1	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	IVREF1	Analog input	—	×	1	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SO11	Output	—	0/1	0	—	—	0	1	×	TRDIOB1 = 0, TKBO00 = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TRDIOB1	Input	PIOR23 = 0 Note 2	×	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	—	—	0	0	SO11 = 1	TKBO00 = 0										
	TKBO00	Output	—	0	0	—	—	0	0	SO11 = 1	TRDIOB1 = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(INTP5)	Input	PIOR04 = 1	×	0	—	—	1	×	×	×	—	—	—	—	—	—	—	—	—	✓
	(TxD0_1)	Output	PIOR06 = 1	0/1	0	—	—	0	1	—	—	—	—	—	—	—	—	—	✓	✓	✓
P13	P13	Input	—	×	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	—	—	0	0/1	TxD2/SO20 = 1	TRDIOA1 = 0, TKBO01 = 0, (TRDIOC0) = 0										
		N-ch open drain output	—	1	0	—	—	0	0/1												
	ANI23	Analog input	—	×	1	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TxD2	Output	PIOR07 = 0	0/1	0	—	—	0	1	×	TRDIOA1 = 0, TKBO01 = 0, (TRDIOC0) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SO20	Output	PIOR07 = 0	0/1	0	—	—	0	1	×	TRDIOA1 = 0, TKBO01 = 0, (TRDIOC0) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TRDIOA1	Input	PIOR22 = 0 Note 1	×	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output		0	0	—	—	0	0	TxD2/SO20 = 1	TKBO01 = 0, (TRDIOC0) = 0										
	TKBO01	Output	—	0	0	—	—	0	0	TxD2/SO20 = 1	TRDIOA1 = 0, (TRDIOC0) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(TRDIOC0)	Input	PIOR27 = 1	×	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output		0	0	—	—	0	0	TxD2/SO20 = 1	TRDIOA1 = 0, TKBO01 = 0										

Note 1. This setting is only applicable in the 24- to 64-pin products.

Note 2. This setting is only applicable in the 30- to 64-pin products.

Table 7 - 5 Examples of Register and Output Latch Settings for Alternate Functions (5/22)

Pin Name	Function Used		PIORx	POMxx	PMCAxx	CCDE	CCSx	PMxx	Pxx	Alternate Function Output		20-pin	24-pin	25-pin	30-pin	32-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O								SAU and DALI	Other Than SAU and DALI										
P14	P14	Input	—	×	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	—	—	0	0/1	SDA20 = 1	TRDIOD0 = 0, TKBO10 = 0, VCOUT0 = 0, (SCLA0) = 0										
		N-ch open drain output	—	1	0	—	—	0	0/1												
	ANI24	Analog input	—	×	1	—	—	1	×	×	×	✓	—	✓	✓	✓	✓	✓	✓	✓	✓
	TRDIOD0	Input	PIOR26 = 0 Note 1	×	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output		0	0	—	—	0	0	SDA20 = 1	TKBO10 = 0, VCOUT0 = 0, (SCLA0) = 0										
	TKBO10	Output	—	0	0	—	—	0	0	SDA20 = 1	TRDIOD0 = 0, VCOUT0 = 0, (SCLA0) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	VCOUT0	Output	PIOR30 = 0 Note 2	0	0	—	—	0	0	SDA20 = 1	TRDIOD0 = 0, TKBO10 = 0, (SCLA0) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	RxD2	Input	PIOR07 = 0	×	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SI20	Input	PIOR07 = 0	×	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SDA20	I/O	PIOR07 = 0	1	0	—	—	0	1	×	TRDIOD0 = 0, TKBO10 = 0, VCOUT0 = 0, (SCLA0) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(SCLA0)	I/O	PIOR02 = 1	1	0	—	—	0	0	SDA20 = 1	TRDIOD0 = 0, TKBO10 = 0, VCOUT0 = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note 1. This setting is only applicable in the 24- to 64-pin products.

Note 2. This setting is only applicable in the 30- to 64-pin products.

Table 7 - 5 Examples of Register and Output Latch Settings for Alternate Functions (6/22)

Pin Name	Function Used		PIORx	POMxx	PMCAxx	CCDE	CCSx	PMxx	Pxx	Alternate Function Output		20-pin	24-pin	25-pin	30-pin	32-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O								SAU and DALI	Other Than SAU and DALI										
P15	P15	Input	—	×	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	—	—	0	0/1	SCK20/SCL20 = 1	PCLBUZ1 = 0 **Note 2**, TRDIOB0 = 0, TKBO11 = 0, VCOUT1 = 0, (SDAA0) = 0, (TO01) = 0 **Note 1**										
		N-ch open drain output	—	1	0	—	—	0	0/1												
	ANI25	Analog input	—	×	1	—	—	1	×	×	×	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	PCLBUZ1	Output	—	0	0	—	—	0	0	SCK20/SCL20 = 1	TRDIOB0 = 0, TKBO11 = 0, VCOUT1 = 0, (SDAA0) = 0, (TO01) = 0 **Note 1**	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
	SCK20	Input	PIOR07 = 0	×	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output		0/1	0	—	—	0	1	×	PCLBUZ1 = 0 **Note 2**, TRDIOB0 = 0, TKBO11 = 0, VCOUT1 = 0, (SDAA0) = 0, (TO01) = 0 **Note 1**	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SCL20	Output	PIOR07 = 0	0/1	0	—	—	0	1	×	PCLBUZ1 = 0 **Note 2**, TRDIOB0 = 0, TKBO11 = 0, VCOUT1 = 0, (SDAA0) = 0, (TO01) = 0 **Note 1**	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note 1. This setting is only applicable in the 20- to 32-pin products.

Note 2. This setting is only applicable in the 20- to 52-pin products.

Table 7 - 5 Examples of Register and Output Latch Settings for Alternate Functions (7/22)

Pin Name	Function Used		PIORx	POMxx	PMCAxx	CCDE	CCSx	PMxx	Pxx	Alternate Function Output		20-pin	24-pin	25-pin	30-pin	32-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O								SAU and DALI	Other Than SAU and DALI										
P15	TRDIOB0	Input	PIOR21 = 0 **Note 4**	×	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output		0	0	—	—	0	0	SCK20/SCL20 = 1	PCLBUZ1 = 0 **Note 2**, TKBO11 = 0, VCOUT1 = 0, (SDAA0) = 0, (TO01) = 0 **Note 1**										
	TKBO11	Output	—	0	0	—	—	0	0	SCK20/SCL20 = 1	PCLBUZ1 = 0 **Note 2**, TRDIOB0 = 0, VCOUT1 = 0, (SDAA0) = 0, (TO01) = 0 **Note 1**	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	VCOUT1	Output	PIOR31 = 0 **Note 3**	0	0	—	—	0	0	SCK20/SCL20 = 1	PCLBUZ1 = 0 **Note 2**, TRDIOB0 = 0, TKBO11 = 0, (SDAA0) = 0, (TO01) = 0 **Note 1**	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(SDAA0)	I/O	PIOR02 = 1	1	0	—	—	0	0	SCK20/SCL20 = 1	PCLBUZ1 = 0 **Note 2**, TRDIOB0 = 0, TKBO11 = 0, VCOUT1 = 0, (TO01) = 0 **Note 1**	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(TI01)	Input	PIOR35 = 1	×	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	—	—	—	—	—
	(TO01)	Output	PIOR35 = 1	0	0	—	—	0	0	SCK20/SCL20 = 1	PCLBUZ1 = 0 **Note 2**, TRDIOB0 = 0, TKBO11 = 0, VCOUT1 = 0, (SDAA0) = 00	✓	✓	✓	✓	✓	—	—	—	—	—

Note 1. This setting is only applicable in the 20- to 32-pin products.

Note 2. This setting is only applicable in the 20- to 52-pin products.

Note 3. This setting is only applicable in the 30- to 64-pin products.

Note 4. This setting is only applicable in the 32- to 64-pin products.

Table 7 - 5 Examples of Register and Output Latch Settings for Alternate Functions (8/22)

Pin Name	Function Used		PIORx	POMxx	PMCAxx	CCDE	CCSx	PMxx	Pxx	Alternate Function Output		20-pin	24-pin	25-pin	30-pin	32-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O								SAU and DALI	Other Than SAU and DALI										
P16	P16	Input	—	—	0	CCDE00 = 0	CCS0[2:0] = xxxB	1	×	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	0	CCDE00 = 0	CCS0[2:0] = xxxB	0	0/1	×	TO01 = 0, TRDIOC0 = 0, TKBO20 = 0, (TRGIOB) = 0, (TRDIOA1) = 0										
	ANI26	Analog input	—	—	1	CCDE00 = 0	CCS0[2:0] = xxxB	1	×	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	IVREF0	Analog input	—	—	1	CCDE00 = 0	CCS0[2:0] = xxxB	1	×	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	CCD00	Output	—	—	0	CCDE00 = 1	CCS0[2:0] = 001B to 011B	0	0	×	TO01 = 0, TRDIOC0 = 0, TKBO20 = 0, (TRGIOB) = 0, (TRDIOA1) = 0 **Note 1**	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TI01	Input	PIOR35 = 0	—	0	CCDE00 = 0	CCS0[2:0] = xxxB	1	×	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TO01	Output	PIOR35 = 0	—	0	CCDE00 = 0	CCS0[2:0] = xxxB	0	0	×	TRDIOC0 = 0, TKBO20 = 0, (TRGIOB) = 0, (TRDIOA1) = 0 **Note 1**	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	INTP5	Input	PIOR04 = 0 **Note 2**	—	0	CCDE00 = 0	CCS0[2:0] = xxxB	1	×	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TRDIOC0	Input	PIOR27 = 0	—	0	CCDE00 = 0	CCS0[2:0] = xxxB	1	×	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output		—	0	CCDE00 = 0	CCS0[2:0] = xxxB	0	0	×	TO01 = 0, TKBO20 = 0, (TRGIOB) = 0, (TRDIOA1) = 0										
	TKBO20	Output	—	—	0	CCDE00 = 0	CCS0[2:0] = xxxB	0	0	×	TO01 = 0, TRDIOC0 = 0, (TRGIOB) = 0, (TRDIOA1) = 0 **Note 1**	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(SI00)	Input	PIOR01 = 1	—	0	CCDE00 = 0	CCS0[2:0] = xxxB	1	×	×	×	—	—	—	—	—	—	—	—	—	✓
	(RxD0)	Input	PIOR01 = 1	—	0	CCDE00 = 0	CCS0[2:0] = xxxB	1	×	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(TRGIOB)	Input	PIOR20 = 1	×	0	CCDE00 = 0	CCS0[2:0] = xxxB	1	×	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output		—	0	CCDE00 = 0	CCS0[2:0] = xxxB	0	0	×	TO01 = 0, TRDIOC0 = 0, TKBO20 = 0, (TRDIOA1) = 0 **Note 1**										
	(TRDIOA1)	Input	PIOR22 = 1	—	0	CCDE00 = 0	CCS0[2:0] = xxxB	1	×	×	×	—	✓	✓	✓	✓	—	—	—	—	—
		Output		—	0	CCDE00 = 0	CCS0[2:0] = xxxB	0	0	×	TO01 = 0, TRDIOC0 = 0, TKBO20 = 0, (TRGIOB) = 0										

Note 1. This setting is only applicable in the 24- to 32-pin products.

Note 2. This setting is only applicable in the 64-pin products.

Table 7 - 5 Examples of Register and Output Latch Settings for Alternate Functions (9/22)

Pin Name	Function Used		PIORx	POMxx	PMCAxx	CCDE	CCSx	PMxx	Pxx	Alternate Function Output		20-pin	24-pin	25-pin	30-pin	32-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O								SAU and DALI	Other Than SAU and DALI										
P17	P17	Input	—	×	0	CCDE01 = 0	CCS0[2:0] = xxxB	1	×	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	CCDE01 = 0	CCS0[2:0] = xxxB	0	0/1	(TxD0) = 1 (SO00) = 1 Note 2	TO02 = 0, TRDIOA0 = 0, TKBO21 = 0, (TRGIOA) = 0, (TRDIOD0) = 0 Note 1										
		N-ch open drain output	—	1	0	CCDE01 = 0	CCS0[2:0] = xxxB	0	0/1												
	ANI27	Analog input	—	×	1	CCDE01 = 0	CCS0[2:0] = xxxB	1	×	×	×	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	CCD01	Output	—	0/1	0	CCDE01 = 1	CCS0[2:0] = 001B to 011B	0	0	×	TO02 = 0, TRDIOA0 = 0, TKBO21 = 0, (TRGIOA) = 0, (TRDIOD0) = 0 Note 1	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TI02	Input	PIOR36 = 0	×	0	CCDE01 = 0	CCS0[2:0] = xxxB	1	×	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TO02	Output	PIOR36 = 0	0	0	CCDE01 = 0	CCS0[2:0] = xxxB	0	0	(TxD0) = 1 (SO00) = 1 Note 2	TRDIOA0 = 0, TKBO21 = 0, (TRGIOA) = 0, (TRDIOD0) = 0 Note 1	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TRDIOA0	Input	—	×	0	CCDE01 = 0	CCS0[2:0] = xxxB	1	×	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	CCDE01 = 0	CCS0[2:0] = xxxB	0	0	(TxD0) = 1 (SO00) = 1 Note 2	TO02 = 0, TKBO21 = 0, (TRGIOA) = 0, (TRDIOD0) = 0 Note 1										
	TRDCLK	Input	—	0	0	CCDE01 = 0	CCS0[2:0] = xxxB	1	×	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TKBO21	Output	—	0	0	CCDE01 = 0	CCS0[2:0] = xxxB	0	0	(TxD0) = 1 (SO00) = 1 Note 2	TO02 = 0, TRDIOA0 = 0, (TRGIOA) = 0, (TRDIOD0) = 0 Note 1	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(TxD0)	Output	PIOR01 = 1	0/1	0	CCDE01 = 0	CCS0[2:0] = xxxB	0	1	×	TO02 = 0, TRDIOA0 = 0, TKBO21 = 0, (TRGIOA) = 0, (TRDIOD0) = 0 Note 1	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(SO00)	Output	PIOR01 = 1	0/1	0	CCDE01 = 0	CCS0[2:0] = xxxB	0	1	×	TO02 = 0, TRDIOA0 = 0, TKBO21 = 0, (TRGIOA) = 0, (TRDIOD0) = 0 Note 1	—	—	—	—	—	—	—	—	—	✓
	(TRGIOA)	Input	PIOR20 = 1	×	0	CCDE01 = 0	CCS0[2:0] = xxxB	1	×	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output		0	0	CCDE01 = 0	CCS0[2:0] = xxxB	0	0	(TxD0) = 1 (SO00) = 1 Note 2	TO02 = 0, TRDIOA0 = 0, TKBO21 = 0, (TRDIOD0) = 0 Note 1										
	(TRDIOD0)	Input	PIOR26 = 1	×	0	CCDE01 = 0	CCS0[2:0] = xxxB	1	×	×	×	—	✓	✓	✓	✓	—	—	—	—	—
		Output		0	0	CCDE01 = 0	CCS0[2:0] = xxxB	0	0	(TxD0) = 1 (SO00) = 1 Note 2	TO02 = 0, TRDIOA0 = 0, TKBO21 = 0, (TRGIOA) = 0										

Note 1. This setting is only applicable in the 24- to 32-pin products.

Note 2. This setting is only applicable in the 64-pin products.

Table 7 - 5 Examples of Register and Output Latch Settings for Alternate Functions (10/22)

Pin Name	Function Used		PIORx	PMCAxx	PMxx	Pxx	20-pin	24-pin	25-pin	30-pin	32-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O														
P20	P20	Input	—	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0/1										
	ANI0	Analog input	—	1	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	AVREFP	Reference voltage	—	1	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(INTP6)	Input	PIOR06 = 1	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P21	P21	Input	—	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0/1										
	ANI1	Analog input	—	1	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	AVREFM	Reference voltage	—	1	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(INTP7)	Input	PIOR06 = 1	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P22	P22	Input	—	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0/1										
	ANI2	Analog input	—	1	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	ANO0	Analog output	—	1	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	PGAI4	Analog input	—	1	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P23	P23	Input	—	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0/1										
	ANI3	Analog input	—	1	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	ANO1	Analog output	—	1	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	PGAGND	Analog input	—	1	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P24	P24	Input	—	0	1	×	—	—	—	—	—	✓	✓	✓	✓	✓
		Output	—	0	0	0/1										
	ANI4	Analog input	—	1	1	×	—	—	—	—	—	✓	✓	✓	✓	✓
P25	P25	Input	—	0	1	×	—	—	—	—	—	✓	✓	✓	✓	✓
		Output	—	0	0	0/1										
	ANI5	Analog input	—	1	1	×	—	—	—	—	—	✓	✓	✓	✓	✓
P26	P26	Input	—	0	1	×	—	—	—	—	—	✓	✓	✓	✓	✓
		Output	—	0	0	0/1										
	ANI6	Analog input	—	1	1	×	—	—	—	—	—	✓	✓	✓	✓	✓
P27	P27	Input	—	0	1	×	—	—	—	—	—	—	✓	✓	✓	✓
		Output	—	0	0	0/1										
	ANI7	Analog input	—	1	1	×	—	—	—	—	—	—	✓	✓	✓	✓

Table 7 - 5 Examples of Register and Output Latch Settings for Alternate Functions (11/22)

Pin Name	Function Used		PIORx	POMxx	PMxx	Pxx	Alternate Function Output		20-pin	24-pin	25-pin	30-pin	32-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O					SAU and DALI	Other Than SAU and DALI										
P30	P30	Input	—	×	1	×	×	×	—	—	—	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0/1	SCK00/SCL00 = 1, (TxD1) = 1Note 1	RTC1HZ = 0, TRJO0 = 0, (TRDIOB1) = 0 Note 2, (VCOUT1) = 0										
		N-ch open drain output	—	1	0	0/1												
	INTP3	Input	PIOR00 = 0 Note 3	×	1	×	×	×	—	—	—	✓	✓	✓	✓	✓	✓	✓
	RTC1HZ	Output	—	0	0	0	SCK00/SCL00 = 1, (TxD1) = 1Note 1	TRJO0 = 0, (TRDIOB1) = 0 Note 2, (VCOUT1) = 0	—	—	—	✓	✓	✓	✓	✓	✓	✓
	SCK00	Input	PIOR01 = 0	×	1	×	×	×	—	—	—	✓	✓	✓	✓	✓	✓	✓
		Output		0/1	0	1	×	RTC1HZ = 0, TRJO0 = 0, (TRDIOB1) = 0 Note 2, (VCOUT1) = 0	—	—	—	✓	✓	✓	✓	✓	✓	✓
	SCL00	Output	PIOR01 = 0	0/1	0	1	×	RTC1HZ = 0, TRJO0 = 0, (TRDIOB1) = 0 Note 2, (VCOUT1) = 0	—	—	—	✓	✓	✓	✓	✓	✓	✓
	TRJO0	Output	PIOR13, PIOR12 = 00B	0	0	0	SCK00/SCL00 = 1, (TxD1) = 1Note 1	RTC1HZ = 0, (TRDIOB1) = 0 Note 2, (VCOUT1) = 0	—	—	—	✓	✓	✓	✓	✓	✓	✓
	(TxD1)	Output	PIOR05 = 1	0/1	0	1	×	RTC1HZ = 0, TRJO0 = 0, (TRDIOB1) = 0 Note 2, (VCOUT1) = 0	—	—	—	✓	✓	—	—	—	—	—
	(TRDIOB1)	Input	PIOR23 = 1	×	1	×	×	×	—	—	—	✓	✓	✓	✓	—	—	—
		Output		0	0	0	SCK00/SCL00 = 1, (TxD1) = 1Note 1	RTC1HZ = 0, TRJO0 = 0, (VCOUT1) = 0										
	(VCOUT1)	Output	PIOR31 = 1	0	0	0	SCK00/SCL00 = 1, (TxD1) = 1Note 1	RTC1HZ = 0, TRJO0 = 0, (TRDIOB1) = 0	—	—	—	✓	✓	✓	✓	✓	✓	✓

Note 1. This setting is only applicable in the 30- to 32-pin products.

Note 2. This setting is only applicable in the 30- to 44-pin products.

Note 3. This setting is only applicable in the 64-pin products.

Table 7 - 5 Examples of Register and Output Latch Settings for Alternate Functions (12/22)

Pin Name	Function Used		PIORx	POMxx	PMxx	Pxx	Alternate Function Output		20-pin	24-pin	25-pin	30-pin	32-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O					SAU and DALI	Other Than SAU and DALI										
P31	P31	Input	—	—	1	×	—	×	—	—	—	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0/1	—	TO03 = 0, PLCBUZ0 = 0 **Note 1**, (PCLBUZ0) = 0 **Note 2**, (TRJIO0) = 0, (VCOUT0) = 0										
	TI03	Input	PIOR34 = 0	—	1	×	—	×	—	—	—	✓	✓	✓	✓	✓	✓	✓
	TO03	Output	PIOR34 = 0	—	0	0	—	PLCBUZ0 = 0 **Note 1**, (PCLBUZ0) = 0 **Note 2**, (TRJIO0) = 0, (VCOUT0) = 0	—	—	—	✓	✓	✓	✓	✓	✓	✓
	INTP4	Input	PIOR00 = 0 **Note 3**	—	1	×	—	×	—	—	—	✓	✓	✓	✓	✓	✓	✓
	(TRJIO0)	Input	PIOR11, PIOR10 = 01B	—	1	×	—	×	—	—	—	✓	✓	✓	✓	✓	✓	✓
		Output		—	0	0	—	TO03 = 0, PLCBUZ0 = 0 **Note 1**, (PCLBUZ0) = 0 **Note 2**, (VCOUT0) = 0	—	—	—	✓	✓	✓	✓	✓	✓	✓
	PCLBUZ0	Output	—	—	0	0	—	TO03 = 0, (PCLBUZ0) = 0 **Note 2**, (TRJIO0) = 0, (VCOUT0) = 0	—	—	—	✓	✓	✓	✓	—	—	—
	(PCLBUZ0)	Output	PIOR03 = 1	—	0	0	—	TO03 = 0, PLCBUZ0 = 0 **Note 1**, (TRJIO0) = 0, (VCOUT0) = 0	—	—	—	—	—	—	—	✓	✓	✓
	$\overline{\text{SSI00}}$	Input	—	—	1	×	—	×	—	—	—	✓	—	—	—	—	—	—
	(RxD1)	Input	PIOR05 = 1	—	1	×	—	×	—	—	—	✓	✓	—	—	—	—	—
	(VCOUT0)	Output	PIOR30 = 1	—	0	0	—	TO03 = 0, PLCBUZ0 = 0 **Note 1**, (PCLBUZ0) = 0 **Note 2**, (TRJIO0) = 0	—	—	—	✓	✓	✓	✓	✓	✓	✓

Note 1. This setting is only applicable in the 30- to 44-pin products.

Note 2. This setting is only applicable in the 48- to 64-pin products.

Note 3. This setting is only applicable in the 64-pin products.

Table 7 - 5 Examples of Register and Output Latch Settings for Alternate Functions (13/22)

Pin Name	Function Used		PIORx	POMxx	PMxx	Pxx	Alternate Function Output		20-pin	24-pin	25-pin	30-pin	32-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O					SAU and DALI	Other Than SAU and DALI										
P40	P40	Input	—	—	1	×	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0/1	—	—										
P41	P41	Input	—	—	1	×	—	×	—	—	—	—	—	—	✓	✓	✓	✓
		Output	—	—	0	0/1	—	(TRJIO0) = 0										
	(TRJIO0)	Input	PIOR11, PIOR10 = 10B	—	1	×	—	×	—	—	—	—	—	—	✓	✓	✓	✓
		Output		—	0	0	—	×										
P42	P42	Input	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	0	0/1	—	—	—	—	—	—	—	—	—	—	—	✓
	(INTP8)	Input	PIOR00 = 1	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓
P43	P43	Input	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	0	0/1	—	—	—	—	—	—	—	—	—	—	—	✓
	(INTP9)	Input	PIOR00 = 1	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓

Table 7 - 5 Examples of Register and Output Latch Settings for Alternate Functions (14/22)

Pin Name	Function Used		PIORx	POMxx	PMxx	Pxx	Alternate Function Output		20-pin	24-pin	25-pin	30-pin	32-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O					SAU and DALI	Other Than SAU and DALI										
P50	P50	Input	—	×	1	×	×	×	—	—	—	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0/1	SDA00 = 1	TRGIOA = 0, (TRJO0) = 0, (TRDIOC1) = 0 **Note 1**, (VCOUT3) = 0										
		N-ch open drain output	—	1	0	0/1												
	INTP1	Input	PIOR00 = 0 **Note 2**	×	1	×	×	×	—	—	—	✓	✓	✓	✓	✓	✓	✓
	SI00	Input	PIOR01 = 0	×	1	×	×	×	—	—	—	✓	✓	✓	✓	✓	✓	✓
	RxD0	Input	PIOR01 = 0, PIOR06 = 0	×	1	×	×	×	—	—	—	✓	✓	✓	✓	✓	✓	✓
	DALIRxD0	Input	PIOR37 = 0	×	1	×	×	×	—	—	—	✓	✓	✓	✓	✓	✓	✓
	SDA00	I/O	PIOR01 = 0	1	0	1	×	TRGIOA = 0, (TRJO0) = 0, (TRDIOC1) = 0 **Note 1**, (VCOUT3) = 0	—	—	—	✓	✓	✓	✓	✓	✓	✓
	TRGIOA	Input	PIOR20 = 0	×	1	×	×	×	—	—	—	✓	✓	✓	✓	✓	✓	✓
		Output		0	0	0	SDA00 = 1	(TRJO0) = 0, (TRDIOC1) = 0 **Note 1**, (VCOUT3) = 0										
	(TRJO0)	Output	PIOR13, PIOR12 = 01B	0	0	0	SDA00 = 1	TRGIOA = 0, (TRDIOC1) = 0 **Note 1**, (VCOUT3) = 0	—	—	—	✓	✓	✓	✓	✓	✓	✓
	(TI03)	Input	PIOR34 = 1	×	1	×	×	×	—	—	—	✓	✓	✓	✓	✓	✓	✓
	(TRDIOC1)	Input	PIOR24 = 1	×	1	×	×	×	—	—	—	✓	✓	—	—	—	—	—
		Output		0	0	0	SDA00 = 1	TRGIOA = 0, (TRJO0) = 0, (VCOUT3) = 0										
	(VCOUT3)	Output	PIOR33 = 1	0	0	0	SDA00 = 1	TRGIOA = 0, (TRJO0) = 0, (TRDIOC1) = 0 **Note 1**	—	—	—	✓	✓	✓	✓	✓	✓	✓

Note 1. This setting is only applicable in the 30- to 32-pin products.

Note 2. This setting is only applicable in the 64-pin products.

Table 7 - 5 Examples of Register and Output Latch Settings for Alternate Functions (15/22)

Pin Name	Function Used		PIORx	POMxx	PMxx	Pxx	Alternate Function Output		20-pin	24-pin	25-pin	30-pin	32-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O					SAU and DALI	Other Than SAU and DALI										
P51	P51	Input	—	×	1	×	×	×	—	—	—	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0/1	TxD0/SO00 = 1, DALITxD0 = 1	TRGIOB = 0, (TRDIOD1) = 0 **Note 1**, (VCOUT2) = 0										
		N-ch open drain output	—	1	0	0/1												
	INTP2	Input	PIOR00 = 0 **Note 2**	×	1	×	×	×	—	—	—	✓	✓	✓	✓	✓	✓	✓
	SO00	Output	PIOR01 = 0	0/1	0	1	×	TRGIOB = 0, (TRDIOD1) = 0 **Note 1**, (VCOUT2) = 0	—	—	—	✓	✓	✓	✓	✓	✓	✓
	TxD0	Output	PIOR01 = 0, PIOR06 = 0	0/1	0	1	×	TRGIOB = 0, (TRDIOD1) = 0 **Note 1**, (VCOUT2) = 0	—	—	—	✓	✓	✓	✓	✓	✓	✓
	DALITxD0	Output	PIOR37 = 0	0/1	0	1	×	TRGIOB = 0, (TRDIOD1) = 0 **Note 1**, (VCOUT2) = 0	—	—	—	✓	✓	✓	✓	✓	✓	✓
	TRGIOB	Input	PIOR20 = 0	×	1	×	×	×	—	—	—	✓	✓	✓	✓	✓	✓	✓
		Output		0	0	0	TxD0/SO00 = 1, DALITxD0 = 1	(TRDIOD1) = 0 **Note 1**, (VCOUT2) = 0										
	(TRDIOD1)	Input	PIOR25 = 1	×	1	×	×	×	—	—	—	✓	✓	✓	✓	—	—	—
		Output		0	0	0	TxD0/SO00 = 1, DALITxD0 = 1	TRGIOB = 0, (VCOUT2) = 0										
	(VCOUT2)	Output	PIOR32 = 1	0	0	0	TxD0/SO00 = 1, DALITxD0 = 1	TRGIOB = 0, (TRDIOD1) = 0 **Note 1**	—	—	—	✓	✓	✓	✓	✓	✓	✓
P52	P52	Input	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	0	0/1	—	—										
	(INTP1)	Input	PIOR00 = 1	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓
P53	P53	Input	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	0	0/1	—	—										
	(INTP2)	Input	PIOR00 = 1	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓
P54	P54	Input	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	0	0/1	—	—										
	(INTP3)	Input	PIOR00 = 1	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓
P55	P55	Input	—	×	1	×	×	×	—	—	—	—	—	—	—	—	—	✓
		Output	—	0	0	0/1	(SCK00) = 1	(PCLBUZ1) = 0										
		N-ch open drain output	—	1	0	0/1												
	(INTP4)	Input	PIOR00 = 1	×	1	×	×	×	—	—	—	—	—	—	—	—	—	✓
	(PCLBUZ1)	Output	PIOR04 = 1	0	0	0	×	×	—	—	—	—	—	—	—	—	—	✓
	(SCK00)	Input	PIOR01 = 1	×	1	×	×	×	—	—	—	—	—	—	—	—	—	✓
		Output		0/1	0	1	×	(PCLBUZ1) = 0	—	—	—	—	—	—	—	—	—	✓

Note 1. This setting is only applicable in the 30- to 44-pin products.

Note 2. This setting is only applicable in the 64-pin products.

Table 7 - 5 Examples of Register and Output Latch Settings for Alternate Functions (16/22)

Pin Name	Function Used		PIORx	CCDE	CCSx	PMxx	Pxx	Alternate Function Output		20-pin	24-pin	25-pin	30-pin	32-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O						SAU and DALI	Other Than SAU and DALI										
P60	P60	Input	—	CCDE04 = 0	CCS4[2:0] = 000B	1	×	—	×	—	—	—	✓	✓	✓	✓	✓	✓	✓
		N-ch open drain output [withstand voltage of 6 V]	—	CCDE04 = 0	CCS4[2:0] = 000B	0	0/1	—	SCLA0 = 0										
	CCD04	Output	—	CCDE04 = 1	CCS4[2:0] = 001B to 100B	0	0	—	SCLA0 = 0	—	—	—	✓	✓	✓	✓	✓	✓	✓
	SCLA0	I/O	PIOR02 = 0	CCDE04 = 0	CCS4[2:0] = 000B	0	0	—	×	—	—	—	✓	✓	✓	✓	✓	✓	✓
P61	P61	Input	—	CCDE05 = 0	CCS5[2:0] = 000B	1	×	—	×	—	—	—	✓	✓	✓	✓	✓	✓	✓
		N-ch open drain output [withstand voltage of 6 V]	—	CCDE05 = 0	CCS5[2:0] = 000B	0	0/1	—	SDAA0 = 0										
	CCD05	Output	—	CCDE05 = 1	CCS5[2:0] = 001B to 100B	0	0	—	SDAA0 = 0	—	—	—	✓	✓	✓	✓	✓	✓	✓
	SDAA0	I/O	PIOR02 = 0	CCDE05 = 0	CCS5[2:0] = 000B	0	0	—	×	—	—	—	✓	✓	✓	✓	✓	✓	✓
P62	P62	Input	—	CCDE02 = 0	CCS0[2:0] = 000B	1	×	—	×	—	—	—	—	✓	✓	✓	✓	✓	✓
		Output	—	CCDE02 = 0	CCS0[2:0] = 000B	0	0/1	—	×										
	CCD02	Output	—	CCDE02 = 1	CCS0[2:0] = 001B to 100B	0	0	—	×	—	—	—	—	✓	✓	✓	✓	✓	✓
	$\overline{\text{SSI00}}$	Input	—	CCDE02 = 0	CCS0[2:0] = 000B	1	×	—	×	—	—	—	—	✓	✓	✓	✓	✓	✓
P63	P63	Input	—	CCDE03 = 0	CCS0[2:0] = 000B	1	×	—	×	—	—	—	—	—	—	✓	✓	✓	✓
		Output	—	CCDE03 = 0	CCS0[2:0] = 000B	0	0/1	—	×										
	CCD03	Output	—	CCDE03 = 1	CCS0[2:0] = 001B to 100B	0	0	—	×	—	—	—	—	—	—	✓	✓	✓	✓

Table 7 - 5 Examples of Register and Output Latch Settings for Alternate Functions (17/22)

Pin Name	Function Used		PIORx	POMxx	PMxx	Pxx	Alternate Function Output		20-pin	24-pin	25-pin	30-pin	32-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O					SAU and DALI	Other Than SAU and DALI										
P70	P70	Input	—	—	1	×	×	×	—	—	—	—	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0/1	SCK21/SCL21 = 1Note 1	(TRDIOB0) = 0										
	KR0	Input	—	—	1	×	×	×	—	—	—	—	—	✓	✓	✓	✓	✓
	SCK21	Input	—	—	1	×	×	×	—	—	—	—	—	✓	✓	✓	✓	✓
		Output	—	—	0	1	×	(TRDIOB0) = 0	—	—	—	—	—	✓	✓	✓	✓	✓
	SCL21	Output	—	—	0	1	×	(TRDIOB0) = 0	—	—	—	—	—	✓	✓	✓	✓	✓
	(TRDIOB0)	Input	PIOR21 = 1	—	1	×	×	×	—	—	—	—	✓	✓	✓	✓	✓	✓
		Output		—	0	0	SCK21/SCL21 = 1Note 1	×										
P71	P71	Input	—	×	1	×	×	×	—	—	—	—	—	✓	✓	✓	✓	✓
		Output	—	0	0	0/1	SDA21 = 1	(TO01) = 0 (TRDIOD0) = 0										
		N-ch open drain output	—	1	0	0/1												
	KR1	Input	—	×	1	×	×	×	—	—	—	—	—	✓	✓	✓	✓	✓
	SI21	Input	—	×	1	×	×	×	—	—	—	—	—	✓	✓	✓	✓	✓
	SDA21	I/O	—	1	0	1	×	(TO01) = 0 (TRDIOD0) = 0	—	—	—	—	—	✓	✓	✓	✓	✓
	(TI01)	Input	PIOR35 = 1	×	1	×	×	×	—	—	—	—	—	✓	✓	✓	✓	✓
	(TO01)	Output	PIOR35 = 1	0	0	0	SDA21 = 1	(TRDIOD0) = 0	—	—	—	—	—	✓	✓	✓	✓	✓
	(TRDIOD0)	Input	PIOR26 = 1	×	1	×	×	×	—	—	—	—	—	✓	✓	✓	✓	✓
		Output		0	0	0	SDA21 = 1	(TO01) = 0										
P72	P72	Input	—	×	1	×	×	×	—	—	—	—	—	✓	✓	✓	✓	✓
		Output	—	0	0	0/1	SO21 = 1	(TRDIOA1) = 0										
		N-ch open drain output	—	1	0	0/1												
	KR2	Input	—	×	1	×	×	×	—	—	—	—	—	✓	✓	✓	✓	✓
	SO21	Output	—	0/1	0	1	×	(TRDIOA1) = 0	—	—	—	—	—	✓	✓	✓	✓	✓
	(TxD1)	Output	PIOR05 = 1	0/1	0	1	SO21 = 1	×	—	—	—	—	—	✓	✓	✓	✓	✓
	(TRDIOA1)	Input	PIOR22 = 1	×	1	×	×	×	—	—	—	—	—	✓	✓	✓	✓	✓
		Output		0	0	0	×	×										
P73	P73	Input	—	×	1	×	×	×	—	—	—	—	—	✓	✓	✓	✓	✓
		Output	—	0	0	0/1	SO01 = 1Note 2	(TRDIOC1) = 0										
		N-ch open drain output	—	1	0	0/1												
	KR3	Input	—	×	1	×	×	×	—	—	—	—	—	✓	✓	✓	✓	✓
	SO01	Output	—	0/1	0	1	×	(TRDIOC1) = 0	—	—	—	—	—	—	—	✓	✓	✓
	(RxD1)	Input	PIOR05 = 1	×	1	×	×	×	—	—	—	—	—	✓	✓	✓	✓	✓
	(TRDIOC1)	Input	PIOR24 = 1	×	1	×	×	×	—	—	—	—	—	✓	✓	✓	✓	✓
		Output		0	0	0	SO01 = 1Note 2	×										

Note 1. This setting is only applicable in the 40- to 64-pin products.

Note 2. This setting is only applicable in the 48- to 64-pin products.

Table 7 - 5 Examples of Register and Output Latch Settings for Alternate Functions (18/22)

Pin Name	Function Used		PIORx	POMxx	PMxx	Pxx	Alternate Function Output		20-pin	24-pin	25-pin	30-pin	32-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O					SAU and DALI	Other Than SAU and DALI										
P74	P74	Input	—	×	1	×	×	×	—	—	—	—	—	—	—	✓	✓	✓
		Output	—	0	0	0/1	SDA01 = 1	(TRDIOB1) = 0										
		N-ch open drain output	—	1	0	0/1												
	KR4	Input	—	×	1	×	×	×	—	—	—	—	—	—	—	✓	✓	✓
	INTP8	Input	PIOR00 = 0Note	×	1	×	×	×	—	—	—	—	—	—	—	✓	✓	✓
	SI01	Input	—	×	1	×	×	×	—	—	—	—	—	—	—	✓	✓	✓
	SDA01	I/O	—	1	0	1	×	(TRDIOB1) = 0	—	—	—	—	—	—	—	✓	✓	✓
	(TRDIOB1)	Input	PIOR23 = 1	×	1	×	×	×	—	—	—	—	—	—	—	✓	✓	✓
		Output		0	0	0	SDA01 = 1	×										
P75	P75	Input	—	—	1	×	×	×	—	—	—	—	—	—	—	✓	✓	✓
		Output	—	—	0	0/1	SCK01/SCL01 = 1	(TRDIOD1) = 0										
	KR5	Input	—	—	1	×	×	×	—	—	—	—	—	—	—	✓	✓	✓
	INTP9	Input	PIOR00 = 0Note	—	1	×	×	×	—	—	—	—	—	—	—	✓	✓	✓
	SCK01	Input	—	—	1	×	×	×	—	—	—	—	—	—	—	✓	✓	✓
		Output	—	—	0	1	×	(TRDIOD1) = 0	—	—	—	—	—	—	—	✓	✓	✓
	SCL01	Output	—	—	0	1	×	(TRDIOD1) = 0	—	—	—	—	—	—	—	✓	✓	✓
	(TRDIOD1)	Input	PIOR25 = 1	—	1	×	×	×	—	—	—	—	—	—	—	✓	✓	✓
		Output	—	—	0	0	SCK01/SCL01 = 1	×										
P76	P76	Input	—	—	1	×	—	—	—	—	—	—	—	—	—	—	✓	✓
		Output	—	—	0	0/1	—	—										
	KR6	Input	—	—	1	×	—	—	—	—	—	—	—	—	—	—	✓	✓
	INTP10	Input	PIOR01 = 0	—	1	×	—	—	—	—	—	—	—	—	—	—	✓	✓
	(RxD2)	Input	PIOR07 = 1	—	1	×	—	—	—	—	—	—	—	—	—	—	✓	✓
P77	P77	Input	—	—	1	×	×	—	—	—	—	—	—	—	—	—	✓	✓
		Output	—	—	0	0/1	(TxD2) = 1	—										
	KR7	Input	—	—	1	×	×	—	—	—	—	—	—	—	—	—	✓	✓
	INTP11	Input	PIOR01 = 0	—	1	×	×	—	—	—	—	—	—	—	—	—	✓	✓
	(TxD2)	Output	PIOR07 = 1	—	0	1	×	—	—	—	—	—	—	—	—	—	✓	✓

Note This setting is only applicable in the 64-pin products.

Table 7 - 5 Examples of Register and Output Latch Settings for Alternate Functions (19/22)

<R>

Pin Name	Function Used		PIORx	POMxx	PMCAxx	PMxx	Pxx	Alternate Function Output		20-pin	24-pin	25-pin	30-pin	32-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O						SAU and DALI	Other Than SAU and DALI										
P120	P120	Input	—	—	0	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0	0/1	—	TO02 = 0Note										
	ANI19	Analog input	—	—	1	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	IVCMP0	Analog input	—	—	1	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	PGAI0	Analog input	—	—	1	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(TI02)	Input	PIOR36 = 1Note	—	0	1	×	—	×	✓	✓	✓	✓	✓	—	—	—	—	—
	(TO02)	Output	PIOR36 = 1Note	—	0	0	0	—	×	✓	✓	✓	✓	✓	—	—	—	—	—
	TRGIDZ	Input	—	—	0	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TRGTRG	Input	—	—	0	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note This setting is only applicable in the 20- to 32-pin products.

Table 7 - 5 Examples of Register and Output Latch Settings for Alternate Functions (20/22)

Pin Name	Function Used		PIORx	CMC		PMxx	Pxx	20-pin	24-pin	25-pin	30-pin	32-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O		EXCLK, OSCSEL, EXCLKS, OSCSELS	XTSEL												
P121	P121	Input	—	00xxB/10xxB/11xxB	0	1	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
				xx00B/xx10B/xx11B	1Note												
		Output	—	00xxB/10xxB/11xxB	0	0	0/1										
				xx00B/xx10B/xx11B	1Note												
	VBAT	Input	—	00xxB/10xxB/11xxB	0	0	1	—	—	—	—	—	✓	✓	✓	✓	✓
	X1	—	—	01xxB	0	1	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	XT1	—	—	xx01B	1	1	×	—	✓	✓	✓	✓	—	—	—	—	—
	INTP21	Input	—	00xxB/10xxB/11xxB	0	1	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
				xx00B/xx10B/xx11B	1Note												
	(TI02)	Input	PIOR36 = 1	00xxB/10xxB/11xxB	0	1	×	—	—	—	—	—	✓	✓	✓	✓	✓
				xx00B/xx10B/xx11B	1Note												
	(TO02)	Output	PIOR36 = 1	00xxB/10xxB/11xxB	0	0	0	—	—	—	—	—	✓	✓	✓	✓	✓
				xx00B/xx10B/xx11B	1Note												
P122	P122	Input	—	00xxB/10xxB	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
				xx00B/xx10B	1Note												
		Output	—	00xxB/10xxB	0	0	0/1										
				xx00B/xx10B	1Note												
	X2	—	—	01xxB	0	1	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	XT2	—	—	xx01B	1	1	×	—	✓	✓	✓	✓	—	—	—	—	—
	EXCLK	Input	—	11xxB	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	EXCLKS	Input	—	xx11B	1	1	×	✓	✓	✓	✓	✓	—	—	—	—	—
	INTP20	Input	—	00xxB/10xxB	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
				xx00B/xx10B	1Note												
	(TO00)	Output	PIOR36 = 1	00xxB/10xxB	0	0	0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
				xx00B/xx10B	1Note												
P123	P123	Input	—	xx00B/xx10B/xx11B	0	—	×	—	—	—	—	—	✓	✓	✓	✓	✓
	XT1	—	—	xx01B	0	—	×	—	—	—	—	—	✓	✓	✓	✓	✓
P124	P124	Input	—	xx00B/xx10B	0	—	×	—	—	—	—	—	✓	✓	✓	✓	✓
	XT2	—	—	xx01B	0	—	×	—	—	—	—	—	✓	✓	✓	✓	✓
	EXCLKS	Input	—	xx11B	0	—	×	—	—	—	—	—	✓	✓	✓	✓	✓

Note This setting is only applicable in the 24- to 32-pin products.

Table 7 - 5 Examples of Register and Output Latch Settings for Alternate Functions (21/22)

Pin Name	Function Used		PMx	Pxx	20-pin	24-pin	25-pin	30-pin	32-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O												
P130	P130	Output	—	0/1	—	—	✓	—	—	—	—	✓	✓	✓
P137	P137	Input	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	INTP0	Input	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 7 - 5 Examples of Register and Output Latch Settings for Alternate Functions (22/22)

Pin Name	Function Used		PIORx	POMxx	PMCAxx	PMxx	Pxx	Alternate Function Output		20-pin	24-pin	25-pin	30-pin	32-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O						SAU and DALI	Other Than SAU and DALI										
P140	P140	Input	—	—	—	1	×	—	×	—	—	—	—	—	—	—	✓	✓	✓
		Output	—	—	—	0	0/1	—	PCLBUZ0 = 0										
	PCLBUZ0	Output	PIOR03 = 0	—	—	0	0	—	×	—	—	—	—	—	—	—	✓	✓	✓
	INTP6	Input	PIOR06 = 0	—	—	1	×	—	×	—	—	—	—	—	—	—	✓	✓	✓
P141	P141	Input	—	—	—	1	×	—	×	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	—	0	0/1	—	PCLBUZ1 = 0										
	PCLBUZ1	Output	PIOR04 = 0	—	—	0	0	—	×	—	—	—	—	—	—	—	—	—	✓
	INTP7	Input	PIOR06 = 0	—	—	1	×	—	×	—	—	—	—	—	—	—	—	—	✓
P146	P146	Input	—	—	0	1	×	—	—	—	—	—	—	—	—	✓	✓	✓	✓
		Output	—	—	0	0	0/1	—	—										
	ANI28	Analog input	—	—	1	1	×	—	—	—	—	—	—	—	—	✓	✓	✓	✓
P147	P147	Input	—	—	0	1	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0	0/1	—	—										
	ANI18	Analog input	—	—	1	1	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	ANO2	Analog output	—	—	1	1	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	IVCMP3	Analog input	—	—	1	1	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	PGAI3	Analog input	—	—	1	1	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓

7.6 Cautions When Using Port Function

7.6.1 Cautions on 1-bit manipulation instruction for port register xx (Pxx)

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 before switching a port from input mode to output mode.

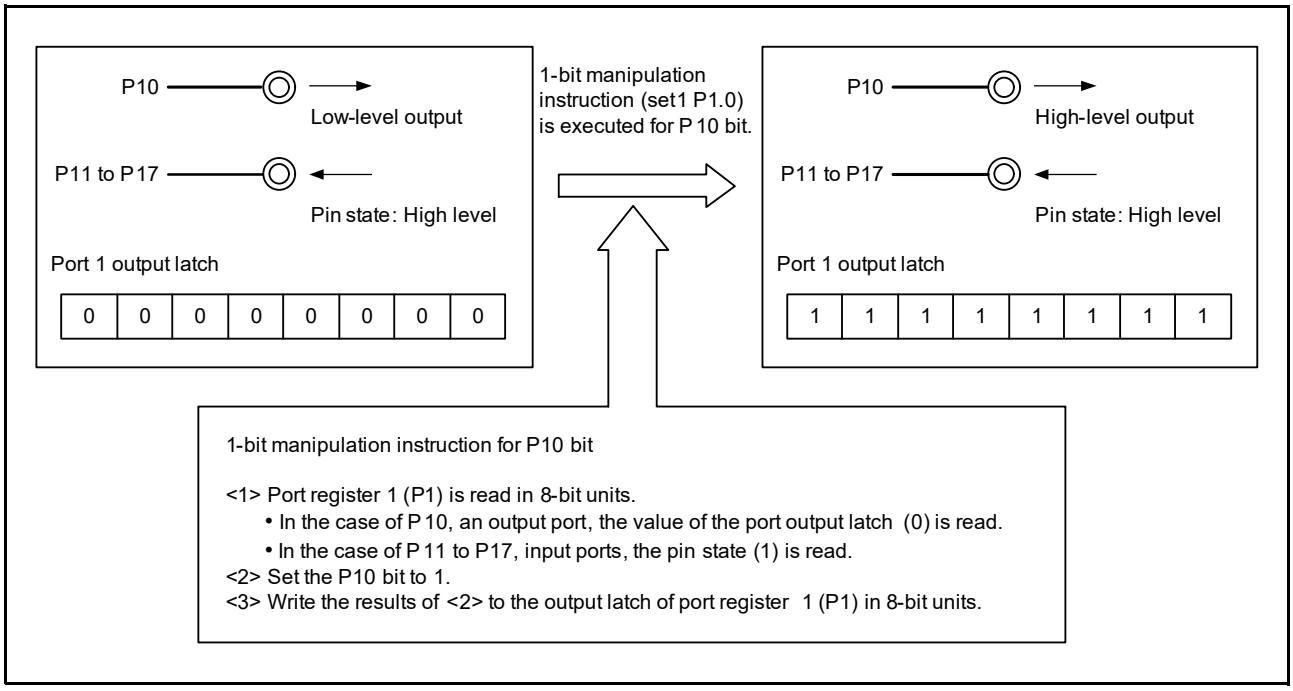
<Example> When P10 is an output port, P11 to P17 are input ports (all pins are at the high level), and the output latch value of port 1 is 00H, if the output of output port P10 is changed from low level to high level via a 1-bit manipulation instruction, the output latch value of port 1 becomes FFH.

Explanation: The targets of writing to and reading from the Pxx register of a port whose PMnm bit is 1 are the output latch and pin state, respectively.
A 1-bit manipulation instruction is executed in the following order in the RL78/G24.

<1> The Pxx register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pxx 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 states of P11 to P17, which are input ports, are read. If the states 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 7 - 16 One-bit Manipulation Instruction (P10)



7.6.2 Notes on specifying the pin settings

For an output pin to which multiple functions are assigned, the output of the unused alternate functions must be set to the initial state of the pin so as to prevent conflicting outputs. This also applies to the functions assigned by using peripheral I/O redirection registers x (PIORx). For details about the alternate function output, see **7.5 Register Settings When Using Alternate Function**.
No specific setting is required for input pins because the output of their alternate functions is disabled (the buffer output is Hi-Z).

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

Section 8 Operation State Control

The operating voltage, operating timing, and operating current of the internal circuit are optimized using flash operation modes. Select an appropriate flash operation mode in accord with the operating voltage range and clock frequencies of the MCU.
The flash operation mode set by the option byte is selected for operation immediately after a reset is released. The mode can be changed by setting of the relevant register.

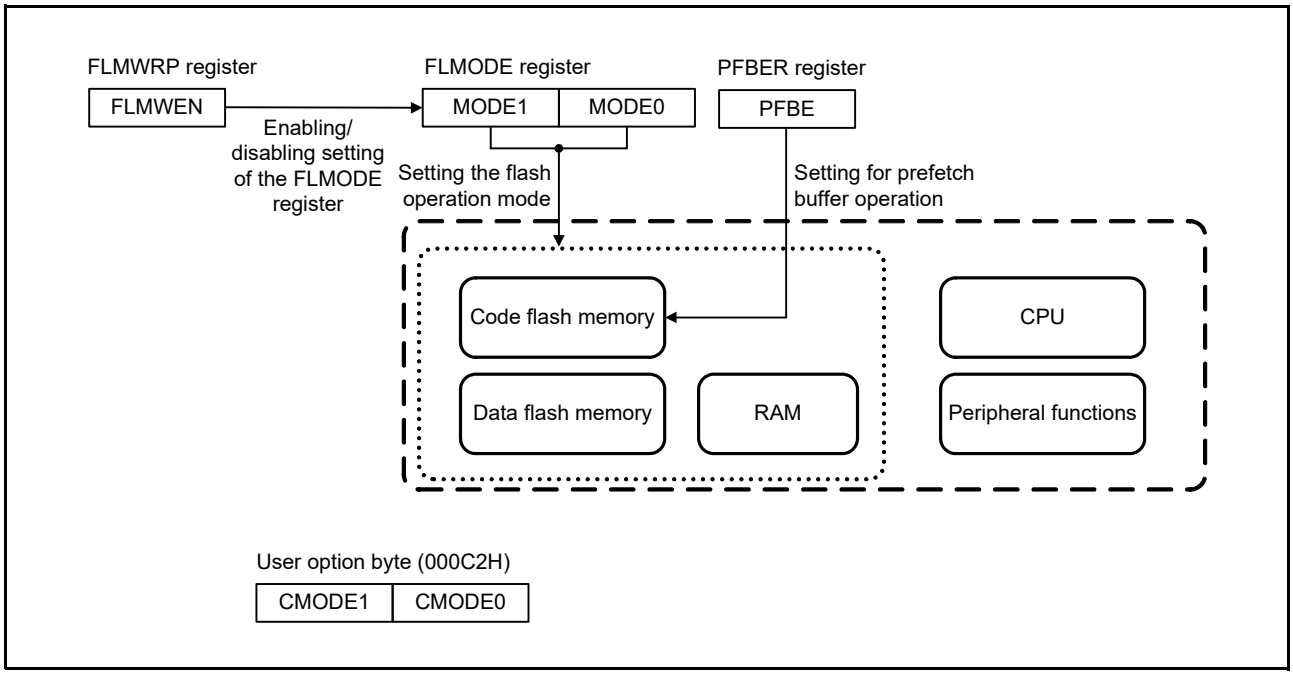
8.1 Configuration of Operation State Control

Operation state control is supported by the following hardware blocks.

Table 8 - 1 Configuration of Operation State Control

Item	Configuration
Option byte	• Address of the user option byte: 000C2H
Control registers	• Flash operating mode select register (FLMODE) • Flash operating mode protect register (FLMWRP) • Prefetch buffer enable register (PFBER)

Figure 8 - 1 Block Diagram of Operation State Control



There are the following five flash operation modes.

- HS (high-speed main) mode (with prefetching off)
- HS (high-speed main) mode (with prefetching on)
- LS (low-speed main) mode
- LP (low-power main) mode
- SUB mode

The MCU can be operated efficiently by setting these flash operation modes according to MCU operating conditions.
Table 8 - 2 lists the features of each flash operation mode.

Table 8 - 2 Features of Each Flash Operation Mode

Flash Operation Mode	Recommended Operating Range		Description
HS (high-speed main) mode (with prefetching off)	1.6 V to 1.8 V	1 MHz to 4 MHz (Rewriting of the flash memory is not possible.)	High-speed CPU operation (at 32 MHz (max.)) is possible in this mode. Suitable when CPU processing capacity is required.
	1.8 V to 5.5 V	1 MHz to 32 MHz	
HS (high-speed main) mode (with prefetching on)	2.4 V to 5.5 V	1 MHz to 48 MHz	High-speed CPU operation (at 48 MHz (max.)) is possible in this mode. Actually using this mode requires enabling the prefetch buffer.
LS (low-speed main) mode	1.6 V to 1.8 V	1 MHz to 4 MHz (Rewriting of the flash memory is not possible.)	The operating current and CPU operation processing (at 24 MHz (max.)) are well-balanced in this mode.
	1.8 V to 5.5 V	1 MHz to 24 MHz	
LP (low-power main) mode	1.6 V to 5.5 V	1 MHz to 2 MHz (Rewriting of the flash memory is not possible.)	The CPU operates at 1 MHz to 2 MHz in this mode. Low operating current is realized at 1 MHz to 2 MHz.
SUB mode	1.6 V to 5.5 V	32.768 kHz (Rewriting of the flash memory is not possible.)	CPU operation is driven by the subsystem clock.**Note** This mode enables low-power operation.

Note The subsystem clock can be derived from the subsystem clock X (f_{SX}) or the low-speed on-chip oscillator clock (f_{IL}).

8.2 Registers to Control Operation State Control

The following registers are used to control operation state control.

- Flash operating mode select register (FLMODE)
- Flash operating mode protect register (FLMWRP)
- Prefetch buffer enable register (PFBER)

8.2.1 Flash operating mode select register (FLMODE)

The FLMODE register is an 8-bit register for controlling the flash operation modes. This register can be set by a 1-bit or 8-bit memory manipulation instruction. Note that the value of this register cannot be changed when the FLMWEN bit in the flash operating mode protect register (FLMWRP) is 0. The setting of the MODE[1:0] bits is updated to that of the CMODE[1:0] bits in the user option byte at address 000C2H following a reset.

Figure 8 - 2 Format of Flash Operating Mode Select Register (FLMODE)

Address: F00AAH
After reset: 40H/80H/C0H**Note**
R/W: R/W

Symbol	<7>	<6>	5	4	3	2	1	0
FLMODE	MODE1	MODE0	0	0	0	0	0	0

MODE1	MODE0	Selection of flash operation mode
0	0	Setting prohibited
0	1	LP (low-power main) mode (Selectable from LS (low-speed main) mode with 1 MHz ≤ f_{CLK} ≤ 2 MHz)
1	0	LS (low-speed main) mode (Selectable from HS (high-speed main) mode with 1 MHz ≤ f_{CLK} ≤ 24 MHz or from LP (low-power main) mode)
1	1	HS (high-speed main) mode (Selectable from LS (low-speed main) mode)

Note The initial value of the FLMODE register is set to the value of the MODE[1:0] bits updated with the set value of the CMODE[1:0] bits in the user option byte (address: 000C2H).

Caution 1. The value of the FLMODE register can be changed when the FLMWEN bit in the flash operating mode protect register (FLMWRP) is 1. After the value of the FLMODE register is changed, set the FLMWEN bit to 0.

Caution 2. Operation is in SUB mode when the setting of the CSS bit in the system clock control register (CKC) is 1 (operation of the CPU and peripheral functions is driven by the subsystem clock) regardless of the setting of the MODE[1:0] bits.

Caution 3. Do not change the value of the MODE[1:0] bits by using the DTC.

Caution 4. When changing the flash operation mode, make sure that operation is possible within the usable voltage range and operating frequency range for the flash operation mode following the transition before changing the mode.

(Cautions are listed on the next page.)

Caution 5. When the flash operation mode is changed by the MODE[1:0] bits, the CPU is placed in the wait state for the following time until the mode changes. Interrupt requests are held pending during this wait period.

Times for switching between the flash operation modes

Switching between the Flash Operation Modes	Transition Time
LS (low-speed main) mode → HS (high-speed main) mode	225 cycles**Note**
LP (low-power main) mode → LS (low-speed main) mode	10 cycles**Note**
LS (low-speed main) mode → LP (low-power main) mode	10 cycles**Note**
HS (high-speed main) mode → LS (low-speed main) mode	30 cycles**Note**

Note Cycles of the CPU/peripheral hardware clock (f_{CLK})

Caution 6. When rewriting the FLMODE register, insert at least one clock cycle of the CPU/peripheral hardware clock (f_{CLK}) after having rewritten the FLMODE register. Do not write to the FLMODE register successively.

Caution 7. Do not change the FLMODE register while rewriting the flash memory.

Caution 8. Before changing the flash operation mode, set the DFLEN bit in the data flash control register (DFLCTL) to 1 to enable access to the data flash memory.

Caution 9. Before rewriting the contents of the code flash memory or data flash memory area by self-programming, be sure to place the MCU in HS (high-speed main) or LS (low-speed main) mode.

8.2.2 Flash operating mode protect register (FLMWRP)

The FLMWRP is an 8-bit register for controlling access to the flash operating mode select register. This register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 8 - 3 Format of Flash Operating Mode Protect Register (FLMWRP)

Address: F00ABH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	<0>
FLMWRP	0	0	0	0	0	0	0	FLMWEN

FLMWEN	Control of flash operating mode select register (FLMODE)
0	Rewriting the FLMODE register is disabled.
1	Rewriting the FLMODE register is enabled.

8.2.3 Prefetch buffer enable register (PFBER)

The PFBER is an 8-bit register for enabling or disabling operation of the prefetch buffer. This register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 8 - 4 Format of Prefetch Buffer Enable Register (PFBER)

Address: F007CH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	<0>
PFBER	0	0	0	0	0	0	0	PFBE

PFBE	Control by the prefetch buffer enable register (PFBER)
0	Disables the prefetch buffer.
1	Enables the prefetch buffer.

Caution 1. Only enable the prefetch buffer when the HS (high-speed main) mode has been selected.
Caution 2. After having enabled the prefetch buffer, change the voltage and operating frequency while ensuring that they are within the usable range for the flash operation mode following the transition.
Caution 3. Disable the prefetch buffer before a transition to the STOP mode.
Caution 4. Changing the setting to enable or disable the prefetch buffer leads to the CPU being placed in the wait state for 2 clock cycles.
Caution 5. Change the setting to enable or disable the prefetch buffer after having stopped supply of the clock signal to the peripheral functions.
Caution 6. When fixing of the flash read protection settings is enabled, writing to the PFBER register is ignored and the PFBE bit is fixed to 0 (the prefetch buffer is disabled).

8.3 Initial Setting of Flash Operation Modes

The user option byte (000C2H) is used to set the initial state of flash operation mode and the high-speed on-chip oscillator after a reset is released.

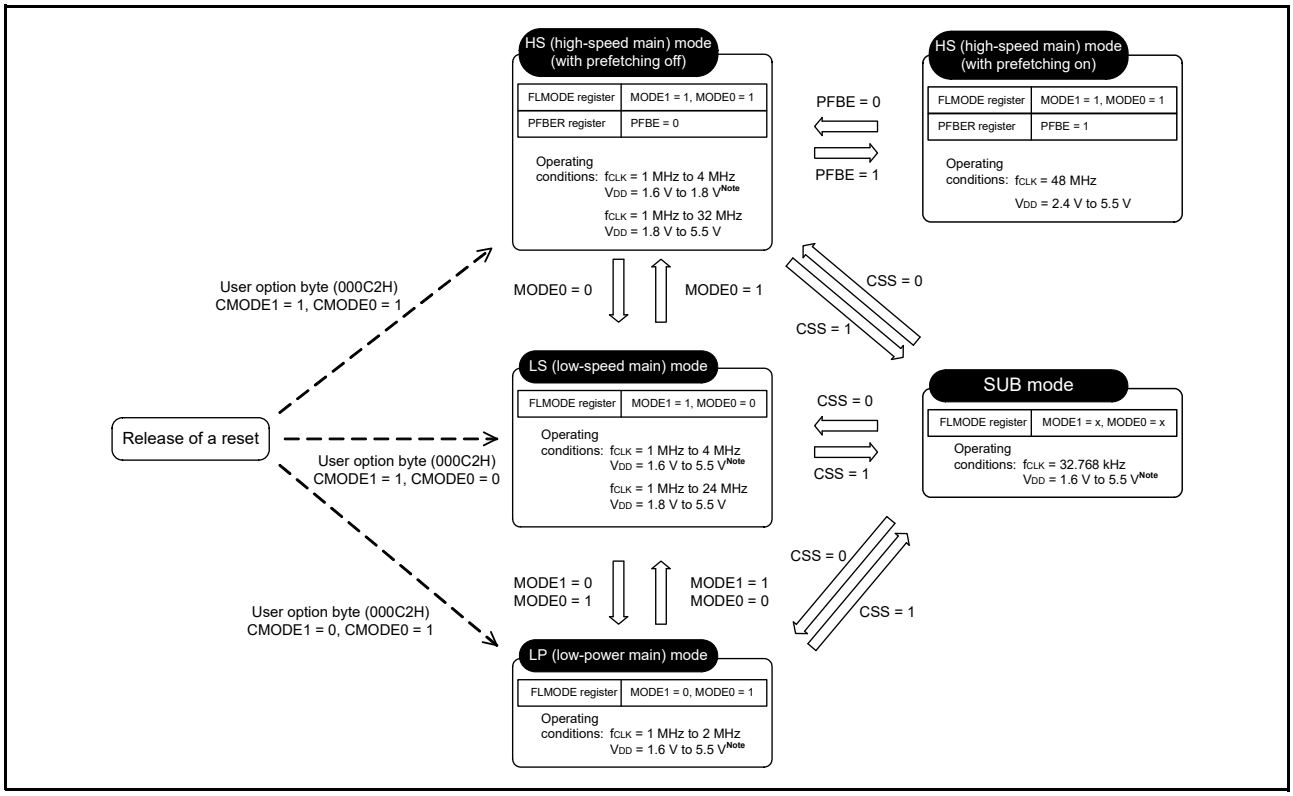
Set an appropriate flash operation mode according to the V_{DD} voltage and the high-speed on-chip oscillator frequency at a reset release.

When a reset is released, the value of the CMODE[1:0] bits is reflected in the MODE[1:0] bits in the flash operating mode select register (FLMODE) and the value of the FRQSEL[4:0] bits is reflected in the high-speed on-chip oscillator frequency select register (HOCODIV). For details on the user option byte (000C2H), see **Section 38 Option Bytes**.

8.4 Transitions between Flash Operation Modes

Setting of the CMODE[1:0] bits in the user option byte (000C2H) determines the initial flash operation mode immediately after a reset is released. The initial state can be selected from among the HS (high-speed main) mode (with prefetching off), LS (low-speed main) mode, or LP (low-power main) mode.
The value of CMODE[1:0] bits is reflected in the MODE[1:0] bits in the flash operating mode select register (FLMODE). After that, the flash operation mode can be changed by changing the values of the FLMODE and PFBER registers during CPU operation. Setting the CSS bit in the CKC register to 1 enables automatically placing the MCU in SUB mode.

Figure 8 - 5 State Transitions between Flash Operation Modes



Note Rewriting of the flash memory is not possible.

Caution **When a reset is applied while the MCU operates, operation always starts in the flash operation mode set by the option byte after release from the reset state. Therefore, make sure that operation does not start outside the operating voltage range when a reset is released by setting the LVD detection voltage within the usable operating voltage range of the flash operation mode set in the option byte.**

8.4.1 Prefetching instructions from the flash memory

The flash memory can handle instruction prefetching to accelerate the execution of code. The use of prefetching requires that the prefetch buffer be enabled. To enable the prefetch buffer, set the PFBE bit in the PFBER register to 1. Switch the prefetch buffer between enabled and disabled in accord with the flowcharts shown as **Figure 8 - 6 Flowchart for Switching the Prefetch Buffer from Disabled to Enabled (When f_{IH} Is to Be Selected as f_{MAIN})** to **Figure 8 - 9 Flowchart for Switching the Prefetch Buffer from Enabled to Disabled (When f_{PLL} Is Selected as f_{MAIN})**.

Figure 8 - 6 Flowchart for Switching the Prefetch Buffer from Disabled to Enabled (When f_{IH} Is to Be Selected as f_{MAIN})

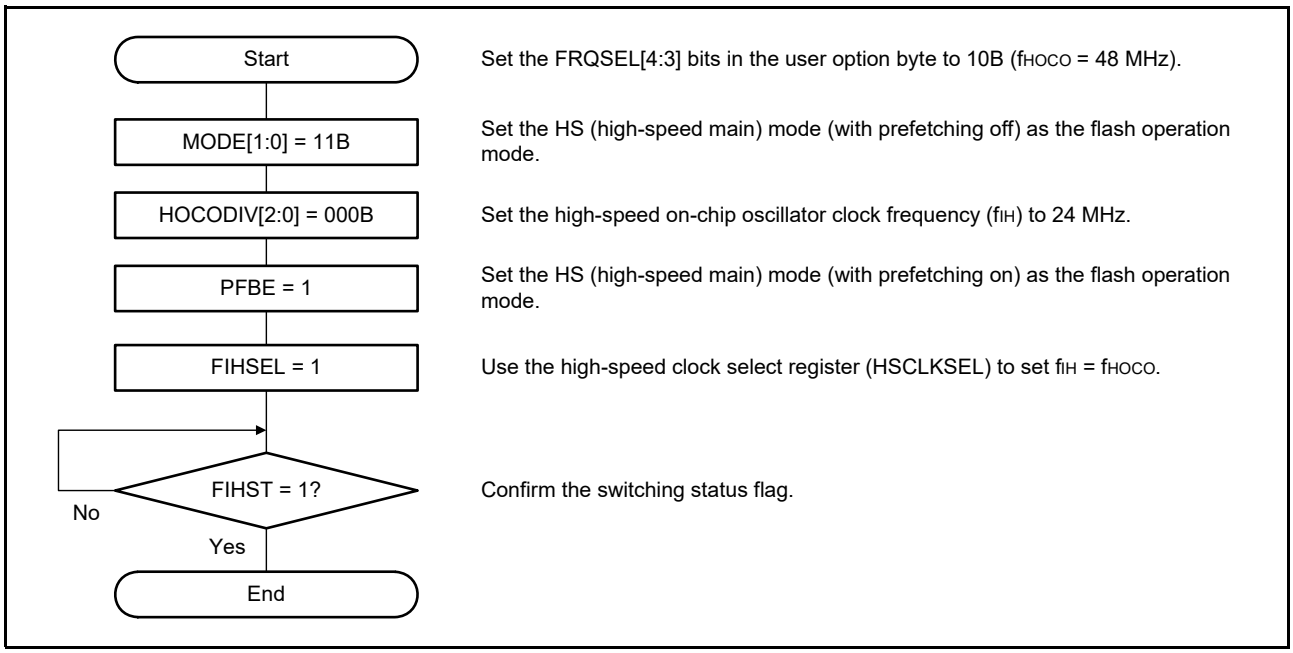


Figure 8 - 7 Flowchart for Switching the Prefetch Buffer from Enabled to Disabled (When f_{IH} Is Selected as f_{MAIN})

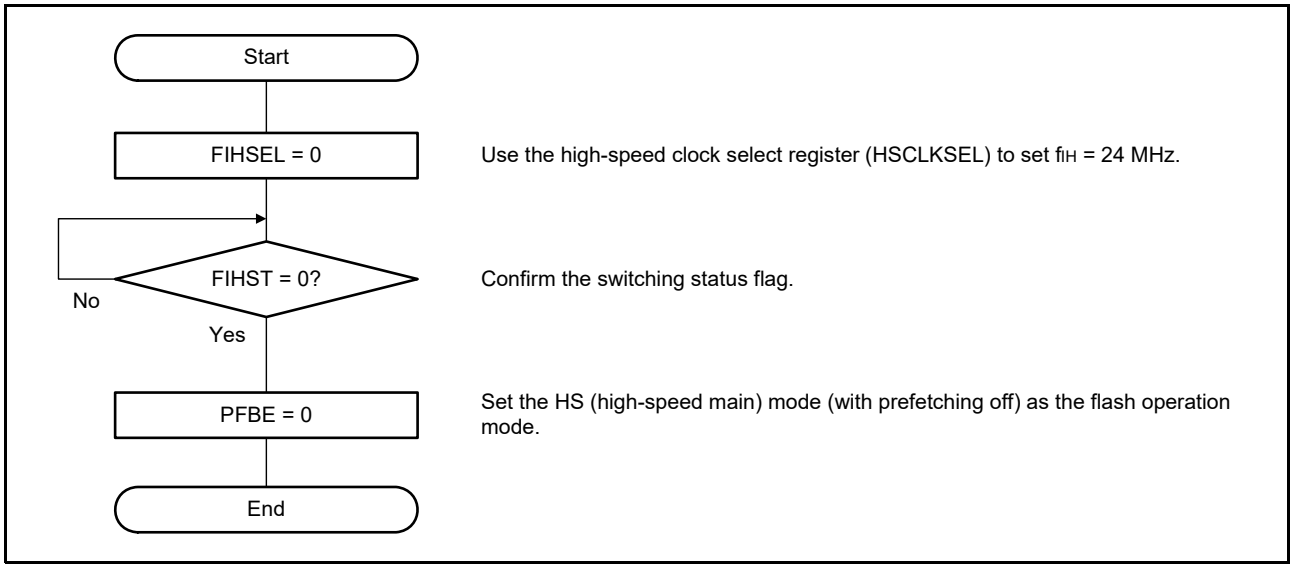


Figure 8 - 8 Flowchart for Switching the Prefetch Buffer from Disabled to Enabled (When f_{PLL} Is to Be Selected as f_{MAIN})

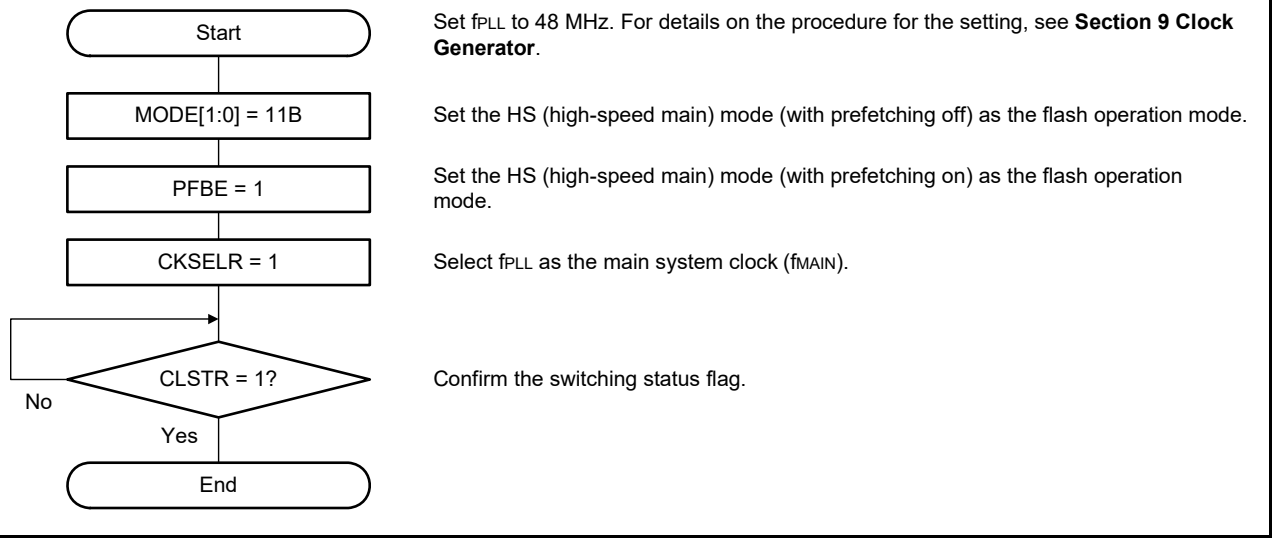
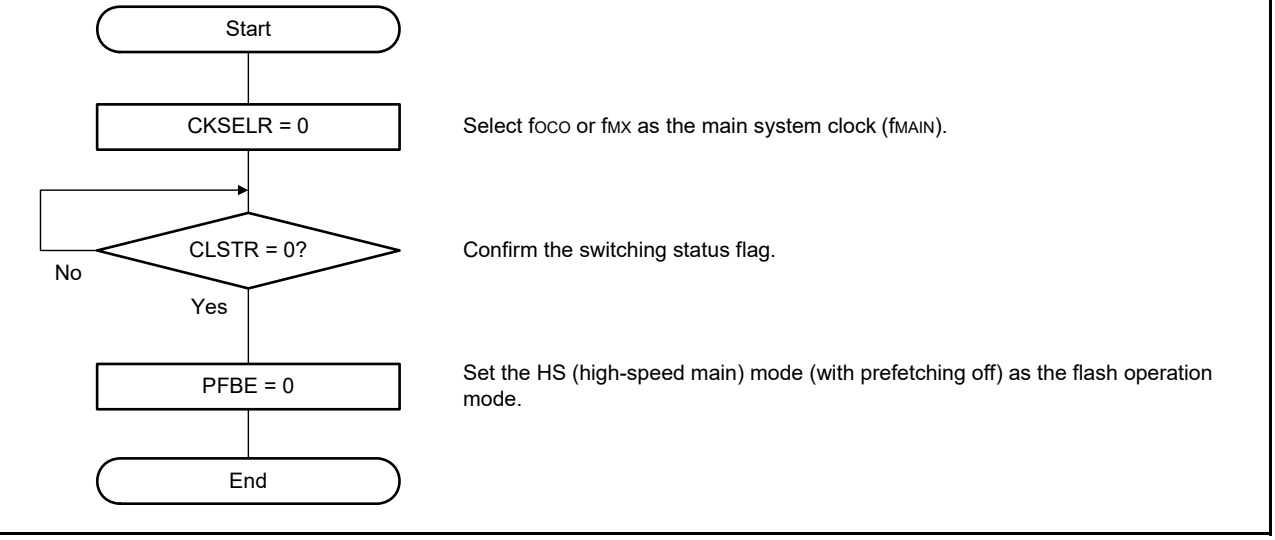


Figure 8 - 9 Flowchart for Switching the Prefetch Buffer from Enabled to Disabled (When f_{PLL} Is Selected as f_{MAIN})



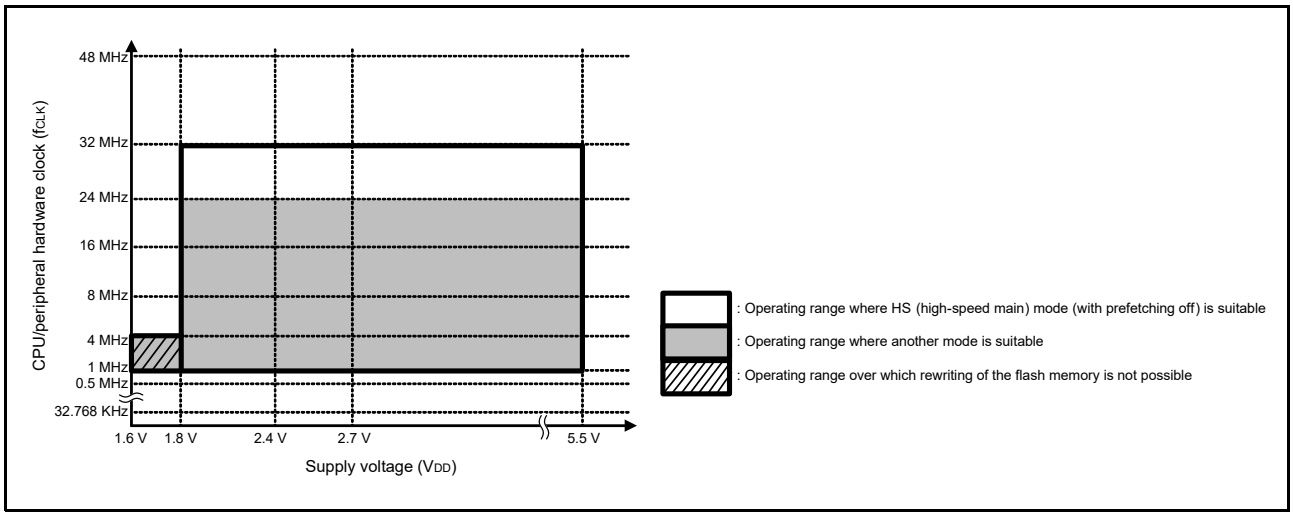
8.5 Details of Flash Operation Modes

8.5.1 Details of HS (high-speed main) mode (with prefetching off)

HS (high-speed main) mode (with prefetching off) is suitable for applications that require CPU high-speed processing. HS (high-speed main) mode (with prefetching off) can be operated immediately after release from the reset state. Also, this mode can be entered from LS (low-speed main) mode.

The suitable operating range in HS (high-speed main) mode (with prefetching off) is when the supply voltage is 1.8 V ≤ V_{DD} ≤ 5.5 V and the operating frequency is 24 MHz < f_{CLK} ≤ 32 MHz. When 24 MHz or lower is used for operation, another mode can be used as the suitable flash operation mode.

Figure 8 - 10 Operating Range in HS (High-speed Main) Mode (with Prefetching Off)

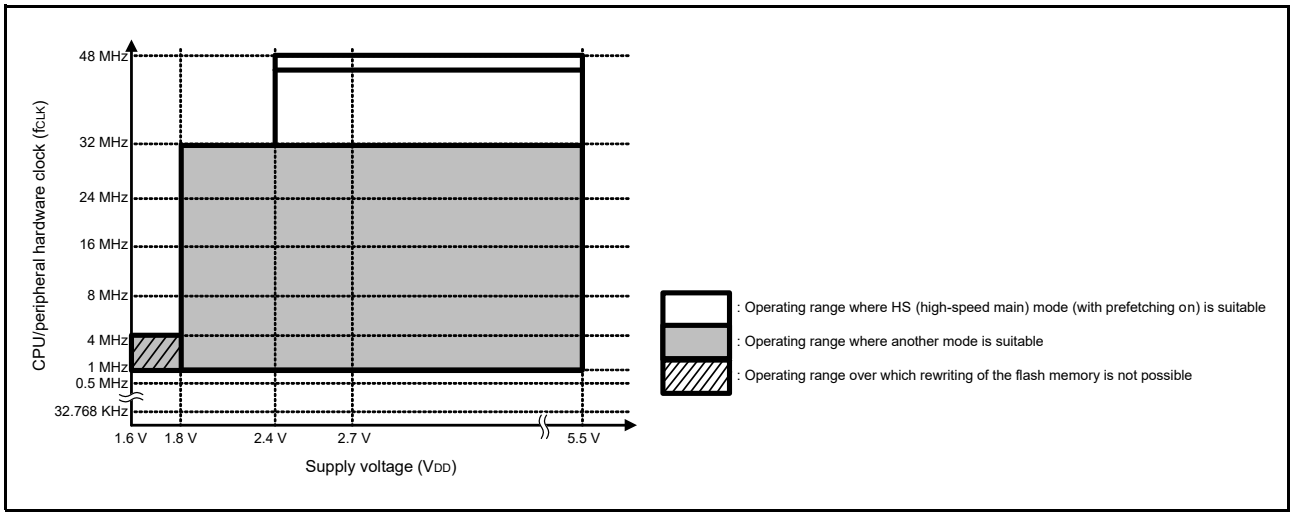


Caution FAA core operations and the rewriting of values for the peripheral functions are prohibited for 20 µs after release from the stop state.

8.5.2 Details of HS (high-speed main) mode (with prefetching on)

In HS (high-speed main) mode (with prefetching on), the CPU operates at 48 MHz.
HS (high-speed main) mode (with prefetching on) can be entered from HS (high-speed main) mode (with prefetching off).
To change the operating frequency to 48 MHz, enable prefetching after having ensured that the supply voltage is 2.4 V ≤ V_{DD} ≤ 5.5 V.

Figure 8 - 11 Operating Range in HS (High-speed Main) Mode (with Prefetching On)



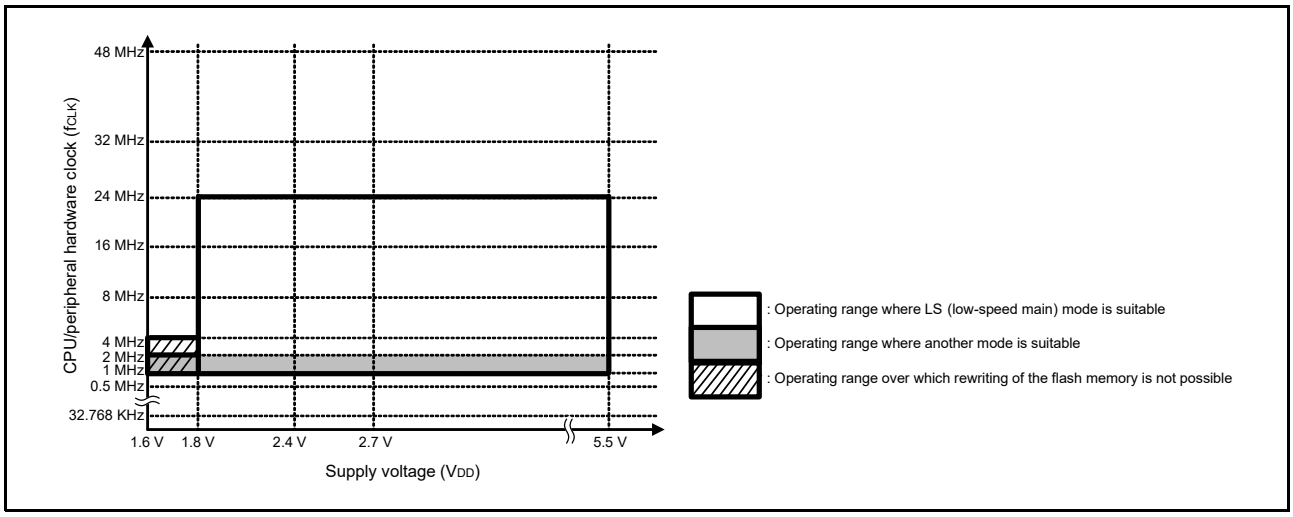
Caution FAA core operations and the rewriting of values for the peripheral functions are prohibited for 20 µs after release from the stop state.

8.5.3 Details of LS (low-speed main) mode

LS (low-speed main) mode supports both CPU processing capacity and operating voltage performance, suitable for applications that require low-power consumption at 2 to 24 MHz.
LS (low-speed main) mode can be operated immediately after release from the reset state. Also, this mode can be entered from HS (high-speed main) mode, or LP (low-power main) mode. For a transition from HS (high-speed main) mode to LS (low-speed main) mode, make sure that the operating frequency is 1 MHz ≤ f_{CLK} ≤ 24 MHz.
The suitable operating range in LS (low-speed main) mode is when the supply voltage is 1.8 V ≤ V_{DD} ≤ 5.5 V and the operating frequency is 2 MHz < f_{CLK} ≤ 24 MHz, or when the supply voltage is 1.6 V ≤ V_{DD} < 1.8 V and the operating frequency is 2 MHz < f_{CLK} ≤ 4 MHz**Note**.

Note Rewriting of the flash memory is not possible.

Figure 8 - 12 Operating Range in LS (Low-speed Main) Mode



Caution FAA core operations and the rewriting of values for the peripheral functions are prohibited for 20 µs after release from the stop state.

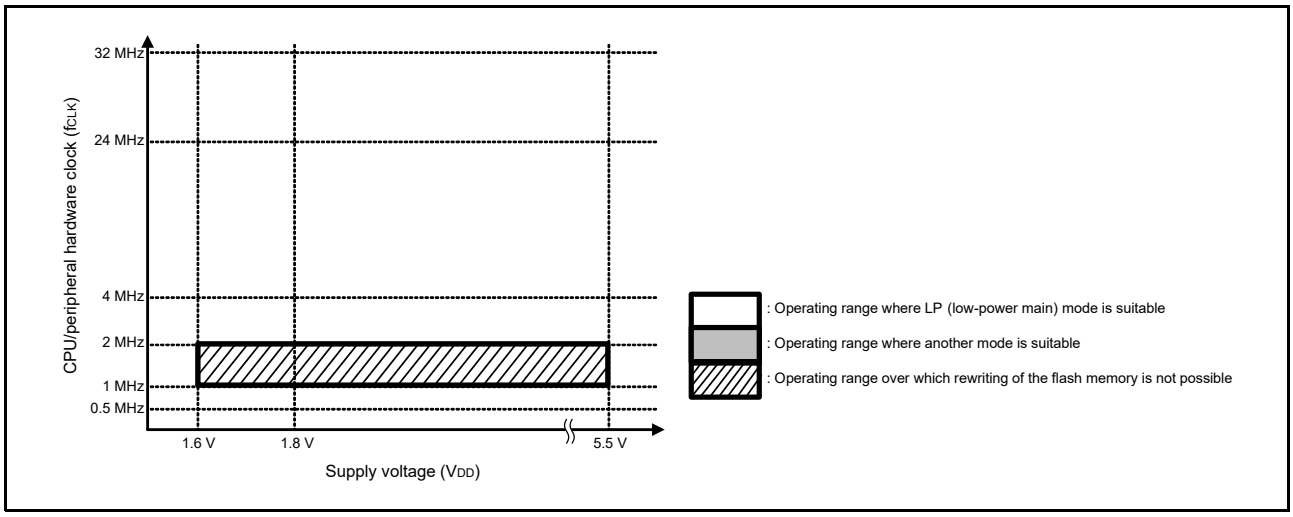
8.5.4 Details of LP (low-power main) mode

In LP (low-power main) mode, the CPU operates with a low voltage and at a frequency from 1 MHz to 2 MHz. The CPU can operate in LP (low-power main) mode immediately after a reset is released. LP (low-power main) mode can be entered from LS (low-speed main) mode. For a transition from LS (low-speed main) mode to LP (low-power main) mode, make sure the operating frequency is 1 MHz ≤ f_{CLK} ≤ 2 MHz.

The suitable operating range in LP (low-power main) mode is when the supply voltage is 1.6 V ≤ V_{DD} ≤ 5.5 V and the operating frequency is 1 MHz ≤ f_{CLK} ≤ 2 MHz.

When rewriting of flash memory is required, place the CPU in LS (low-speed main) mode.

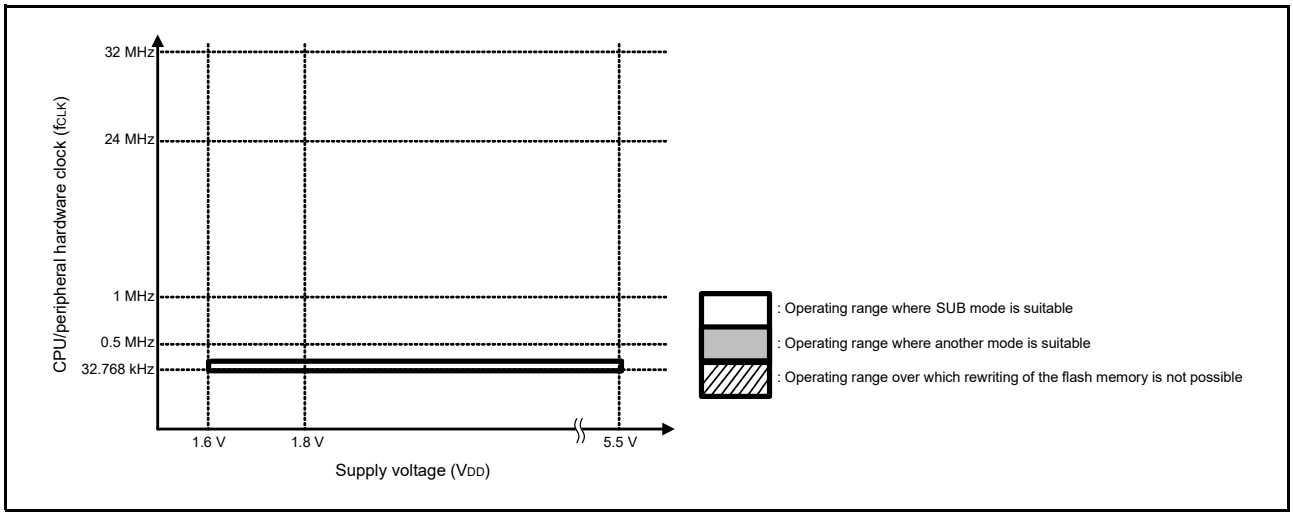
Figure 8 - 13 Operating Range in LP (Low-power Main) Mode



8.5.5 Details on SUB mode

In SUB mode, the CPU operates at a frequency of 32.768 kHz.
Transition to SUB mode is possible from HS (high-speed main) mode, LS (low-speed main) mode, or LP (low-power main) mode. Setting the CSS bit in the system clock control register (CKC) to 1 places the CPU in SUB mode. Rewriting of the flash memory is not possible in SUB mode. When rewriting of flash memory is required, place the CPU in HS (high-speed main) mode or LS (low-speed main) mode.

Figure 8 - 14 Operating Range in SUB Mode



Section 9 Clock Generator

The presence or absence of the following pin functions depends on the product: the X1 and X2 pin functions for connecting a resonator for the main system clock, the EXCLK pin function for external clock input, the XT1 and XT2 pin functions for connecting a resonator for the subsystem clock, and the EXCLKS pin function for external clock input.

	Products with 20 Pins	Products with 24 to 32 Pins	Products with 40 to 64 Pins
X1 pin	—	—	✓
X2/EXCLK multiplexed pin	—	—	✓
XT1 pin	—	—	✓
XT2/EXCLKS multiplexed pin	—	—	✓
X1/XT1 multiplexed pin	—	✓	—
X2/XT2/EXCLK/EXCLKS multiplexed pin	—	✓	—
EXCLK/EXCLKS multiplexed pin	✓	—	—

9.1 Functions of Clock Generator

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

1. Main system clock
   <1> X1 oscillator
   This circuit oscillates to produce a clock at fX = 1 to 20 MHz by connecting a resonator to the X1 pin and X2 pin. Oscillation can be stopped by executing a STOP instruction or setting of the MSTOP bit (bit 7 of the clock operation status control register (CSC)).
   <2> High-speed on-chip oscillator
   The frequency of oscillation can be selected from among fHOCO = 64, 48, 32, 24, 16, 12, 8, 6, 4, 3, 2, or 1 MHz (typ.) by using a user option byte (000C2H). If 64 MHz or 48 MHz is selected for fHOCO, fIH is 32 MHz or 24 MHz, respectively. When 32 MHz or a lower frequency is selected for fHOCO, fIH is not divided, thus having the same frequency as fHOCO. After release from the reset state, the CPU always starts operating with this high-speed on-chip oscillator clock. Oscillation can be stopped by executing a STOP instruction or setting of the HIOSTOP bit (bit 0 of the CSC register).
   The frequency specified by using the option byte can be changed by using the high-speed on-chip oscillator frequency select register (HOCODIV). For details about the frequency, see **Figure 9 - 12 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	32	48	64
2.4 V ≤ VDD ≤ 5.5 V	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
1.8 V ≤ VDD ≤ 5.5 V	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
1.6 V ≤ VDD ≤ 5.5 V	✓	✓	—	—	—	—	—	—	—	—	—	—

   <3> Middle-speed on-chip oscillator
   The frequency of oscillation can be selected from among fIM = 4, 2, or 1 MHz (typ.) by using the MOCODIV[1:0] bits (bits 1, 0 of the MOCODIV register). Oscillation can be stopped by executing a STOP instruction or clearing of the MIOEN bit (bit 1 of the CSC register).
   <4> PLL oscillator
   This circuit oscillates to produce a clock at fPLL = 96 or 64 MHz as the main system clock by using the main clock control register (MCKC) to select the PLL clock and setting the PLL control register (DSCCTL) for either of these frequencies.

An external main system clock (fEX = 1 to 20 MHz) can also be supplied from the EXCLK/X2/P122 pin. External main system clock input can be disabled by executing a 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 a main on-chip oscillator clock (high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock) can be selected by setting of the MCM0 and MCM1 bits (bits 4 and 0 of the system clock control register (CKC)).

Note that the usable frequency range of the CPU/peripheral hardware clock depends on the flash operation mode and VDD power supply voltage settings.
When the main system clock is to be used as the CPU/peripheral hardware clock, select the frequency of the main system clock to suit the flash operation mode specified in the CMODE[1:0] bits of a user option byte (000C2H) (see **Section 38 Option Bytes**) or the flash operating mode select register (FLMODE).

2. Subsystem clock

<1> XT1 oscillator

This circuit oscillates to produce a clock at f_{XT} = 32.768 kHz by connecting a 32.768-kHz resonator to XT1 pin and XT2 pin. Oscillation can be stopped by setting the XTSTOP bit (bit 6 of the clock operation status control register (CSC)).

An external subsystem clock (f_{EXS} = 32.768 kHz) can also be supplied from the EXCLKS/XT2/P124 pin. External subsystem clock input can be disabled by the setting of the XTSTOP bit.

<2> Low-speed on-chip oscillator

This circuit oscillates to produce a clock at f_{IL} = 32.768 kHz (typ.).

The low-speed on-chip oscillator operates when the following condition is met.

- The value of one or more of the following bits is 1: bit 4 (WDTON) of a user option byte (000C0H), bit 4 (WUTMMCK0) of the subsystem clock supply mode control register (OSMC), and bit 0 (SELLOSC) of the subsystem clock select register (CKSEL)

Remark f_X: X1 clock oscillation frequency
f_{HOCO}: High-speed on-chip oscillator clock frequency (64 MHz max.)
f_{IH}: High-speed on-chip oscillator clock frequency (48 MHz max.)**Note**
f_{IM}: Middle-speed on-chip oscillator clock frequency (4 MHz max.)
f_{EX}: External main system clock frequency
f_{XT}: XT1 clock oscillation frequency
f_{EXS}: External subsystem clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency
f_{PLL}: PLL clock frequency (96 MHz max.)

Note When f_{HOCO} is set to 64 or 48 MHz, f_{IH} is obtained by dividing f_{HOCO} by 2 under hardware control. When 32 MHz or a lower frequency is selected for f_{HOCO}, f_{IH} is not divided under hardware control, thus having the same frequency as f_{HOCO}.
When a 48-MHz clock is to be supplied as f_{IH} while f_{HOCO} is set to 48 MHz, enable the prefetch buffer by using the PFBE bit (bit 0 of the prefetch buffer enable register (PFBER)) before switching the clock.

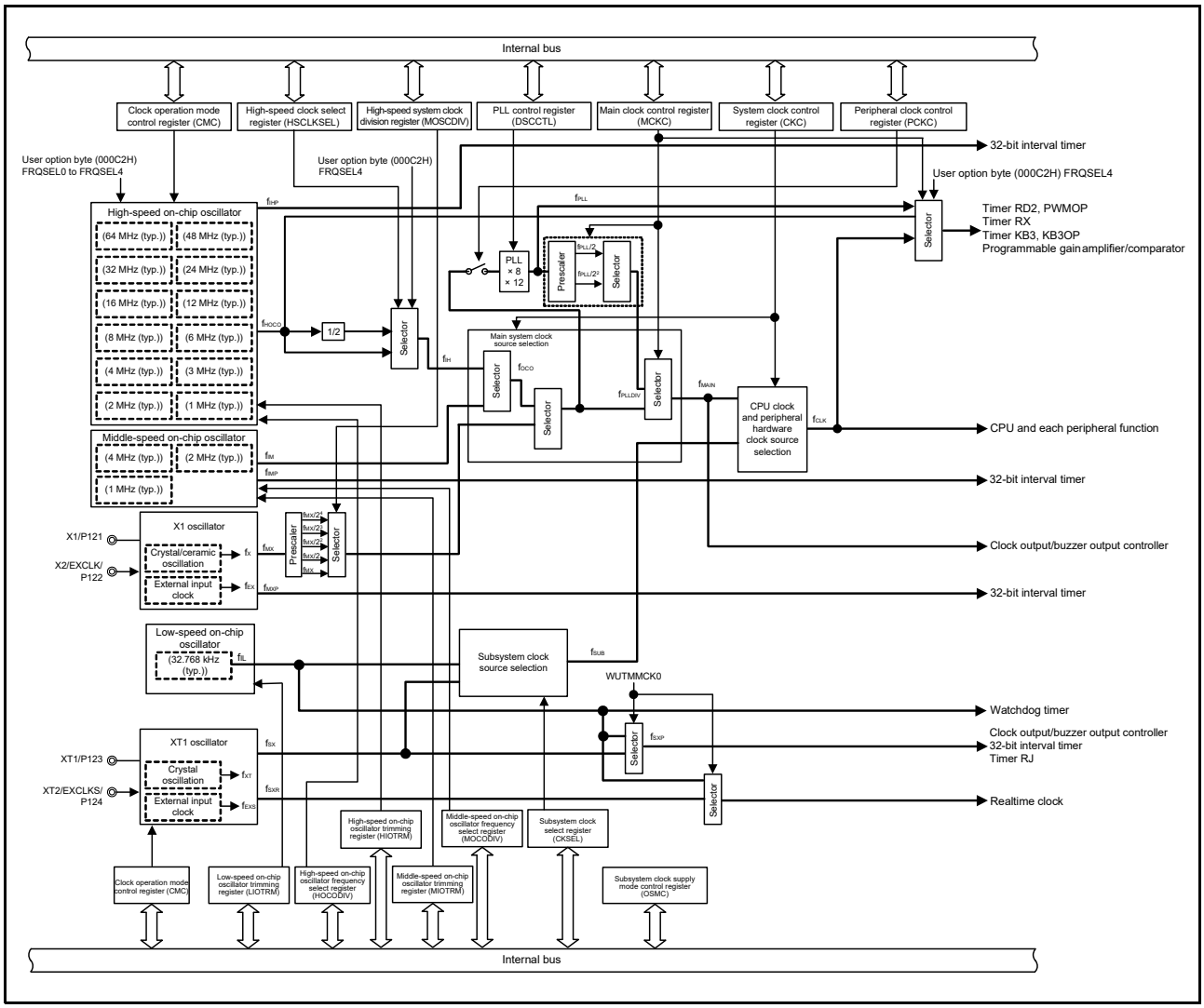
9.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 9 - 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 to 2 (PER0 to PER2) • Subsystem clock supply mode control register (OSMC) • Subsystem clock select register (CKSEL) • High-speed on-chip oscillator frequency select register (HOCODIV) • Middle-speed on-chip oscillator frequency select register (MOCODIV) • High-speed system clock division register (MOSCDIV) • High-speed on-chip oscillator trimming register (HIOTRM) • Middle-speed on-chip oscillator trimming register (MIOTRM) • Low-speed on-chip oscillator trimming register (LIOTRM) • Standby mode release setting register (WKUPMD) • High-speed clock select register (HSCLKSEL) • PLL control register (DSCCTL) • Main clock control register (MCKC) • Peripheral clock control register (PCKC)
Oscillators	• X1 oscillator • XT1 oscillator • High-speed on-chip oscillator • Middle-speed on-chip oscillator • Low-speed on-chip oscillator • PLL oscillator

Figure 9 - 1 Block Diagram of Clock Generator



Remark 1. f_X: X1 clock oscillation frequency
f_{EX}: External main system clock frequency
f_{HOCO}: High-speed on-chip oscillator clock frequency (64 MHz max.)
f_{IH}: High-speed on-chip oscillator clock frequency (48 MHz max.)**Note**
f_{IHP}: High-speed on-chip oscillator peripheral clock frequency
(48 MHz when FRQSEL4 = 1 and FRQSEL3 = 0, 32 MHz when FRQSEL3 = 1,
24 MHz when FRQSEL4 = 0 and FRQSEL3 = 0)
f_{IM}: Middle-speed on-chip oscillator clock frequency (4 MHz max.)
f_{IMP}: Middle-speed on-chip oscillator peripheral clock frequency (4 MHz)
f_{MX}: High-speed system clock frequency
f_{MXP}: High-speed peripheral clock frequency
f_{MAIN}: Main system clock frequency
f_{XT}: XT1 clock oscillation frequency
f_{EXS}: External subsystem clock frequency
f_{SX}: Subsystem clock X frequency
f_{SXR}: Subsystem clock XR frequency
f_{SXP}: Low-speed peripheral clock frequency
f_{SUB}: Subsystem clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency
f_{OCO}: Main on-chip oscillator clock frequency (f_{IH} or f_{IM})

f_{PLL}: PLL clock frequency (96 MHz max.)

f_{PLLDIV}: PLL frequency-divided clock frequency (48 MHz max.)

Note When f_{HOCO} is set to 64 or 48 MHz, f_{IH} is obtained by dividing f_{HOCO} by 2 under hardware control. When 32 MHz or a lower frequency is selected for f_{HOCO}, f_{IH} is not divided under hardware control, thus having the same frequency as f_{HOCO}.

When a 48-MHz clock is to be supplied as f_{IH} while f_{HOCO} is set to 48 MHz, enable the prefetch buffer by using the PFBE bit (bit 0 of the prefetch buffer enable register (PFBER)) before switching the clock.

Remark 2. The table below shows selection of the clock signal for supply to timer RD2, the PWMOP, timer RX, 16-bit timers KB30, KB31, and KB32, the optional functions of 16-bit timers KB30, KB31, and KB32, comparators, and the programmable gain amplifier.

Bit Value in the User Option Byte (000C2H)	Bit Value in the MCKC Register	Clock Signal for Supply to Timer RD2, the PWMOP, Timer RX, 16-bit Timers KB30, KB31, and KB32, the Optional Functions of 16-bit Timers KB30, KB31, and KB32, Comparators, and the Programmable Gain Amplifier
FRQSEL4	CLSTR	
0	0	System clock (f_{CLK})
1	0	High-speed on-chip oscillator (f_{HOCO})
0	1	PLL clock (f_{PLL})
1	1	Setting prohibited

9.3 Registers to Control the 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 to 2 (PER0 to PER2)
- Subsystem clock supply mode control register (OSMC)
- Subsystem clock select register (CKSEL)
- High-speed on-chip oscillator frequency select register (HOCODIV)
- Middle-speed on-chip oscillator frequency select register (MOCODIV)
- High-speed system clock division register (MOSCDIV)
- High-speed on-chip oscillator trimming register (HIOTRM)
- Middle-speed on-chip oscillator trimming register (MIOTRM)
- Low-speed on-chip oscillator trimming register (LIOTRM)
- Standby mode release setting register (WKUPMD)
- High-speed clock select register (HSCLKSEL)
- PLL control register (DSCCTL)
- Main clock control register (MCKC)
- Peripheral clock control register (PCKC)

Caution Which registers and bit functions are included depends on the product. Be sure to set bits to which no functions are allocated to their initial values.

9.3.1 Clock operation mode control register (CMC)

The CMC register is used to set the operation mode of the X1/P121, X2/EXCLK/P122, XT1/P123, and XT2/EXCLKS/P124 pins, and to select the gain of the oscillator. The CMC register can be written only once by an 8-bit memory manipulation instruction after release from the reset state. This register can be read by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 9 - 2 Format of Clock Operation Mode Control Register (CMC) (1/2)

Address: FFFA0H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS **Note 1**	OSCSELS **Note 1**	XTSEL **Notes 1, 2**	AMPHS1 **Note 1**	AMPHS0	AMPH

Products with 20 pins

XTSEL **Notes 1, 2**	EXCLK	OSCSEL	EXCLKS **Note 1**	OSCSELS **Note 1**	System clock pin operation mode	EXCLK/P122/ EXCLKS pin
0	0	0	0	0	Port mode	Port
0	1	0	0	0	Port mode	Port
0	1	1	0	0	External clock input mode	External clock EXCLK input
1	0	0	0	0	Port mode	Port
1	0	0	1	0	Port mode	Port
1	0	0	1	1	External clock input mode	External clock EXCLKS input
Other than above					Setting prohibited	

Products with 24 to 32 pins

XTSEL **Notes 1, 2**	EXCLK	OSCSEL	EXCLKS **Note 1**	OSCSELS **Note 1**	System clock pin operation mode	X1/P121/ XT1 pin	X2/EXCLK/ P122/XT2/ EXCLKS pin
0	0	0	0	0	Port mode	Port	Port
0	0	1	0	0	X1 oscillation mode	Crystal/ceramic resonator connection	
0	1	0	0	0	Port mode	Port	Port
0	1	1	0	0	External clock input mode	Port	External clock EXCLK input
1	0	0	0	0	Port mode	Port	Port
1	0	0	0	1	XT1 oscillation mode	Crystal resonator connection	
1	0	0	1	0	Port mode	Port	Port
1	0	0	1	1	External clock input mode	Port	External clock EXCLKS input
Other than above					Setting prohibited		

Figure 9 - 2 Format of Clock Operation Mode Control Register (CMC) (2/2)

Products with 40 to 64 pins

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

EXCLKS Note 1	OSCSELS Note 1	Subsystem clock pin operation mode	XT1/P123 pin	XT2/EXCLKS/P124 pin
0	0	Input port mode	Input port	Input port
0	1	XT1 oscillation mode	Crystal resonator connection	
1	0	Input port mode	Input port	Input port
1	1	External clock input mode	Input port	External clock input

AMPHS1 Note 1	AMPHS0 Note 1	Selection of the oscillation mode of the XT1 oscillator
0	0	Low power consumption oscillation 1 (default)Note 3
0	1	Normal oscillation
1	0	Low power consumption oscillation 2Note 3
1	1	Low power consumption oscillation 3Note 3

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

Note 1. The EXCLKS, OSCSELS, XTSEL, and AMPHS[1:0] bits are only initialized by a power-on reset; they retain the values when a reset caused by another source occurs.

Note 2. The XTSEL bit can be written only in the products with 20 to 32 pins. Be sure to set this bit to 0 in the products with 40 to 64 pins.

Note 3. The gain and operating current of the XT1 clock oscillator decrease in the following order: low power consumption oscillation 1 > low power consumption oscillation 2 > low power consumption oscillation 3.

Caution 1. The CMC register can be written only once by an 8-bit memory manipulation instruction after release from the reset state. Even if you intend to use the CMC register with its initial value (00H), be sure to write 00H to the register after release from the reset state as a precaution against malfunctions (since returning the value to 00H after erroneously having written a value other than 00H to it is not possible).

Caution 2. After release from the reset state, set the CMC register before X1 or XT1 oscillation is started by setting 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. Make the settings of the AMPH and AMPHS[1:0] bits while f_{IH} is selected as f_{CLK} after release from the reset state (before f_{CLK} is switched to f_{MX} or f_{SUB}).

Caution 5. Count the oscillation stabilization time of f_{XT} by software.

(Cautions and Remark are listed on the next page.)

Caution 6. The XT1 oscillator is a circuit with low amplification in order to achieve low power consumption. Note the following points when designing the circuit.

- **Pins and circuit boards include parasitic capacitance. Therefore, perform oscillation evaluation using a circuit board to be actually used and confirm that there are no problems.**
- **When using low power consumption oscillation 2 (AMPHS[1:0] = 10B) or low power consumption oscillation 3 (AMPHS[1:0] = 11B) as the XT1 oscillator mode, sufficiently evaluate the resonator as described in 9.7 Resonator and Oscillator Constants, before using it in either of these modes.**
- **Make the wiring runs between the XT1 and XT2 pins and the resonator as short as possible to minimize the parasitic capacitance and wiring resistance. Take care with this particularly when low power consumption oscillation 2 (AMPHS[1:0] = 10B) or low power consumption oscillation 3 (AMPHS[1:0] = 11B) is selected.**
- **Configure the circuit of the circuit board, using material with little parasitic capacitance and wiring resistance.**
- **Place a ground pattern that has the same voltage as V_{SS} as much as possible near the XT1 oscillator.**
- **Be sure that the signal lines between the XT1 and XT2 pins and the resonator do not cross the other signal lines. Do not route the wiring near a signal line through which a strong fluctuating current flows.**
- **The impedance between the XT1 and XT2 pins may drop and oscillation may be disturbed due to moisture absorption of the circuit board in a high-humidity environment or dew condensation on the board. When using the circuit board in such an environment, take measures to damp-proof the circuit board, such as by coating.**
- **When coating the circuit board, use material that does not cause capacitance or leakage between the XT1 and XT2 pins.**

Caution 7. When selecting the XT1 oscillation mode for the P121/X1/XT1 and P122/X2/EXCLK/XT2/EXCLKS pins while the value of the XTSEL bit is 1 in a product with 20 to 32 pins, make sure that V_{DD} is no less than 2.4 V.

Remark fx: X1 clock oscillation frequency

9.3.2 System clock control register (CKC)

The CKC 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. The value of this register following a reset is 00H.

Figure 9 - 3 Format of System Clock Control Register (CKC)

Address: FFFA4H
After reset: 00H
R/W: R/W**Note 1**

Symbol	<7>	<6>	<5>	<4>	3	2	<1>	<0>
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1

CLS	State of the CPU/peripheral hardware clock (f_{CLK})
0	Main system clock (f_{MAIN})
1	Subsystem clock (f_{SUB})

CSS**Note 2**	Selection of the CPU/peripheral hardware clock (f_{CLK})
0	Main system clock (f_{MAIN})
1	Subsystem clock (f_{SUB})

MCS	State of the main system clock (f_{MAIN})
0	Main on-chip oscillator clock (f_{OCO})
1	High-speed system clock (f_{MX})

MCM0 **Note 2**	Main system clock (f_{MAIN}) operation control
0	Selects the main on-chip oscillator clock (f_{OCO}) as the main system clock (f_{MAIN})
1	Selects the high-speed system clock (f_{MX}) as the main system clock (f_{MAIN})

MCS1	State of the main on-chip oscillator clock (f_{OCO})
0	High-speed on-chip oscillator clock
1	Middle-speed on-chip oscillator clock

MCM1 **Note 2**	Main on-chip oscillator clock (f_{OCO}) operation control
0	High-speed on-chip oscillator clock
1	Middle-speed on-chip oscillator clock

Note 1. Bits 7, 5, and 1 are read-only.
Note 2. Changing the value of the MCM[1:0] bits is prohibited while the CSS bit is set to 1.

Caution 1. Be sure to set bits 3 and 2 of the CKC register to 0.
Caution 2. The clock set by the CSS bit is supplied to the CPU and peripheral hardware. If the CPU clock is changed, therefore, the clock supplied to peripheral hardware (except the realtime clock, clock output/buzzer output, 32-bit interval timer, and watchdog timer) is also changed at the same time. Consequently, stop each peripheral function when changing the CPU/peripheral hardware clock.

(Caution and Remark are listed on the next page.)

Caution 3. If the subsystem clock is used as the peripheral hardware clock, the operations of the A/D converter and serial interface IICA are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the sections describing each item of peripheral hardware as well as Section 43 Electrical Characteristics (T_A = –40 to +105°C) or Section 44 Electrical Characteristics (T_A = –40 to +125°C).

Remark f_{IH}: High-speed on-chip oscillator clock frequency (48 MHz max.)
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency
f_{SUB}: Subsystem clock frequency
f_{OCO}: Main on-chip oscillator clock frequency (f_{IH} or f_{IM})

9.3.3 Clock operation status control register (CSC)

The CSC register is used to control the operations of the high-speed system clock, high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, and subsystem 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. The value of this register following a reset is C0H.

Figure 9 - 4 Format of Clock Operation Status Control Register (CSC)

Address: FFFA1H
After reset: C0H
R/W: R/W

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

MSTOP	High-speed system clock operation control		
	X1 oscillation mode	External clock input mode	Port mode
0	The X1 oscillator runs.	An external clock signal on the EXCLK pin is effective.	I/O port
1	The X1 oscillator is stopped.	An external clock signal on the EXCLK pin is ineffective.	

XTSTOP **Note**	Subsystem clock operation control		
	XT1 oscillation mode	External clock input mode	Port mode
0	The XT1 oscillator runs.	An external clock signal on the EXCLKS pin is effective.	Input port
1	The XT1 oscillator is stopped.	An external clock signal on the EXCLKS pin is ineffective.	

MIOEN	Middle-speed on-chip oscillator clock operation control
0	The middle-speed on-chip oscillator is stopped.
1	The middle-speed on-chip oscillator runs.

HIOSTOP	High-speed on-chip oscillator clock operation control
0	The high-speed on-chip oscillator runs.
1	The high-speed on-chip oscillator is stopped.

Note The XTSTOP bit is only initialized by a power-on reset; it retains its value when a reset caused by another source occurs.

Caution 1. After release from the reset state, 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 release from the reset state. 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. When starting X1 oscillation by using the MSTOP bit, check the oscillation stabilization time of the X1 clock by using the oscillation stabilization time counter status register (OSTC).

Caution 4. When starting XT1 oscillation by using the XTSTOP bit, include code to wait for oscillation of the subsystem clock to become stable.

Caution 5. Do not stop the clock selected for the CPU/peripheral hardware clock (f_{CLK}) with the CSC register.

Caution 6. When stopping the clock, confirm the conditions before clock oscillation is stopped. For details on how to stop the clock, see Table 9 - 8 Conditions Before the Clock Oscillation Is Stopped and Flag Settings.

9.3.4 Oscillation stabilization time counter status register (OSTC)

The OSTC register indicates the counter value by the X1 clock oscillation stabilization time counter. The X1 clock oscillation stabilization time can be checked in the following cases:

- If the X1 clock starts oscillation while the main on-chip oscillator clock or subsystem clock is in use as the CPU clock.
- If entry to and then release from the STOP mode proceed while the main on-chip oscillator clock is in use as the CPU clock and the X1 clock is oscillating.

The OSTC register can be read by a 1-bit or 8-bit memory manipulation instruction. The value of this register is 00H following a reset, STOP instruction, or the MSTOP bit (bit 7 of the clock operation status control register (CSC)) being set to 1.

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

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

Figure 9 - 5 Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFFA2H
After reset: 00H
R/W: R

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

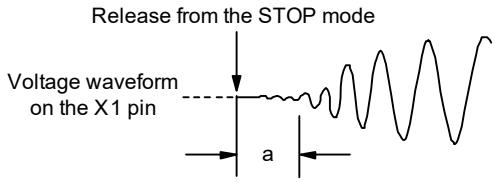
MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	State of the oscillation stabilization time		
									fx = 10 MHz	fx = 20 MHz
0	0	0	0	0	0	0	0	Less than 2^8/fx	Less than 25.6 μs	Less than 12.8 μs
1	0	0	0	0	0	0	0	2^8/fx min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	2^9/fx min.	51.2 μs min.	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.

(Cautions and Remark are listed on the next page.)

Caution 2. The value counted by the OSTC register will only have reached the oscillation stabilization time setting in 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 counter value which is to be checked by using the OSTC register.
- **If the X1 clock starts oscillation while the main on-chip oscillator clock or subsystem clock is in use as the CPU clock.**
- **If entry to and then release from the STOP mode proceed while the main on-chip oscillator clock is in use as the CPU clock and the X1 clock is oscillating**
  Therefore, note that the value counted by the OSTC register will only have reached the oscillation stabilization time setting in the OSTS register after release from the STOP mode.

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



Remark fx: X1 clock oscillation frequency

9.3.5 Oscillation stabilization time select register (OSTS)

The OSTS register is used to select the X1 clock oscillation stabilization time. When the X1 clock is made to oscillate by clearing the MSTOP bit to start operation of the X1 oscillator, actual operation is automatically delayed for the time set in the OSTS register. Use the oscillation stabilization time counter status register (OSTC) to confirm that the specified oscillation stabilization time has elapsed when the CPU clock is switched from the main on-chip oscillator clock or the subsystem clock to the X1 clock or entry to and then release from the STOP mode proceed while the main on-chip oscillator clock is in use as the CPU clock and the X1 clock is oscillating. The OSTC register can be used to check the counter value when counting has reached the time set beforehand in the OSTS register. The OSTS register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 07H.

Figure 9 - 6 Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFFA3H
After reset: 07H
R/W: R/W

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

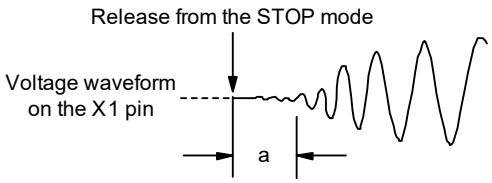
OSTS2	OSTS1	OSTS0	Selection of the oscillation stabilization time		
				fx = 10 MHz	fx = 20 MHz
0	0	0	2^8/fx	25.6 μs	12.8 μs
0	0	1	2^9/fx	51.2 μs	25.6 μs
0	1	0	2^{10}/fx	102 μs	51.2 μs
0	1	1	2^{11}/fx	204 μs	102 μs
1	0	0	2^{13}/fx	819 μs	409 μs
1	0	1	2^{15}/fx	3.27 ms	1.63 ms
1	1	0	2^{17}/fx	13.1 ms	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 value counted by the OSTC register will only have reached the oscillation stabilization time setting in the OSTS register.
In the following cases, set the oscillation stabilization time of the OSTS register to the value greater than the counter value which is to be checked by using the OSTC register after the oscillation starts.
- **If the X1 clock starts oscillation while the main on-chip oscillator clock or subsystem clock is in use as the CPU clock.**
- **If entry to and then release from the STOP mode proceed while the main on-chip oscillator clock is in use as the CPU clock and the X1 clock is oscillating.**
  Therefore, note that the value counted by the OSTC register will only have reached the oscillation stabilization time setting in the OSTS register after release from the STOP mode.

(Caution and Remark are listed on the next page.)

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



Remark fx: X1 clock oscillation frequency

9.3.6 Peripheral enable registers 0 to 2 (PER0 to PER2)

The PER0 to PER2 registers are used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. To use any of the on-chip peripheral modules listed below, the clock supplies to which are controlled by these registers, set the corresponding bit to 1 before making the initial settings of the on-chip peripheral module.

- Realtime clock
- Serial interface IICA
- A/D converter
- Serial array unit n
- Timer array unit
- D/A converter
- Programmable gain amplifier/comparator
- 32-bit interval timer
- DTC
- Digital addressable lighting interface
- FAA
- Data shared memory
- KB3 16-bit timer and forced output stop function of the KB3 16-bit timer
- RG2 timer
- RD2 timer
- PWM option unit
- RX timer
- RJ timer

Remark n = 0, 1

The PER0 to PER2 registers can be set by a 1-bit or 8-bit memory manipulation instruction.
The value of each of these registers following a reset is 00H.

Figure 9 - 7 Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

RTCWEN	Control of access to the realtime clock
0	The SFRs used by the realtime clock cannot be written.
1	The SFRs used by the realtime clock can be read and written.

ADCEN	Control of supply of an input clock to the A/D converter
0	Stops supply of an input clock. • The SFRs used by the A/D converter cannot be written.**Note**
1	Enables supply of an input clock. • The SFRs used by the A/D converter can be read and written.

IICA0EN	Control of supply of an input clock to the IICA serial interface
0	Stops supply of an input clock. • The SFRs used by the IICA serial interface cannot be written.**Note**
1	Enables supply of an input clock. • The SFRs used by the IICA serial interface can be read and written.

SAU1EN	Control of supply of an input clock to serial array unit 1
0	Stops supply of an input clock. • The SFRs used by serial array unit 1 cannot be written.**Note**
1	Enables supply of an input clock. • The SFRs used by serial array unit 1 can be read and written.

SAU0EN	Control of supply of an input clock to serial array unit 0
0	Stops supply of an input clock. • The SFRs used by serial array unit 0 cannot be written.**Note**
1	Enables supply of an input clock. • The SFRs used by serial array unit 0 can be read and written.

TAU0EN	Control of supply of an input clock to the timer array unit
0	Stops supply of an input clock. • The SFRs used by the timer array unit cannot be written.**Note**
1	Enables supply of an input clock. • The SFRs used by the timer array unit can be read and written.

Note When an SFR is read, the value returned is 00H or 0000H regardless of its setting.

Caution 1. Be sure to set bits 6 and 1 to 0.

Caution 2. Do not change the value of a bit of the PER0 register while operation of the corresponding on-chip peripheral module is enabled. Only change a value while the corresponding on-chip peripheral module is stopped.

Figure 9 - 8 Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	<3>	2	1	<0>
PER1	DACEN	0	PGACMPEN	TML32EN	DTCEN	0	0	DALIEN

DACEN	Control of supply of an input clock to the D/A converter
0	Stops supply of an input clock. • The SFRs used by the D/A converter cannot be written.**Note**
1	Enables supply of an input clock. • The SFRs used by the D/A converter can be read and written.

PGACMPEN **Note**	Control of supply of an input clock to the programmable gain amplifier/comparator
0	Stops supply of an input clock. • The SFRs used by the programmable gain amplifier/comparator cannot be written.**Note**
1	Enables supply of an input clock. • The SFRs used by the programmable gain amplifier/comparator can be read and written.

TML32EN	Control of supply of an input clock to the 32-bit interval timer
0	Stops supply of an input clock. • The SFRs used by the 32-bit interval timer cannot be written.**Note**
1	Enables supply of an input clock. • The SFRs used by the 32-bit interval timer can be read and written.

DTCEN	Control of supply of an input clock to the DTC
0	Stops supply of an input clock. • The DTC cannot operate.
1	Enables supply of an input clock. • The DTC can operate.

DALIEN	Control of supply of an input clock to the digital addressable lighting interface
0	Stops supply of an input clock. • The SFRs used by the digital addressable lighting interface cannot be written.**Note**
1	Enables supply of an input clock. • The SFRs used by the digital addressable lighting interface can be read and written.

Note When an SFR is read, the value returned is 00H or 0000H regardless of its setting.

Caution 1. Be sure to set bits 6, 2, and 1 to 0.

Caution 2. Do not change the value of a bit of the PER1 register while operation of the corresponding on-chip peripheral module is enabled. Only change a value while the corresponding on-chip peripheral module is stopped.

Caution 3. When the setting of the FRQSEL4 bit of the user option byte (000C2H) is 1, select f_{IH} as f_{CLK} before setting bit 5 (PGACMPEN) of peripheral enable register 1 (PER1).
When changing the source clock of f_{CLK} to a clock other than f_{IH}, clear bit 5 (PGACMPEN) of peripheral enable register 1 (PER1) before doing so.

Figure 9 - 9 Format of Peripheral Enable Register 2 (PER2) (1/2)

Address: F00FCH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER2	FAAEN	MEMEN	TKBEN	TRGEN	TRD0EN	PWMOPEN	TRXEN	TRJ0EN

FAAEN	Control of supply of an input clock to the FAA core
0	Stops supply of an input clock. • With the following exceptions, the SFRs used by the FAA can be read but not written. Note that only writing to the WIND register, ENB bit in the SYSC register, and ADBSEL register is possible.
1	Enables supply of an input clock. • The SFRs used by the FAA core can be read and written.

MEMEN	Control of supply of an input clock to the data shared memory
0	Stops supply of an input clock. • The SFRs used by the data shared memory can be read but not written.
1	Enables supply of an input clock. • The SFRs used by the data shared memory can be read and written.

TKBEN**Note**	Control of supply of an input clock to the KB3 16-bit timer and forced output stop function of the KB3 16-bit timer
0	Stops supply of an input clock. • The SFRs used by the KB3 16-bit timer and forced output stop function of the KB3 16-bit timer cannot be written.**Note**
1	Enables supply of an input clock. • The SFRs used by the KB3 16-bit timer and forced output stop function of the KB3 16-bit timer can be read and written.

TRGEN	Control of supply of an input clock to the RG2 timer
0	Stops supply of an input clock. • The SFRs used by the RG2 timer cannot be written.**Note**
1	Enables supply of an input clock. • The SFRs used by the RG2 timer can be read and written.

TRD0EN **Note**	Control of supply of an input clock to the RD2 timer
0	Stops supply of an input clock. • The SFRs used by the RD2 timer cannot be written.**Note**
1	Enables supply of an input clock. • The SFRs used by the RD2 timer can be read and written.

Figure 9 - 9 Format of Peripheral Enable Register 2 (PER2) (2/2)

PWMOPEN	Control of supply of an input clock to the PWMOPA
0	Stops supply of an input clock. • The SFRs used by the PWMOPA cannot be written.**Note**
1	Enables supply of an input clock. • The SFRs used by the PWMOPA can be read and written.

TRXEN**Note**	Control of supply of an input clock to the RX timer
0	Stops supply of an input clock. • The SFRs used by the RX timer cannot be written.**Note**
1	Enables supply of an input clock. • The SFRs used by the RX timer can be read and written.

TRJ0EN	Control of supply of an input clock to the RJ timer
0	Stops supply of an input clock. • The SFRs used by the RJ timer cannot be written.**Note**
1	Enables supply of an input clock. • The SFRs used by the RJ timer can be read and written.

Note When an SFR is read, the value returned is 00H or 0000H regardless of its setting.

Caution 1. Do not change the value of a bit of the PER2 register while operation of the corresponding on-chip peripheral module is enabled. Only change a value while the corresponding on-chip peripheral module is stopped.

Caution 2. When the setting of the FRQSEL4 bit of the user option byte (000C2H) is 1, select f_{IH} as f_{CLK} before setting bits 5 (TKBEN), 3 (TRD0EN), and 1 (TRXEN) of peripheral enable register 2 (PER2).
When changing the source clock of f_{CLK} to a clock other than f_{IH}, clear bits 5 (TKBEN), 3 (TRD0EN), and 1 (TRXEN) of peripheral enable register 2 (PER2) before doing so.

9.3.7 Subsystem clock supply mode control register (OSMC)

The OSMC register is used to reduce power consumption by stopping unnecessary clock functions. If the RTCLPC bit is set to 1, power consumption can be reduced, because clock supply to the peripheral functions except the realtime clock is stopped in STOP mode or in HALT mode while the CPU is operating with the subsystem clock. In addition, the OSMC register can be used to select the operating clock for the realtime clock, clock output/buzzer output controller, and 32-bit interval timer. The OSMC register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is undefined**Note**.

Note The RTCLPC and WUTMMCK bits have the value 0 and the HIPREC bit has the value 1 following a reset.

Figure 9 - 10 Format of Subsystem Clock Supply Mode Control Register (OSMC)

Address: F00F3H
After reset: Undefined**Note 1**
R/W: R/W**Note 2**

Symbol	<7>	6	5	<4>	3	2	1	<0>
OSMC	RTCLPC	0	0	WUTMMCK 0	×	×	0	HIPREC

RTCLPC **Note 5**	Setting in STOP mode or in HALT mode while the CPU is operating with the subsystem clock X
0	Enables supply of the subsystem clock X to peripheral functions (See **Table 31 - 1 Operating States in HALT Mode** to **Table 31 - 4 Operating States in SNOOZE Mode** for peripheral functions whose operations are enabled.)
1	Stops supply of the subsystem clock to peripheral functions other than the realtime clock.

WUTMMCK0	Selection of the operating clock for the realtime clock, 32-bit interval timer, timer RJ, and clock output/buzzer output controller
0	Subsystem clock X
1	Low-speed on-chip oscillator clock**Notes 3, 4**

HIPREC	State of the high-speed on-chip oscillator clock**Notes 6, 7**
0	The high-speed on-chip oscillator clock is being started at high speed and waiting for the precision of its oscillation to become stable is in progress.**Note 8**
1	The high-speed on-chip oscillator clock is operating with high precision.

Note 1. The RTCLPC and WUTMMCK bits have the value 0 and the HIPREC bit has the value 1 following a reset.
Note 2. Be sure to set bits 6, 5, and 1 to 0. Bits 3, 2, and 0 are read-only. Writing to these bits is ignored.
Note 3. Do not set the WUTMMCK0 bit to 1 while the subsystem clock X is oscillating.
Note 4. Switching between the subsystem clock X and the low-speed on-chip oscillator clock can be enabled by the WUTMMCK0 bit only when all of the realtime clock, 32-bit interval timer, and clock output/buzzer output function are stopped.
Note 5. When the subsystem clock X is selected (SELLOSC = 0) by bit 0 (SELLOSC) of the CKSEL register and RTCLPC is set to 1, the subsystem clock (f_{SUB}) is stopped. However, when the low-speed on-chip oscillator clock is selected (SELLOSC = 1) and RTCLPC is set to 1, the subsystem clock (f_{SUB}) is not stopped.
Note 6. Undefined while the high-speed on-chip oscillator is stopped.
Note 7. For frequency accuracy of the high-speed on-chip oscillator clock, see **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** or **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.

(Note and Caution are listed on the next page.)

Note 8. When the RL78/G24 is released from the STOP mode while the setting for starting the high-speed on-chip oscillator at high speed is in place (WKUPMD.FWKUP = 1), the high-speed on-chip oscillator clock runs at low precision while it is starting up. After the oscillation accuracy stabilization time, the value of the HIPREC bit changes automatically to 1.
The table below shows the frequency of the high-speed on-chip oscillator when FRQSEL3 = 0 and HIPREC = 0.

Setting of FRQSEL2 or HOCODIV2	Setting of FRQSEL1 or HOCODIV1	Setting of FRQSEL0 or HOCODIV0	Frequency of the High-speed On-chip Oscillator
0	0	0	16 MHz
0	0	1	8 MHz
0	1	0	4 MHz
0	1	1	2 MHz
1	0	0	Setting prohibited
1	0	1	Setting prohibited

Caution Do not execute a STOP instruction when HIPREC = 0.

9.3.8 Subsystem clock select register (CKSEL)

The CKSEL register is used to select the subsystem clock X or low-speed on-chip oscillator clock as the subsystem clock. The CKSEL register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 9 - 11 Format of Subsystem Clock Select Register (CKSEL)

Address: FFFA7H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	<0>
CKSEL	0	0	0	0	0	0	0	SELLOSC

SELLOSC	Selection of the subsystem clock X or low-speed on-chip oscillator clock
0	Subsystem clock X
1	Low-speed on-chip oscillator clock[Note]

Note Do not set SELLOSC to 1 when the subsystem clock X (f_{SX}) or XR (f_{SXR}) is operating.

Caution When changing the value of the SELLOSC bit, be sure to set the CSS bit to 0 (selecting f_{MAIN}) and confirm that the value of the CLS bit is 0 before doing so.

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

The HOCODIV register is used to change the frequency of the high-speed on-chip oscillator set in a user option byte (000C2H). Note that the selectable frequencies depend on the values of the FRQSEL[4:3] bits of the user option byte (000C2H) and bit 0 (FIHSEL) of the high-speed clock select register (HSCLKSEL). The HOCODIV register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is that set in the FRQSEL[2:0] bits of the user option byte (000C2H).

Figure 9 - 12 Format of High-speed On-chip Oscillator Frequency Select Register (HOCODIV)

Address: F00A8H
After reset: The value set in the FRQSEL[2:0] bits of a user option byte (000C2H)
R/W: 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 the high-speed on-chip oscillator clock frequency					
			FIHSEL = 0				FIHSEL = 1	
			FRQSEL4 = 0		FRQSEL4 = 1		FRQSEL4 = 1	
			FRQSEL3 = 0	FRQSEL3 = 1	FRQSEL3 = 0	FRQSEL3 = 1	FRQSEL3 = 0	FRQSEL3 = 1
0	0	0	f_{IH} = 24 MHz	f_{IH} = 32 MHz	f_{IH} = 24 MHz f_{HOCO} = 48 MHz	f_{IH} = 32 MHz f_{HOCO} = 64 MHz	f_{IH} = 48 MHz f_{HOCO} = 48 MHz	Setting prohibited
0	0	1	f_{IH} = 12 MHz	f_{IH} = 16 MHz	f_{IH} = 12 MHz f_{HOCO} = 24 MHz	f_{IH} = 16 MHz f_{HOCO} = 32 MHz	Setting prohibited	Setting prohibited
0	1	0	f_{IH} = 6 MHz	f_{IH} = 8 MHz	f_{IH} = 6 MHz f_{HOCO} = 12 MHz	f_{IH} = 8 MHz f_{HOCO} = 16 MHz	Setting prohibited	Setting prohibited
0	1	1	f_{IH} = 3 MHz	f_{IH} = 4 MHz	f_{IH} = 3 MHz f_{HOCO} = 6 MHz	f_{IH} = 4 MHz f_{HOCO} = 8 MHz	Setting prohibited	Setting prohibited
1	0	0	Setting prohibited	f_{IH} = 2 MHz	Setting prohibited	f_{IH} = 2 MHz f_{HOCO} = 4 MHz	Setting prohibited	Setting prohibited
1	0	1	Setting prohibited	f_{IH} = 1 MHz	Setting prohibited	f_{IH} = 1 MHz f_{HOCO} = 2 MHz	Setting prohibited	Setting prohibited
Other than above			Setting prohibited					

Caution 1. Set the HOCODIV register while ensuring that the voltage is within the usable range for the flash operation mode set in the flash operating mode select register (FLMODE) both before and after the frequency change. For details about the FLMODE register, see 8.2.1 Flash operating mode select register (FLMODE).

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

Caution 3. After use of the HOCODIV register to change the frequency, actually switching of the frequency only proceeds after the following transition times have elapsed.

- **The CPU/peripheral hardware clock continues to operate at the frequency before the change for up to three cycles.**
- **Up to three cycles of waiting are required for the CPU/peripheral hardware clock to be at the post-change frequency.**

Caution 4. When using the HOCODIV register to change f_{HOCO} to 64 or 48 MHz (FRQSEL4 = 1 and HOCODIV[2:0] = 000B), clear bit 5 (PGACMPEN) of peripheral enable register 1 (PER1) and bits 5 (TKBEN), 3 (TRD0EN), 2 (PWMOPEN), and 1 (TRXEN) of peripheral enable register 2 (PER2) before doing so.

9.3.10 Middle-speed on-chip oscillator frequency select register (MOCODIV)

The MOCODIV register is used to select the frequency of the middle-speed on-chip oscillator. The MOCODIV register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 9 - 13 Format of Middle-speed On-chip Oscillator Frequency Select Register (MOCODIV)

Address: F00F2H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
MOCODIV	0	0	0	0	0	0	MOCODIV1	MOCODIV0

MOCODIV 1	MOCODIV 0	Selection of the middle-speed on-chip oscillator clock frequency
0	0	4 MHz
0	1	2 MHz
1	0	1 MHz
Other than above		Setting prohibited

Caution Set the MOCODIV register while ensuring that the voltage is within the usable range for the flash operation mode set in the flash operating mode select register (FLMODE) both before and after the frequency change.

9.3.11 High-speed system clock division register (MOSCDIV)

The MOSCDIV register is used to select the division ratio of the high-speed system clock. The MOSCDIV register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 9 - 14 Format of High-speed System Clock Division Register (MOSCDIV)

Address: F0214H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
MOSCDIV	0	0	0	0	0	MOSCDIV2	MOSCDIV1	MOSCDIV0

MOSCDIV2	MOSCDIV1	MOSCDIV0	Selected division ratio for the high-speed system clock	f_{MX} = 20 MHz
0	0	0	f_{MX}	20 MHz
0	0	1	f_{MX}/2	10 MHz
0	1	0	f_{MX}/4	5 MHz
0	1	1	f_{MX}/8	2.5 MHz
1	0	0	f_{MX}/16	1.25 MHz
Other than above			Setting prohibited	

Caution Set the MOSCDIV register while ensuring that the voltage is within the usable range for the flash operation mode set in the flash operating mode select register (FLMODE) both before and after the frequency change.

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

The HIOTRM register is used to adjust the accuracy of the high-speed on-chip oscillator. The accuracy of the high-speed on-chip oscillator frequency can be adjusted through self-measurement of the frequency by using a timer with high accuracy (timer array unit or 32-bit interval timer) for external clock input or in other ways. 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 9 - 15 Format of High-speed On-chip Oscillator Trimming Register (HIOTRM)

Address: F00A0H
After reset: **Note**
R/W: 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 of this register following a reset is that adjusted at shipment.

Remark 1. The HIOTRM register holds a six-bit value used to adjust the high-speed on-chip oscillator with an increment of 1 corresponding to an increase of frequency by 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 (R01AN2833)**.

9.3.13 Middle-speed on-chip oscillator trimming register (MIOTRM)

The MIOTRM register is used to adjust the accuracy of the middle-speed on-chip oscillator. The accuracy of the middle-speed on-chip oscillator frequency can be adjusted through self-measurement of the frequency by using a timer with high accuracy (timer array unit or 32-bit interval timer) for external clock input or in other ways. The MIOTRM register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 90H.

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 9 - 16 Format of Middle-speed On-chip Oscillator Trimming Register (MIOTRM)

Address: F0212H
After reset: 90H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
MIOTRM	MIOTRM7	MIOTRM6	MIOTRM5	MIOTRM4	MIOTRM3	MIOTRM2	MIOTRM1	MIOTRM0

MIOTRM7	MIOTRM6	MIOTRM5	MIOTRM4	MIOTRM3	MIOTRM2	MIOTRM1	MIOTRM0	Middle-speed on-chip oscillator
0	0	0	0	0	0	0	0	Minimum speed
0	0	0	0	0	0	0	1	↑
								↑
1	0	0	0	1	1	1	1	↑
1	0	0	1	0	0	0	0	Initial value
1	0	0	1	0	0	0	1	↓
								↓
1	1	1	1	1	1	1	0	↓
1	1	1	1	1	1	1	1	Maximum speed

Remark For details about the accuracy adjustment resolution of the middle-speed on-chip oscillator clock, see **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** or **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.

9.3.14 Low-speed on-chip oscillator trimming register (LIOTRM)

The LIOTRM register is used to adjust the accuracy of the low-speed on-chip oscillator. The accuracy of the low-speed on-chip oscillator frequency can be adjusted through self-measurement of the frequency by using a timer with high accuracy (timer array unit or 32-bit interval timer) for external clock input or in other ways. The LIOTRM register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 80H.

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 9 - 17 Format of Low-speed On-chip Oscillator Trimming Register (LIOTRM)

Address: F0213H
After reset: 80H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
LIOTRM	LIOTRM7	LIOTRM6	LIOTRM5	LIOTRM4	LIOTRM3	LIOTRM2	LIOTRM1	LIOTRM0

LIOTRM7	LIOTRM6	LIOTRM5	LIOTRM4	LIOTRM3	LIOTRM2	LIOTRM1	LIOTRM0	Low-speed on-chip oscillator
0	0	0	0	0	0	0	0	Minimum speed
0	0	0	0	0	0	0	1	↑
0	1	1	1	1	1	1	1	
1	0	0	0	0	0	0	0	Initial value
1	0	0	0	0	0	0	1	↓
1	1	1	1	1	1	1	0	
1	1	1	1	1	1	1	1	Maximum speed

Remark For details about the accuracy adjustment resolution of the low-speed on-chip oscillator clock, see **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** or **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.

9.3.15 Standby mode release setting register (WKUPMD)

The WKUPMD register is used to set the operation when the standby mode is released. The WKUPMD register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 9 - 18 Format of Standby Mode Release Setting Register (WKUPMD)

Address: F0215H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	<0>
WKUPMD	0	0	0	0	0	0	0	FWKUP

FWKUP	Setting for starting the high-speed on-chip oscillator at the times of release from STOP mode and of transitions to SNOOZE mode**Notes 1, 2**
0	Starting of the high-speed on-chip oscillator is at normal speed.**Note 3**
1	Starting of the high-speed on-chip oscillator is at high speed.**Notes 3, 4**

Note 1. This setting is only available when the high-speed on-chip oscillator is selected for the CPU clock.

Note 2. This register is initialized when the RL78/G24 is released from STOP mode in response to the generation of a reset signal, so starting of the high-speed on-chip oscillator is at normal speed.

Note 3. For the activation time, see **Section 31 Standby Function**.
The accuracy of the high-speed on-chip oscillator frequency depends on whether starting of the high-speed on-chip oscillator is at normal speed or at high speed. See **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** or **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.

Note 4. This is only specifiable when the setting of the FRQSEL4 bit of the user option byte (000C2H) is 0.

9.3.16 High-speed clock select register (HSCLKSEL)

The HSCLKSEL register enables supply of a 48-MHz clock as the high-speed on-chip oscillator clock. When a 48-MHz clock is to be supplied as the high-speed on-chip oscillator clock while the high-speed on-chip oscillator clock is set to 48 MHz, enable the prefetch buffer by using the PFBE bit (bit 0 of the prefetch buffer enable register (PFBER)) before switching the clock. The HSCLKSEL register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H or 02H**Note**.

Note The value after a reset depends on a setting in an option byte. That is, the value of the FRQSEL4 bit of the user option byte is reflected in the FIHST bit after release from the reset state.

Figure 9 - 19 Format of High-speed Clock Select Register (HSCLKSEL)

Address: F0219H
After reset: 00H or 02H**Note 1**
R/W: R/W**Note 2**

Symbol	7	6	5	4	3	2	<1>	<0>
HSCLKSEL	0	0	0	0	0	0	FIHST	FIHSEL

Value of the user option byte (000C2H)		FIHSEL	Selection of f_{IH}
FRQSEL4	FRQSEL3		
0	0/1	0	$f_{IH} = f_{HOCO}$
1	0/1	0	$f_{IH} = f_{HOCO}/2$
1	0	1	$f_{IH} = f_{HOCO}$**Note 3**
Other than above			Setting prohibited

FIHST	State of f_{IH}
0	$f_{IH} = f_{HOCO}$
1	$f_{IH} = f_{HOCO}/2$

Note 1. The value after a reset depends on a setting in an option byte. That is, the value of the FRQSEL4 bit of the user option byte is reflected in the FIHST bit after release from the reset state.

Note 2. Bit 1 is read-only.

Note 3. While the prefetch buffer is disabled (by the PFBE bit of the PFBER register being set to 0), do not set the FIHSEL bit to 1.

Remark Running the main system clock (f_{MAIN}) at 48 MHz requires that prefetching be enabled.
For the flowcharts for setting up and disabling prefetching, see **Section 8 Operation State Control**.

9.3.17 PLL control register (DSCCTL)

The DSCCTL register controls operation of the PLL oscillator. When the PLL function is to be used, set the high-speed on-chip oscillator clock or high-speed system clock to run at 8 MHz. The DSCCTL register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 56H.

Figure 9 - 20 Format of PLL Control Register (DSCCTL)

Address: F02E5H
After reset: 56H
R/W: R/W

Symbol	7	<6>	5	4	3	2	1	<0>
DSCCTL	0	PLLDIV0	0	DSCM4	DSCM3	DSCM2	DSCM1	DSCON

PLLDIV0	PLL division ratio selection bit[Note 1]
1	Frequency division by two
Other than above	Setting prohibited

DSCM4	DSCM3	DSCM2	DSCM1	PLL multiplication factor selection bits[Notes 1, 2]
0	1	1	1	Times 16/2 (times 8): f_{PLL} = 64 MHz
1	0	1	1	Times 24/2 (times 12): f_{PLL} = 96 MHz
Other than above				Setting prohibited

DSCON	Control of PLL oscillation and output[Note 3]
0	The PLL is stopped.
1	The PLL is oscillating and producing a clock signal.[Note 4]

Note 1. Do not change the setting while the PLL is operating. Stop the PLL before changing it.

Note 2. Outputs from the PLL oscillator are frequency-divided by two at the final stage, so the actual multiplication factors become the values enclosed in parentheses.

Note 3. For transitions to the STOP mode, switch from the PLL clock to the high-speed on-chip oscillator clock (with f_{IH} = 8 MHz) and then stop the PLL.

Note 4. After PLL operation has started, an interval of waiting for lock-in (50 μs) is inserted for stabilization of the PLL clock frequency.

Caution 1. Be sure to set bits 7 and 5 to 0.

Caution 2. If the setting of the FRQSEL4 bit of the user option byte (000C2H) is 1, do not make the setting to start PLL operation (DSCON = 1).

Caution 3. If the CPU is operating with the subsystem clock, do not make the setting to start PLL operation (DSCON = 1).

9.3.18 Main clock control register (MCKC)

The MCKC register controls operation of the main clock. The MCKC register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 9 - 21 Format of Main Clock Control Register (MCKC)

Address: F02E6H
After reset: 00H
R/W: R/W**Note 1**

Symbol	<7>	6	5	4	3	<2>	<1>	<0>
MCKC	CLSTR	0	0	0	0	RDIV1	RDIV0	CKSELR

CLSTR	State of switching the source of the main system clock (f_{MAIN})
0	On-chip oscillator clock (f_{OCO})/high-speed system clock (f_{MX})**Note 2**
1	PLL frequency-divided clock (f_{PLLDIV})

RDIV1	RDIV0	Division ratio for the PLL frequency-divided clock (f_{PLLDIV})**Note 3**
0	0	Frequency division by two (f_{PLLDIV} = 48/32 MHz, f_{PLL} = 96/64 MHz)
0	1	Frequency division by four (f_{PLLDIV} = 24/16 MHz, f_{PLL} = 96/64 MHz)
Other than above		Setting prohibited

CKSELR	Selection of the main system clock (f_{MAIN})
0	On-chip oscillator clock (f_{OCO})/high-speed system clock (f_{MX})**Note 2**
1	PLL frequency-divided clock (f_{PLLDIV})**Notes 4, 5**

Note 1. Bit 7 is read-only.

Note 2. Use bit 4 (MCM0) of the system clock control register (CKC) to select the on-chip oscillator clock (f_{OCO}) or high-speed system clock (f_{MX}).

Note 3. Do not change the setting of the RDIV bits while the value of the CKSELR bit is 1.

Note 4. Do not stop the on-chip oscillator clock (f_{OCO}) or high-speed system clock (f_{MX}) while the PLL clock is selected.

Note 5. While the prefetch buffer is disabled (by the PFBE bit of the PFBER register being set to 0), the main system clock (f_{MAIN}) cannot run at 48 MHz (selected by setting the RDIV[1:0] bits and DSCM[4:1] bits of the DSCCTL register to 00B and 1011B, respectively). Therefore, do not set the CKSELR bit to 1 to select f_{PLL} as the main system clock (f_{MAIN}) in such cases.

Remark 1. The setting of the CKSELR bit is given priority over that of the MCM0 bit in selecting the source of the main system clock (f_{MAIN}).

Remark 2. Running the main system clock (f_{MAIN}) at 48 MHz requires that prefetching be enabled.
For the flowcharts for setting up and disabling prefetching, see **Section 8 Operation State Control**.

9.3.19 Peripheral clock control register (PCKC)

The PCKC register controls supply of the clock signal to the PLL oscillator. The PCKC register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 9 - 22 Format of Peripheral Clock Control Register (PCKC)

Address: F0098H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	<1>	0
PCKC	0	0	0	0	0	0	PLLCK	0

PLLCK	Control of the PLL operating clock supply
0	Stops supply of the on-chip oscillator clock (fOCO)/high-speed system clock (fMX).
1	Enables supply of the on-chip oscillator clock (fOCO)/high-speed system clock (fMX).**Note**

Note Use bit 4 (MCM0) of the system clock control register (CKC) to select the on-chip oscillator clock (fOCO) or high-speed system clock (fMX). Waiting for the PLL input clock to be stabilized (1 μs) is required after having set the PLLCK bit to 1.

9.4 System Clock Oscillator

9.4.1 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.

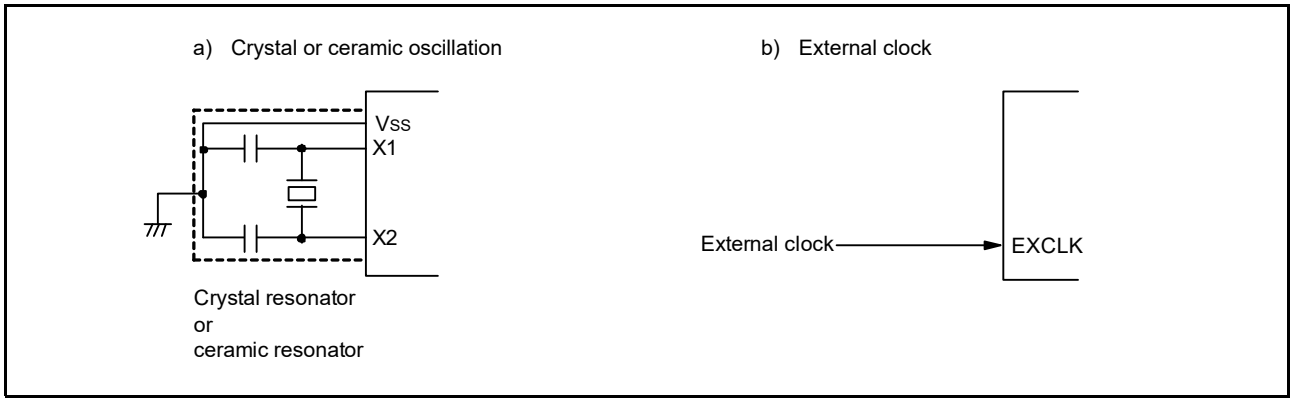
- Crystal or ceramic oscillation: EXCLK, OSCSEL = 0, 1
- External clock input: EXCLK, OSCSEL = 1, 1

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

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

Figure 9 - 23 shows examples of external circuits for the X1 oscillator.

Figure 9 - 23 Examples of External Circuits for the X1 Oscillator



Caution When using the X1 oscillator, wire as follows in the area enclosed in broken lines in the Figure 9 - 23 Examples of External Circuits for the X1 Oscillator 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 strong fluctuating current flows.**
- **Always apply the same voltage to the ground point for capacitors in the oscillator as that on Vss. Do not ground the capacitors to a ground pattern through which a strong current flows.**
- **Do not bring a signal line out from the oscillator.**

9.4.2 XT1 oscillator

The XT1 oscillator oscillates with a crystal resonator (32.768 kHz (typ.)) connected to the XT1 and XT2 pins.

To use the XT1 oscillator, set bit 4 (OSCSELS) of the clock operation mode control register (CMC) to 1.

An external clock can also be input. In this case, input the clock signal to the EXCLKS pin. To use the XT1 oscillator, set bits 5 and 4 (EXCLKS, OSCSELS) of the clock operation mode control register (CMC) as follows.

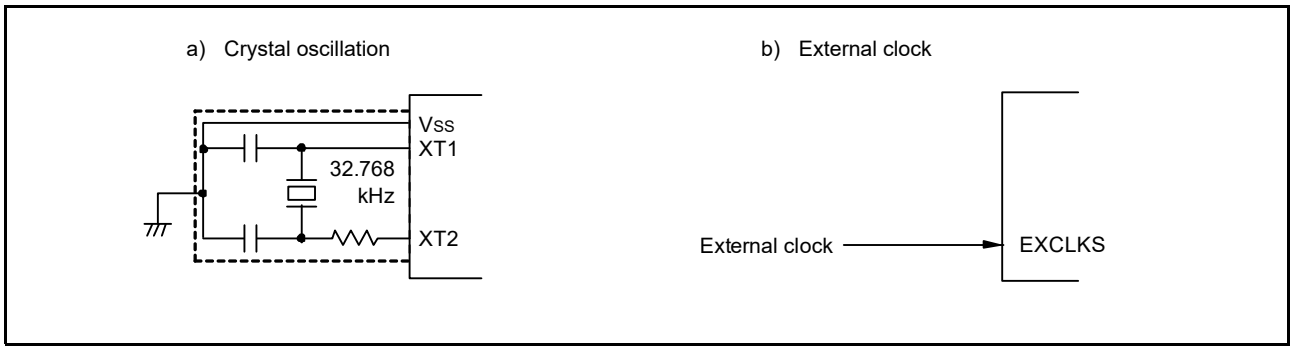
- Crystal oscillation: EXCLKS, OSCSELS = 0, 1
- External clock input: EXCLKS, OSCSELS = 1, 1

When the XT1 oscillator is not used, set the input port mode (EXCLKS, OSCSELS = 0, 0).

When XT1 oscillator is not used, and the pins are not used as input port pins, either, see **Table 2 - 2 Connections of Unused Pins**.

Figure 9 - 24 shows examples of external circuits for the XT1 oscillator.

Figure 9 - 24 Examples of External Circuits for the XT1 Oscillator



Caution 1. When using the XT1 oscillator, wire as follows in the area enclosed in broken lines in Figure 9 - 24 Examples of External Circuits for the XT1 Oscillator to avoid an adverse effect from wiring capacitance.

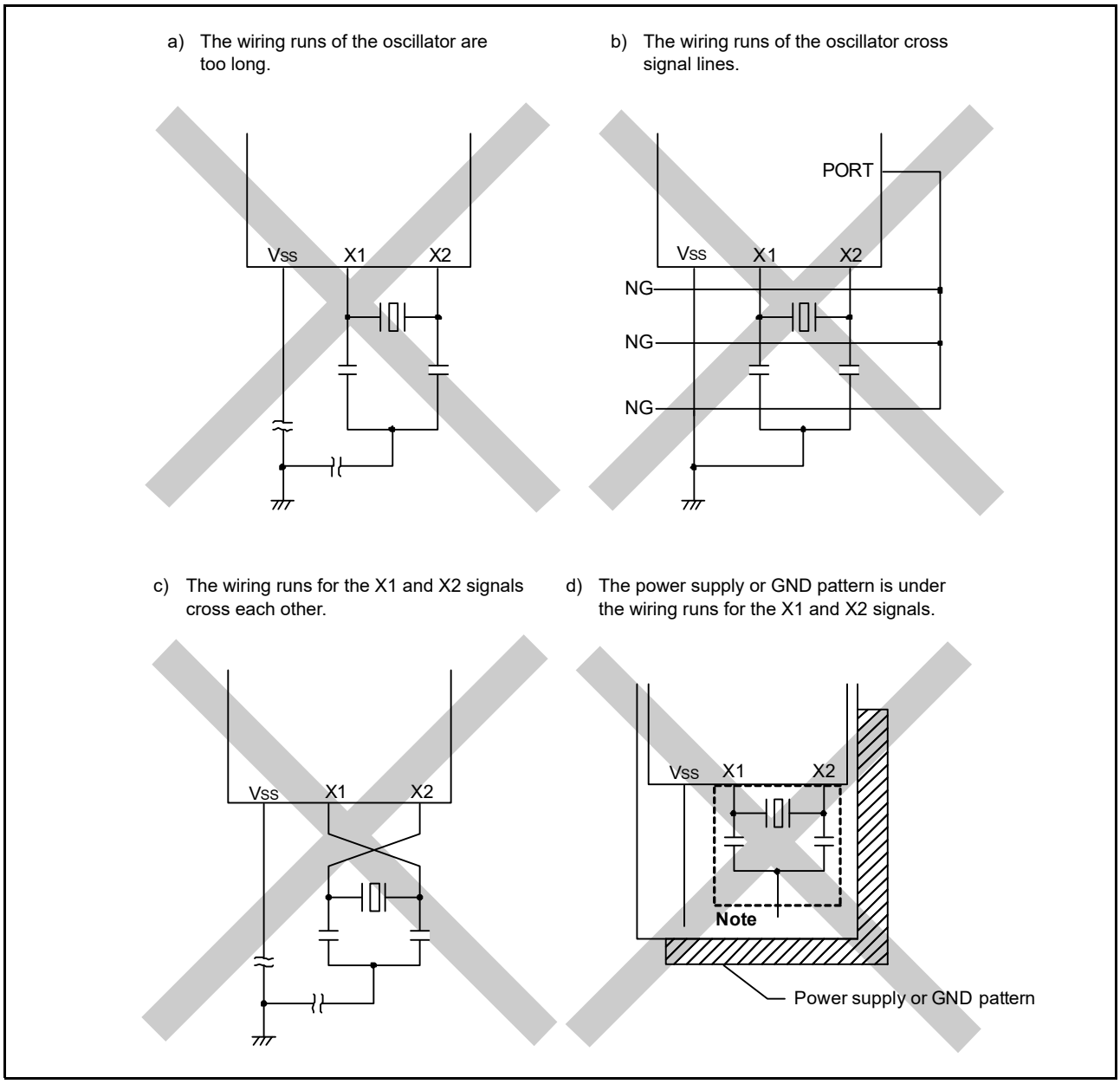
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a strong fluctuating current flows.**
- **Always apply the same voltage to the ground point for capacitors in the oscillator as that on Vss. Do not ground the capacitors to a ground pattern through which a strong current flows.**
- **Do not bring a signal line out from the oscillator.**

Caution 2. The XT1 oscillator is a circuit with low amplification in order to achieve low power consumption. Note the following points when designing the circuit.

- **Pins and circuit boards include parasitic capacitance. Therefore, perform oscillation evaluation using a circuit board to be actually used and confirm that there are no problems.**
- **When using low power consumption oscillation 2 (AMPHS[1:0] = 10B) or low power consumption oscillation 3 (AMPHS[1:0] = 11B) as the XT1 oscillator mode, sufficiently evaluate the resonator as described in 9.7 Resonator and Oscillator Constants, before using it in either of these modes.**
- **Make the wiring runs between the XT1 and XT2 pins and the resonator as short as possible to minimize the parasitic capacitance and wiring resistance. Take care with this particularly when low power consumption oscillation 2 (AMPHS[1:0] = 10B) or low power consumption oscillation 3 (AMPHS[1:0] = 11B) is selected.**
- **Configure the circuit of the circuit board, using material with little parasitic capacitance and wiring resistance.**
- **Place a ground pattern that has the same voltage as Vss as much as possible near the XT1 oscillator.**
- **Be sure that the signal lines between the XT1 and XT2 pins and the resonator do not cross the other signal lines. Do not route the wiring near a signal line through which a strong fluctuating current flows.**
- **The impedance between the XT1 and XT2 pins may drop and oscillation may be disturbed due to moisture absorption of the circuit board in a high-humidity environment or dew condensation on the board. When using the circuit board in such an environment, take measures to damp-proof the circuit board, such as by coating.**
- **When coating the circuit board, use material that does not cause capacitance or leakage between the XT1 and XT2 pins.**

Figure 9 - 25 shows examples of incorrect resonator connection.

Figure 9 - 25 Examples of Incorrect Resonator Connection (1/2)

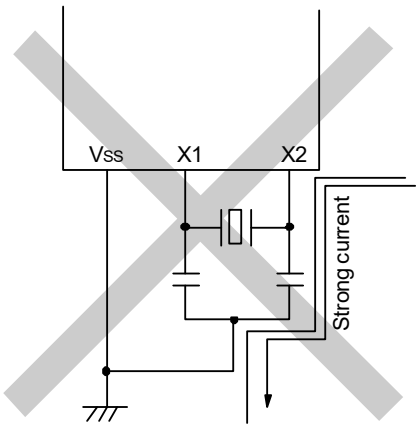


Note Do not place a power supply or GND pattern under the wiring section (section enclosed in broken lines in the figure) of the X1 and X2 pins and the resonator in a multi-layer board or double-sided board.
Do not configure a layout that will cause capacitance elements and affect the oscillation characteristics.

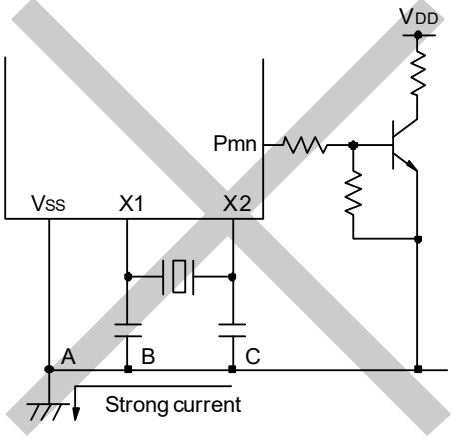
Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert a resistor in series on the XT2 side.

Figure 9 - 25 Examples of Incorrect Resonator Connection (2/2)

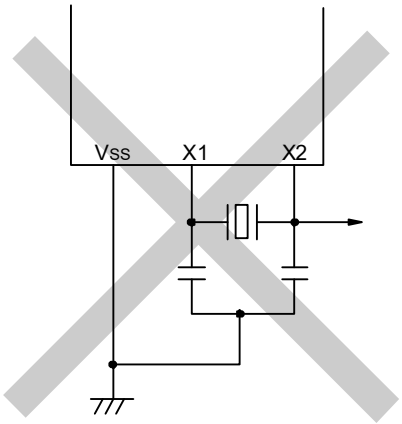
e) A signal line for the oscillator and a signal line through which a strong fluctuating current flows are close to each other.



f) Current flows through the ground line of the oscillator (making the voltage at points A, B, and C fluctuate).



g) A signal line is brought out.



Caution **When the wiring runs for the X2 and X1 signals are in parallel, crosstalk noise from the X2 signal line may be imposed on the X1 signal, resulting in malfunctions.**

Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert a resistor in series on the XT2 side.

9.4.3 High-speed on-chip oscillator

The high-speed on-chip oscillator is incorporated in the RL78/G24. The frequency can be selected from among 64, 48, 32, 24, 16, 12, 8, 6, 4, 3, 2, or 1 MHz by using a user option byte (000C2H). 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 release from the reset state.

9.4.4 Middle-speed on-chip oscillator

The middle-speed on-chip oscillator is incorporated in the RL78/G24. Oscillation can be controlled by bit 1 (MIOEN) of the clock operation status control register (CSC).

9.4.5 Low-speed on-chip oscillator

The low-speed on-chip oscillator is incorporated in the RL78/G24.
The low-speed on-chip oscillator operates when either of the following conditions is met.

- The watchdog timer is operating.
- The value of one or both of bit 4 (WUTMMCK0) of the subsystem clock supply mode control register (OSMC) and bit 0 (SELLOSC) of the subsystem clock select register (CKSEL) is 1.

The low-speed on-chip oscillator is stopped when the watchdog timer is stopped and both WUTMMCK0 and SELLOSC are set to 0.

9.4.6 PLL (phase-locked loop)

The PLL circuit is incorporated in the RL78/G24.
The PLL uses the 8-MHz high-speed on-chip oscillator clock or high-speed system clock as a reference clock and generates either of the two types of PLL clock (f_{PLL}) by multiplying the selected reference clock; 96- or 64-MHz clock is generated through multiplication by 24/2 (times 12) or 16/2 (times 8), respectively.
The multiplication factor can be controlled by bits 4 to 1 (DSCM[4:1]) of the PLL control register (DSCCTL).
Operation can be controlled by bit 0 (DSCON) of the PLL control register (DSCCTL).

Caution If the CPU is operating with the subsystem clock, do not make the setting to start PLL operation (DSCON = 1).
If the setting of the FRQSEL4 bit of the user option byte (000C2H) is 1, do not make the setting to start PLL operation (DSCON = 1).

9.5 Operations of the Clock Generator

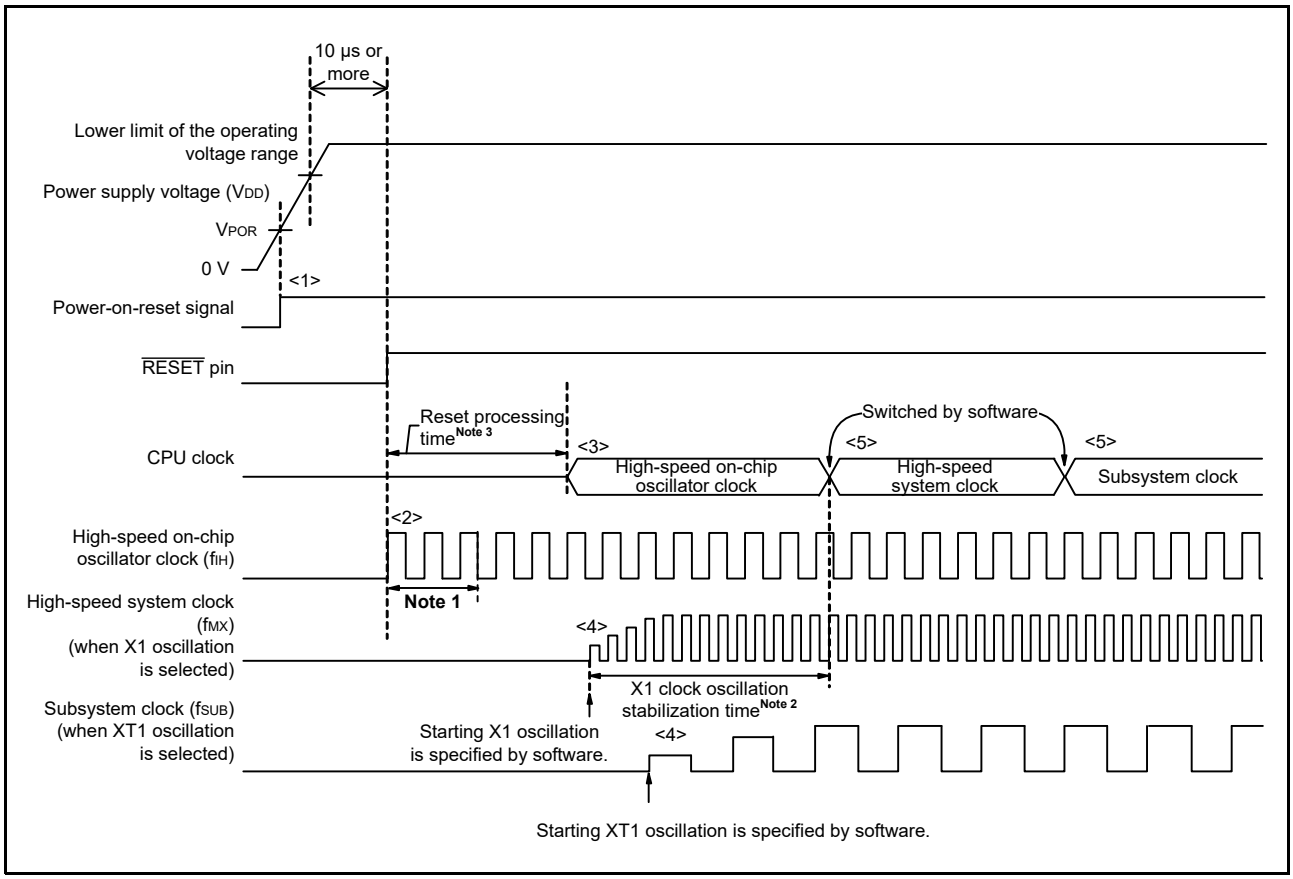
The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode (see **Figure 9 - 1 Block Diagram of Clock Generator**).

- Main system clocks (f_{MAIN})
  - High-speed system clocks (f_{MX})
    X1 clock (f_X)
    External main system clock (f_{EX})
  - High-speed on-chip oscillator clock (f_{IH})
  - Middle-speed on-chip oscillator clock (f_{IM})
  - PLL clock (f_{PLL})
- Subsystem clocks (f_{SUB})
  - XT1 clock (f_{XT})
  - External subsystem clock (f_{EXS})
  - Low-speed on-chip oscillator clock (f_{IL})
- CPU/peripheral hardware clock (f_{CLK})
- Subsystem clock X (f_{SX})
- Peripheral clocks
  - High-speed on-chip oscillator peripheral clock (f_{IHP})
  - Middle-speed on-chip oscillator peripheral clock (f_{IMP})
  - High-speed peripheral clock (f_{MXP})
  - Low-speed peripheral clock (f_{SXP})
  - Subsystem clock XR (f_{SXR})

The CPU starts operation when the high-speed on-chip oscillator starts outputting after release from the reset state in the RL78/G24.

Figure 9 - 26 shows the clock generator operation when power supply voltage is turned on.

Figure 9 - 26 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 **43.4 AC Characteristics** or **44.4 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 with the high-speed on-chip oscillator clock after waiting for the voltage to become stable and a reset processing have been performed after release from the reset state.

<4> Set the start of oscillation of the X1 or XT1 clock via software (see **9.6.2 Example of setting the X1 oscillator clock** and **9.6.3 Example of setting the XT1 oscillator clock**).

<5> When switching the CPU clock to the X1 or XT1 clock, wait for the clock oscillation to become stable, and then make the setting for switching via software (see **9.6.2 Example of setting the X1 oscillator clock** and **9.6.3 Example of setting the XT1 oscillator clock**).

Note 1. The internal 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 **Section 33 Power-on-reset Circuit (POR)**.

Caution Waiting for the oscillation stabilization time is not required when external clock input through the EXCLK pin is used.

9.6 Controlling Clocks

9.6.1 Example of setting the high-speed on-chip oscillator

After release from the reset state, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock as the source. The frequency of the high-speed on-chip oscillator can be selected from among 64, 48, 32, 24, 16, 12, 8, 6, 4, 3, 2, or 1 MHz by using the FRQSEL[4:0] bits of a user option byte (000C2H). In addition, the frequency can be changed by the high-speed on-chip oscillator frequency select register (HOCODIV).

[User option byte setting]

Address: 000C2H

	7	6	5	4	3	2	1	0
User 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

FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator clock	
					f_{HOCO}	f_{IH}
1	1	0	0	0	64 MHz	32 MHz
1	0	0	0	0	48 MHz	24 MHz
0	1	0	0	0	32 MHz	32 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	0	0	1	0	6 MHz	6 MHz
0	1	0	1	1	4 MHz	4 MHz
0	0	0	1	1	3 MHz	3 MHz
0	1	1	0	0	2 MHz	2 MHz
0	1	1	0	1	1 MHz	1 MHz
Other than above					Setting prohibited	

[Settings of the high-speed on-chip oscillator frequency select register (HOCODIV) and high-speed clock select register (HSCLKSEL)]

Address: F00A8H

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

Address: F0219H

Symbol	7	6	5	4	3	2	<1>	<0>
HSCLKSEL	0	0	0	0	0	0	FIHST	FIHSEL

HOCODIV2	HOCODIV1	HOCODIV0	Selection of the high-speed on-chip oscillator clock frequency					
			FIHSEL = 0				FIHSEL = 1	
			FRQSEL4 = 0		FRQSEL4 = 1		FRQSEL4 = 1	
			FRQSEL3 = 0	FRQSEL3 = 1	FRQSEL3 = 0	FRQSEL3 = 1	FRQSEL3 = 0	FRQSEL3 = 1
0	0	0	f_{IH} = 24 MHz	f_{IH} = 32 MHz	f_{IH} = 24 MHz f_{HOCO} = 48 MHz	f_{IH} = 32 MHz f_{HOCO} = 64 MHz	f_{IH} = 48 MHz f_{HOCO} = 48 MHz	Setting prohibited
0	0	1	f_{IH} = 12 MHz	f_{IH} = 16 MHz	f_{IH} = 12 MHz f_{HOCO} = 24 MHz	f_{IH} = 16 MHz f_{HOCO} = 32 MHz	Setting prohibited	Setting prohibited
0	1	0	f_{IH} = 6 MHz	f_{IH} = 8 MHz	f_{IH} = 6 MHz f_{HOCO} = 12 MHz	f_{IH} = 8 MHz f_{HOCO} = 16 MHz	Setting prohibited	Setting prohibited
0	1	1	f_{IH} = 3 MHz	f_{IH} = 4 MHz	f_{IH} = 3 MHz f_{HOCO} = 6 MHz	f_{IH} = 4 MHz f_{HOCO} = 8 MHz	Setting prohibited	Setting prohibited
1	0	0	Setting prohibited	f_{IH} = 2 MHz	Setting prohibited	f_{IH} = 2 MHz f_{HOCO} = 4 MHz	Setting prohibited	Setting prohibited
1	0	1	Setting prohibited	f_{IH} = 1 MHz	Setting prohibited	f_{IH} = 1 MHz f_{HOCO} = 2 MHz	Setting prohibited	Setting prohibited
Other than above			Setting prohibited					

9.6.2 Example of setting the X1 oscillator clock

After release from the reset state, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock as the source. To subsequently change the source clock to the X1 oscillator clock, set and start the oscillator 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 the oscillation to become stable by using the oscillation stabilization time counter status register (OSTC). After the oscillation becomes stable, set the X1 oscillator clock as the source of f_{CLK} by using the system clock control register (CKC).

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

<1> Set the OSCSEL bit of the CMC register to 1 to make the X1 oscillator operate. Also set the AMPH bit to 1 if f_X is more than 10 MHz.

	7	6	5	4	3	2	1	0
CMC	EXCLK 0	OSCSEL 1	EXCLKS 0	OSCSELS 0	XTSEL 0	AMPHS1 0	AMPHS0 0	AMPH 0/1

<2> Use the OSTS register to select the oscillation stabilization time for the X1 oscillator after release from the STOP mode.
Example: Set the register as shown below if waiting is to be for at least 102 μs with 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 the MSTOP bit of the CSC register to 0 to start X1 oscillation.

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

<4> Use the OSTC register to wait for oscillation of the X1 oscillator to become stable.
Example: Wait until counting has reached the following value if waiting is to be for at least 102 μs with 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 oscillator clock as the source of the CPU/peripheral hardware clock.

	7	6	5	4	3	2	1	0
CKC	CLS 0	CSS 0	MCS 0	MCM0 1	0	0	MCS1 0	MCM1 0

Caution When using the system clock control register (CKC) to change the main system clock (f_{MAIN}), do so while the voltage is within the usable range for the flash operation mode set in the option byte (000C2H) and in the flash operating mode select register (FLMODE) both before and after changing the clock. For details about the FLMODE register, see 8.2.1 Flash operating mode select register (FLMODE).

9.6.3 Example of setting the XT1 oscillator clock

After release from the reset state, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock as the source. To subsequently change the source clock to the XT1 oscillator clock, set and start the oscillator by using the subsystem clock supply mode control register (OSMC), clock operation mode control register (CMC), and clock operation status control register (CSC), and set the XT1 oscillator clock as the source of f_{CLK} by using the system clock control register (CKC).

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

<1> To select only running the realtime clock with the subsystem clock (for ultra-low current) in STOP mode or in HALT mode while the CPU is operating with the subsystem clock, set the RTCLPC bit to 1.

	7	6	5	4	3	2	1	0
OSMC	RTCLPC 0/1	0	0	WUTMMCK0 0	×	×	0	HIPREC 0

<2> Set the OSCSELS bit of the CMC register to 1 to make the XT1 oscillator operate. Also set the XTSEL bit to 1 in a product with 20 to 32 pins.

	7	6	5	4	3	2	1	0
CMC	EXCLK 0	OSCSEL 0	EXCLKS 0	OSCSELS 1	XTSEL 0/1	AMPHS1 0/1	AMPHS0 0/1	AMPH 0

AMPHS[1:0] bits: These bits are used to specify the oscillation mode of the XT1 oscillator.

<3> Clear the XTSTOP bit of the CSC register to 0 to start XT1 oscillation.

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

<4> Include code to wait for oscillation of the subsystem clock to become stable by using a timer or in other ways.

<5> Use the CSS bit of the CKC register to specify the XT1 oscillator clock as the source of the CPU/peripheral hardware clock.

	7	6	5	4	3	2	1	0
CKC	CLS 0	CSS 1	MCS 0	MCM0 0	0	0	MCS1 0	MCM1 0

Remark ×: Undefined

9.6.4 Example of setting the PLL circuit

Set the high-speed on-chip oscillator clock or high-speed system clock as the reference clock, and then use the PLL control register (DSCCTL) to make the settings for control over the PLL circuit.
When the PLL function is to be used, set the high-speed on-chip oscillator clock or high-speed system clock to run at 8 MHz.

[Register settings] Set the registers in the order of <1> to <6> below.

<1> Use the PLLDIV0 and DSCM[4:1] bits of the DSCCTL register to make the division ratio and multiplication factor settings for the PLL circuit.

	7	6	5	4	3	2	1	0
DSCCTL	0	PLLDIV0 1	0	DSCM4 0/1	DSCM3 0/1	DSCM2 0/1	DSCM1 0/1	DSCON 0

<2> Set the PLLCK bit in the PCKC register to start supply of the on-chip oscillator clock (fOCO)/high-speed system clock (fMX).

	7	6	5	4	3	2	1	0
PCKC	0	0	0	0	0	0	PLLCK 1	0

<3> Use the RDIV[1:0] bits of the MCKC register to make the division ratio setting for the main system clock.

	7	6	5	4	3	2	1	0
MCKC	CLSTR 0	0	0	0	0	RDIV1 0/1	RDIV0 0/1	CKSELR 0

<4> After waiting for at least 1 μs, set the DSCON bit of the DSCCTL register to 1 to make the PLL circuit operate.**Note**

	7	6	5	4	3	2	1	0
DSCCTL	0	PLLDIV0 1	0	DSCM4 0/1	DSCM3 0/1	DSCM2 0/1	DSCM1 0/1	DSCON 1

<5> After PLL operation has started, an interval of waiting for lock-in (50 μs) is inserted for stabilization of the PLL clock frequency. After the frequency has become stable, set the CKSELR bit of the MCKC register to 1 to select the PLL output as the main system clock.

	7	6	5	4	3	2	1	0
MCKC	CLSTR 0	0	0	0	0	RDIV1 0/1	RDIV0 0/1	CKSELR 1

<6> Check that the setting of the CLSTR status bit of the MCKC register has become 1 to confirm completion of switching of the CPU clock.

	7	6	5	4	3	2	1	0
MCKC	CLSTR 1	0	0	0	0	RDIV1 0/1	RDIV0 0/1	CKSELR 1

Note Only start PLL operation no less than 1 μs after oscillation of the high-speed on-chip oscillator clock or high-speed system clock has become stable.
In addition, when restarting PLL operation after it has been stopped, wait for at least 4 μs before restarting it.

9.6.5 State transitions of the CPU clock

Figure 9 - 27 shows the state transitions of the CPU clock.

Figure 9 - 27 State Transitions of the CPU Clock

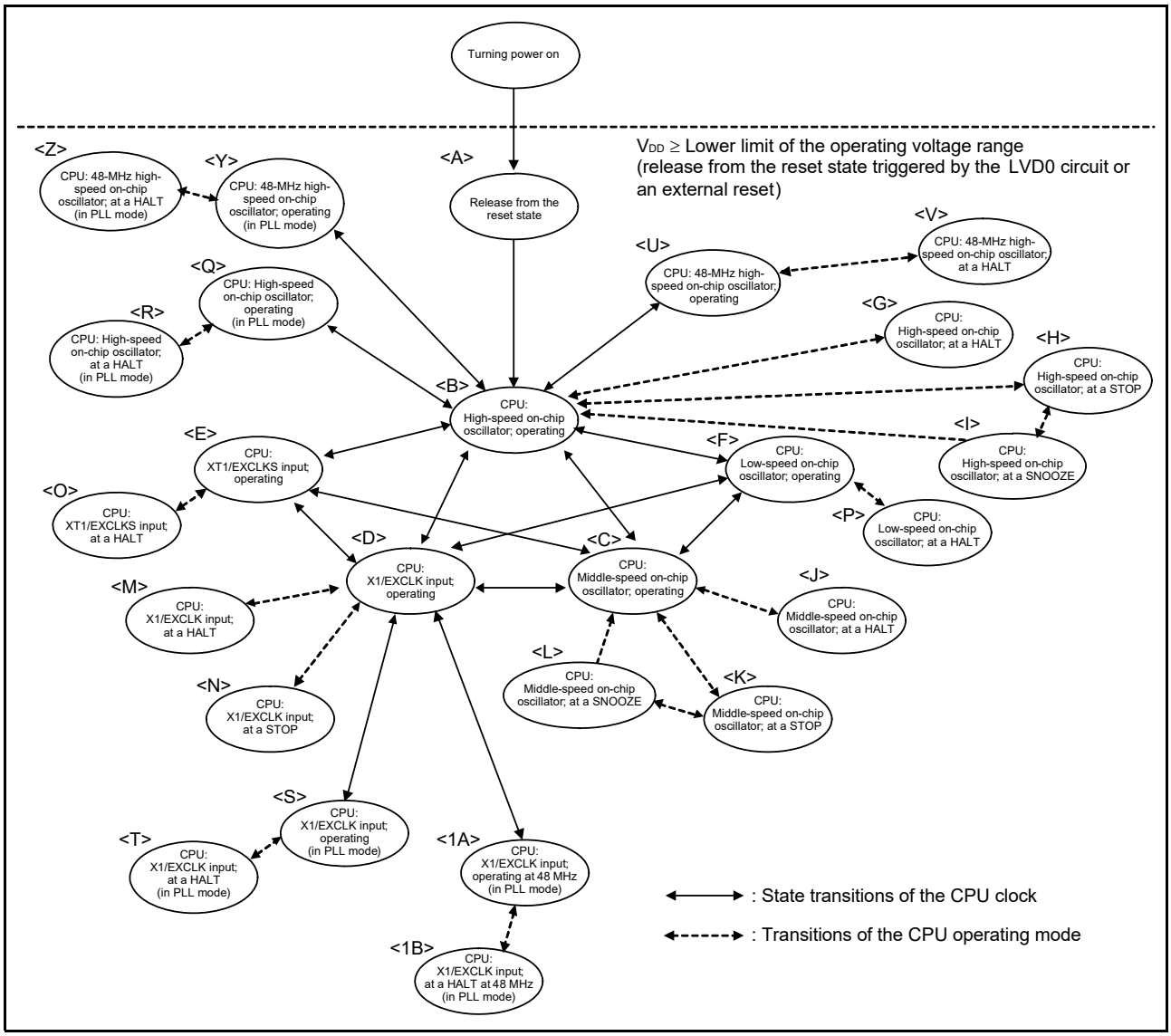


Table 9 - 2 shows examples of transitions of the CPU clock and SFR settings.

Table 9 - 2 Examples of Transitions of the CPU Clock and SFR Settings (1/4)

1. Transition to state <B> where the CPU is operating with the high-speed on-chip oscillator clock after release from the reset state <A>
   Scope of state transitions: <A> → <B>

Clock after Transition	SFR Setting
High-speed on-chip oscillator	SFR setting not required (default state after release from the reset state)

2. Transitions to state <B> where the CPU is operating with the high-speed on-chip oscillator clock
   Scope of state transitions: <C> → <B>, <D> → <B>, <E> → <B>, <F> → <B>

(Sequence of setting the SFRs) →

SFR to Be Set / Clock after Transition	CSC Register	Waiting for Oscillation Stabilization	CKC Register		
	HIOSTOP		CSS	MCM0	MCM1
High-speed on-chip oscillator	0	5 μs	0	0	0

Unnecessary if the high-speed on-chip oscillator clock is already running

   Scope of state transitions: <Q> → <B>, <Y> → <B>

(Sequence of setting the SFRs) →

SFR to Be Set / Clock after Transition	MCKC Register	Checking for Completion of Clock Switching	DSCCTL Register	PCKC Register
	CKSELR		DSCON	PLLCK
High-speed on-chip oscillator	0	Confirming the value of the CLSTR bit of the MCKC register is 0	0	0

3. Transitions to state <C> where the CPU is operating with the middle-speed on-chip oscillator clock
   Scope of state transitions: <B> → <C>, <D> → <C>, <E> → <C>, <F> → <C>

(Sequence of setting the SFRs) →

SFR to Be Set / Clock after Transition	CSC Register	Waiting for Oscillation Stabilization	CKC Register		
	MIOEN		CSS	MCM0	MCM1
Middle-speed on-chip oscillator	1	1 μs	0	0	1

Unnecessary if the middle-speed on-chip oscillator clock is already running

Remark <A> to <1B> in **Table 9 - 2 Examples of Transitions of the CPU Clock and SFR Settings** correspond to <A> to <1B> in **Figure 9 - 27 State Transitions of the CPU Clock**.

Table 9 - 2 Examples of Transitions of the CPU Clock and SFR Settings (2/4)

4. Transitions to states <Q>, <Y>, <S>, and <1A>, in which the PLL clock is operating
   Scope of state transitions: <B> → <Q>, <B> → <Y>, <D> → <S>, <D> → <1A>

(Sequence of setting the SFRs) →

SFR to Be Set / Clock after Transition	DSCCTL Register		PCKC Register	Waiting Time Until the PLL Clock Is Stabilized	DSCCTL Register	Lock-in Waiting Time	MCKC Register	Checking for Completion of Clock Switching
	PLLDIV0	DSCM [4:1]	PLLCK		DSCON		CKSELR	
PLL clock	1	0111B or 1011B	1	1 μs	1	50 μs	1	Confirming the value of the CLSTR bit of the MCKC register is 1

Caution Set 8 MHz for the high-speed on-chip oscillator clock or high-speed system clock.

5. Transitions to state <D> where the CPU is operating with the high-speed system clock
   Scope of state transitions: <B> → <D>, <C> → <D>, <E> → <D>**Note 1**, <F> → <D>

(Sequence of setting the SFRs) →

SFR to Be Set / Clock after Transition	CMC Register**Note 2**			OSTS Register	CSC Register	OSTC Register	CKC Register	
	EXCLK	OSCSEL	AMPH		MSTOP		CSS	MCM0
X1 clock: 1 MHz ≤ fx ≤ 10 MHz	0	1	0	**Note 3**	0	Must be checked	0	1
X1 clock: 10 MHz < fx ≤ 20 MHz	0	1	1	**Note 3**	0	Must be checked	0	1
External main clock	1	1	×	**Note 3**	0	Need not be checked	0	1

Unnecessary if these bits are already set

Unnecessary if the high-speed system clock is already running

Scope of state transitions: <S> → <D>, <1A> → <D>

(Sequence of setting the SFRs) →

SFR to Be Set / Clock after Transition	MCKC Register	Checking for Completion of Clock Switching	DSCCTL Register	PCKC Register
	CKSELR		DSCON	PLLCK
X1 clock or external main clock	0	Confirming the value of the CLSTR bit of the MCKC register is 0	0	0

Note 1. Products with 20 to 32 pins do not support this transition.

Note 2. The clock operation mode control register (CMC) can be changed only once after release from the reset state. This register setting is not necessary if it has already been set.
In the products with 20 to 32 pins, set XTSEL to 0.

(Note, Caution, and Remark are listed on the next page.)

Note 3. Set the oscillation stabilization time as follows.

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

Caution **Set a clock after the supply voltage has reached the operating voltage range for the clock to be set (see Section 43 Electrical Characteristics (T_A = –40 to +105°C) or Section 44 Electrical Characteristics (T_A = –40 to +125°C)).**

Remark <A> to <1B> in **Table 9 - 2 Examples of Transitions of the CPU Clock and SFR Settings** correspond to <A> to <1B> in **Figure 9 - 27 State Transitions of the CPU Clock**.

Table 9 - 2 Examples of Transitions of the CPU Clock and SFR Settings (3/4)

6. Transitions to state <E> where the CPU is operating with the subsystem clock
   Scope of state transitions: <B> → <E>, <C> →<E>, <D> → <E>**Note 1**

(Sequence of setting the SFRs) →

SFR to Be Set / Clock after Transition	CMC Register**Note 2**				CSC Register	Waiting for Oscillation Stabilization	CKC Register
	EXCLKS	OSCSELS	AMPHS1	AMPHS0	XTSTOP		CSS
XT1 clock	0	1	0/1	0/1	0	Necessary	1
External subsystem clock	1	1	×	×	×	Necessary	1

Unnecessary if these bits are already set

Unnecessary if the subsystem clock is already running

Note 1. Products with 20 to 32 pins do not support this transition.

Note 2. The clock operation mode control register (CMC) can be written only once by an 8-bit memory manipulation instruction after release from the reset state.
In the products with 20 to 32 pins, set XTSEL to 1.

7. Transitions to state <F> where the CPU is operating with the low-speed on-chip oscillator clock
   Scope of state transitions: <B> → <F>, <C> → <F>, <D> → <F>

(Sequence of setting the SFRs) →

SFR to Be Set / Clock after Transition	CKSEL	Waiting for Oscillation Accuracy Stabilization	CKC Register
	SELLOSC		CSS
Low-speed on-chip oscillator	1	80 μs	1

Unnecessary if the low-speed on-chip oscillator clock is already running

Remark 1. ×: Don't care.

Remark 2. <A> to <1B> in **Table 9 - 2 Examples of Transitions of the CPU Clock and SFR Settings** correspond to <A> to <1B> in **Figure 9 - 27 State Transitions of the CPU Clock**.

Table 9 - 2 Examples of Transitions of the CPU Clock and SFR Settings (4/4)

8. Transitions from the CPU operating mode <B>, <C>, <D>, <E>, <F>, <Q>, <S>, <U>, <Y>, or <1A> to the HALT mode <G>, <J>, <M>, <O>, <P>, <R>, <T>, <V>, <Z>, or <1B>
   Scope of state transitions: <B> → <G>, <C> → <J>, <D> → <M>, <E> → <O>, <F> → <P>, <Q> → <R>, <S> → <T>, <U> → <V>, <Y> → <Z>, <1A> → <1B>

Mode after Transition	Description
HALT mode	Executing a HALT instruction

9. Transitions from the CPU operating mode <B>, <C>, or <D> to the STOP mode <H>, <K>, or <N>
   Scope of state transitions: <B> → <H>, <C> →<K>, <D> → <N>

(Setting sequence) →

Mode after Transition	Description			
STOP mode	Stopping peripheral functions that cannot operate in STOP mode	Setting the OSTS register	Confirming the value of the HIPREC bit is 1.	Executing a STOP instruction

When STOP or SNOOZE mode is released
- FWKUP = 0 if the high-speed on-chip oscillator is started at normal speed.
- FWKUP = 1 if the high-speed on-chip oscillator is started at high speed.

Only necessary if the CPU is shifting from the state of operating with the high-speed on-chip oscillator to the STOP mode

Only necessary if the CPU is shifting from the state of operating with the high-speed system clock to the STOP mode

Only necessary if the high-speed on-chip oscillator is started at high speed

10. Transitions between a STOP mode <H> or <K> and SNOOZE mode <I> or <L>
    For details about the setting for switching from the STOP mode to the SNOOZE mode, see **Section 31 Standby Function** and descriptions of the SNOOZE mode functions of the on-chip peripheral modules.

Remark <A> to <1B> in **Table 9 - 2 Examples of Transitions of the CPU Clock and SFR Settings** correspond to <A> to <1B> in **Figure 9 - 27 State Transitions of the CPU Clock**.

9.6.6 Conditions before changing the CPU clock and processing after changing the CPU clock

The conditions before changing the CPU clock and processing after changing the CPU clock are shown below.

Table 9 - 3 Changing the CPU Clock (1/5)

CPU Clock		Conditions before Change	Processing after Change
Before Change	After Change		
High-speed on-chip oscillator clock	Middle-speed on-chip oscillator clock	The middle-speed on-chip oscillator is operating. • MIOEN = 1	Operating current can be reduced by stopping the high-speed on-chip oscillator (HIOSTOP = 1) after checking that the CPU clock is changed.
	X1 clock	X1 oscillation has become stable. • OSCSEL = 1, EXCLK = 0, MSTOP = 0, XTSEL = 0Note 1 • The oscillation stabilization time has elapsed.	
	External main system clock	External clock input from the EXCLK pin is enabled. • OSCSEL = 1, EXCLK = 1, MSTOP = 0, XTSEL = 0Note 1	
	PLL clock	The high-speed on-chip oscillator clock is operating at 8 MHz and being supplied as the operating clock for the PLL. • FRQSEL[4:0] = 01010B • HIOSTOP = 0 • PLLCK = 1 PLL oscillation has become stable. • PLLDIV0 = 1, DSCM[4:1] = 0111B or 1011B, DSCON = 1 • The lock-in waiting time (50 μs) has elapsed.	—
	XT1 clock	XT1 oscillation has become stable. • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0, XTSEL = 1Note 1 • The oscillation stabilization time has elapsed.	Operating current can be reduced by stopping the high-speed on-chip oscillator (HIOSTOP = 1) after checking that the CPU clock is changed.
	External subsystem clock	External clock input from the EXCLKS pin is enabled. • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0, XTSEL = 1Note 1	
	Low-speed on-chip oscillator clock	The low-speed on-chip oscillator is selected. • SELLOSC = 1	

Table 9 - 3 Changing the CPU Clock (2/5)

CPU Clock		Conditions before Change	Processing after Change
Before Change	After Change		
Middle-speed on-chip oscillator clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is operating. • HIOSTOP = 0	Operating current can be reduced by stopping the middle-speed on-chip oscillator (MIOEN = 0) after checking that the CPU clock is changed.
	X1 clock	X1 oscillation has become stable. • OSCSEL = 1, EXCLK = 0, MSTOP = 0, XTSEL = 0Note 1 • The oscillation stabilization time has elapsed.	
	PLL clock	Transition is not possible	—
	External main system clock	External clock input from the EXCLK pin is enabled. • OSCSEL = 1, EXCLK = 1, MSTOP = 0, XTSEL = 0Note 1	Operating current can be reduced by stopping the middle-speed on-chip oscillator (MIOEN = 0) after checking that the CPU clock is changed.
	XT1 clock	XT1 oscillation has become stable. • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0, XTSEL = 1Note 1 • The oscillation stabilization time has elapsed.	
	External subsystem clock	External clock input from the EXCLKS pin is enabled. • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0, XTSEL = 1Note 1	
	Low-speed on-chip oscillator clock	The low-speed on-chip oscillator is selected. • SELLOSC = 1	
PLL clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is operating and the high-speed on-chip oscillator clock is selected as the PLL operating clock. • HIOSTOP = 0, MCS = 0, MCS1 = 0	Operating current can be reduced by stopping the PLL (DSCON = 0) after checking that the CPU clock is changed. DSCON = 0 PLLCK = 0
	Middle-speed on-chip oscillator clock	Transition is not possible.	—
	X1 clock	The X1 oscillator is operating and the X1 oscillator clock is selected as the operating clock for the PLL. • OSCSEL = 1, EXCLK = 0, MSTOP = 0, XTSEL = 0Note 1, MCS = 1	Operating current can be reduced by stopping the PLL (DSCON = 0) after checking that the CPU clock is changed. DSCON = 0 PLLCK = 0
	External main system clock	External clock input from the EXCLK pin is enabled and the external main system clock is selected as the PLL source clock. • OSCSEL = 1, EXCLK = 1, MSTOP = 0, XTSEL = 0Note 1, MCS = 1	
	XT1 clock	Transition is not possible.	—
	External subsystem clock		
	Low-speed on-chip oscillator clock		

Table 9 - 3 Changing the CPU Clock (3/5)

CPU Clock		Conditions before Change	Processing after Change
Before Change	After Change		
X1 clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is enabled. • HIOSTOP = 0 • The oscillation stabilization time has elapsed.	X1 oscillation can be stopped (MSTOP = 1) after checking that the CPU clock is changed.
	Middle-speed on-chip oscillator clock	The middle-speed on-chip oscillator is operating. • MIOEN = 1	
	External main system clock	Transition is not possible.	—
	PLL clock	The X1 clock is operating at 8 MHz and being supplied as the operating clock for the PLL. • OSCSEL = 1, EXCLK = 0, MSTOP = 0, XTSEL=0Note 1, MCS = 1 PLL oscillation has become stable. • PLLDIV0 = 1, DSCM[4:1] = 0111B or 1011B, DSCON = 1 • The lock-in waiting time (50 μs) has elapsed.	—
	XT1 clockNote 2	XT1 oscillation has become stable. • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • The oscillation stabilization time has elapsed.	X1 oscillation can be stopped (MSTOP = 1) after checking that the CPU clock is changed.
	External subsystem clockNote 2	External clock input from the EXCLKS pin is enabled. • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
	Low-speed on-chip oscillator clock	The XT1 oscillator is stopped. The low-speed on-chip oscillator is selected. • SELLOSC = 1	
External main system clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is enabled. • HIOSTOP = 0 • The oscillation stabilization time has elapsed.	External main system clock input can be disabled (MSTOP = 1) after checking that the CPU clock is changed.
	Middle-speed on-chip oscillator clock	The middle-speed on-chip oscillator is operating. • MIOEN = 1	
	PLL clock	The external clock from the EXCLK pin is operating at 8 MHz and being supplied as the operating clock for the PLL. • OSCSEL = 1, EXCLK = 0, MSTOP = 0, XTSEL = 0Note 1, MCS = 1 PLL oscillation has become stable. • PLLDIV0 = 1, DSCM[4:1] = 0111B or 1011B, DSCON = 1 • The lock-in waiting time (50 μs) has elapsed.	—
	X1 clock	Transition is not possible.	—
	XT1 clockNote 2	XT1 oscillation has become stable. • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • The oscillation stabilization time has elapsed.	External main system clock input can be disabled (MSTOP = 1) after checking that the CPU clock is changed.
	External subsystem clockNote 2	External clock input from the EXCLKS pin is enabled. • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
	Low-speed on-chip oscillator clock	The XT1 oscillator is stopped. The low-speed on-chip oscillator is selected. SELLOSC = 1	

Table 9 - 3 Changing the CPU Clock (4/5)

CPU Clock		Conditions before Change	Processing after Change
Before Change	After Change		
XT1 clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is operating and the high-speed on-chip oscillator clock is selected as the main system clock. • HIOSTOP = 0, MCS = 0, MCS1 = 0	XT1 oscillation can be stopped (XTSTOP = 1) after checking that the CPU clock is changed.
	Middle-speed on-chip oscillator clock	The middle-speed on-chip oscillator is operating and the middle-speed on-chip oscillator clock is selected as the main system clock. • MIOEN = 1, MCS = 0, MCS1 = 1	
	PLL clock	Setting prohibited. This change requires two steps, that is, switching from the XT1 clock to the high-speed on-chip oscillator clock or X1 clock, and then switching from the high-speed on-chip oscillator clock or X1 clock to the PLL clock.	—
	X1 clock**Note 2**	X1 oscillation has become stable and the high-speed system clock is selected as the main system clock. • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • The oscillation stabilization time has elapsed. • MCS = 1	XT1 oscillation can be stopped (XTSTOP = 1) after checking that the CPU clock is changed.
	External main system clock**Note 2**	External clock input from the EXCLK pin is enabled and the high-speed system clock is selected as the main system clock. • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	External subsystem clock	Transition is not possible.	—
	Low-speed on-chip oscillator clock	Transition is not possible.	
External subsystem clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is operating and the high-speed on-chip oscillator clock is selected as the main system clock. • HIOSTOP = 0, MCS = 0, MCS1 = 0	External subsystem clock input can be disabled (XTSTOP = 1) after checking that the CPU clock is changed.
	Middle-speed on-chip oscillator clock	The middle-speed on-chip oscillator is operating and the middle-speed on-chip oscillator clock is selected as the main system clock. • MIOEN = 1, MCS = 0, MCS1 = 1	
	PLL clock	Setting prohibited. This change requires two steps, that is, switching from the external subsystem clock to the high-speed on-chip oscillator clock or X1 clock, and then switching from the high-speed on-chip oscillator clock or X1 clock to the PLL clock.	—
	X1 clock**Note 2**	X1 oscillation has become stable and the high-speed system clock is selected as the main system clock. • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • The oscillation stabilization time has elapsed. • MCS = 1	External subsystem clock input can be disabled (XTSTOP = 1) after checking that the CPU clock is changed.
	External main system clock**Note 2**	External clock input from the EXCLK pin is enabled and the high-speed system clock is selected as the main system clock. • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	XT1 clock	Transition is not possible.	—
	Low-speed on-chip oscillator clock	Transition is not possible.	

Table 9 - 3 Changing the CPU Clock (5/5)

CPU Clock		Conditions before Change	Processing after Change
Before Change	After Change		
Low-speed on-chip oscillator clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is operating and the high-speed on-chip oscillator clock is selected as the main system clock. • HIOSTOP = 0, MCS = 0, MCS1 = 0	—
	Middle-speed on-chip oscillator clock	The middle-speed on-chip oscillator is operating and the middle-speed on-chip oscillator clock is selected as the main system clock. • MIOEN = 1, MCS = 0, MCS1 = 1	
	PLL clock	Setting prohibited. This change requires two steps, that is, switching from the low-speed on-chip oscillator clock to the high-speed on-chip oscillator clock or X1 clock, and then switching from the high-speed on-chip oscillator clock or X1 clock to the PLL clock.	
	X1 clock	X1 oscillation has become stable and the high-speed system clock is selected as the main system clock. • OSCSEL = 1, EXCLK = 0, MSTOP = 0, XTSEL = 0**Note 1** • The oscillation stabilization time has elapsed. • MCS = 1	
	External main system clock	External clock input from the EXCLK pin is enabled and the high-speed system clock is selected as the main system clock. • OSCSEL = 1, EXCLK = 1, MSTOP = 0, XTSEL = 0**Note 1** • MCS = 1	
	XT1 clock	Transition is not possible.	
	External subsystem clock	Transition is not possible.	

Note 1. Writing to the XTSEL bit is only possible in the products with 20 to 32 pins.

Note 2. Switching to this clock is only possible in the products with 40 to 64 pins.

9.6.7 Time required for switchover of the CPU clock and main system clock

By setting bits 6, 4, 0 (CSS, MCM0, MCM1) of the system clock control register (CKC), the CPU clock can be switched between the main system clock and subsystem clock, the main system clock can be switched between the on-chip oscillator clock and high-speed system clock, and the on-chip oscillator clock can be switched between the high-speed on-chip oscillator clock and middle-speed on-chip oscillator clock.

In actual operation, the clock is not switched immediately after writing to the CKC register; the CPU continues to operate with the prior clock for several clock cycles after writing proceeds (see **Table 9 - 4 Maximum Time Required for Main System Clock Switchover** to **Table 9 - 7 Maximum Number of Clock Cycles Required for f_{MAIN} ↔ f_{SUB}**).

Whether the source of the CPU clock is the main system clock or subsystem clock can be ascertained from bit 7 (CLS) of the CKC register. Whether the source of the main system clock is the high-speed system clock, main on-chip oscillator clock, or PLL clock can be ascertained from bit 5 (MCS) of the CKC register and bit 7 (CLSTR) of the MCKC register. Whether the source of the main on-chip oscillator clock is the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock can be ascertained from bit 1 (MCS1) of the CKC register.

When the CPU clock is switched, the peripheral hardware clock is also switched.

Table 9 - 4 Maximum Time Required for Main System Clock Switchover

Clock A	Switching Directions	Clock B	Remark
f_{OCO}	◄►	f_{MX}	See **Table 9 - 5 Maximum Number of Clock Cycles Required for f_{OCO} ↔ f_{MX}.**
f_{IH}	◄►	f_{IM}	See **Table 9 - 6 Maximum Number of Clock Cycles Required for f_{IH} ↔ f_{IM}.**
f_{MAIN}	◄►	f_{SUB}	See **Table 9 - 7 Maximum Number of Clock Cycles Required for f_{MAIN} ↔ f_{SUB}.**

Table 9 - 5 Maximum Number of Clock Cycles Required for f_{OCO} ↔ f_{MX}

Set Value before Switchover		Set Value after Switchover	
MCM0		MCM0	
		0 (f_{MAIN} = f_{OCO})	1 (f_{MAIN} = f_{MX})
0 (f_{MAIN} = f_{OCO})	f_{MX} ≥ f_{OCO}		2 cycles
	f_{MX} < f_{OCO}		2 f_{OCO}/f_{MX} cycles
1 (f_{MAIN} = f_{MX})	f_{MX} ≥ f_{OCO}	2 f_{MX}/f_{OCO} cycles	
	f_{MX} < f_{OCO}	2 cycles	

Table 9 - 6 Maximum Number of Clock Cycles Required for f_{IH} ↔ f_{IM}

Set Value before Switchover		Set Value after Switchover	
MCM1		MCM1	
		0 (f_{OCO} = f_{IH})	1 (f_{OCO} = f_{IM})
0 (f_{OCO} = f_{IH})	f_{IM} ≥ f_{IH}		2 cycles
	f_{IM} < f_{IH}		2 f_{IH}/f_{IM} cycles
1 (f_{OCO} = f_{IM})	f_{IM} ≥ f_{IH}	2 f_{IM}/f_{IH} cycles	
	f_{IM} < f_{IH}	2 cycles	

Table 9 - 7 Maximum Number of Clock Cycles Required for f_{MAIN} ↔ f_{SUB}

Set Value before Switchover	Set Value after Switchover	
CSS	CSS	
	0 (f_{CLK} = f_{MAIN})	1 (f_{CLK} = f_{SUB})
0 (f_{CLK} = f_{MAIN})		1 + 2 f_{MAIN}/f_{SUB} cycles
1 (f_{CLK} = f_{SUB})	3 cycles	

Remark 1. The number of clock cycles listed in **Table 9 - 5 Maximum Number of Clock Cycles Required for foco ↔ fMX**, **Table 9 - 6 Maximum Number of Clock Cycles Required for fIH ↔ fIM**, and **Table 9 - 7 Maximum Number of Clock Cycles Required for fMAIN ↔ fSUB** is the number of cycles of the CPU clock before switchover.

Remark 2. Calculate the number of clock cycles in **Table 9 - 5 Maximum Number of Clock Cycles Required for foco ↔ fMX**, **Table 9 - 6 Maximum Number of Clock Cycles Required for fIH ↔ fIM**, and **Table 9 - 7 Maximum Number of Clock Cycles Required for fMAIN ↔ fSUB** by rounding up the number after the decimal position.

Example: When switching the main system clock from the high-speed on-chip oscillator clock (8 MHz selected) to the high-speed system clock (in this case, f_{IH} = 8 MHz, f_{MX} = 10 MHz)

1 + f_{IH}/f_{MX} = 1 + 8/10 = 1 + 0.8 = 1.8 → 2 cycles

9.6.8 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.

When stopping the clock, confirm the conditions before clock oscillation is stopped.

Table 9 - 8 Conditions Before the Clock Oscillation Is Stopped and Flag Settings

Clock	Conditions Before Clock Oscillation Is Stopped (External Clock Input Is Disabled)	Flag Settings of the SFRs
High-speed on-chip oscillator clock	MCS1 = 1, MCS = 1 or CLS = 1 (The CPU is operating with a clock other than the high-speed on-chip oscillator clock.)	HIOSTOP = 1
Middle-speed on-chip oscillator clock	MCS1 = 0, MCS = 1 or CLS = 1 (The CPU is operating with a clock other than the middle-speed on-chip oscillator clock.)	MIOEN = 0
X1 clock External main system clock	MCS = 0 or CLS = 1 (The CPU is operating with a clock other than the high-speed system clock.)	MSTOP = 1
XT1 clock External subsystem clock	CLS = 0 (The CPU is operating with a clock other than the subsystem clock.)	XTSTOP = 1
Low-speed on-chip oscillator clock**Note**	CLS = 0 (The CPU is operating with a clock other than the low-speed on-chip oscillator clock.)	SELLOSC = 0 WUTMMCK0 = 0
PLL clock	CLSTR = 0 (The CPU is operating with a clock other than the PLL clock.)	DSCON = 0

Note The low-speed on-chip oscillator clock is not stopped while the WDT is operating.

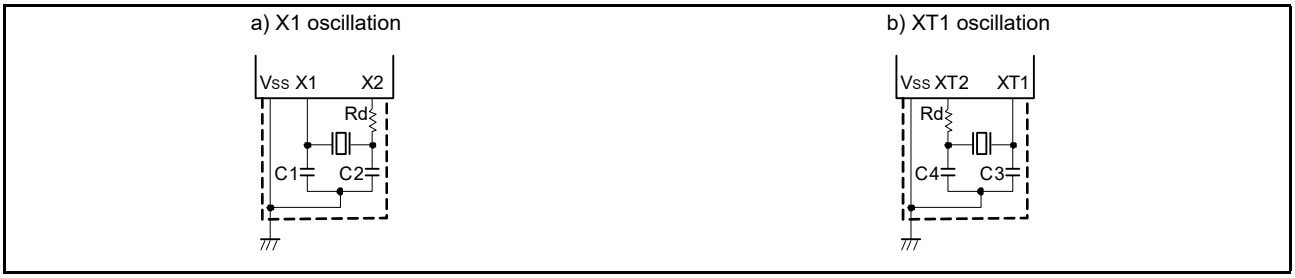
9.7 Resonator and Oscillator Constants

For the resonators for which operation has been verified and their oscillator constants (reference values), see the target product page on the Renesas Web site.

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 9 - 28 Examples of External Circuits



Section 10 Timer Array Unit (TAU)

Timer array unit has one unit with four channels.

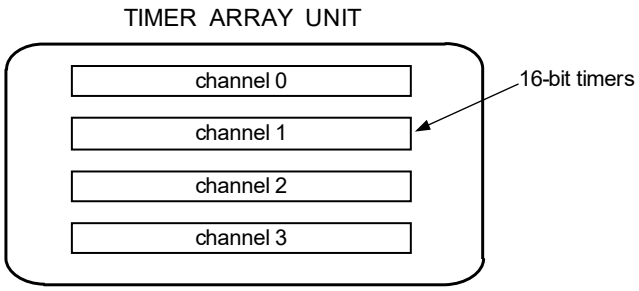
Units	Channels	20-, 24-, 25-, 30-, 32-, 40-, 44-, 48-, 52-, and 64-pin
Unit 0	Channel 0	✓
	Channel 1	✓
	Channel 2	✓
	Channel 3	✓

Caution 1. The presence or absence of timer I/O pins depends on the product. See Table 10 - 2 Timer I/O Pins Provided in Each Product for details.

Caution 2. Most of the following descriptions in this section use the 64-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 **10.8.1 Operation as an interval timer or for square wave output**) • Square wave output (→ refer to **10.8.1 Operation as an interval timer or for square wave output**) • External event counter (→ refer to **10.8.2 Operation as an external event counter**) • Divider[Note] (→ refer to **10.8.3 Operation as a frequency divider (channel 0 only)**) • Input pulse interval measurement (→ refer to **10.8.4 Operation for input pulse interval measurement**) • Measurement of high-/low-level width of input signal (→ refer to **10.8.5 Operation for input signal high-/low-level width measurement**) • Delay counter (→ refer to **10.8.6 Operation as a delay counter**)	• One-shot pulse output (→ refer to **10.9.1 Operation for the one-shot pulse output function**) • PWM output (→ refer to **10.9.2 Operation for the PWM function**) • Multiple PWM output (→ refer to **10.9.3 Operation for the multiple PWM output function**)

Note Channel 0 only supports this.

It is possible to use the 16-bit timer of channels 1 and 3 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)

Channel 3 can be used to realize LIN-bus communication operating in combination with UART0 of the serial array unit.

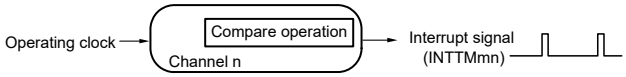
10.1 Functions of Timer Array Unit

Timer array unit has the following functions.

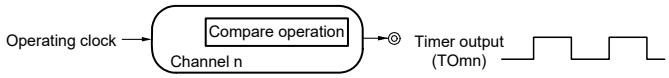
10.1.1 Independent channel operation function

By operating a channel independently, it can be used for the following seven purposes without being affected by the operation mode of other channels.

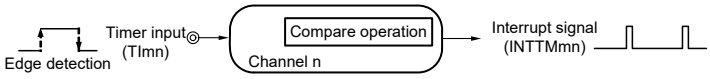
1. Interval timer
   Each timer of a unit can be used as a reference timer that generates an interrupt (INTTMmn) at fixed intervals.



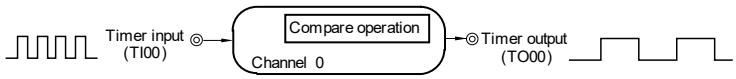
2. Square wave output
   A toggle operation is performed each time INTTMmn interrupt is generated and a square wave with a duty factor of 50% is output from a timer output pin (TOmn).



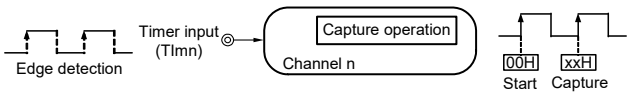
3. External event counter
   Each timer of a unit can be used as an event counter that generates an interrupt when the number of the valid edges of a signal input to the timer input pin (TImn) has reached a specific value.



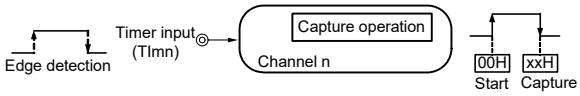
4. Divider function (channel 0 only)
   A clock input from a timer input pin (TI00) is divided and output from an output pin (TO00).



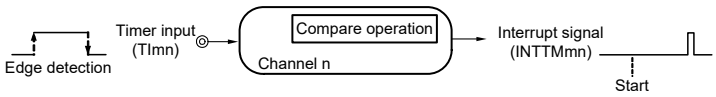
5. Input pulse interval measurement
   Counting is started by the valid edge of a pulse signal input to a timer input pin (TImn). The count value of the timer is captured at the valid edge of the next pulse. In this way, the interval of the input pulse can be measured.



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.



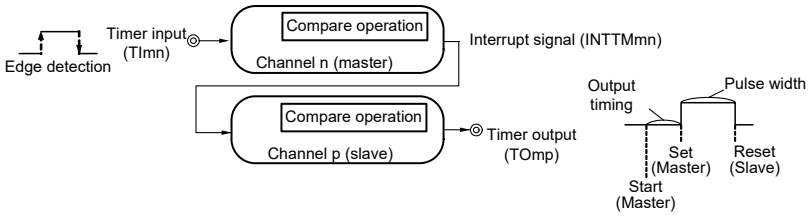
Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Remark 2. The presence or absence of timer I/O pins of channels 0 to 3 depends on the product. See **Table 10 - 2 Timer I/O Pins Provided in Each Product** for details.

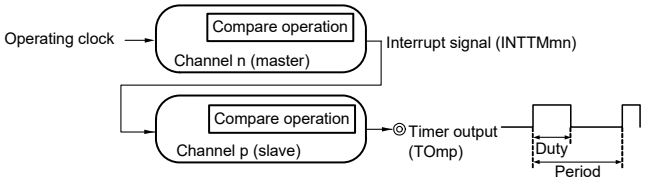
10.1.2 Simultaneous channel operation function

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 three purposes.

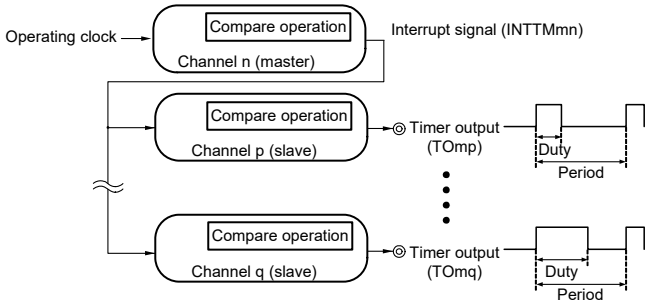
1. One-shot pulse output
   Two channels are used as a set to generate a one-shot pulse with a specified output timing and a specified pulse width.



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.



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 **For details about the rules of simultaneous channel operation function, see 10.4.1 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)

10.1.3 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 10.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only).

10.1.4 LIN-bus supporting function (channel 3 only)

Timer array unit is used to check whether signals received in LIN-bus communication match the LIN-bus communication format.

1. Detection of wakeup signal
   The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD0) of UART0 and the count value of the timer is captured at the rising edge. In this way, a low-level width can be measured. If the low-level width is greater than a specific value, it is recognized as a wakeup signal.
2. Detection of break field
   The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD0) of UART0 after a wakeup signal is detected, and the count value of the timer is captured at the rising edge. In this way, a low-level width is measured. If the low-level width is greater than a specific value, it is recognized as a break field.
3. Measurement of pulse width of sync field
   After a break field is detected, the low-level width and high-level width of the signal input to the serial data input pin (RxD0) of UART0 are measured. From the bit interval of the sync field measured in this way, a baud rate is calculated.

Remark For details about setting up the operations used to implement the LIN-bus, see **10.3.14 Input switch control register (ISC)** and **10.8.5 Operation for input signal high-/low-level width measurement**.

10.2 Configuration of Timer Array Unit

Timer array unit includes the following hardware.

Table 10 - 1 Configuration of Timer Array Unit

Item	Configuration
Timer/counter	Timer counter register mn (TCRmn)
Register	Timer data register mn (TDRmn)
Timer input	TI00 to TI03 pins**Note**, RxD0 pin (for LIN-bus)
Timer output	TO00 to TO03 pins**Note**
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Peripheral reset control register 0 (PRR0) • Timer clock select register m (TPSm) (m = 0) • Timer channel enable status register m (TEm) (m = 0) • Timer channel start register m (TSm) (m = 0) • Timer channel stop register m (TTm) (m = 0) • Timer I/O select register 0 (TIOS0) • Timer output enable register m (TOEm) (m = 0) • Timer output register m (TOm) (m = 0) • Timer output level register m (TOLm) (m = 0) • Timer output mode register m (TOMm) (m = 0)
	<Registers of each channel> • Timer mode register mn (TMRmn) (m = 0, n = 0 to 3) • Timer status register mn (TSRmn) (m = 0, n = 0 to 3) • Input switch control register (ISC) • Noise filter enable register 1 (NFEN1) • Port mode registers xx (PMxx) (xx = 0, 1, 3, 5, 7, 12) • Port registers xx (Pxx) (xx = 0, 1, 3, 7, 12) • Port output mode registers xx (POMxx) (xx = 1, 7) • Port mode control A registers xx (PMCAxx) (xx = 0, 1, 12)

Note The presence or absence of timer I/O pins of channels 0 to 3 depends on the product. See **Table 10 - 2 Timer I/O Pins Provided in Each Product** for details.

The presence or absence of timer I/O pins in each timer array unit channel depends on the product.

Table 10 - 2 Timer I/O Pins Provided in Each Product

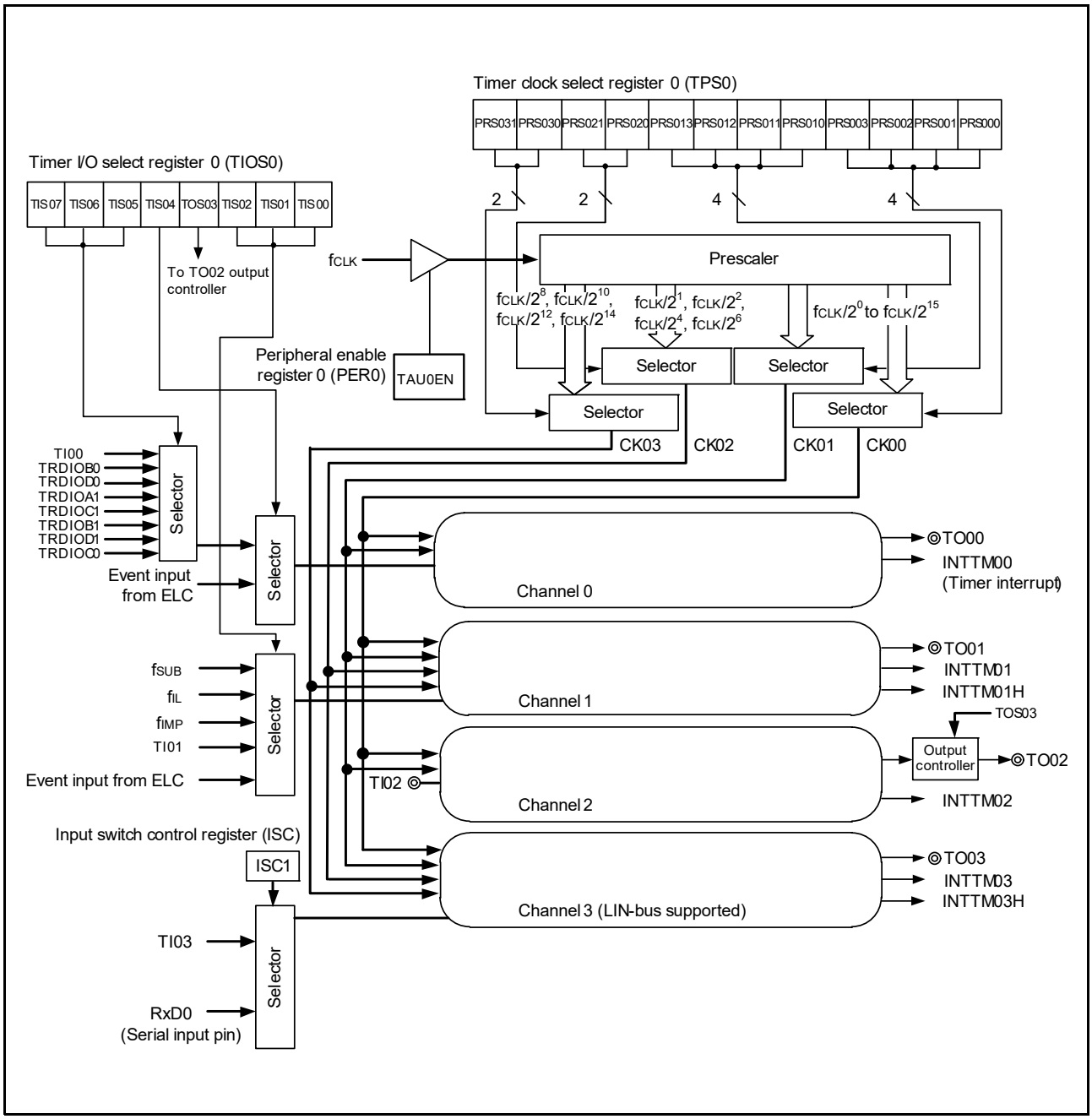
Timer Array Unit Channels		I/O Pins of Each Product		
		20-pin	24- and 25-pin	30-, 32-, 40-, 44-, 48-, 52-, and 64-pin
Unit 0	Channel 0	TI00/TO00 (TI00/TO00)		
	Channel 1	(TI01/TO01)	TI01/TO01 (TI01/TO01)	
	Channel 2	(TI02/TO02)	TI02/TO02 (TI02/TO02)	
	Channel 3	(TI03/TO03)		TI03/TO03 (TI03/TO03)

Remark 1. When timer input and timer output are shared by the same pin, either only timer input or only timer output can be used.

Remark 2. Pins in the parentheses indicate an alternate port when peripheral I/O redirection register 3 (PIOR3) is set. For details, see **7.3.8 Peripheral I/O redirection registers x (PIORx)**.

Figure 10 - 1 shows an entire configuration of the timer array unit for a 64-pin product as an example.

Figure 10 - 1 Entire Configuration of the Timer Array Unit (Example: 64-pin Products)



Remark fSUB: Subsystem clock frequency
fIL: Low-speed on-chip oscillator clock frequency
fIMP: Middle-speed on-chip oscillator peripheral clock frequency

Figure 10 - 2 Internal Block Diagram of Channel 0 of the Timer Array Unit

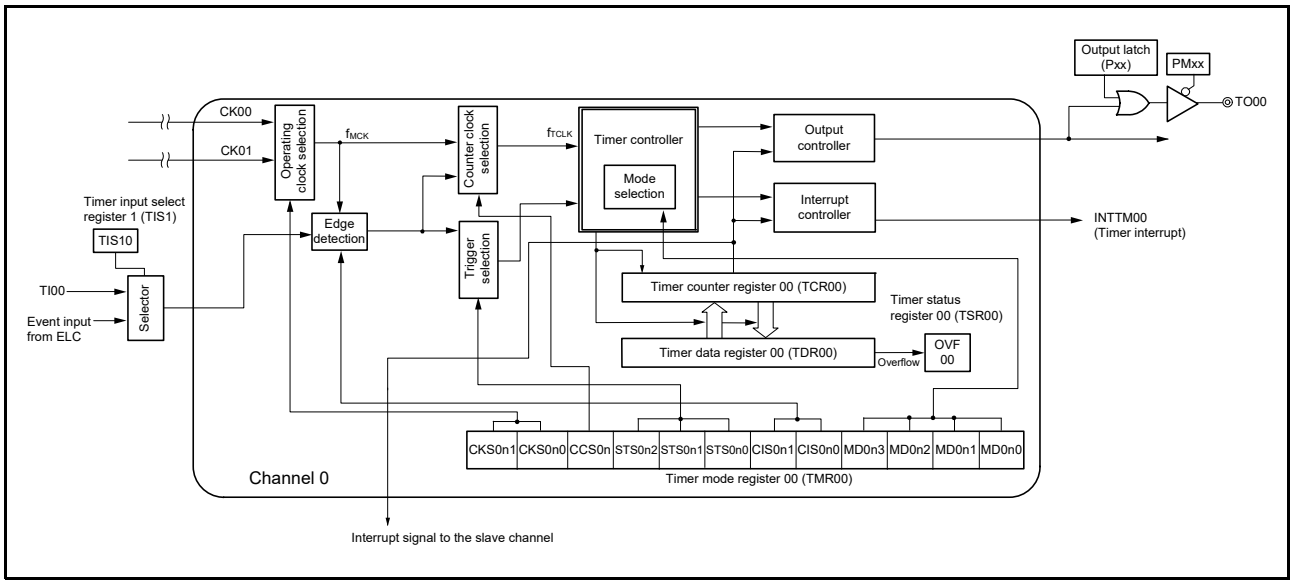


Figure 10 - 3 Internal Block Diagram of Channel 1 of the Timer Array Unit

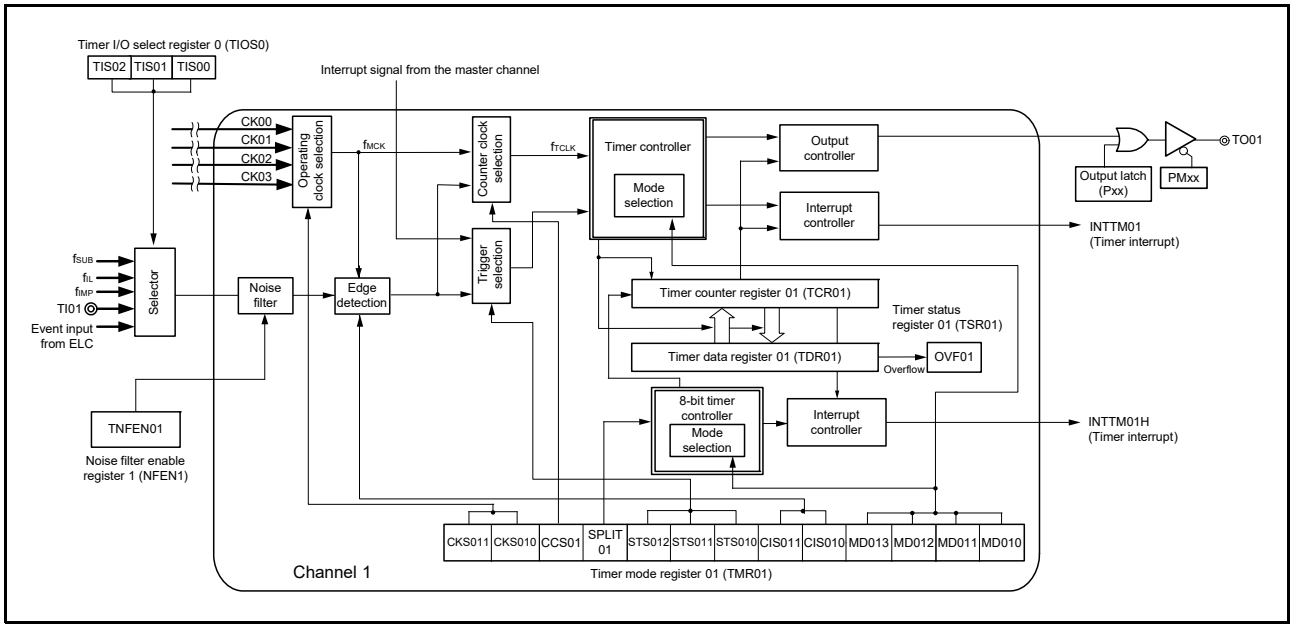


Figure 10 - 4 Internal Block Diagram of Channel 2 of the Timer Array Unit

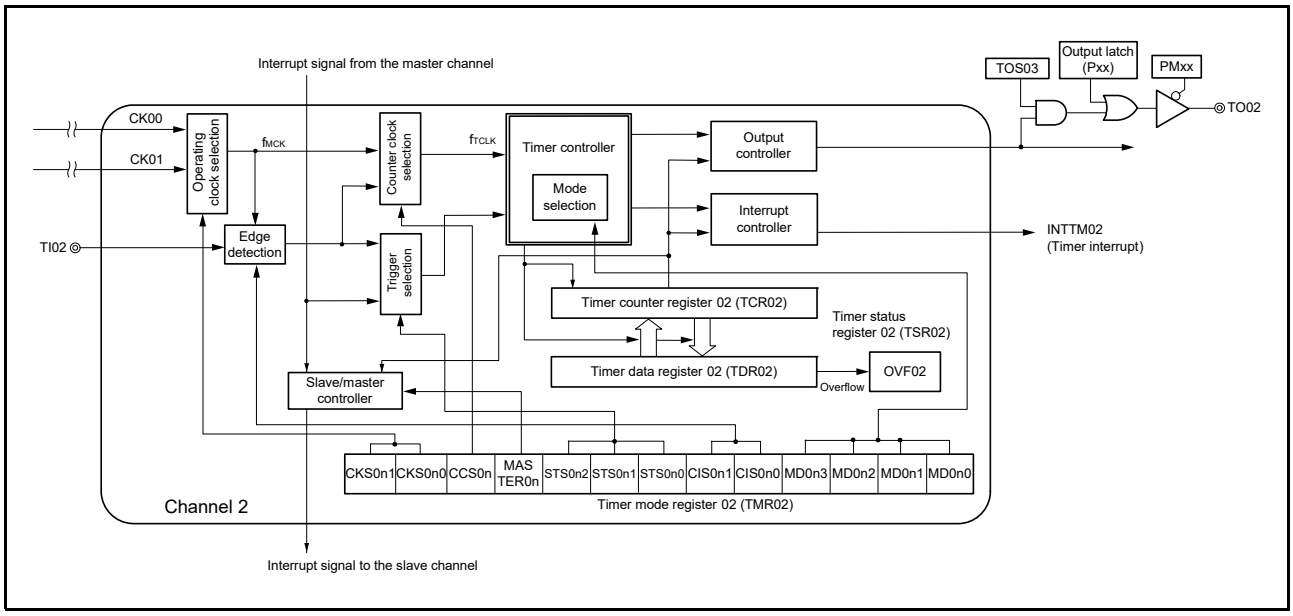
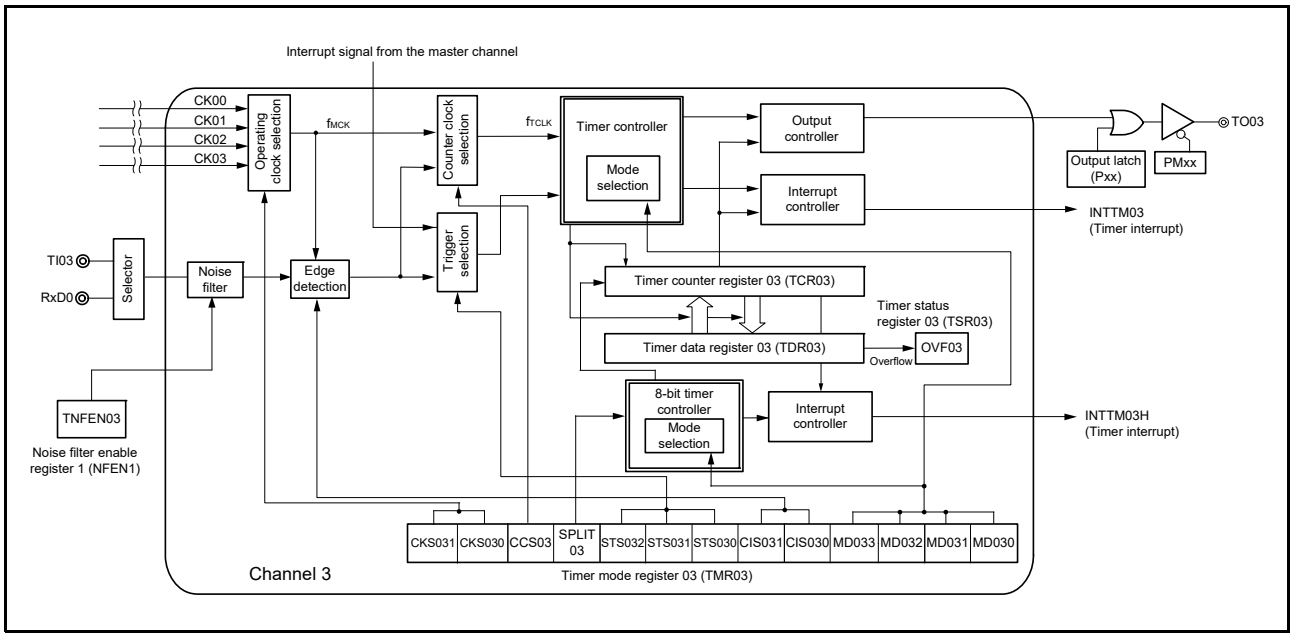


Figure 10 - 5 Internal Block Diagram of Channel 3 of the Timer Array Unit

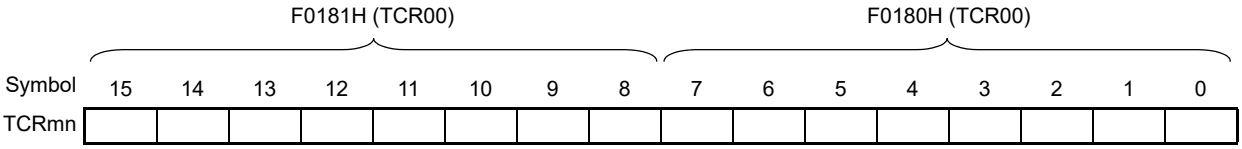


10.2.1 Timer counter register mn (TCRmn) (m = 0, n = 0 to 3)

The TCRmn register is a 16-bit read-only register and is used to count the number of input clock cycles. The value of this counter is incremented or decremented in synchronization with the rising edge of each cycle of the counter 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 **10.3.4 Timer mode register mn (TMRmn) (m = 0, n = 0 to 3)**).

Figure 10 - 6 Format of Timer Counter Register mn (TCRmn)

Address: F0180H, F0181H (TCR00) to F0186H, F0187H (TCR03)
After reset: FFFFH
R/W: R



The count value can be read by reading timer counter register mn (TCRmn).
The count value is set to FFFFH in the following cases.

- When the reset signal is generated
- When the TAUmRES bit of peripheral reset control register 0 (PRR0) 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.**

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

The value read from the TCRmn register varies depending on the change to the operation mode and operating state as shown in the table below.

Table 10 - 3 Timer Counter Register mn (TCRmn) Read Value in Various Operation Modes

Operation Mode	Count Mode	Value Read from the Timer Counter Register mn (TCRmn)[Note]			
		Value When the Operation Mode Is Changed after Releasing Reset	Value When Count Operation Is Temporarily Stopped (TTmn = 1)	Value When the Operation Mode Is Changed after Count Operation Was Temporarily Stopped (TTmn = 1)	Value When Waiting for a Start Trigger after One Count
Interval timer mode	Countdown	FFFFH	Value when counting is stopped	Undefined	—
Capture mode	Count-up	0000H	Value when counting is stopped	Undefined	—
Event counter mode	Countdown	FFFFH	Value when counting is stopped	Undefined	—
One-count mode	Countdown	FFFFH	Value when counting is stopped	Undefined	FFFFH
Capture & one-count mode	Count-up	0000H	Value when counting is stopped	Undefined	Captured 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)

10.2.2 Timer data register mn (TDRmn) (m = 0, n = 0 to 3)

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 SPLITm1, SPLITm3 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. The value of this register following a reset is 0000H.

Figure 10 - 7 Format of Timer Data Register mn (TDRmn) (n = 0, 2)

Address: FFF18H, FFF19H (TDR00), FFF64H, FFF65H (TDR02)
After reset: 0000H
R/W: R/W

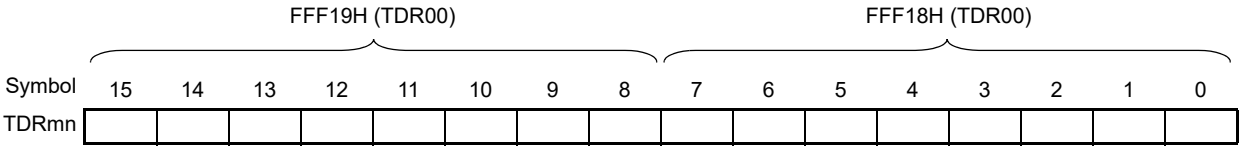
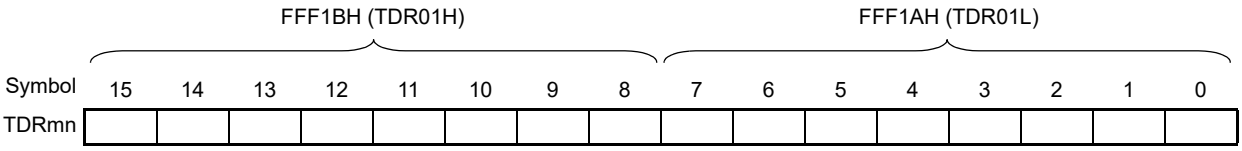


Figure 10 - 8 Format of Timer Data Register mn (TDRmn) (n = 1, 3)

Address: FFF1AH, FFF1BH (TDR01), FFF66H, FFF67H (TDR03)
After reset: 0000H
R/W: R/W



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.

ii) When timer data register mn (TDRmn) is used as capture register

The count value of timer counter 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).

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.**

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

10.3 Registers to Control the Timer Array Unit

The following registers are used to control the timer array unit.

- Peripheral enable register 0 (PER0)
- Peripheral reset control register 0 (PRR0)
- Timer clock select register m (TPSm) (m = 0)
- Timer mode register mn (TMRmn) (m = 0, n = 0 to 3)
- Timer status register mn (TSRmn) (m = 0, n = 0 to 3)
- Timer channel enable status register m (TEm) (m = 0)
- Timer channel start register m (TSm) (m = 0)
- Timer channel stop register m (TTm) (m = 0)
- Timer I/O select register 0 (TIOS0)
- Timer output enable register m (TOEm) (m = 0)
- Timer output register m (TOm) (m = 0)
- Timer output level register m (TOLm) (m = 0)
- Timer output mode register m (TOMm) (m = 0)
- Input switch control register (ISC)
- Noise filter enable register 1 (NFEN1)
- Port mode registers xx (PMxx) (xx = 0, 1, 3, 5, 7, 12)
- Port registers xx (Pxx) (xx = 0, 1, 3, 7, 12)
- Port output mode registers xx (POMxx) (xx = 1, 7)
- Port mode control A registers xx (PMCAxx) (xx = 0, 1, 12)

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

10.3.1 Peripheral enable register 0 (PER0)

The PER0 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. To use the timer array unit, 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. The value of this register following a reset is 00H.

Figure 10 - 9 Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

TAU0EN	Control of supply of an input clock to the timer array unit
0	Stops supply of an input clock. • The SFRs used by the timer array unit cannot be written. • When an SFR used by the timer array unit is read, the value returned is 00H or 0000H.
1	Enables supply of an input clock. • The SFRs used by the timer array unit can be read and written.

Caution When setting the timer array unit, make sure that the setting of the TAUmEN bit is 1 before setting the following registers. If TAUmEN = 0, the value of each register which controls the timer array unit is 00H and writing to any of those registers is ignored (except for timer I/O select register 0 (TIOS0), the input switch control register (ISC), noise filter enable register 1 (NFEN1), port mode control A registers 0, 1, and 12 (PMCA0, PMCA1, and PMCA12), port mode registers 0, 1, 3, 5, 7, and 12 (PM0, PM1, PM3, PM5, PM7, and PM12), port registers 0, 1, 3, 5, 7, and 12 (P0, P1, P3, P5, P7, and P12)).

- **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 output enable register m (TOEm)**
- **Timer output register m (TOm)**
- **Timer output level register m (TOLm)**
- **Timer output mode register m (TOMm)**

10.3.2 Peripheral reset control register 0 (PRR0)

The PRR0 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. To place the timer array unit in the reset state, set bit 0 (TAU0RES) of this register to 1. The PRR0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 10 - 10 Format of Peripheral Reset Control Register 0 (PRR0)

Address: F00F1H
After reset: 00H
R/W: R/W

Symbol	7	6	<5>	<4>	<3>	<2>	1	<0>
PRR0	0	0	ADCRES	IICA0RES	SAU1RES	SAU0RES	0	TAU0RES

TAU0RES	Control resetting of the timer array unit
0	The timer array unit is released from the reset state.
1	The timer array unit is in the reset state. • The SFRs for use with the timer array unit are initialized.

10.3.3 Timer clock select register m (TPSm) (m = 0)

The TPSm register is a 16-bit register used to select two types or four types of operating clocks (CKm0, CKm1, CKm2, CKm3) that are commonly supplied to each channel. CKm0 is selected by using bits 3 to 0 of the TPSm register, and CKm1 is selected by using bits 7 to 4 of the TPSm register. In addition, only for channels 1 and 3, CKm2 and CKm3 can be also selected. CKm2 is selected by using bits 9 and 8 of the TPSm register, and CKm3 is selected by using bits 13 and 12 of the TPSm register. 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 operating clock (CKSmn[1:0] = 00B) are stopped (TEmn = 0).
If the PRSm10 to PRSm13 bits can be rewritten (n = 0 to 3):
  All channels for which CKm1 is selected as the operating clock (CKSmn[1:0] = 01B) are stopped (TEmn = 0).
If the PRSm20 and PRSm21 bits can be rewritten (n = 1, 3):
  All channels for which CKm2 is selected as the operating clock (CKSmn[1:0] = 10B) 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 operating clock (CKSmn[1:0] = 11B) are stopped (TEmn = 0).

The TPSm register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 10 - 11 Format of Timer Clock Select Register m (TPSm) (1/2)

Address: F01B6H, F01B7H (TPS0)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TPSm	0	0	PRSm31	PRSm30	0	0	PRSm21	PRSm20

7	6	5	4	3	2	1	0
PRSm13	PRSm12	PRSm11	PRSm10	PRSm03	PRSm02	PRSm01	PRSm00

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Selection of operating clock (CKmk)Note (k = 0, 1)						
					fCLK = 2 MHz	fCLK = 5 MHz	fCLK = 10 MHz	fCLK = 20 MHz	fCLK = 32 MHz	fCLK = 48 MHz
0	0	0	0	fCLK	2 MHz	5 MHz	10 MHz	20 MHz	32 MHz	48 MHz
0	0	0	1	fCLK/2	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz	24 MHz
0	0	1	0	fCLK/2^2	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz	12 MHz
0	0	1	1	fCLK/2^3	250 kHz	625 kHz	1.25 MHz	2.5 MHz	4 MHz	6 MHz
0	1	0	0	fCLK/2^4	125 kHz	313 kHz	625 kHz	1.25 MHz	2 MHz	3 MHz
0	1	0	1	fCLK/2^5	62.5 kHz	156 kHz	313 kHz	625 kHz	1 MHz	1.5 MHz
0	1	1	0	fCLK/2^6	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 kHz	750 kHz
0	1	1	1	fCLK/2^7	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	250 kHz	375 kHz
1	0	0	0	fCLK/2^8	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz	187.5 kHz
1	0	0	1	fCLK/2^9	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	62.5 kHz	93.8 kHz
1	0	1	0	fCLK/2^{10}	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz	46.9 kHz
1	0	1	1	fCLK/2^{11}	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	15.6 kHz	23.4 kHz
1	1	0	0	fCLK/2^{12}	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz	11.7 kHz
1	1	0	1	fCLK/2^{13}	244 Hz	610 Hz	1.22 kHz	2.44 kHz	3.91 kHz	5.86 kHz
1	1	1	0	fCLK/2^{14}	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz	2.93 kHz
1	1	1	1	fCLK/2^{15}	61.0 Hz	153 Hz	305 Hz	610 Hz	977 Hz	1.46 Hz

Note When changing the clock selected for fCLK (by changing the system clock control register (CKC) value), stop timer array unit (TTm = 00FFH).

Caution 1. Be sure to clear bits 15, 14, 11, 10 to 0.

Caution 2. If fCLK (undivided) is selected as the operating clock (CKmk) and TDRmn is set to 0000H (m = 0, n = 0 to 3), interrupt requests output from timer array units cannot be used.

Remark 1. fCLK: CPU/peripheral hardware clock frequency

Remark 2. Waveform of the clock to be selected in the TPSm register which becomes high level for one period of fCLK from its rising edge (m = 0). For details, see **10.5.1 Counter clock (fTCLK)**.

Figure 10 - 11 Format of Timer Clock Select Register m (TPSm) (2/2)

PRSm21	PRSm20	Selection of operating clock (CKm2)Note						
			fCLK = 2 MHz	fCLK = 5 MHz	fCLK = 10 MHz	fCLK = 20 MHz	fCLK = 32 MHz	fCLK = 48 MHz
0	0	fCLK/2	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz	24 MHz
0	1	fCLK/2^2	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz	12 MHz
1	0	fCLK/2^4	125 kHz	313 kHz	625 kHz	1.25 MHz	2 MHz	3 MHz
1	1	fCLK/2^6	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 kHz	750 kHz

PRSm31	PRSm30	Selection of operating clock (CKm3)Note						
			fCLK = 2 MHz	fCLK = 5 MHz	fCLK = 10 MHz	fCLK = 20 MHz	fCLK = 32 MHz	fCLK = 48 MHz
0	0	fCLK/2^8	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz	188 kHz
0	1	fCLK/2^10	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz	46.9 kHz
1	0	fCLK/2^12	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz	11.7 kHz
1	1	fCLK/2^14	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz	2.93 kHz

Note When changing the clock selected for fCLK (by changing the system clock control register (CKC) value), stop timer array unit (TTm = 00FFH).
The timer array unit must also be stopped if the operating clock (fMCK) 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 operating clock, the interval times shown in **Table 10 - 4** can be achieved by using the interval timer function.

Table 10 - 4 Interval Times Available for Operating Clock CKSm2 or CKSm3

Clock		Interval TimeNote (fCLK = 32 MHz)			
		10 μs	100 μs	1 ms	10 ms
CKm2	fCLK/2	✓	—	—	—
	fCLK/2^2	✓	—	—	—
	fCLK/2^4	✓	✓	—	—
	fCLK/2^6	✓	✓	—	—
CKm3	fCLK/2^8	—	✓	✓	—
	fCLK/2^10	—	✓	✓	—
	fCLK/2^12	—	—	✓	✓
	fCLK/2^14	—	—	✓	✓

Note The margin is within 5%.

Remark 1. fCLK: CPU/peripheral hardware clock frequency

Remark 2. For details of a signal of fCLK/2^i selected with the TPSm register, see **10.5.1 Counter clock (fTCLK)**.

10.3.4 Timer mode register mn (TMRmn) (m = 0, n = 0 to 3)

The TMRmn register sets an operation mode of channel n. This register is used to select the operating clock (f_{MCK}), select the counter 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 (CISmn[1:0]) can be rewritten even while the register is operating with some functions (when TEmn = 1). For details, see **10.8 Independent Channel Operation Function of Timer Array Unit** and **10.9 Simultaneous Channel Operation Function of Timer Array Unit**. The TMRmn register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Caution The bit function assigned to bit 11 of the TMRmn register depends on the channel.
TMRm2: MASTERmn bit (n = 2)
TMRm3, TMRm1: SPLITmn bit (n = 3, 1)
TMRm0: Fixed to 0

Figure 10 - 12 Format of Timer Mode Register mn (TMRmn) (1/3)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TMRmn (n = 2)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0

7	6	5	4	3	2	1	0
CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8
TMRmn (n = 1, 3)	CKSmn1	CKSmn0	0	CCSmn	SPLITmn	STSmn2	STSmn1	STSmn0

7	6	5	4	3	2	1	0
CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8
TMRmn (n = 0)	CKSmn1	CKSmn0	0	CCSmn	0Note 1	STSmn2	STSmn1	STSmn0

7	6	5	4	3	2	1	0
CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

CKSmn1	CKSmn0	Selection of operating clock (f_{MCK}) of channel n
0	0	Operating clock CKm0 set by timer clock select register m (TPSm)
0	1	Operating clock CKm2 set by timer clock select register m (TPSm)
1	0	Operating clock CKm1 set by timer clock select register m (TPSm)
1	1	Operating clock CKm3 set by timer clock select register m (TPSm)
Operating clock (f_{MCK}) is used by the edge detector. A counter clock (f_{TCLK}) and a sampling clock are generated depending on the setting of the CCSmn bit. The operating clocks CKm2 and CKm3 can only be selected for channels 1 and 3.		

CCSmn	Selection of counter clock (f_{TCLK}) of channel n
0	Operating clock (f_{MCK}) specified by the CKSmn[1:0] bits
1	Valid edge of input signal input from the TImn pin In channel 0, valid edge of input signal selected by TIOS0 In channel 1, valid edge of input signal selected by TIOS0 In channel 3, valid edge of input signal selected by ISC
Counter clock (f_{TCLK}) is used for the counter, output controller, and interrupt controller.	

Figure 10 - 12 Format of Timer Mode Register mn (TMRmn) (2/3)

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). Note that, in channel 0, this bit is a read-only bit and fixed to 0 (Regardless of the bit setting, channel 0 operates as master, 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

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 STSmn[2:0] bits is other than 010B, set the CISmn[1:0] bits to 10B.		

Figure 10 - 12 Format of Timer Mode Register mn (TMRmn) (3/3)

MD mn3	MD mn2	MD mn1	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 in each mode varies depending on the MDmn0 bit. See the table below for details.					

Operation mode set by the MDmn3 to MDmn1 bits (see the table above)	MD mn0	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 a read-only bit and fixed to 0. Writing to this bit 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).

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 CKSmn[1:0] bits (f_{MCK}) or the valid edge of the signal input from the TImn pin is selected as the counter clock (f_{TCLK}).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

10.3.5 Timer status register mn (TSRmn) (m = 0, n = 0 to 3)

The TSRmn register indicates the overflow state of the counter of channel n. The TSRmn register is valid only in the capture mode (MDmn[3:1] = 010B) and capture & one-count mode (MDmn[3:1] = 110B). See **Table 10 - 5 OVF Bit Operation and Set/Clear Conditions in Each Operation Mode** 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 8 lower-order bits of the TSRmn register can be handled as the TSRmnL register, which can be read by an 8-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 10 - 13 Format of Timer Status Register mn (TSRmn)

Address: F01A0H, F01A1H (TSR00) to F01A6H, F01A7H (TSR03)
After reset: 0000H
R/W: R

Symbol	15	14	13	12	11	10	9	8
TSRmn	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	OVF

OVF	Counter overflow state 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 10 - 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.

10.3.6 Timer channel enable status register m (TEm) (m = 0)

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 8 lower-order bits of the TEm register can be handled as the TEmL register, which can be read by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 10 - 14 Format of Timer Channel Enable Status Register m (TEm)

Address: F01B0H, F01B1H (TE0)
After reset: 0000H
R/W: R

Symbol	15	14	13	12	11	10	9	8
TEm	0	0	0	0	TEHm3	0	TEHm1	0

7	6	5	4	3	2	1	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 enabled or stopped state 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)

10.3.7 Timer channel start register m (TSm) (m = 0)

The TSm register is a trigger register that is used to initialize timer counter 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 the TSmL register. The value of this register following a reset is 0000H.

Figure 10 - 15 Format of Timer Channel Start Register m (TSm)

Address: F01B2H, F01B3H (TS0)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TSm	0	0	0	0	TSHm3	0	TSHm1	0

7	6	5	4	3	2	1	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 10 - 6 Operations from the Count Operation Enabled State to the Start of Counting by a Timer Counter Register mn (TCRmn)**).

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 10 - 6 Operations from the Count Operation Enabled State to the Start of Counting by a Timer Counter Register mn (TCRmn)**).

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 10 - 6 Operations from the Count Operation Enabled State to the Start of Counting by a Timer Counter Register mn (TCRmn)**). 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.

Caution 1. Be sure to clear bits 15 to 12, 10, and 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 (TNFENnm = 1): Four cycles of the operating clock (f_{MCK})
When the TImn pin noise filter is disabled (TNFENnm = 0): Two cycles of the operating clock (f_{MCK})

(Remarks are listed on the next page.)

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)

10.3.8 Timer channel stop register m (TTm) (m = 0)

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, TEHm1, TEHm3 = 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 the TTmL register. The value of this register following a reset is 0000H.

Figure 10 - 16 Format of Timer Channel Stop Register m (TTm)

Address: F01B4H, F01B5H (TT0)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TTm	0	0	0	0	TTHm3	0	TTHm1	0

7	6	5	4	3	2	1	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, and 8 to 4 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)

10.3.9 Timer I/O select register 0 (TIOS0)

The TIOS0 register is used to select the timer input of channels 0 and 1 and timer output of channel 2. The TIOS0 register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 10 - 17 Format of Timer I/O Select Register 0 (TIOS0)

Address: F0074H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TIOS0	TIS07	TIS06	TIS05	TIS04	TIS03	TIS02	TIS01	TIS00

TIS07	TIS06	TIS05	Selection of timer input used with channel 0
0	0	0	Input signal of timer input pin (TI00)
0	0	1	Timer RD2 output signal that does not pass through PWMOPA (TRDIOB0)
0	1	0	Timer RD2 output signal that does not pass through PWMOPA (TRDIOD0)
0	1	1	Timer RD2 output signal that does not pass through PWMOPA (TRDIOA1)
1	0	0	Timer RD2 output signal that does not pass through PWMOPA (TRDIOC1)
1	0	1	Timer RD2 output signal that does not pass through PWMOPA (TRDIOB1)
1	1	0	Timer RD2 output signal that does not pass through PWMOPA (TRDIOD1)
1	1	1	Timer RD2 output signal that does not pass through PWMOPA (TRDIOC0)

TIS04	Selection of timer input used with channel 0
0	Input signal specified by the TIS07 to TIS05 bits
1	Event input signal from ELC

TIS03	Enable/disable of TAU channel 2 output to P17 pin[Note]
0	Output enabled
1	Output disabled (Fixed to L)

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	Middle-speed on-chip oscillator peripheral clock (f_{IMP})
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
1	0	1	Subsystem clock (f_{SUB})
Other than above			Setting prohibited

(Note and Cautions are listed on the next page.)

Note If the setting of the PIOR3.PIOR36 bit is 1, the following pin is used as the alternative, depending on the products.
For 40-, 44-, 48-, 52-, and 64-pin products: P121 pin
For 20-, 24-, 25-, 30-, and 32-pin products: P120 pin

Caution 1. At least 1/f_{MCK} + 10 ns is necessary as the high-level and low-level widths of the timer input to be selected. Thus, the TIS02 bit cannot be set to 1 when f_{SUB} is selected as f_{CLK} (CSS in CKC register = 1).

Caution 2. When selecting an event input signal from the ELC using timer I/O select register 0 (TIOS0), select f_{CLK} using timer clock select register 0 (TPS0).

10.3.10 Timer output enable register m (TOEm) (m = 0)

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 the TOEmL register. The value of this register following a reset is 0000H.

Figure 10 - 18 Format of Timer Output Enable Register m (TOEm)

Address: F01BAH, F01BBH (TOE0)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TOEm	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	TOEm3	TOEm2	TOEm1	TOEm0

TOEmn	Enabling/disabling timer output for channel n
0	Disables timer output. The corresponding TOmn bit does not reflect timer operation with this setting, so the output level of a TOmn bit is fixed to the level written to the TOm register. Writing to the TOmn bit is enabled and the level set in the TOmn bit is output from the TOmn pin.
1	Enables timer output. The corresponding TOmn bit reflects timer operation with this setting, so the 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)

10.3.11 Timer output register m (TOm) (m = 0)

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 on 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 TI00, TO00, TI01/TO01, TI02/TO02, or TI03/TO03 pin 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 the TOmL register. The value of this register following a reset is 0000H.

Figure 10 - 19 Format of Timer Output Register m (TOm)

Address: F01B8H, F01B9H (TO0)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TOm	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	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)

10.3.12 Timer output level register m (TOLm) (m = 0)

The TOLm register 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 the TOLmL register. The value of this register following a reset is 0000H.

Figure 10 - 20 Format of Timer Output Level Register m (TOLm)

Address: F01BCH, F01BDH (TOL0)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TOLm	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	TOLm3	TOLm2	TOLm1	0

TOLmn	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)

10.3.13 Timer output mode register m (TOMm) (m = 0)

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 the TOMmL register. The value of this register following a reset is 0000H.

Figure 10 - 21 Format of Timer Output Mode Register m (TOMm)

Address: F01BEH, F01BFH (TOM0)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TOMm	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	TOMm3	TOMm2	TOMm1	0

TOMmn	Control of timer output mode of channel n
0	Master channel output mode (to produce toggled 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 < p ≤ 3
(For details of the relation between the master channel and slave channel, refer to **10.4.1 Basic rules of simultaneous channel operation function**.)

10.3.14 Input switch control register (ISC)

The ISC1 and ISC0 bits of the ISC register are used to implement LIN-bus communication operation by using channel 3 in association with the serial array unit. When the ISC1 bit is set to 1, the input signal of the serial data input pin (RxD0) is selected as a timer input signal. The ISC register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 10 - 22 Format of Input Switch Control Register (ISC)

Address: F0073H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ISC	SSIE00	0	0	0	0	0	ISC1	ISC0

SSIE00	Setting $\overline{SSI00}$ pin input when CSI00 communication and slave mode are applied
0	$\overline{SSI00}$ pin input is invalid.
1	$\overline{SSI00}$ pin input is valid.

ISC1	Switching the input signal of channel 3 in timer array unit 0
0	Uses the input signal of the TI03 pin as a timer input (normal operation).
1	Input signal of the RxD0 pin is used as timer input (detects the wakeup signal and measures the low-level width of the break field and the pulse width of the sync field).

ISC0	Switching external interrupt (INTP0) input
0	Uses the input signal of the INTP0 pin as an external interrupt (normal operation).
1	Uses the input signal of the RxD0 pin as an external interrupt (wakeup signal detection).

Remark When the LIN-bus communication function is used, select the input signal of the RxD0 pin by setting ISC1 to 1.

10.3.15 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 enabled, after synchronization with the operating clock (f_{MCK}) for the target channel, whether the signal keeps the same value for two clock cycles is detected. When the noise filter is disabled, the input signal is only synchronized with the operating clock (f_{MCK}) for the target channel[Note]. The NFEN1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Note For details, see the following sections.

- **10.5.1 Counter clock (f_{TCLK}), 2. When valid edge of input signal via the TImn pin is selected (CCSmn = 1)**
- **10.5.2 Timing of the start of counting**
- **10.7 Timer Input (TImn) Control**

Figure 10 - 23 Format of Noise Filter Enable Register 1 (NFEN1)

Address: F0071H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
NFEN1	0	0	0	0	TNFEN03	TNFEN02	TNFEN01	TNFEN00

TNFEN03	Enabling/disabling use of the noise filter for the TI03 pin[Note]
0	Turns the noise filter off.
1	Turns the noise filter on.

TNFEN02	Enabling/disabling use of the noise filter for the TI02 pin
0	Turns the noise filter off.
1	Turns the noise filter on.

TNFEN01	Enabling/disabling use of the noise filter for the TI01 pin
0	Turns the noise filter off.
1	Turns the noise filter on.

TNFEN00	Enabling/disabling use of the noise filter for the TI00 pin
0	Turns the noise filter off.
1	Turns the noise filter on.

Note The applicable pin can be switched by setting the ISC1 bit of the ISC register.
ISC1 = 0: Whether or not to use the noise filter of the TI03 pin can be selected.
ISC1 = 1: Whether or not to use the noise filter of the RxD0 pin can be selected.

Remark The presence or absence of timer I/O pins of channels 0 to 3 depends on the product. See **Table 10 - 2 Timer I/O Pins Provided in Each Product** for details.

10.3.16 Registers for controlling the port functions multiplexed with the inputs and outputs of the timer array unit

Set the following registers to control the port functions multiplexed with the inputs and outputs of the timer array unit.

- Port mode registers xx (PMxx)
- Port registers xx (Pxx)
- Port output mode registers xx (POMxx)
- Port mode control A registers xx (PMCAxx)

For details, see the following sections.

- **7.3.1 Port mode registers xx (PMxx)**
- **7.3.2 Port registers xx (Pxx)**
- **7.3.5 Port output mode registers xx (POMxx)**
- **7.3.7 Port mode control A registers xx (PMCAxx)**

When the pins multiplexed with TO00 to TO03 are to be used for outputs of timers, set the following register bits corresponding to each port to 0.

- Port mode control A register xx (PMCAxx)
- Port output mode register xx (POMxx)
- Port mode register xx (PMxx)
- Port register xx (Pxx)

Example: When using P01/TO00 for timer output
Set the PMCA01 bit of port mode control A register 0 to 0.
Set the PM01 bit of port mode register 0 to 0.
Set the P01 bit of port register 0 to 0.

When the pins multiplexed with TI00 to TI03 are to be used for inputs of timers, set the port mode register xx (PMxx) bit corresponding to each port to 1 and the port mode control A register xx (PMCAxx) bit corresponding to each port to 0. The port register xx (Pxx) bits can be set to either 0 or 1.

Example: When using P00/TI00 for timer input
Set the PMCA00 bit of port mode control A register 0 to 0.
Set the PM00 bit of port mode register 0 to 1.
Set the P00 bit of port register 0 to 0 or 1.

Remark xx = 0, 1, 3, 5, 7, 12
Note that PMCA3, PMCA5, PMCA7, and POM12 are not present in the RL78/G24 products.

10.4 Basic Rules of Timer Array Unit

10.4.1 Basic rules of simultaneous channel operation function

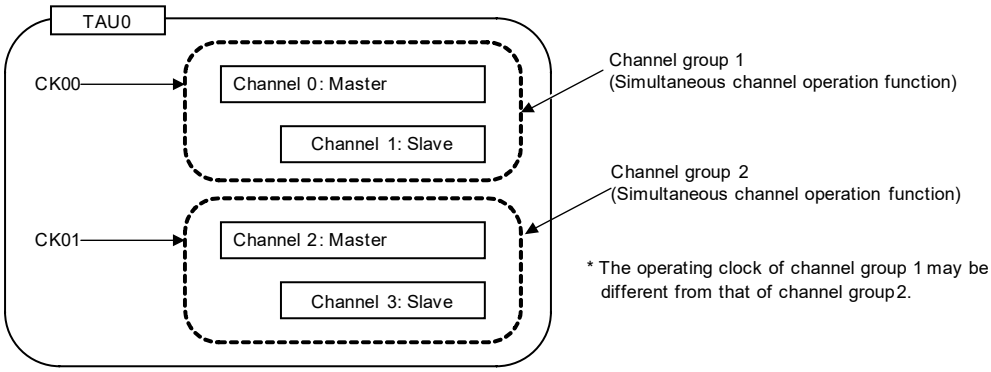
When simultaneously using multiple channels, namely, a combination of a master channel (a reference timer mainly counting the cycle) and slave channels (timers operating according to the master channel), the following rules apply.

1. Only an even channel (channels 0 and 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, and 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, channel 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 CKSmn[1:0] 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 counter clock to the lower channels.
8. A slave channel can use INTTMmn (interrupt), a start software trigger, or the counter clock of the master channel as a source clock, but cannot transmit its own INTTMmn (interrupt), start software trigger, or counter clock to channels with lower channel numbers.
9. A master channel cannot use INTTMmn (interrupt), a start software trigger, or the counter 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 to 0). However, as channel 0 is the highest channel, it can be used as a master channel during simultaneous operation.

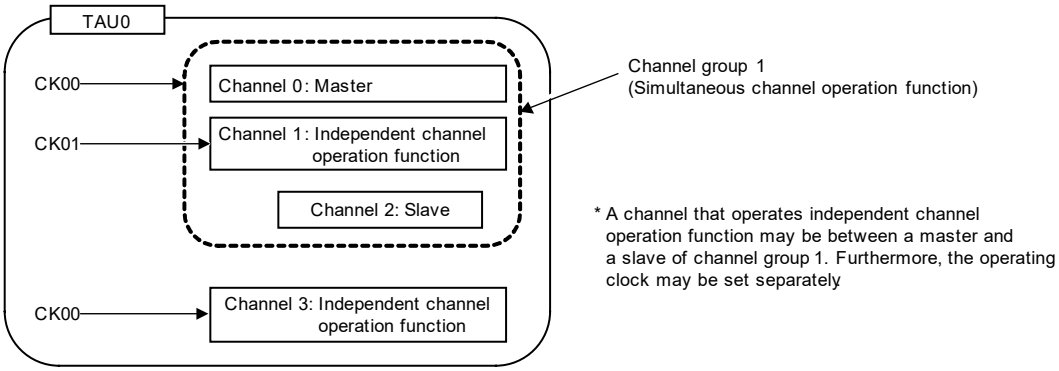
The rules of the simultaneous channel operation function are applied in a channel group (a master channel and slave channels forming one simultaneous channel operation function).
If two or more channel groups that do not operate in combination are specified, the basic rules of the simultaneous channel operation function in **10.4.1 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



Example 2



10.4.2 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 and INTTMm3H (interrupts) (which is the same operation performed when MDmn0 is set to 1).
5. The operating clock of the higher 8 bits is selected according to the CKSmn[1:0] bits of the lower-bit TMRmn register.
6. For the higher 8 bits, the TSHm1 and TSHm3 bits are manipulated to start channel operation and the TTHm1 and TTHm3 bits are manipulated to stop channel operation. The channel state can be checked using the TEHm1 and TEHm3 bits.
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 output function
   - External event counter function
   - Delay count function
8. For the lower 8 bits, the TSm1 and TSm3 bits are manipulated to start channel operation and the TTm1 and TTm3 bits are manipulated to stop channel operation. The channel state can be checked using the TEm1 and TEm3 bits.
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)

10.5 Operations of Counters

10.5.1 Counter clock (fTCLK)

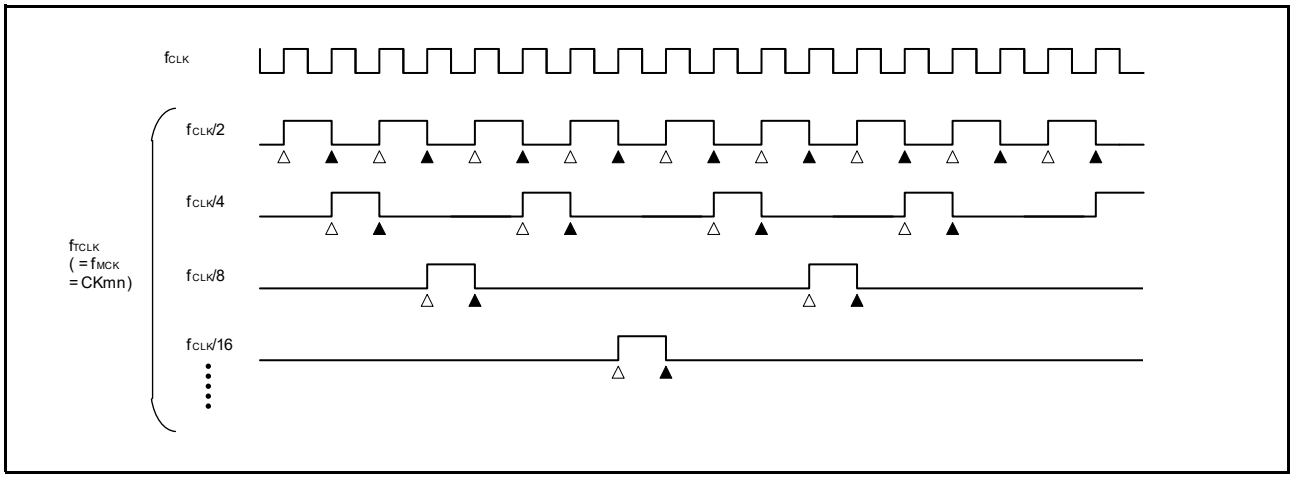
The counter clock (fTCLK) of the timer array unit can be selected between following by CCSmn bit of timer mode register mn (TMRmn).

- Operating clock (fMCK) specified by the CKSmn[1:0] 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 counter clock (fTCLK) are shown below.

1. When operating clock (fMCK) specified by the CKSmn[1:0] bits is selected (CCSmn = 0)
   The counter 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, the clock signal is fixed to the high level.
   Counting of timer counter register mn (TCRmn) delayed by one period of fCLK from rising edge of the counter clock, because of synchronization with fCLK. But, this is described as “counting at rising edge of the counter clock”, as a matter of convenience.

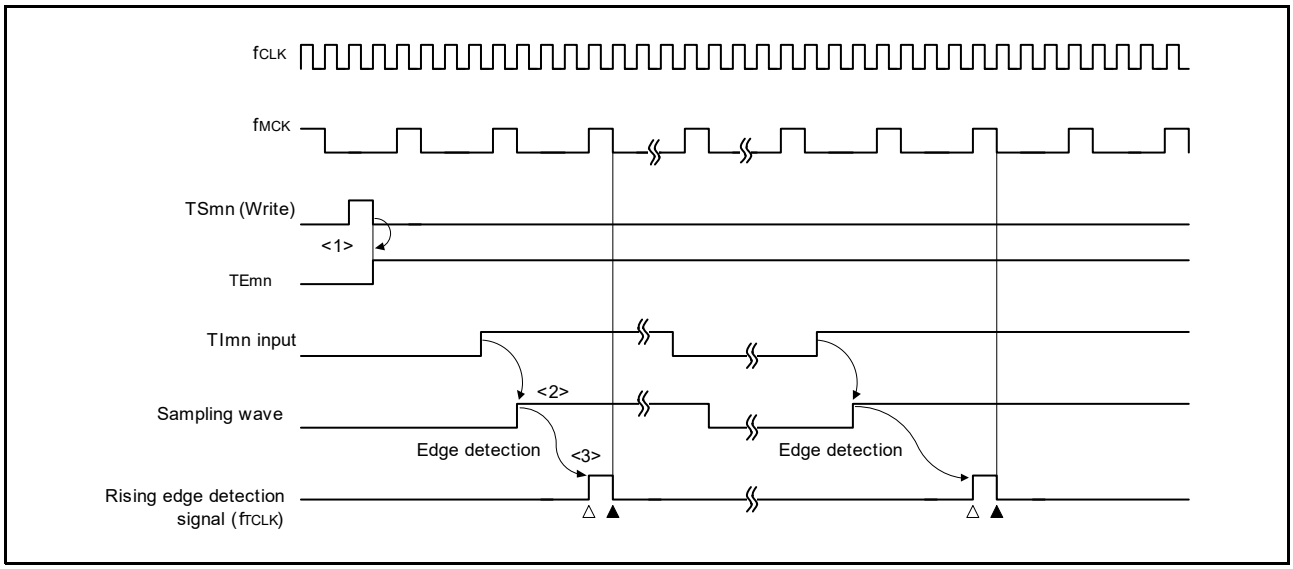
Figure 10 - 24 Timing of fCLK and Counter Clock (fTCLK) (When CCSmn = 0)



Remark 1. △ : Rising edge of the counter 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 counter clock (fTCLK) becomes the signal that detects valid edge of input signal via the TImn pin and synchronizes next rising fMCK. The counter clock (fTCLK) is delayed for 1 to 2 period of fMCK from the input signal via the TImn pin (when a noise filter is used, the delay becomes 3 to 4 clock cycles).
   Counting of timer counter register mn (TCRmn) delayed by one period of fCLK from rising edge of the counter clock, because of synchronization with fCLK. But, this is described as “counting at valid edge of input signal via the TImn pin”, as a matter of convenience.

Figure 10 - 25 Timing of fCLK and Counter 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 fMCK.
<3> The edge is detected by the rising of the sampled signal and the detection signal (counter clock) is output.

Remark 1. △ : Rising edge of the counter clock
▲ : Synchronization, increment/decrement of counter
Remark 2. fCLK: CPU/peripheral hardware clock
fMCK: Operating 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, the one-shot pulse output are the same.

10.5.2 Timing of the start of counting

Timer counter register mn (TCRmn) becomes enabled to operation by setting of TSmn bit of timer channel start register m (TSm).

Table 10 - 6 shows operations from the count operation enabled state to the start of counting by timer counter register mn (TCRmn).

Table 10 - 6 Operations from the Count Operation Enabled State to the Start of Counting by a Timer Counter Register mn (TCRmn)

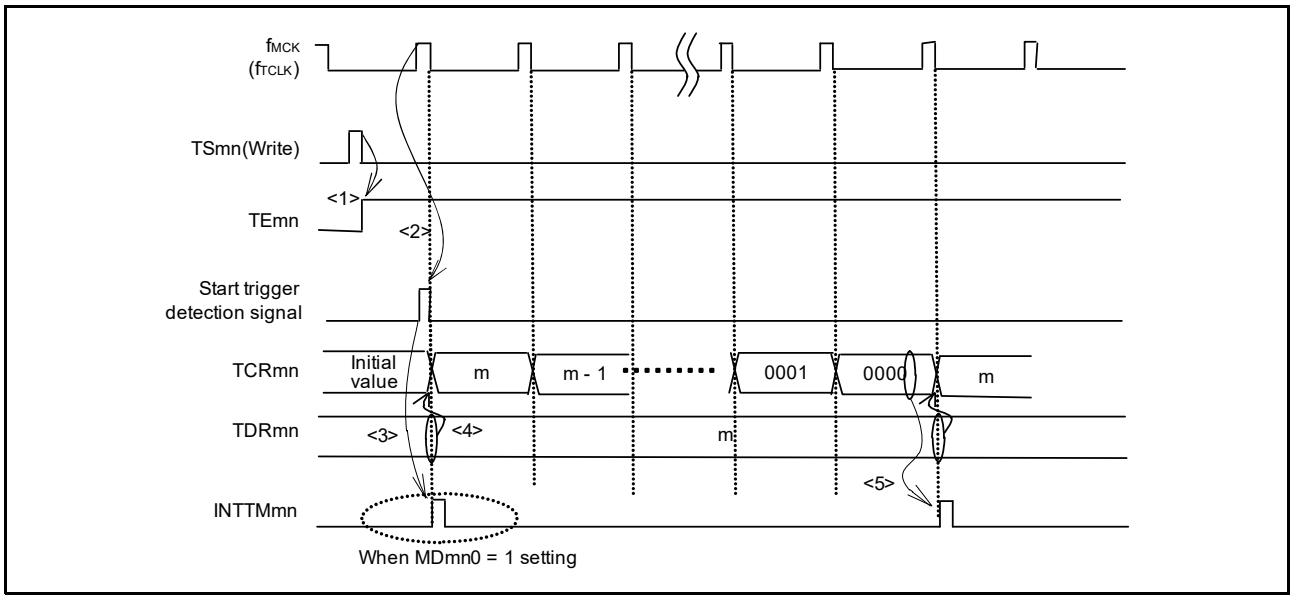
Timer Operation Mode	Operation When TSmn = 1 Is Set
Interval timer mode	No operation is carried out from start trigger detection (TSmn=1) until counter clock generation. The first counter clock loads the value of the TDRmn register to the TCRmn register and the subsequent counter clock performs count down operation (see **10.5.3 Operations of counters**, **1. Operation in 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 counter clock performs count down operation (see **10.5.3 Operations of counters**, **2. Operation in event counter mode**).
Capture mode	No operation is carried out from start trigger detection (TSmn = 1) until counter clock generation. The first counter clock loads 0000H to the TCRmn register and the subsequent counter clock performs count up operation (see **10.5.3 Operations of counters**, **3. Operation in 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 counter clock generation. The first counter clock loads the value of the TDRmn register to the TCRmn register and the subsequent counter clock performs count down operation (see **10.5.3 Operations of counters**, **4. Operation in 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 counter clock generation. The first counter clock loads 0000H to the TCRmn register and the subsequent counter clock performs count up operation (see **10.5.3 Operations of counters**, **5. Operation in capture & one-count mode (high-level width measurement)**).

10.5.3 Operations of counters

Here, the counter operation in each mode is explained.

1. Operation in interval timer mode
   - <1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit. Timer counter register mn (TCRmn) holds the initial value until counter clock generation.
   - <2> A start trigger is generated on the first cycle of the counter clock (f_{MCK}) after operation is enabled.
   - <3> When the MDmn0 bit is set to 1, INTTMmn is generated by the start trigger.
   - <4> By the first cycle of the counter 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 on the next cycle of the counter clock (f_{MCK}) and the value of timer data register mn (TDRmn) is loaded to the TCRmn register and counting keeps on.

Figure 10 - 26 Timing during Operation in Interval Timer Mode

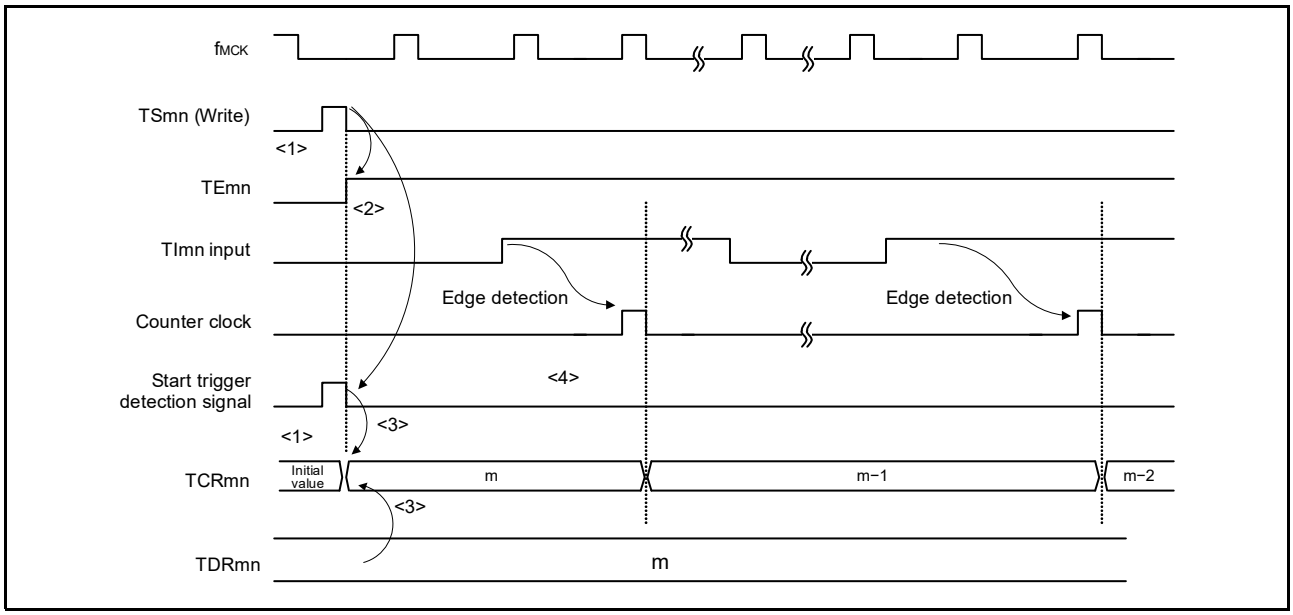


Caution **In the first cycle operation of counter clock after writing the TSmn bit, an error of up to one clock cycle is generated since count start delays until counter clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.**

Remark f_{MCK}, the start trigger detection signal, and INTTMmn become active for a period of one clock cycle in synchronization with f_{CLK}.

2. Operation in event counter mode
   - <1> Timer counter 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 counter clock of the valid edge of the TImn input.

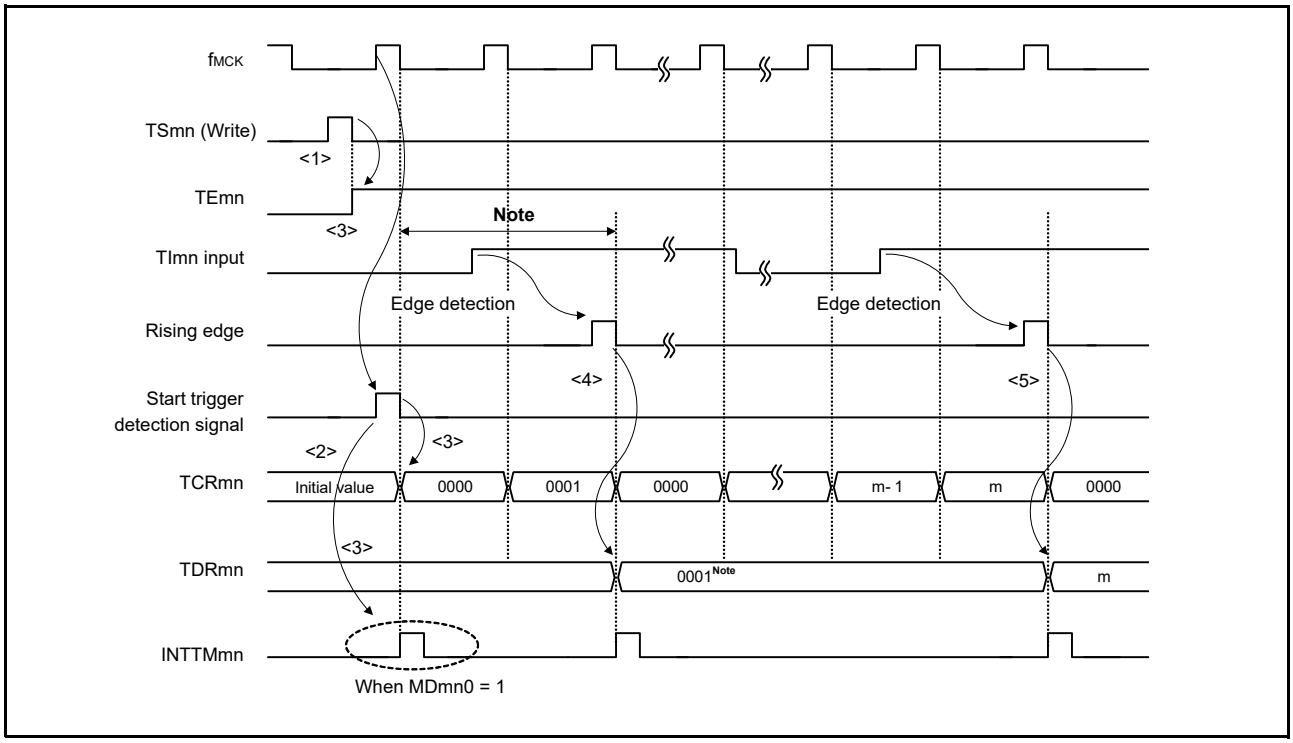
Figure 10 - 27 Timing during Operation in Event Counter Mode



Remark **Figure 10 - 27** 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 counter clock (fMCK).

3. Operation in capture mode (input pulse interval measurement)
   - <1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.
   - <2> Timer counter register mn (TCRmn) holds the initial value until counter clock generation.
   - <3> A start trigger is generated on the first cycle of the counter clock (fMCK) 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 captured value is meaningless. 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 10 - 28 Timing during Operation in Capture Mode (Input Pulse Interval Measurement)



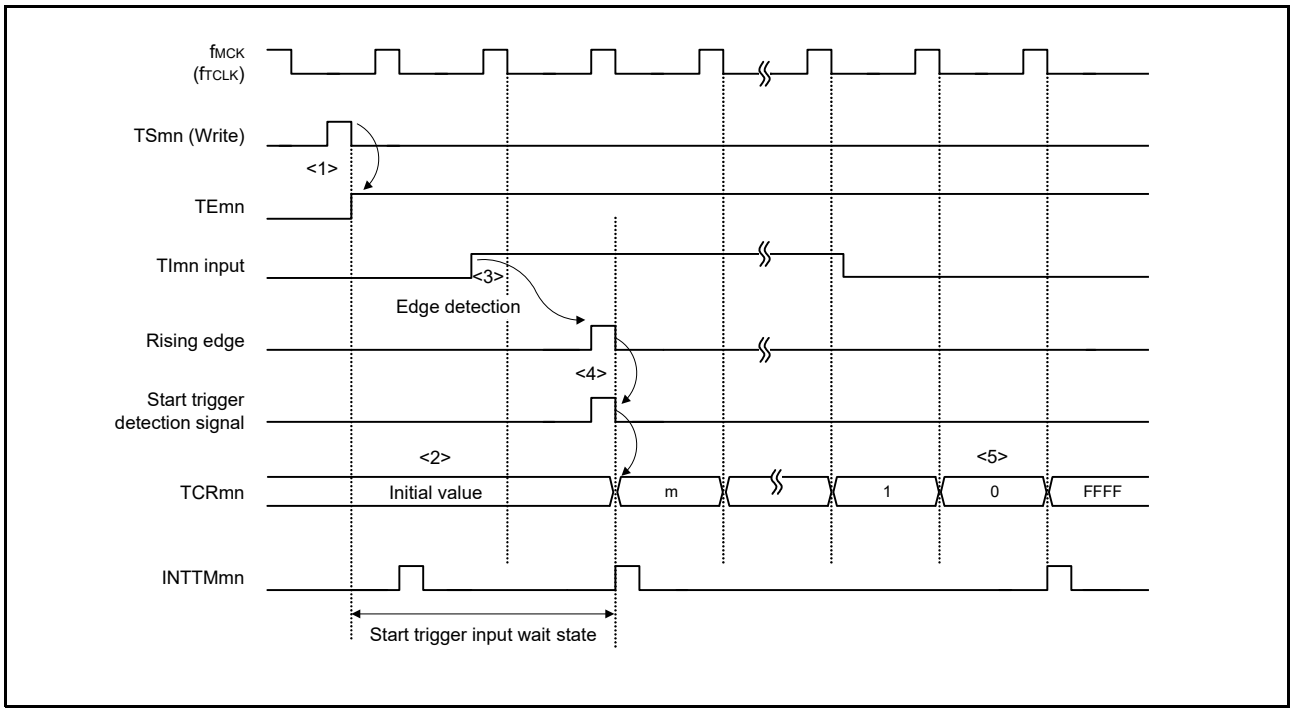
Note If a clock has been input to TImn (the trigger exists) when capturing starts, counting starts when a trigger is detected, even if no edge is detected. Therefore, the first captured value (<4>) does not determine a pulse interval (in the above figure, 0001 just indicates two clock cycles but does not determine the pulse interval) and so the user can ignore it.

Caution **In the first cycle operation of counter clock after writing the TSmn bit, an error of up to one clock cycle is generated since count start delays until counter 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 10 - 28** 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 counter clock (fMCK).

4. Operation in one-count mode
   - <1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.
   - <2> Timer counter 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.

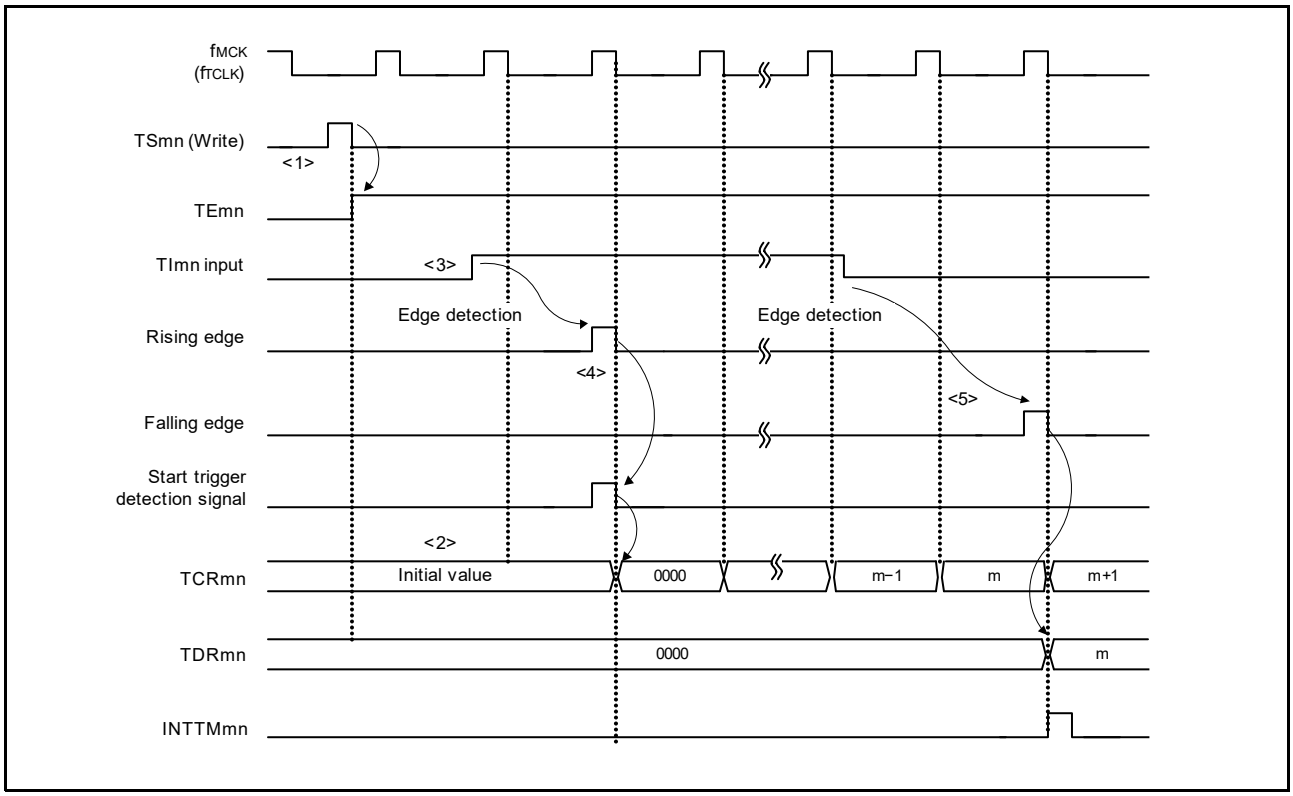
Figure 10 - 29 Timing during Operation in One-count Mode



Remark **Figure 10 - 29** 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 counter clock (fMCK).

5. Operation in 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 counter 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 10 - 30 Timing during Operation in Capture & One-count Mode (High-level Width Measurement)

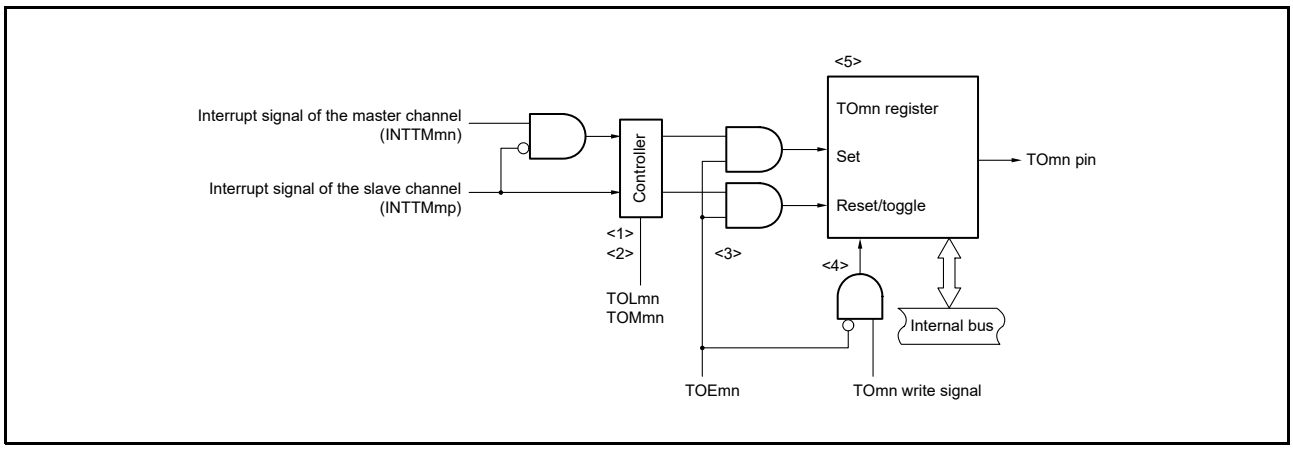


Remark **Figure 10 - 30** shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the counter clock (fMCK).

10.6 Channel Output (TOmn Pin) Control

10.6.1 TOmn pin output circuit configuration

Figure 10 - 31 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: Positive logic output (INTTMmn → set, INTTM0p → reset)
- When TOLmn = 1: Negative logic output (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.

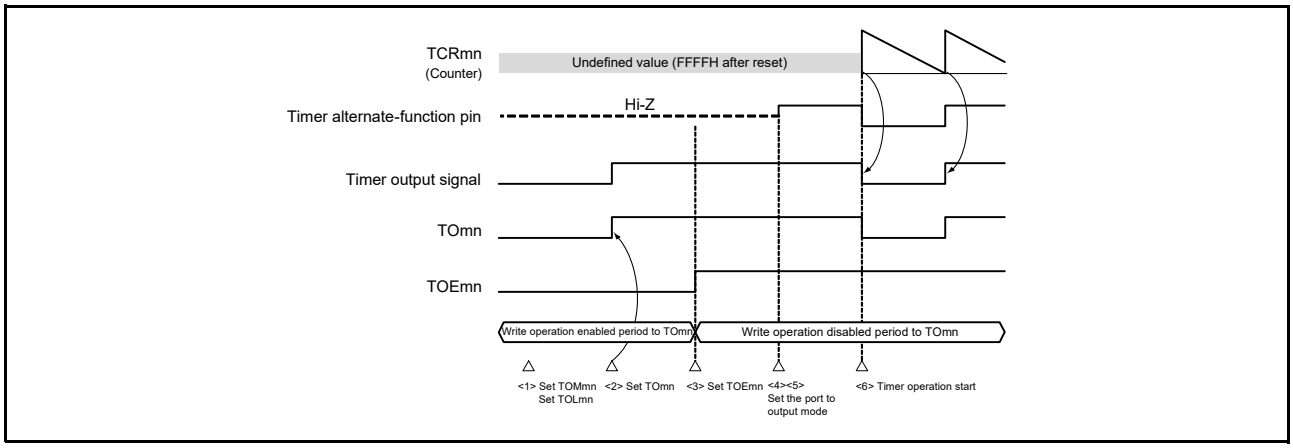
<5> The TOm register can always be read, and the TOmn pin output level can be checked.

Remark m: Unit number (m = 0)
n: Channel number
n = 0 to 3 (n = 0, 2 for master channel)
p: Slave channel number
n < p ≤ 3

10.6.2 TOmn pin output setting

The following figure shows the procedure and state transitions from the initial settings of a TOmn output pin to the start of timer operation.

Figure 10 - 32 State Transitions from the Settings for Timer Output to the Start of Timer Operation



<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 state 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 the port mode control A register xx (PMCAxx) (see **10.3.16 Registers for controlling the port functions multiplexed with the inputs and outputs of the timer array unit**).

<5> The port I/O setting is set to output (see **10.3.16 Registers for controlling the port functions multiplexed with the inputs and outputs of the timer array unit**).

<6> The timer operation is enabled (TSmn = 1).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

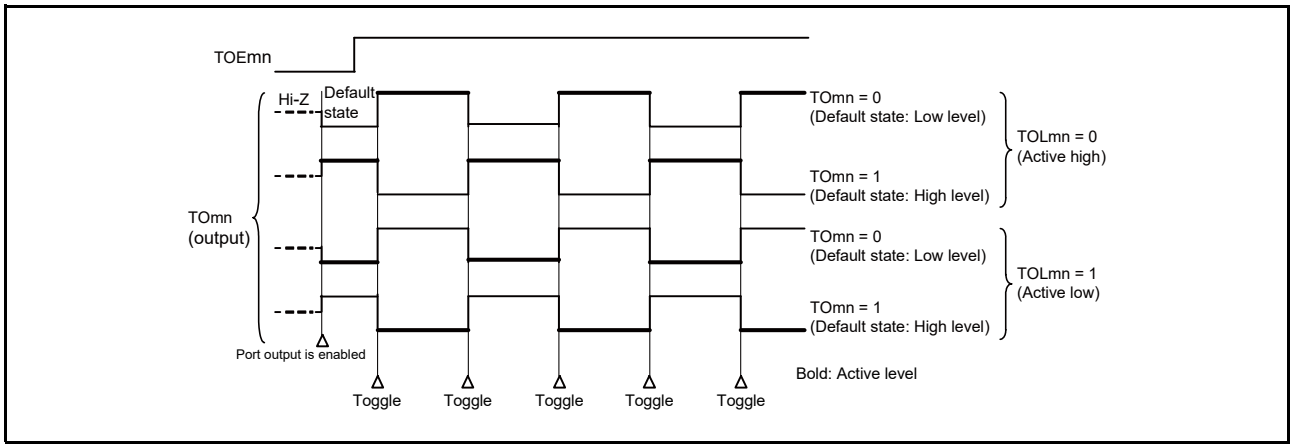
10.6.3 Cautions on channel output operation

1. Changing values set in the registers TOm, TOEm, and TOLm during timer operation
   Since the timer operations (operations of timer counter 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), and timer output level register m (TOLm) 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 in **10.7 Timer Input (TImn) Control** and **10.8 Independent Channel Operation Function of Timer Array Unit**.
   When the values set to the TOEm, 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 inverted.

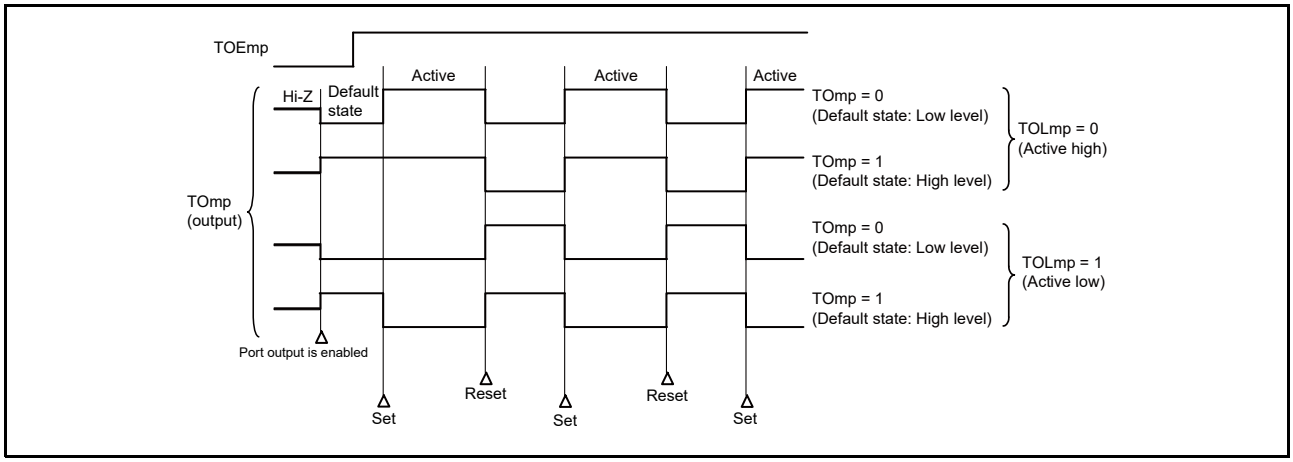
Figure 10 - 33 TOmn Pin Output States with Toggled Output (TOMmn = 0)



Remark 1. Toggle: Toggle signal to invert the output on the TOmn pin
Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

   b) When operation starts with slave channel output mode (TOMmp = 1) setting (PWM output)
      When slave channel output mode (TOMmp = 1), the active level is determined by timer output level register m (TOLm) setting.

Figure 10 - 34 TOmp Pin Output States with PWM Output (TOMmp = 1)



Remark 1. Set: The output signal of the TOmp pin changes from inactive level to active level.
Reset: The output signal of the TOmp pin changes from active level to inactive level.
Remark 2. m: Unit number (m = 0), p: Channel number (p = 1 to 3)

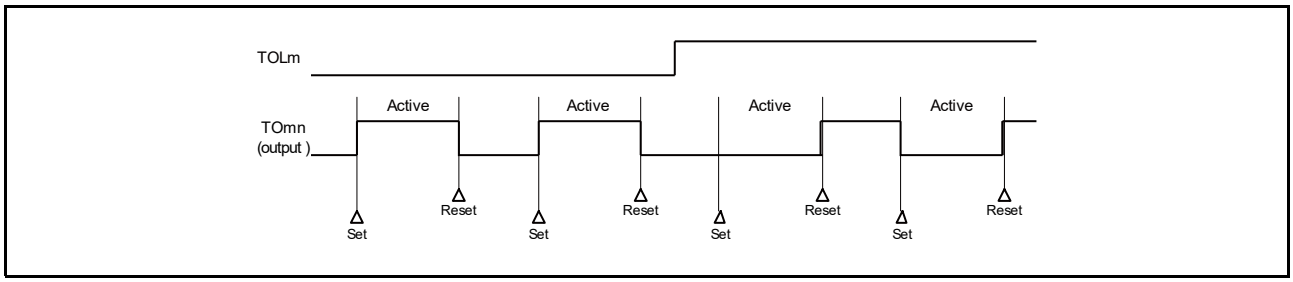
3. Operation of TOmn pin in slave channel output mode (TOMmn = 1)

a) When the relevant bit of timer output level register m (TOLm) is 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 10 - 35 Operation When the Relevant Bit of the TOLm Register Is 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 cycle of the counter 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.

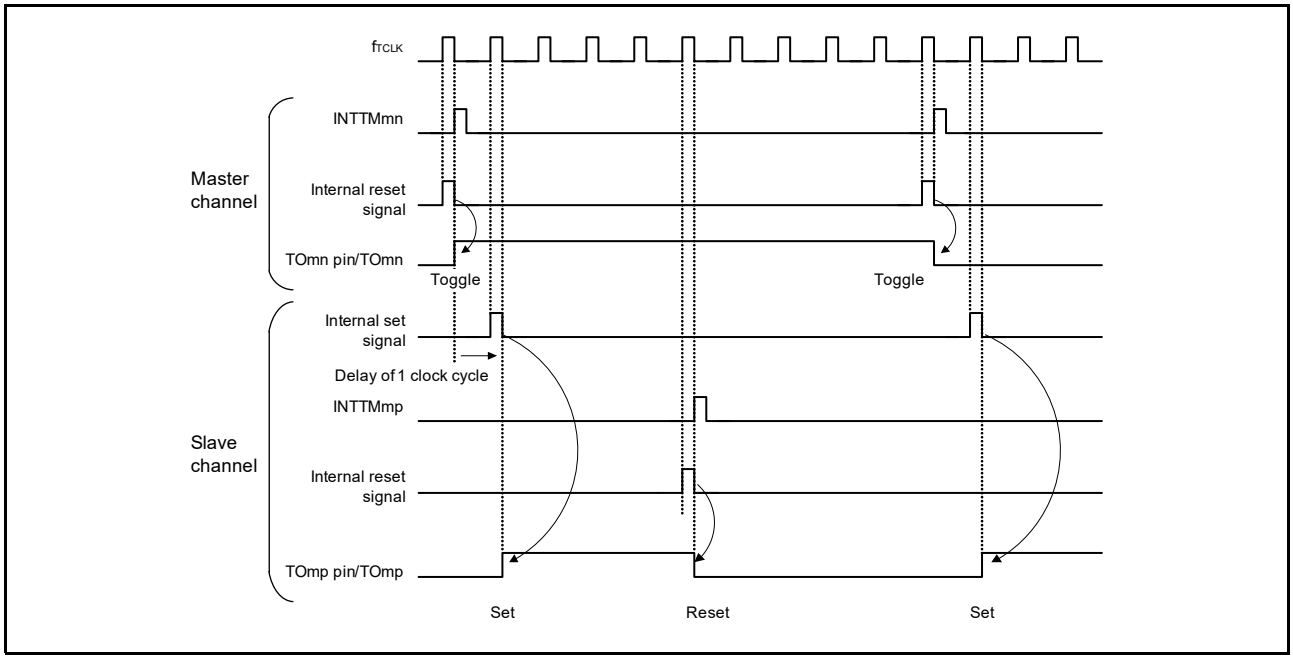
Figure 10 - 36 shows the states of operation following set and reset signals when the master and slave channels are set as follows.

- Master channel: TOEmn = 1, TOMmn = 0, TOLmn = 0
- Slave channel: TOEmp = 1, TOMmp = 1, TOLmp = 0

Figure 10 - 36 States of Operation following Set and Reset Signals

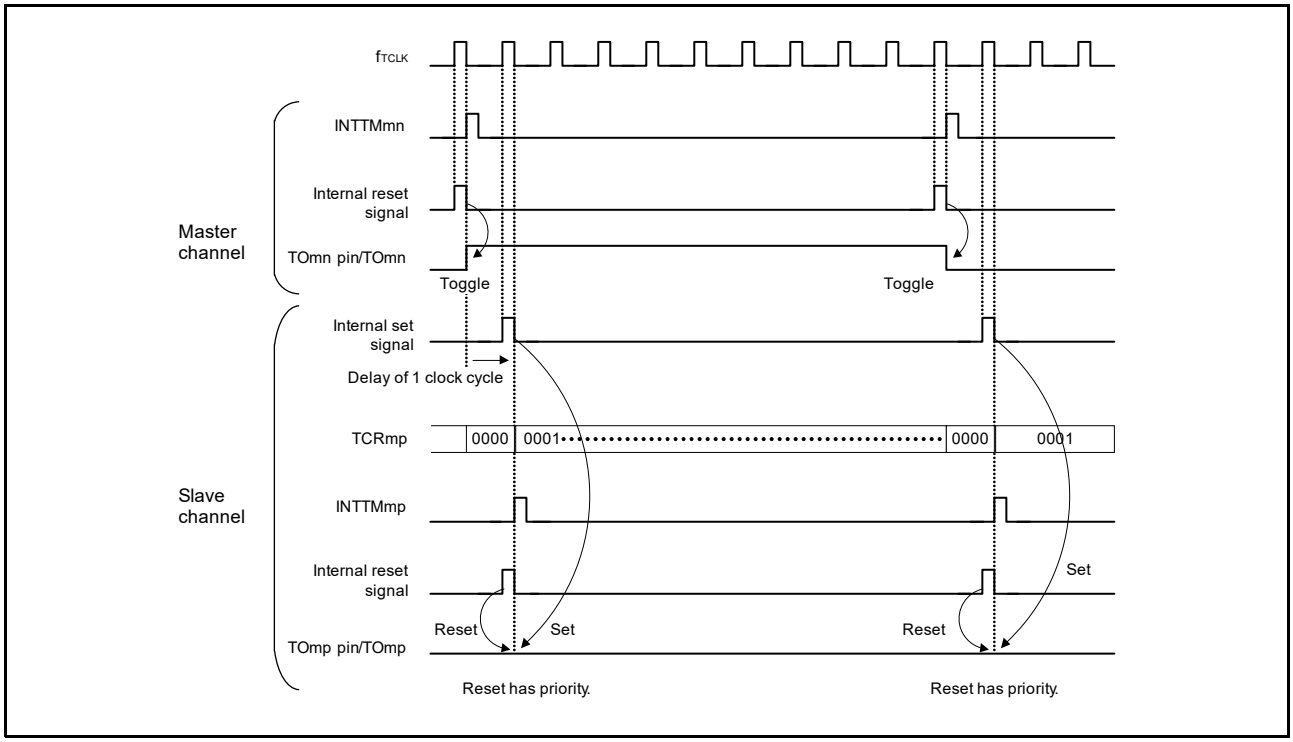
1. Basic timing during operation

<R>



2. Timing during operation with the duty cycle set to 0%

<R>



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 < p ≤ 3

10.6.4 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 10 - 37 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

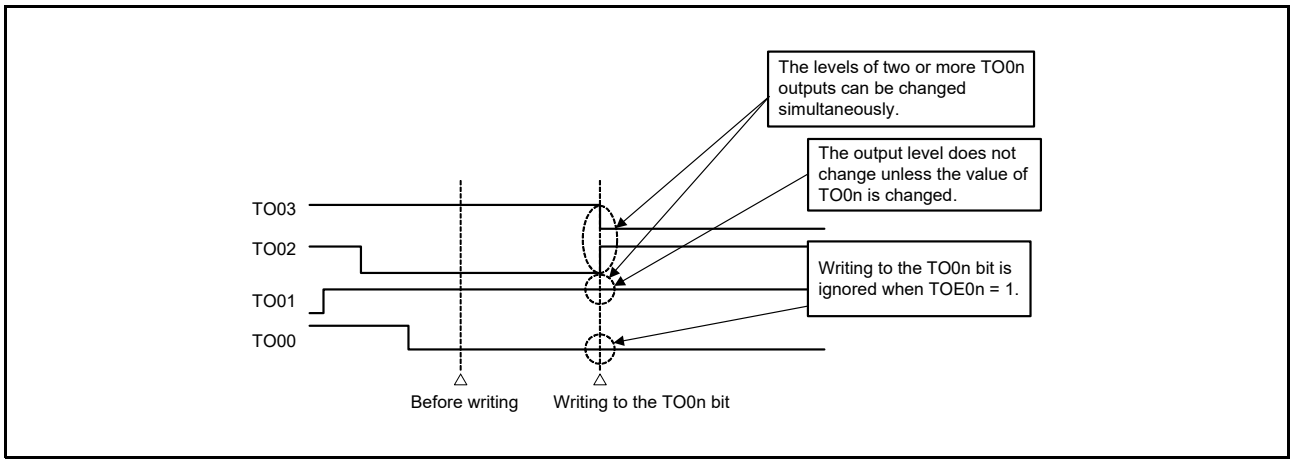
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

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 10 - 38 TO0n Pin States by Collective Manipulation of TO0n Bits



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

10.6.5 Timer interrupts and TOmn outputs when counting is started

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.

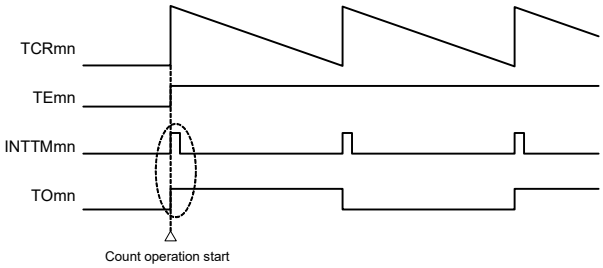
When MDmn0 is set to 1, the count operation start timing can be known by the timer interrupt (INTTMmn) generation.

In the other modes, neither timer interrupt at count operation start nor TOmn output is controlled.

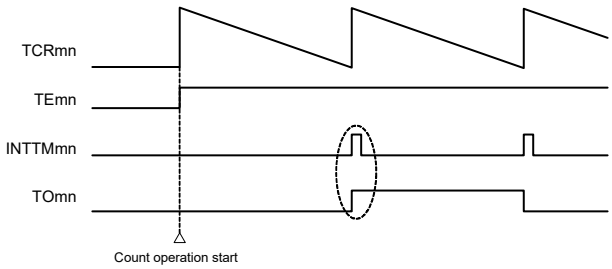
Figure 10 - 39 shows operation examples when the interval timer mode (TOEmn = 1, TOMmn = 0) is set.

Figure 10 - 39 Examples of the Operation of Timer Interrupts and TOmn Outputs When Counting Is Started

a) When MDmn0 is set to 1



b) When MDmn0 is set to 0



When MDmn0 is set to 1, a timer interrupt (INTTMmn) is output at count operation start, and TOmn performs a toggle operation.

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.

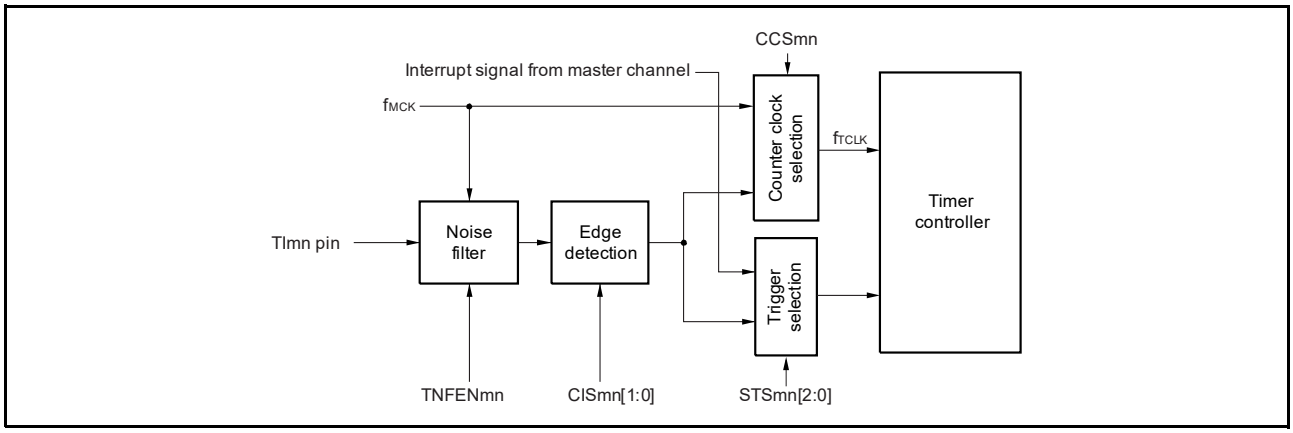
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

10.7 Timer Input (TImn) Control

10.7.1 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.

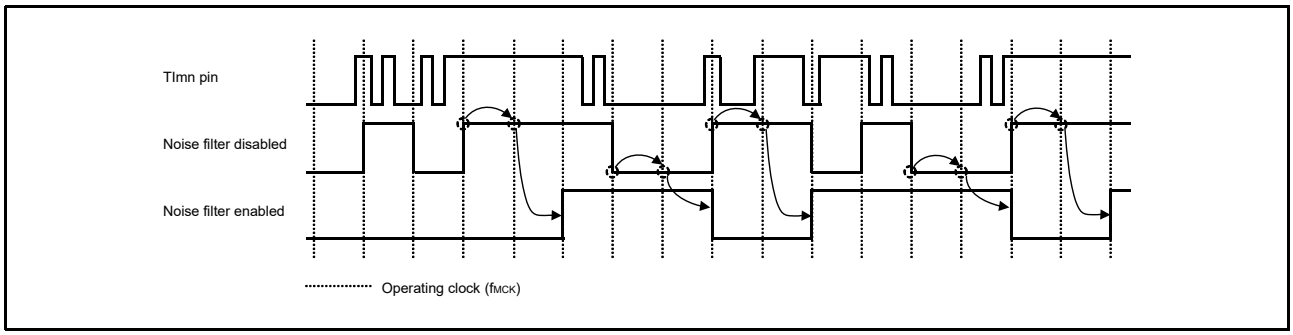
Figure 10 - 40 Input Circuit Configuration



10.7.2 Noise filter

When the noise filter is disabled, the input signal is only synchronized with the operating clock (fMCK) for channel n. When the noise filter is enabled, after synchronization with the operating clock (fMCK) 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 10 - 41 Sampling Waveforms through TImn Input Pin with Noise Filter Enabled and Disabled



10.7.3 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 any 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 any 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).

10.8 Independent Channel Operation Function of Timer Array Unit

10.8.1 Operation as an interval timer or for 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 counter clock × (Set value of TDRmn + 1)

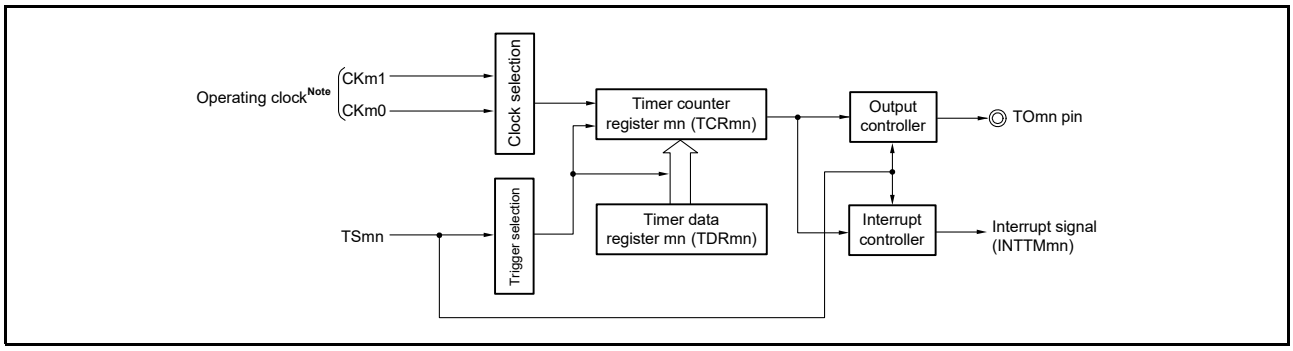
2. Operation for square wave output
   TOmn performs a toggle operation as soon as INTTMmn has been generated, and outputs a square wave with a duty factor of 50%.
   The period and frequency for outputting a square wave from TOmn can be calculated by the following expressions.

• Period of square wave output from TOmn = Period of counter clock × (Set value of TDRmn + 1) × 2
• Frequency of square wave output from TOmn = Frequency of counter clock/{(Set value of TDRmn + 1) × 2}

Timer counter 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) on the first cycle of the counter 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 the output on TOmn is not toggled. If the MDmn0 bit of the TMRmn register is 1, INTTMmn is output and the output on TOmn is toggled.
After that, the TCRmn register count down in synchronization with the counter clock.
When TCRmn = 0000H, INTTMmn is output and the output on TOmn is toggled on the next cycle of the counter 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.

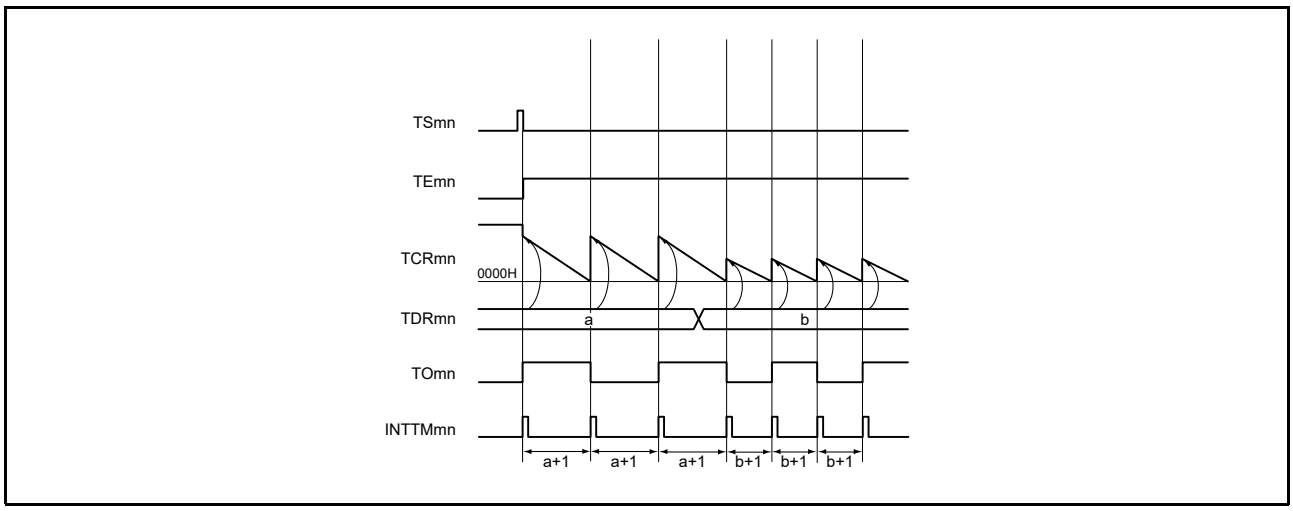
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 10 - 42 Block Diagram for Operation as an Interval Timer or for Square Wave Output



Note For channels 1 and 3, the clock can be selected as CKm0, CKm1, CKm2, or CKm3.

Figure 10 - 43 Example of Basic Timing during Operation as an Interval Timer or for Square Wave Output (MDmn0 = 1)

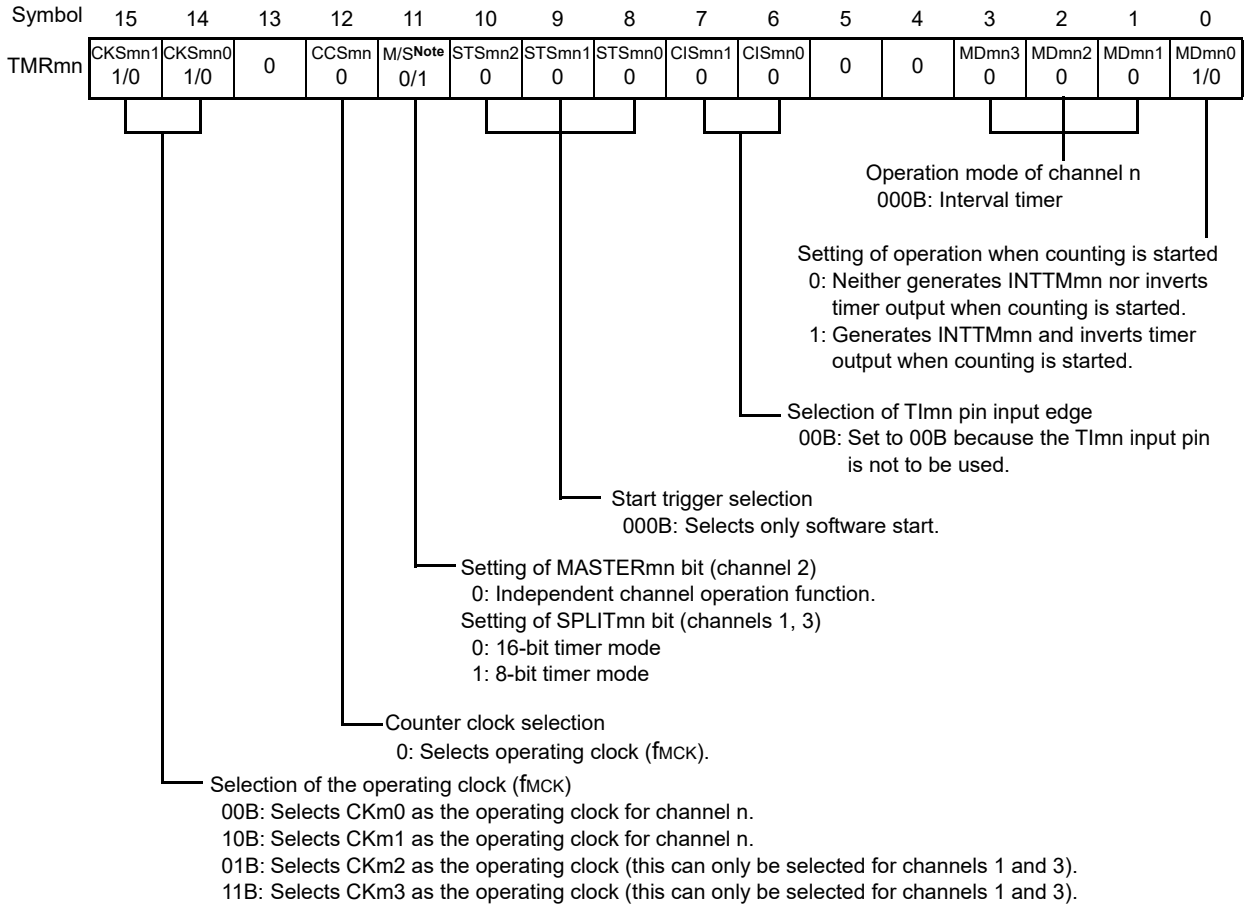


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

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 counter register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
TOmn: TOmn pin output signal

Figure 10 - 44 Example of Register Settings for Operation as an Interval Timer or for Square Wave Output

a) Timer mode register mn (TMRmn)



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: Set this bit 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 10 - 45 Procedure for Operations When the Interval Timer or Square Wave Output Function Is to Be Used

	Software Operation	Hardware State
TAU default setting		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on state. 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 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 counter register mn (TCRmn). 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) = 0, and count operation stops. The TCRmn register holds count value and stops. The TOmn output is not initialized and retains its current state.
	The TOEmn bit is cleared to 0 and value is set to the TOmn bit. →	The TOmn pin outputs the TOmn bit set level.
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. → When holding the TOmn pin output level is not necessary Setting not required.	The TOmn pin output level is held by port function.
	The TAUmEN bit of the PER0 register is cleared to 0. → Set the TAUmRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	This stops supply of the input clock to timer array unit m. 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.)

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

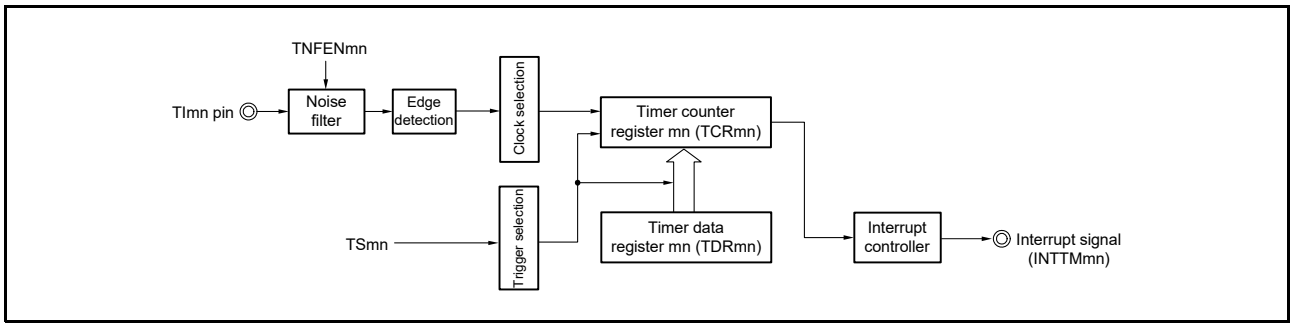
10.8.2 Operation as an external event counter

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

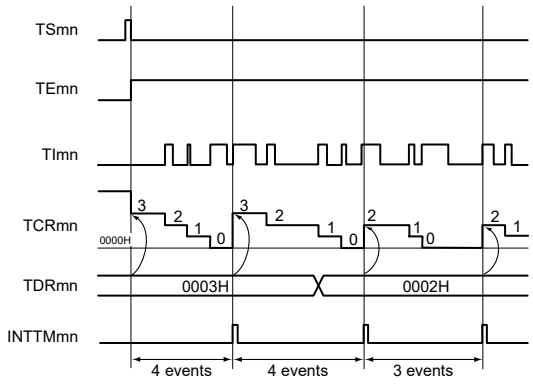
Timer counter register mn (TCRmn) operates as a down counter in the event counter mode.
The TCRmn register loads the value of timer data register mn (TDRmn) by setting any channel start trigger bit (TSmn) of timer channel start register m (TSm) to 1.
The TCRmn register counts down each time the valid 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 10 - 46 Block Diagram for Operation as an External Event Counter



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 10 - 47 Example of Basic Timing during Operation as an External Event Counter

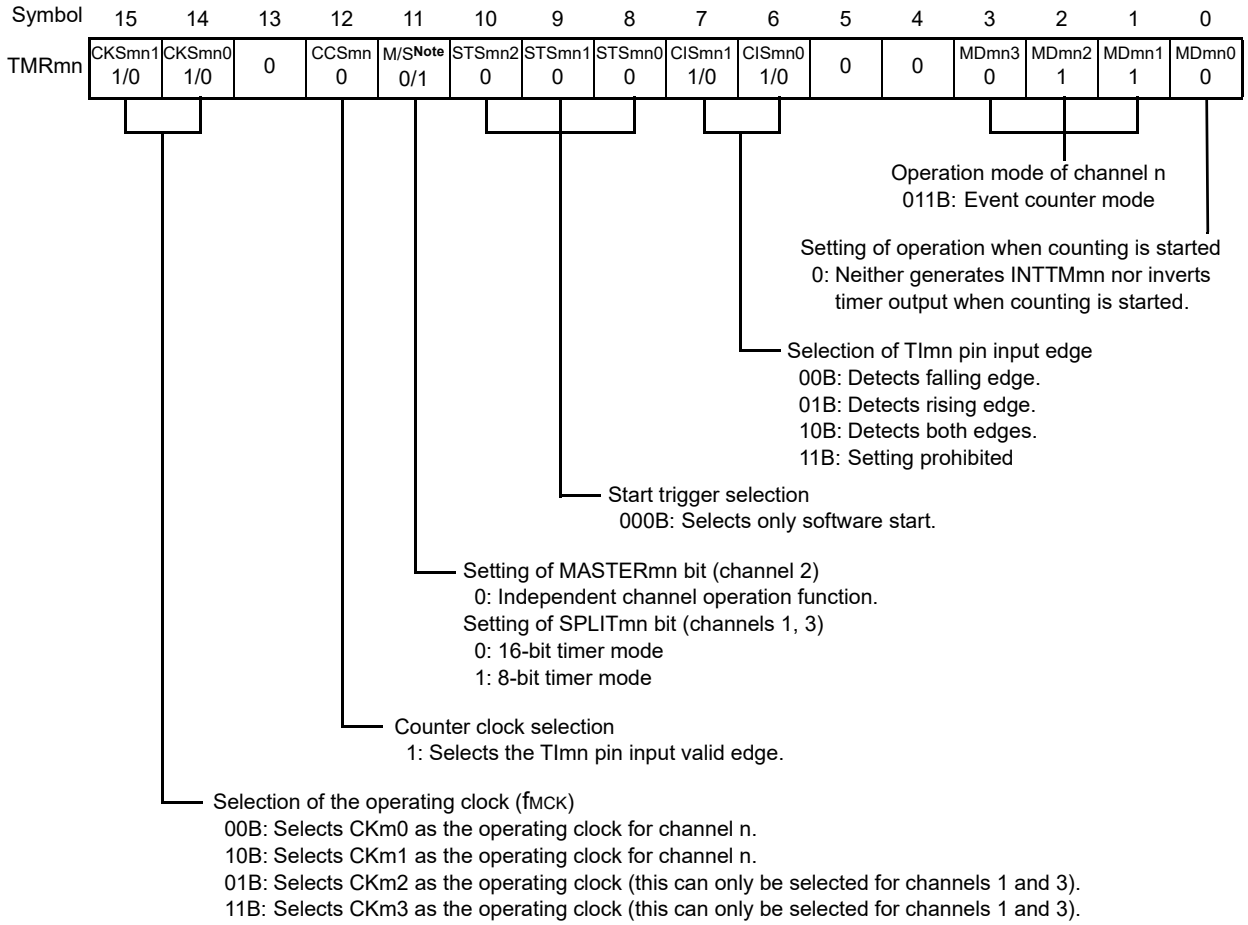


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

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 counter register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)

Figure 10 - 48 Example of Register Settings in External Event Counter Mode

a) Timer mode register mn (TMRmn)



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: Set this bit 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 and Remark are listed on the next page.)

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 10 - 49 Procedure for Operations When the External Event Counter Function Is to Be Used

	Software Operation	Hardware State
TAU default setting		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on state. 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 counter 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. → Set the TAUmRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	This stops supply of the input clock to timer array unit m. 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)

10.8.3 Operation as a frequency divider (channel 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 counter 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 the output on TO00 is not toggled. If the MD000 bit of timer mode register 00 (TMR00) is 1, INTTM00 is output and the output on TO00 is toggled.

After that, the TCR00 register counts down at the valid edge of the TI00 pin. When TCR00 = 0000H, it toggles the output on 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 operating clock.

Clock period of TO00 output = Ideal TO00 output clock period ± Operating 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 10 - 50 Block Diagram for Operation as a Frequency Divider

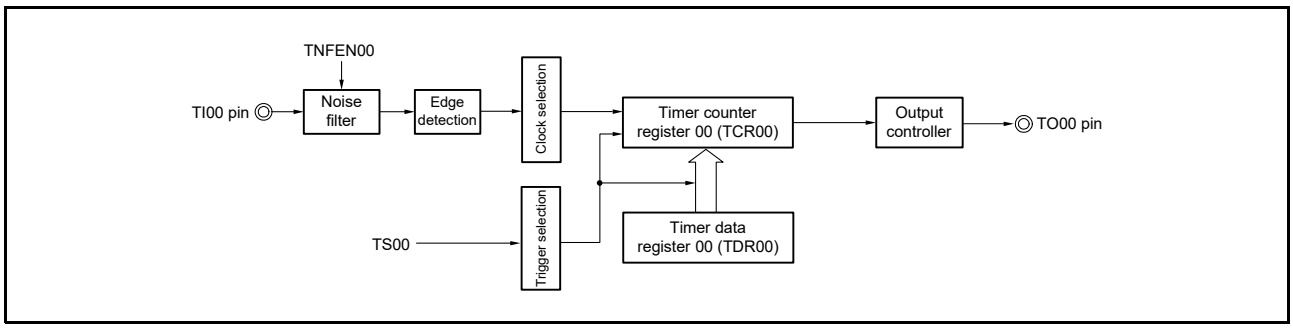
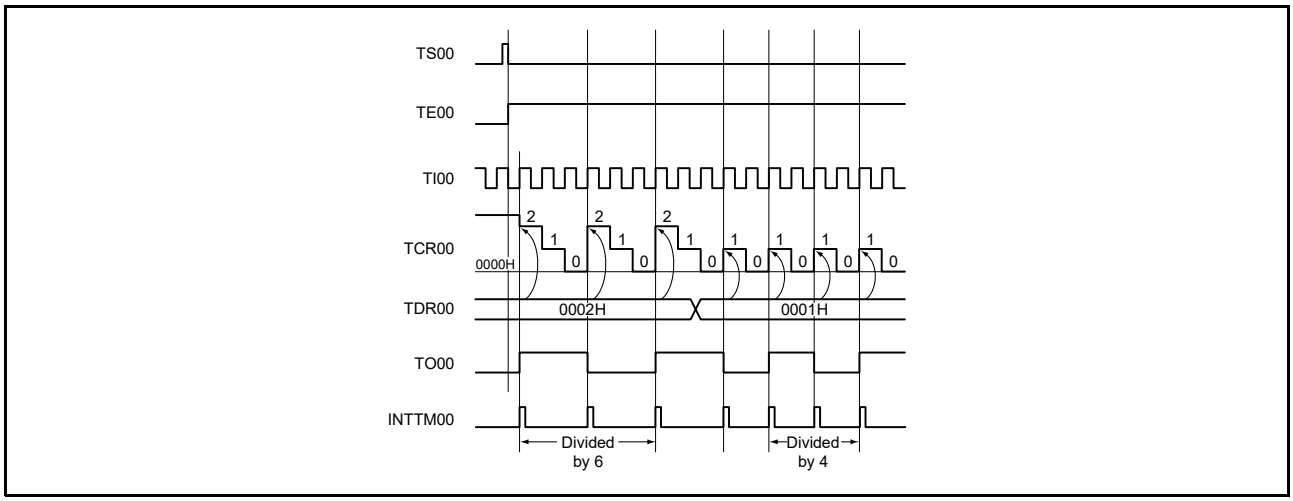


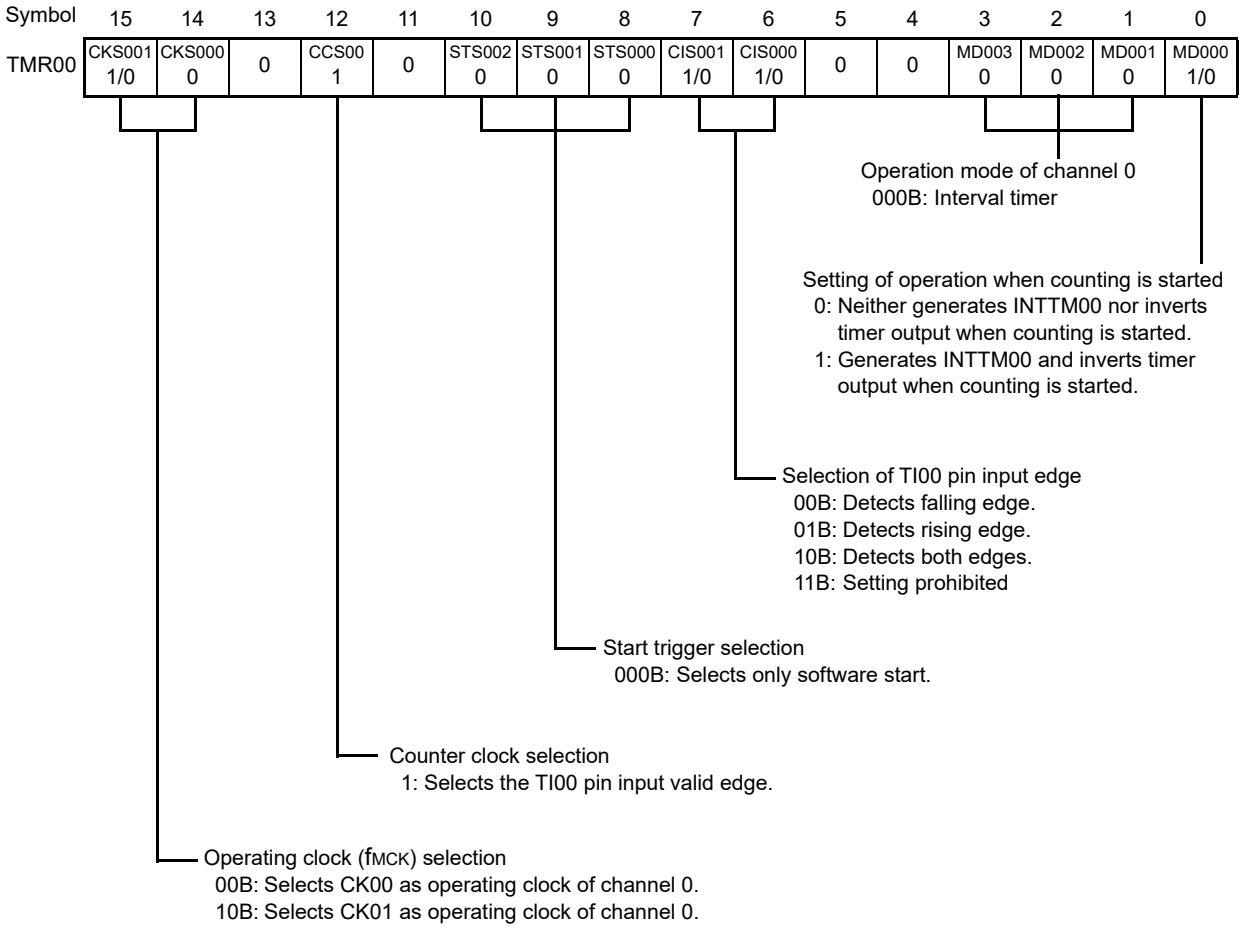
Figure 10 - 51 Example of Basic Timing during Operation as a Frequency Divider (MD000 = 1)



Remark TS00: Bit 0 of timer channel start register 0 (TS0)
TE00: Bit 0 of timer channel enable status register 0 (TE0)
TI00: TI00 pin input signal
TCR00: Timer counter register 00 (TCR00)
TDR00: Timer data register 00 (TDR00)
TO00: TO00 pin output signal

Figure 10 - 52 Example of Register Settings for Operation as a Frequency Divider

a) Timer mode register 00 (TMR00)



b) Timer output register 0 (TO0)

	Bit 0	
TO0	TO00 1/0	0: Outputs 0 from TO00. 1: Outputs 1 from TO00.

c) Timer output enable register 0 (TOE0)

	Bit 0	
TOE0	TOE00 1/0	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	TOL00 0	0: Set this bit to 0 when master channel output mode (TOM00 = 0)

e) Timer output mode register 0 (TOM0)

	Bit 0	
TOM0	TOM00 0	0: Sets master channel output mode.

Figure 10 - 53 Procedure for Operations When the Frequency Divider Function Is to Be Used

	Software Operation	Hardware State
TAU default setting		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on state. 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 0n (TMR0n) (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 counter register 00 (TCR00). 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 and retains its current state.
	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. → Set the TAUmRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	This stops supply of the input clock to timer array unit m. 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

10.8.4 Operation for 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.

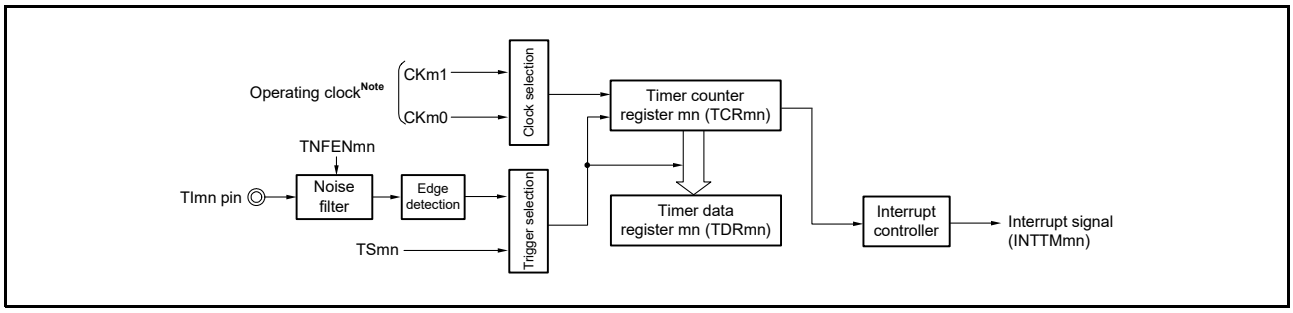
The pulse interval can be calculated by the following expression.

TImn input pulse interval = Period of counter clock × ((10000H × TSRmn: OVF) + (Captured 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 cycle of the operating clock occurs.

Timer counter 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 counter 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 state 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 STSmn[2:0] bits of the TMRmn register to 001B to use the valid edges of TImn as a start trigger and a capture trigger.

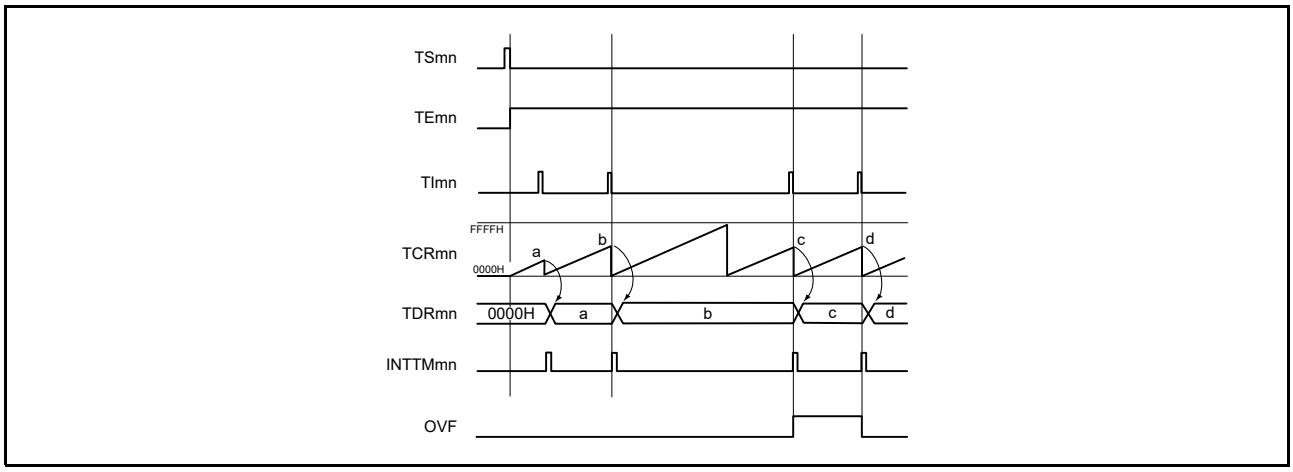
Figure 10 - 54 Block Diagram for Operation for Input Pulse Interval Measurement



Note For channels 1 and 3, the clock can be selected as CKm0, CKm1, CKm2, or CKm3.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 10 - 55 Example of Basic Timing during Operation for Input Pulse Interval Measurement (MDmn0 = 0)

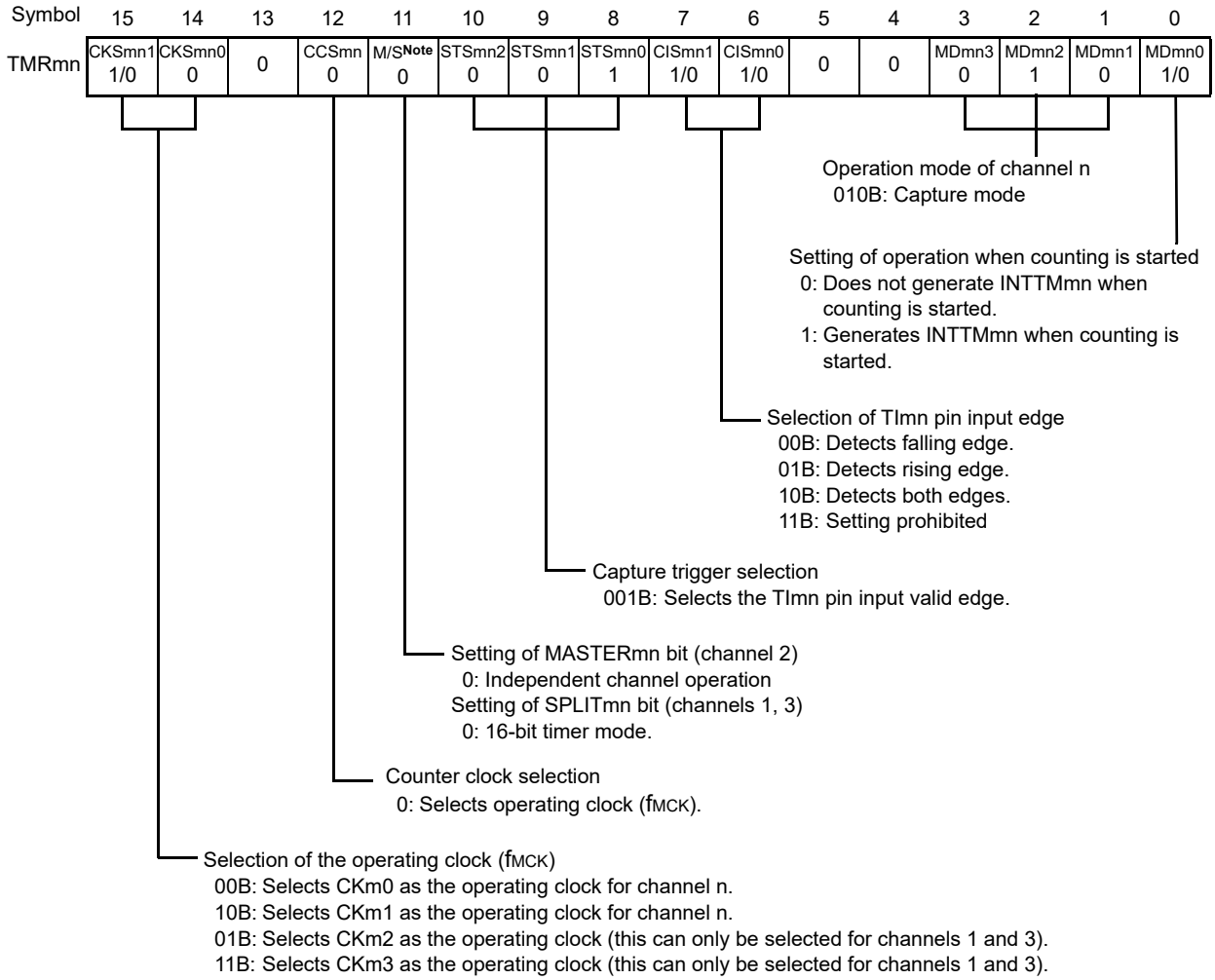


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

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 counter register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
OVF: Bit 0 of timer status register mn (TSRmn)

Figure 10 - 56 Example of Register Settings for Operation for Input Pulse Interval Measurement

a) Timer mode register mn (TMRmn)



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: Set this bit 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 and Remark are listed on the next page.)

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 10 - 57 Procedure for Operations When the Input Pulse Interval Measurement Function Is to Be Used

	Software Operation	Hardware State
TAU default setting		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on state. 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).	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 counter register mn (TCRmn) is cleared to 0000H. When the MDmn0 bit of the TMRmn register is 1, INTTMmn is generated.
During operation	Set values of only the CISmn[1:0] 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 valid edge of the TImn pin input is detected or the TSmn bit is set to 1, 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. → Set the TAUmRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	This stops supply of the input clock to timer array unit m. 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)

10.8.5 Operation for input signal high-/low-level width measurement

Caution When using a channel to implement the LIN-bus, set bit 1 (ISC1) of the input switch control register (ISC) to 1. In the following descriptions, read TImn as RxD0.

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.

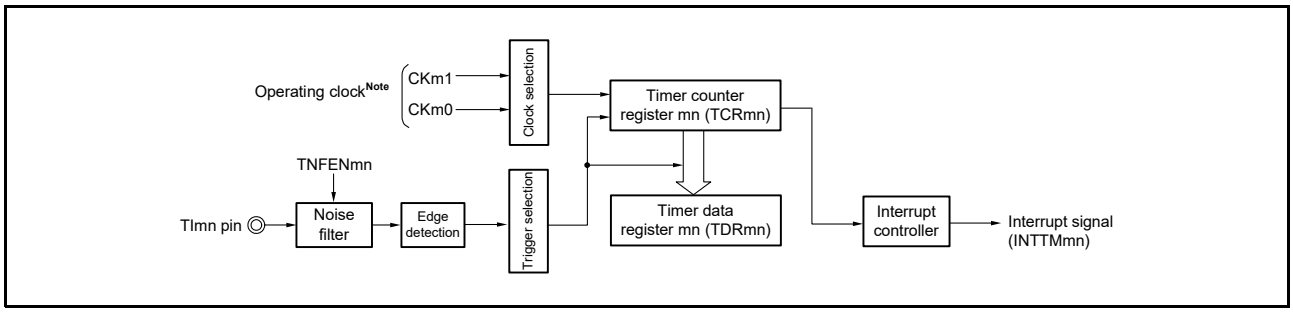
Signal width of TImn input = Period of counter clock × ((10000H × TSRmn: OVF) + (Captured value of TDRmn + 1))

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 cycle of the operating clock occurs.

Timer counter 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 state 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 counter 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 state 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 state 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 CISmn[1:0] 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.

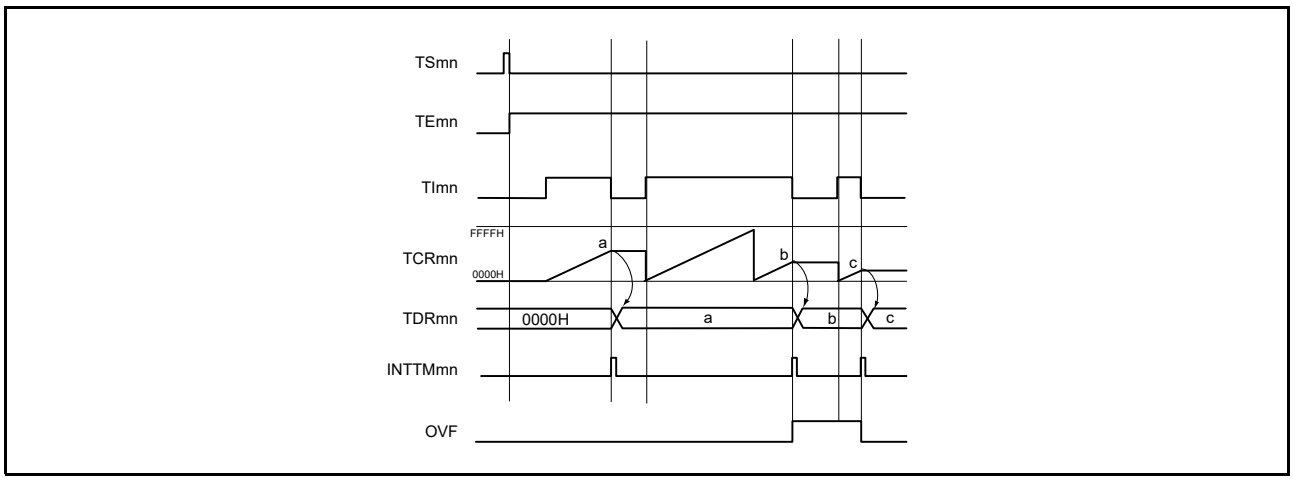
- CISmn[1:0] of TMRmn register = 10B: Low-level width is measured.
- CISmn[1:0] of TMRmn register = 11B: High-level width is measured.

Figure 10 - 58 Block Diagram for Operation for Input Signal High-/Low-level Width Measurement



Note For channels 1 and 3, the clock can be selected as CKm0, CKm1, CKm2, or CKm3.

Figure 10 - 59 Example of Basic Timing during Operation for Input Signal High-/Low-level Width Measurement

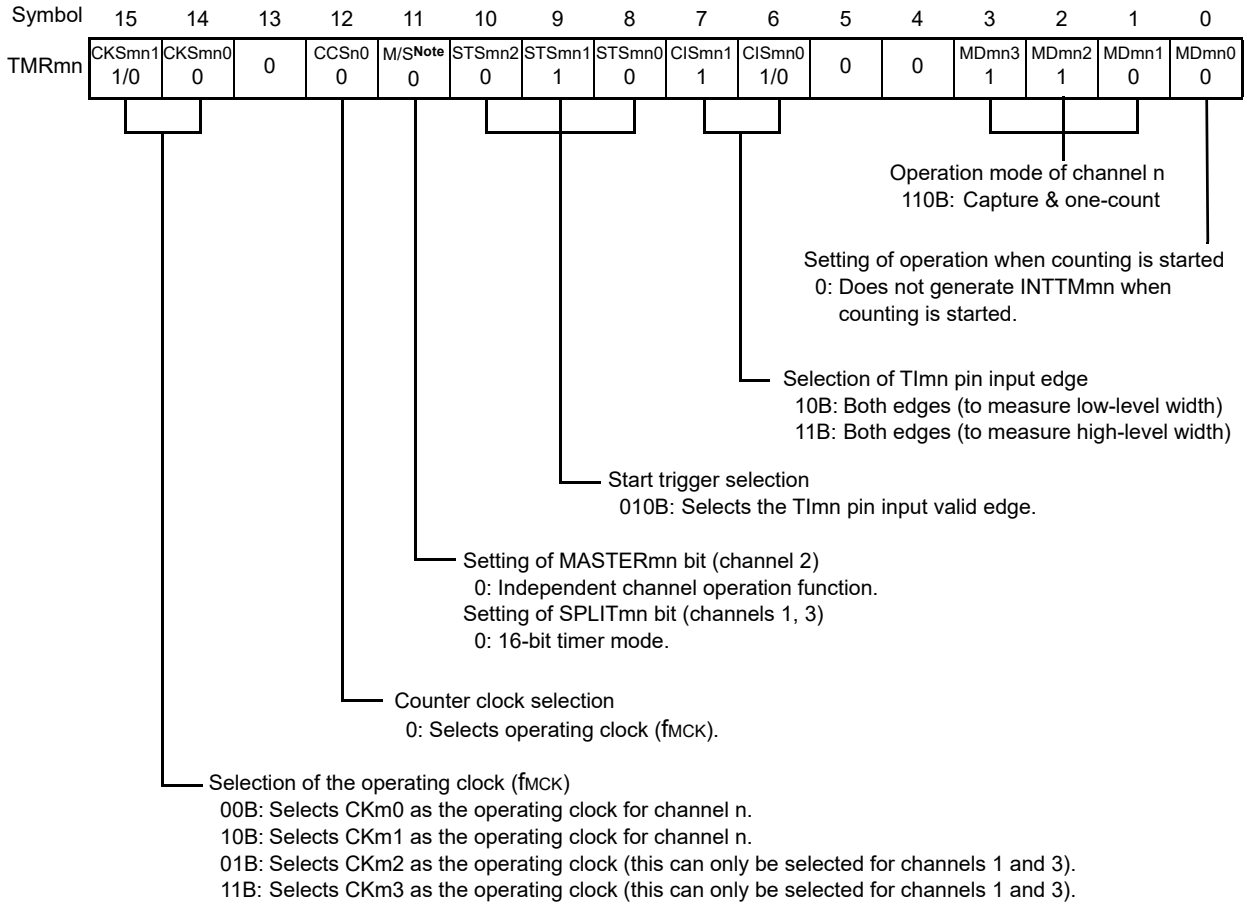


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

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 counter register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
OVF: Bit 0 of timer status register mn (TSRmn)

Figure 10 - 60 Example of Register Settings for Operation for Input Signal High-/Low-level Width Measurement

a) Timer mode register mn (TMRmn)



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: Set this bit 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 10 - 61 Procedure for Operations When the Input Signal High-/Low-level Width Measurement Function Is to Be Used

	Software Operation	Hardware State
TAU default setting		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on state. 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 state is set.
	Detects the TImn pin input count start valid edge. →	Clears timer counter register mn (TCRmn) to 0000H and starts counting up.
During operation	Set value of 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. → Set the TAUmRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	This stops supply of the input clock to timer array unit m. 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)

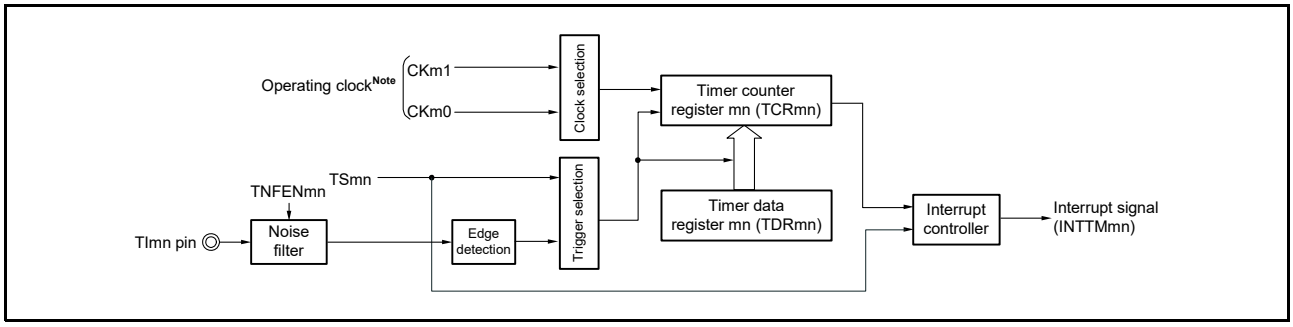
10.8.6 Operation as a delay counter

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 is also possible to start counting down and generate INTTMmn (timer interrupt) at any interval by setting TSmn to 1 by software while TEmn = 1.
The interrupt generation period can be calculated by the following expression.

Generation period of INTTMmn (timer interrupt) = Period of counter clock × (Set value of TDRmn + 1)

Timer counter 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 state is set.
Timer counter 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 counter 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.

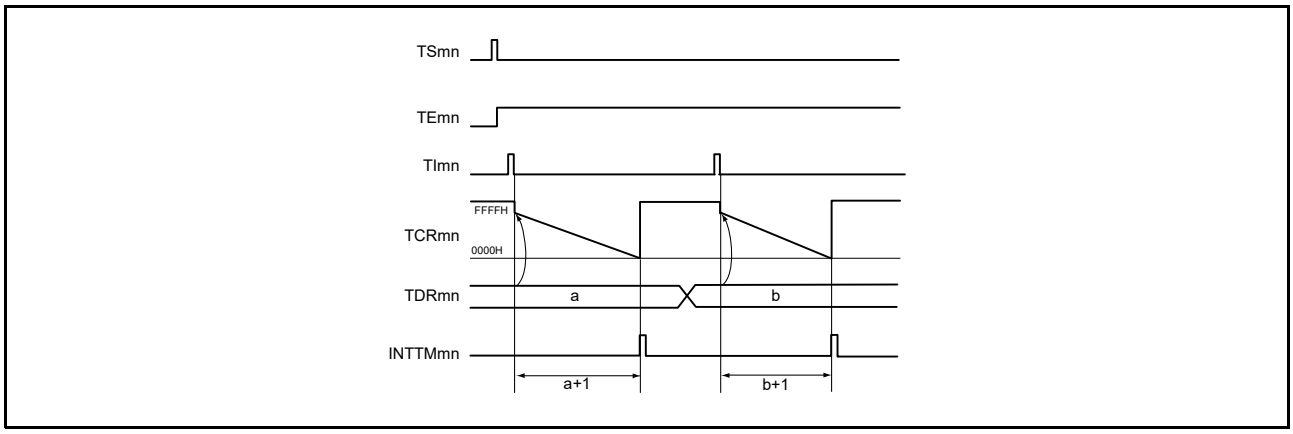
Figure 10 - 62 Block Diagram for Operation as a Delay Counter



Note For channels 1 and 3, the clock can be selected as CKm0, CKm1, CKm2, or CKm3.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 10 - 63 Example of Basic Timing during Operation as a Delay Counter

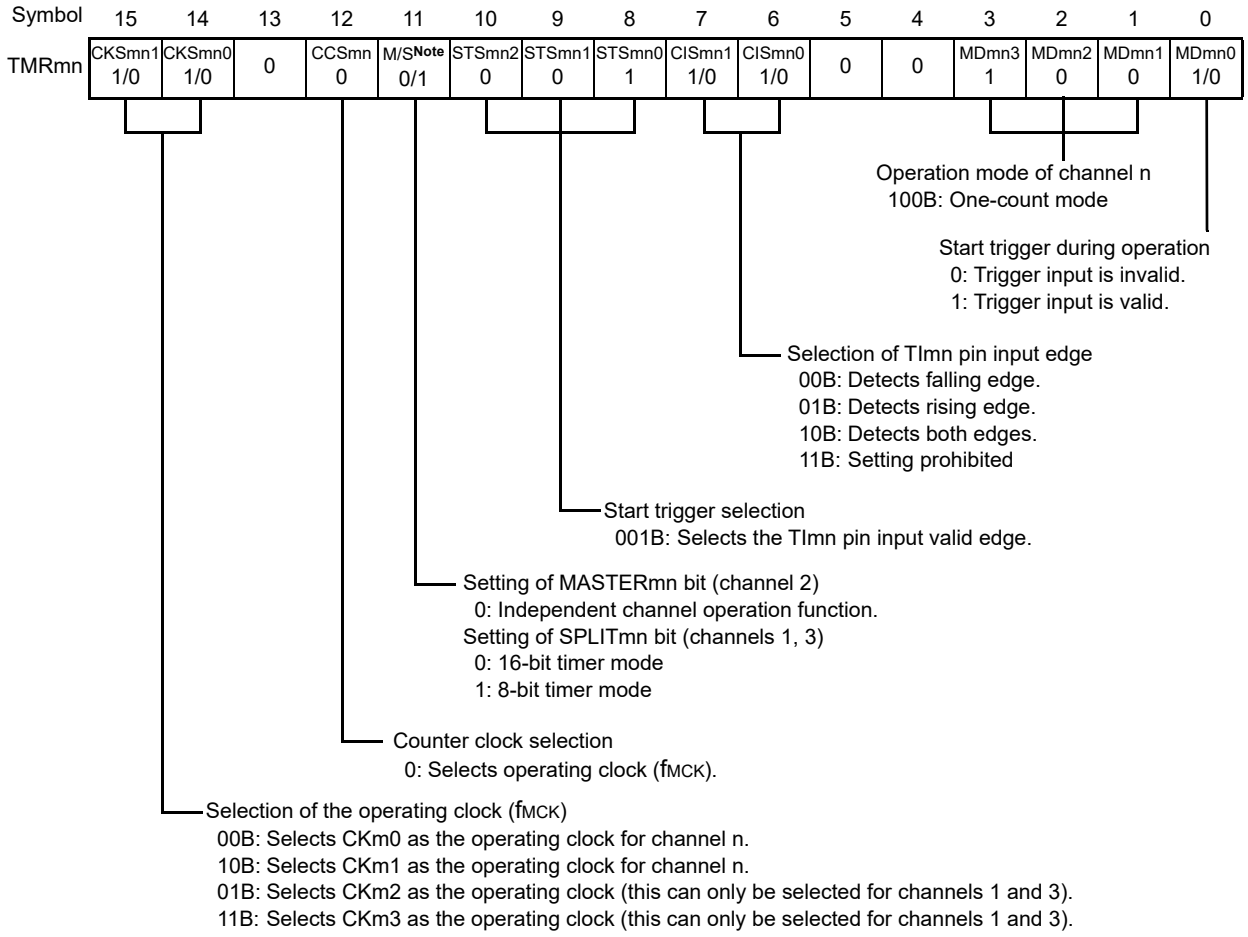


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

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 counter register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)

Figure 10 - 64 Example of Register Settings for Operation as a Delay Counter

a) Timer mode register mn (TMRmn)



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: Set this bit 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 and Remark are listed on the next page.)

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 10 - 65 Procedure for Operations When the Delay Counter Function Is to Be Used

	Software Operation	Hardware State
TAU default setting		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on state. 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 state 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 counter 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. → Set the TAUmRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	This stops supply of the input clock to timer array unit m. 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)

10.9 Simultaneous Channel Operation Function of Timer Array Unit

10.9.1 Operation for the 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.

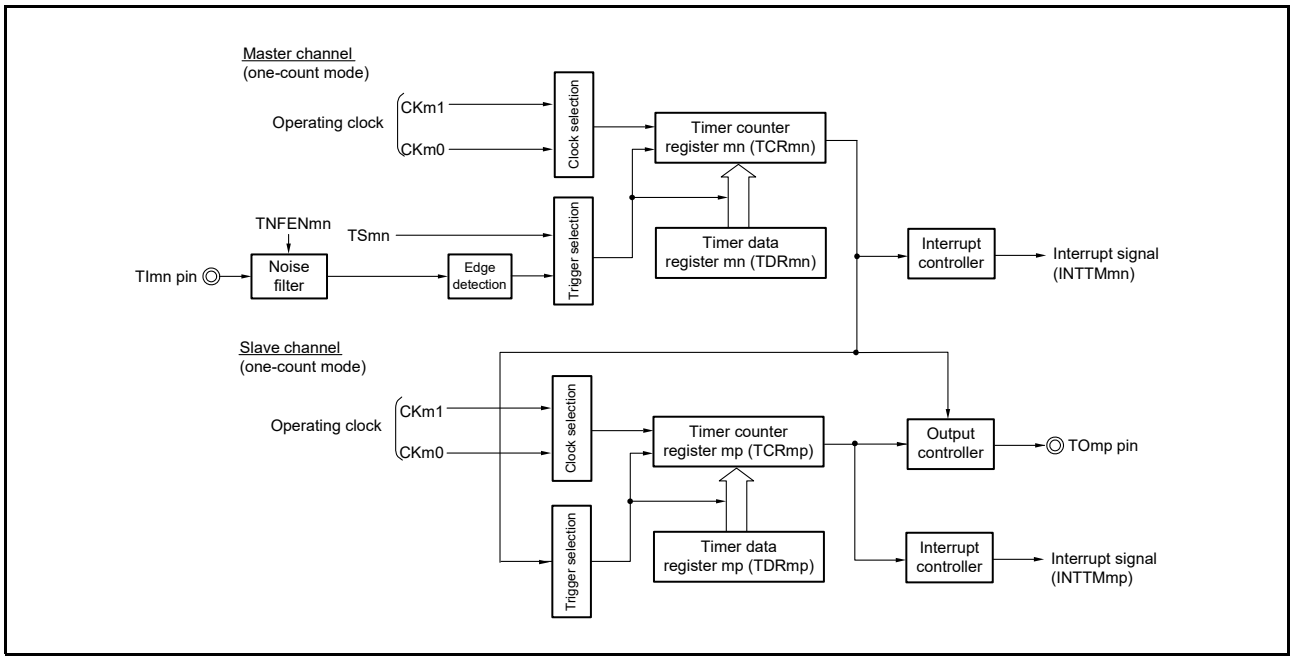
Delay time = {Set value of TDRmn (master) + 2} × Counter clock period Pulse width = {Set value of TDRmp (slave)} × Counter clock period

The master channel operates in the one-count mode and counts the delays. Timer counter 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 counter 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 counter clock. 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 cycle of the counter 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 is output. Rewrite the TDRmn register after INTTMmn is generated and the TDRmp register after INTTMmp is generated.**

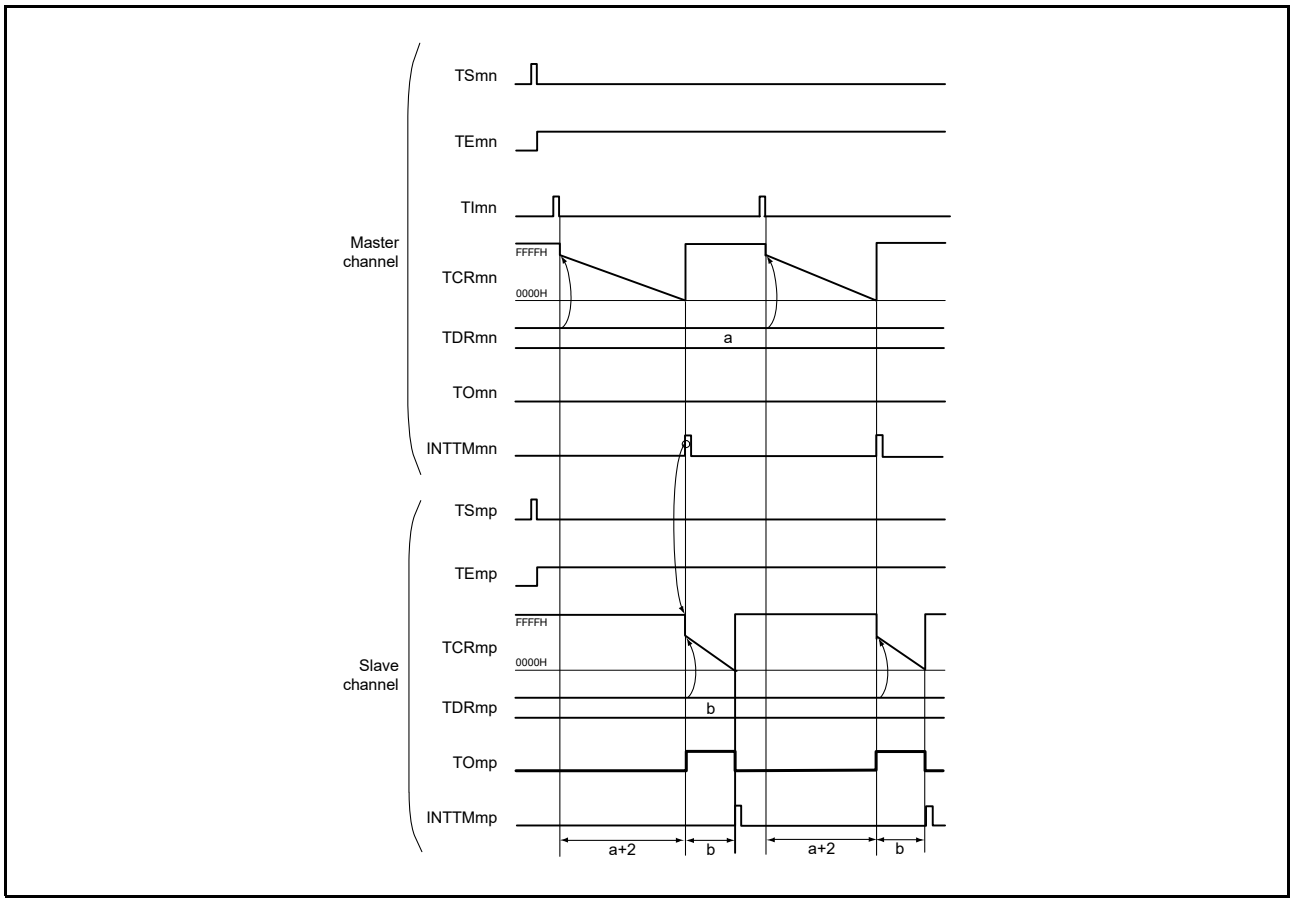
Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

Figure 10 - 66 Block Diagram for Operation for the One-shot Pulse Output Function



Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

Figure 10 - 67 Example of Basic Timing during Operation for the One-shot Pulse Output Function

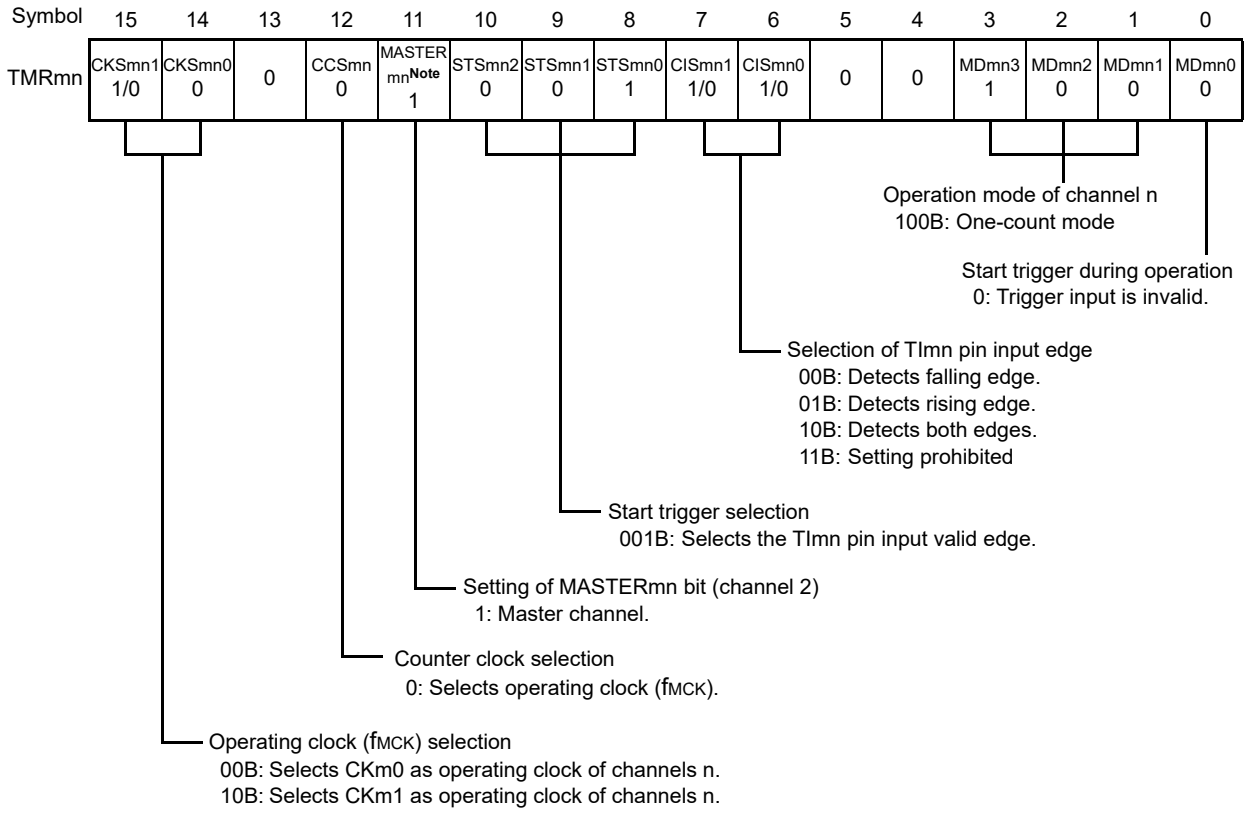


Remark 1. m: Unit number (m = 0), n: Master channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

Remark 2. TSmn, TSmp: Bits n and p of timer channel start register m (TSm)
TEmn, TEmp: Bits n and p of timer channel enable status register m (TEm)
TImn, TImp: Signals on the TImn and TImp input pins
TCRmn, TCRmp: Timer counter registers mn, mp (TCRmn, TCRmp)
TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)
TOmn, TOmp: Signals on the TOmn and TOmp output pins

Figure 10 - 68 Example of Register Settings for the Master Channel When the One-shot Pulse Output Function Is to Be Used

a) Timer mode register mn (TMRmn)



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: Set this bit to 0 when TOMmn = 0 (master channel output mode).

e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn 0

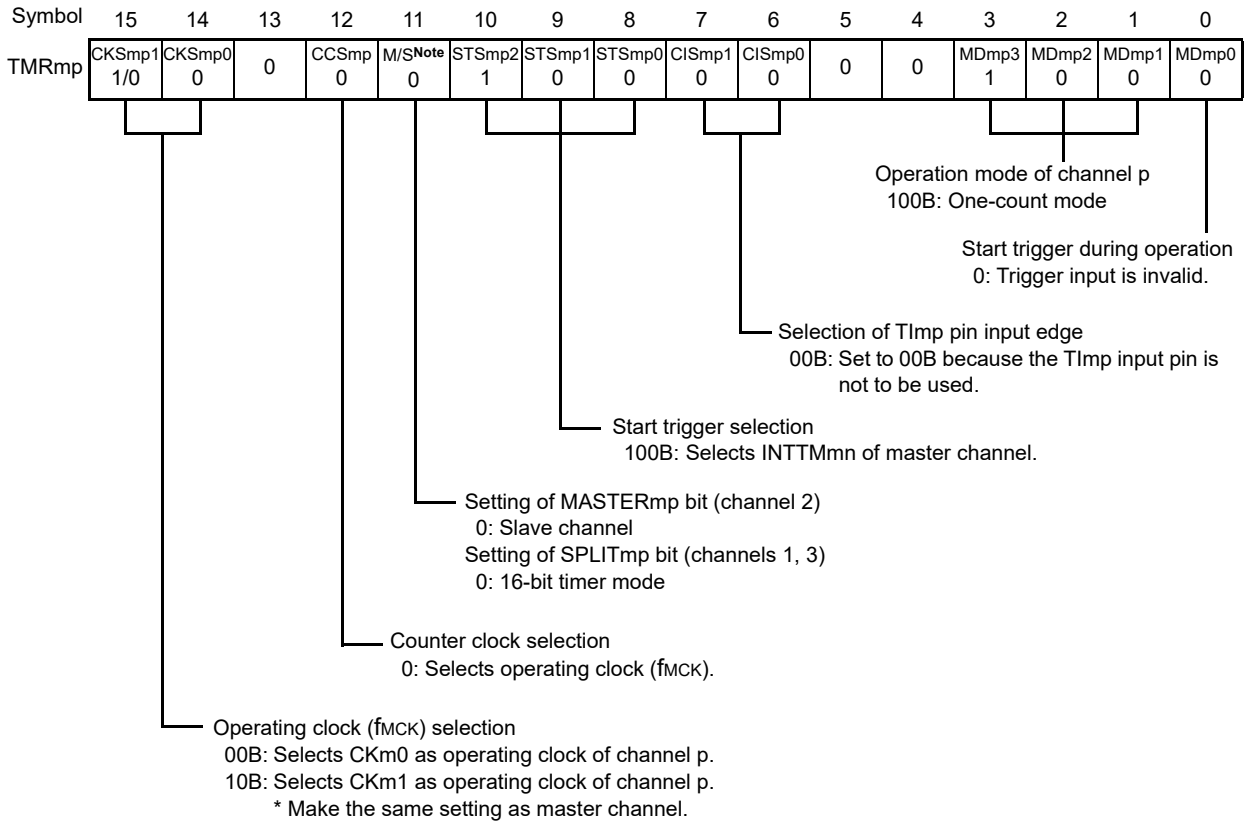
0: Sets master channel output mode.

Note TMRm2: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2)

Figure 10 - 69 Example of Register Settings for the Slave Channel When the One-shot Pulse Output Function Is to Be Used

a) Timer mode register mp (TMRmp)



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: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

Figure 10 - 70 Procedure for Operations When the One-shot Pulse Output Function Is to Be Used (1/2)

	Software Operation	Hardware State
TAU default setting		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable registers 0 (PER0) to 1. →	Power-on state. 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.

Figure 10 - 70 Procedure for Operations When the One-shot Pulse Output Function Is to Be Used (2/2)

	Software Operation	Hardware State
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 state. 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 CISmn[1:0] 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 counter register mn (TCRmn) by 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), 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 start trigger detection. 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 cycle of the counter 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 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 and retains its current state.
	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. → Set the TAUmRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	This stops supply of the input clock to timer array unit m. 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.)

Operation is resumed.

Note Do not set the TSmn bit of the slave channel to 1.

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

10.9.2 Operation for the 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} × Counter 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}

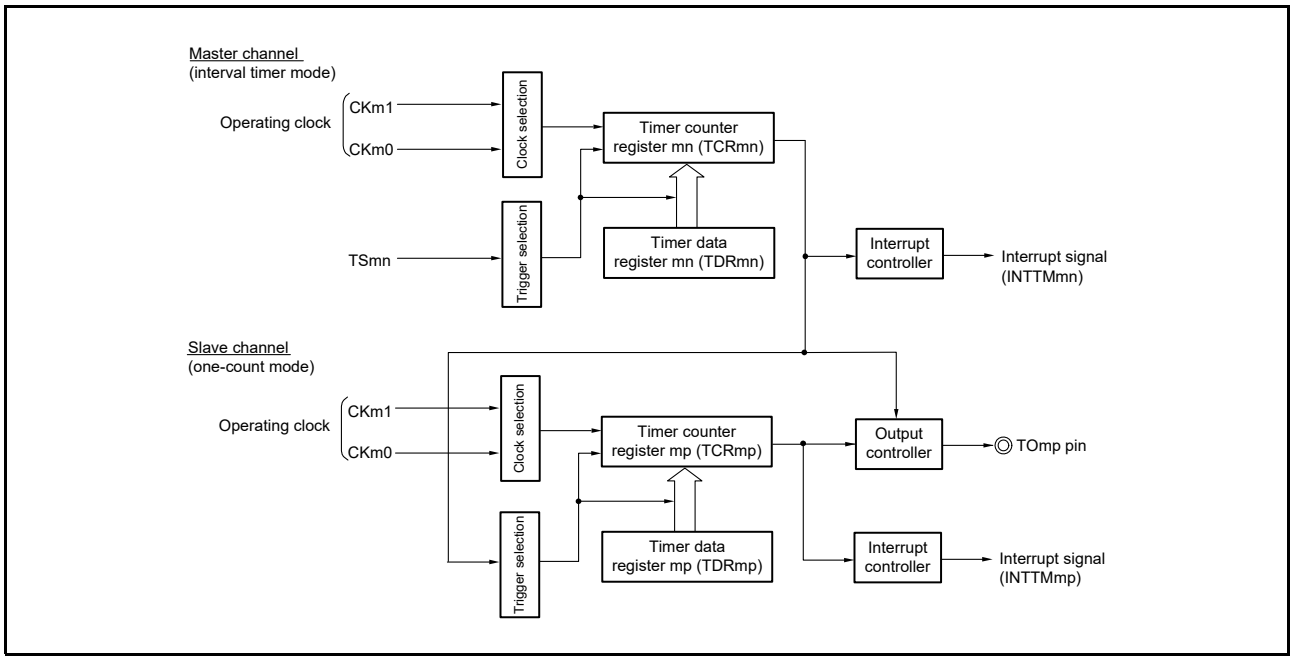
Remark The duty factor exceeds 100% if the set value of TDRmp (slave) > (set value of TDRmn (master) + 1), it summarizes to 100% output.

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 counter register mn (TCRmn), and the counter counts down in synchronization with the counter 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 cycle 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.**

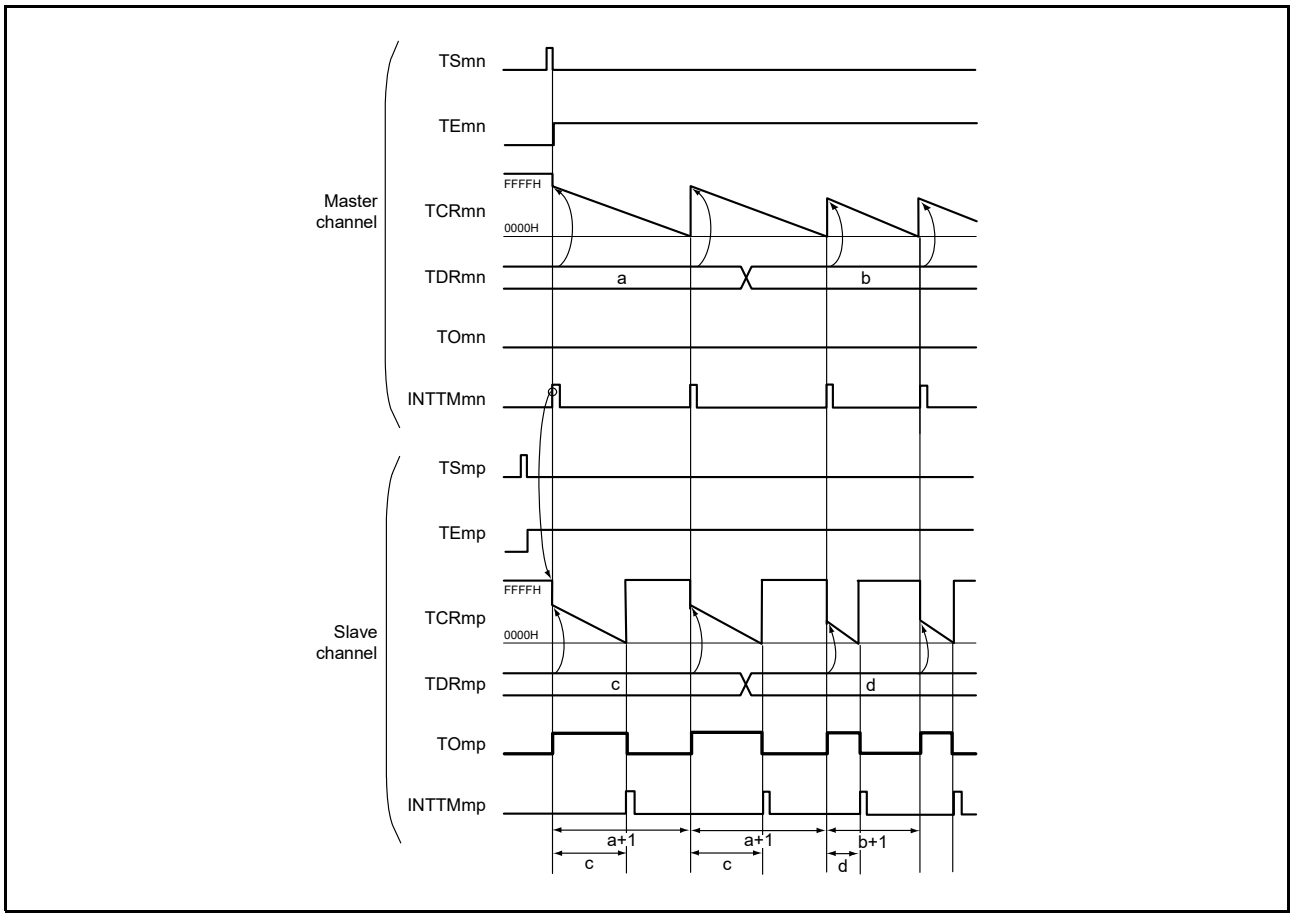
Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

Figure 10 - 71 Block Diagram for Operation for the PWM Function



Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

Figure 10 - 72 Example of Basic Timing during Operation for the PWM Function

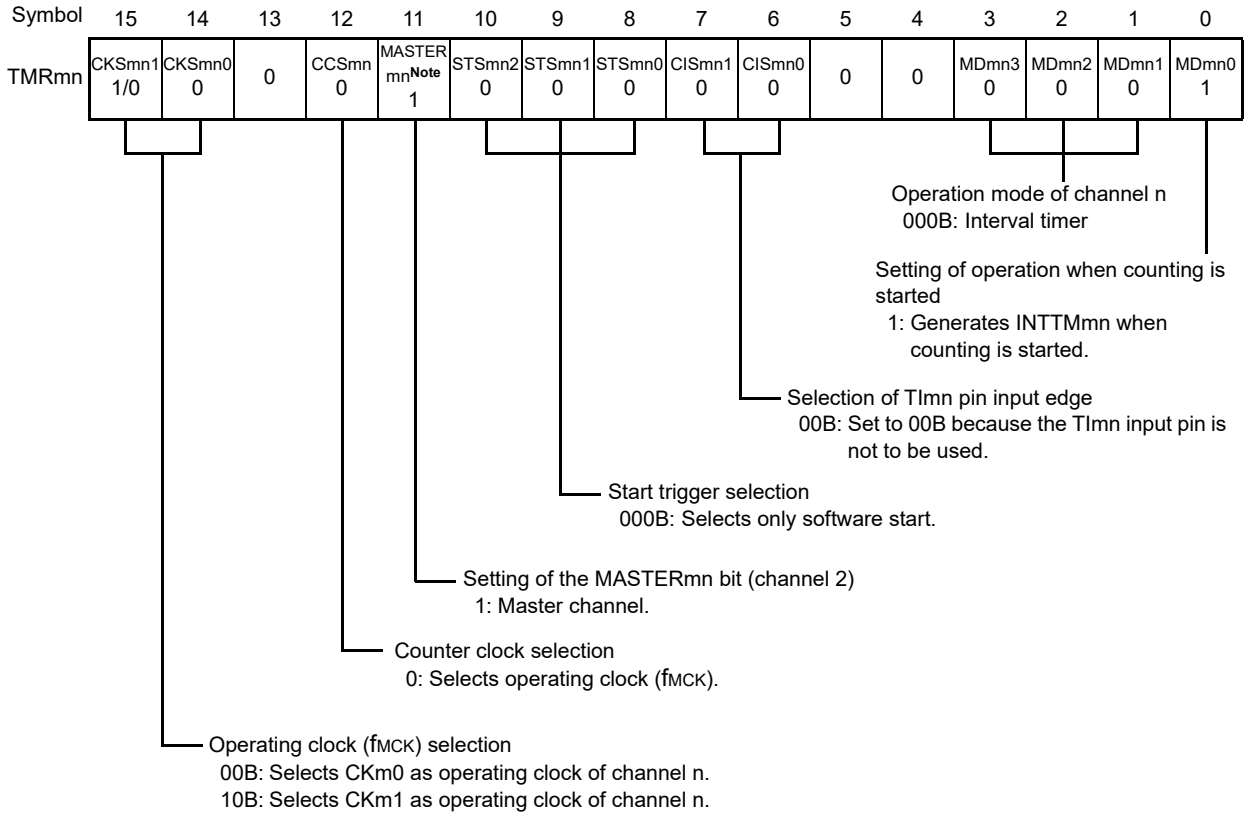


Remark 1. m: Unit number (m = 0), n: Master channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

Remark 2. TSmn, TSmp: Bits n and p of timer channel start register m (TSm)
TEmn, TEmp: Bits n and p of timer channel enable status register m (TEm)
TCRmn, TCRmp: Timer counter registers mn, mp (TCRmn, TCRmp)
TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)
TOmn, TOmp: Signals on the TOmn and TOmp output pins

Figure 10 - 73 Example of Register Settings for the Master Channel When the PWM Function Is to Be Used

a) Timer mode register mn (TMRmn)



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: Set this bit to 0 when TOMmn = 0 (master channel output mode).

e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn 0

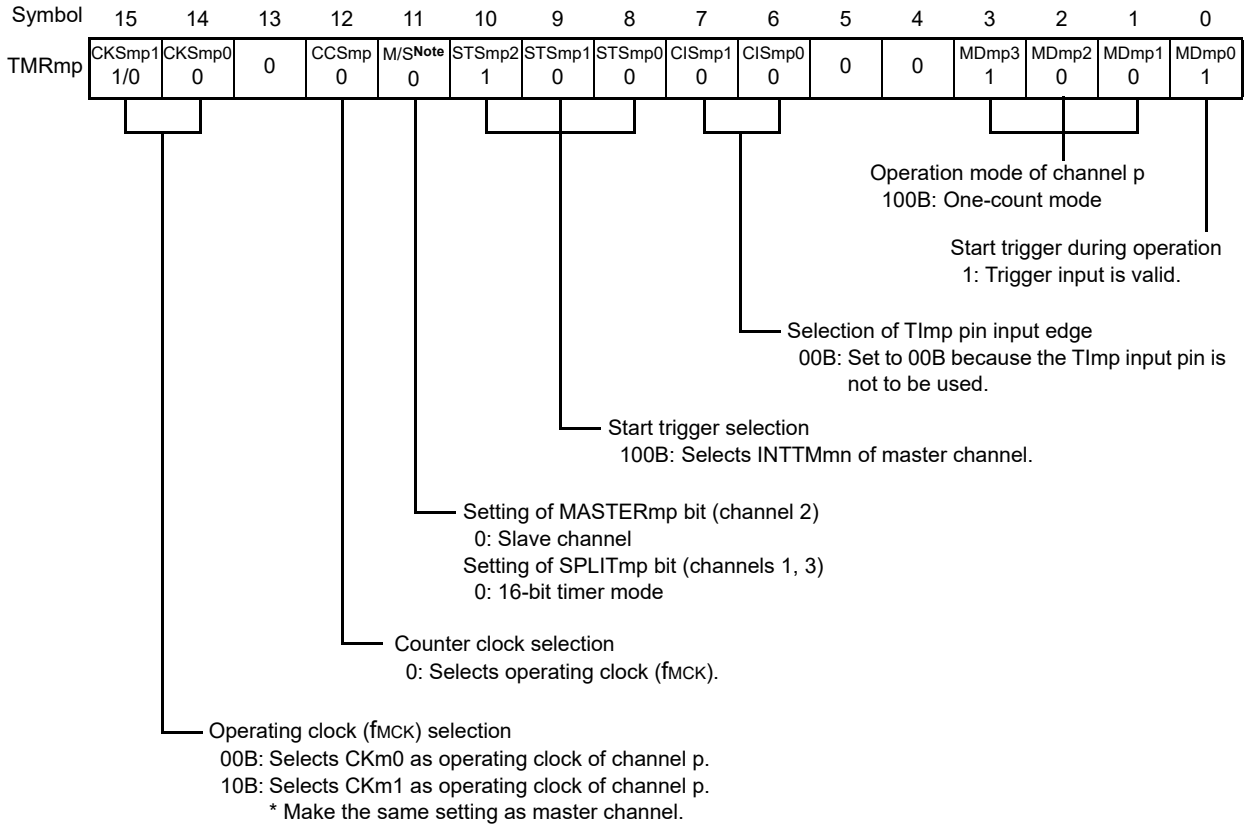
0: Sets master channel output mode.

Note TMRm2: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2)

Figure 10 - 74 Example of Register Settings for the Slave Channel When the PWM Function Is to Be Used

a) Timer mode register mp (TMRmp)



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: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

Figure 10 - 75 Procedure for Operations When the PWM Function Is to Be Used (1/2)

	Software Operation	Hardware State
TAU default setting		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on state. 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. Sets the TOmp bit and determines default level of the TOmp output. →	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.
	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.

Figure 10 - 75 Procedure for Operations When the PWM Function Is to Be Used (2/2)

	Software Operation	Hardware State
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 counter 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 cycle of the counter 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 and retains its current state.
	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. → Set the TAUmRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	This stops supply of the input clock to timer array unit m. 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.)

Operation is resumed.

<R> **Remark** m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 3)

10.9.3 Operation for the 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} × Counter 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
```

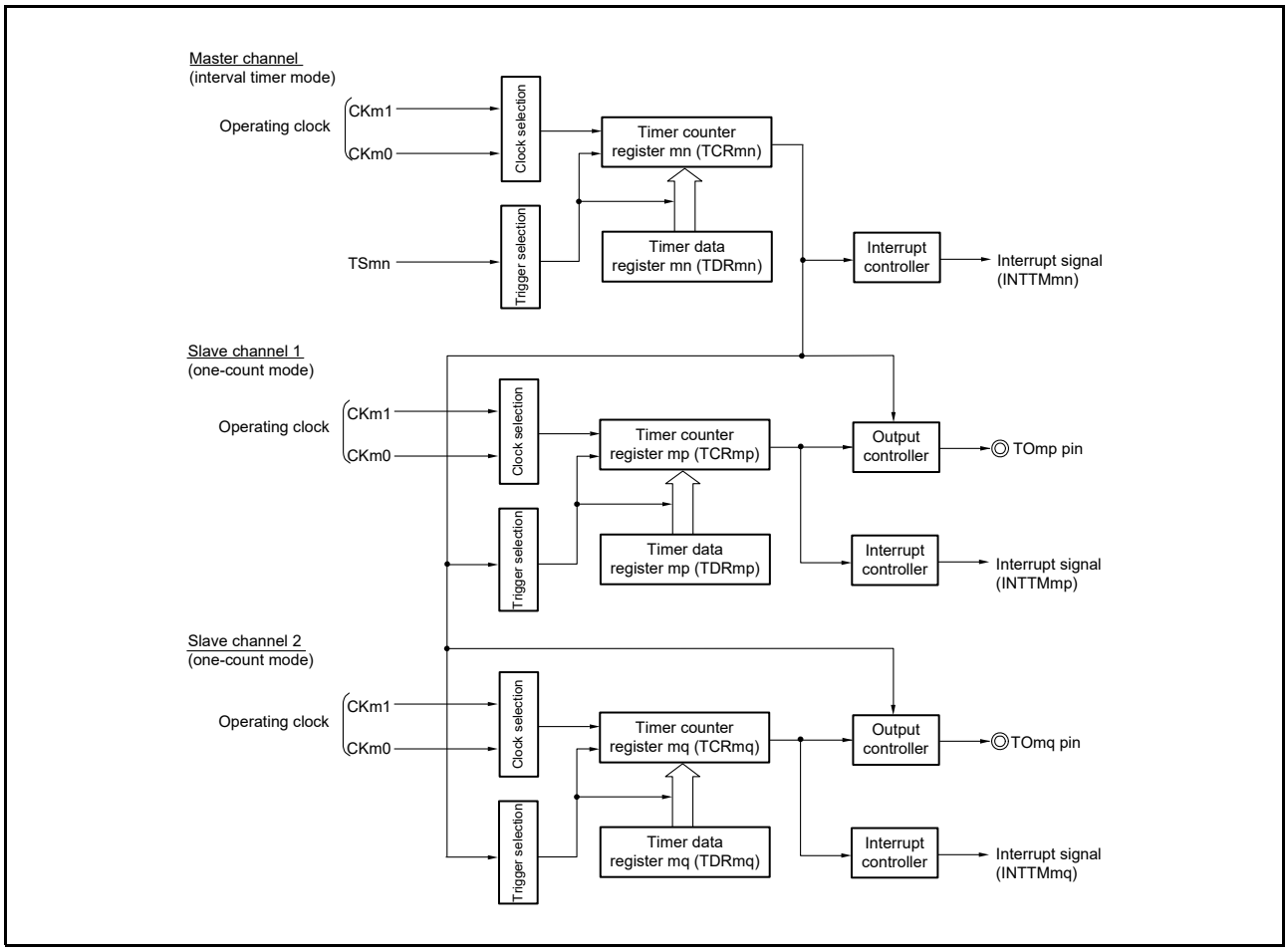
Remark Although the duty factor exceeds 100% if the set value of TDRmp (slave 1) > {set value of TDRmn (master) + 1} or if the {set value of TDRmq (slave 2)} > {set value of TDRmn (master) + 1}, it is summarized into 100% output.

Timer counter 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 cycle of the counter 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 cycle of the counter 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).**

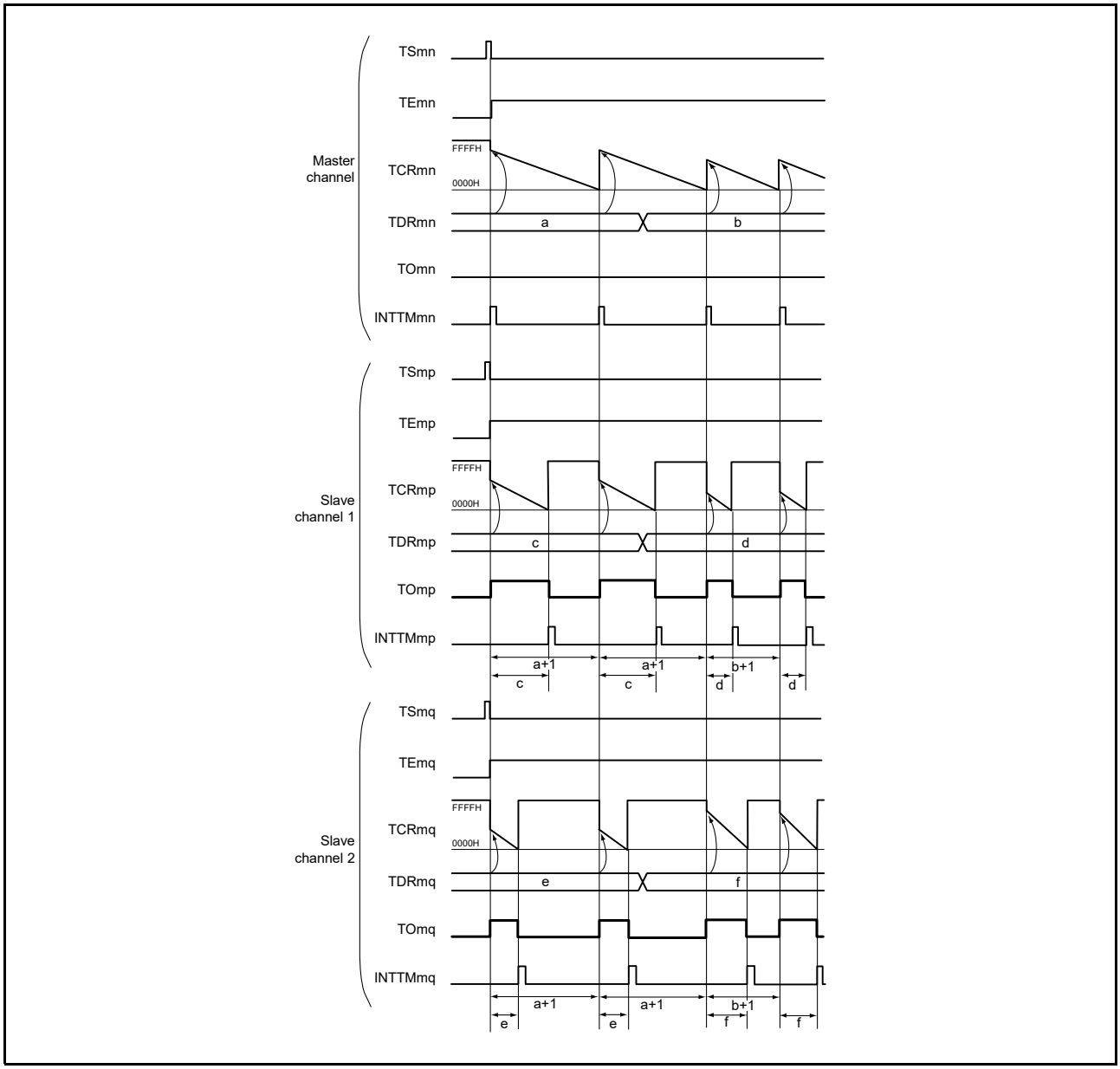
Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 3 (Where p and q are integers)

Figure 10 - 76 Block Diagram for Operation for the Multiple PWM Output Function (for Two Types of PWM Output)



Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 3 (Where p and q are integers)

Figure 10 - 77 Example of Basic Timing during Operation for the Multiple PWM Output Function (for Two Types of PWM Output)

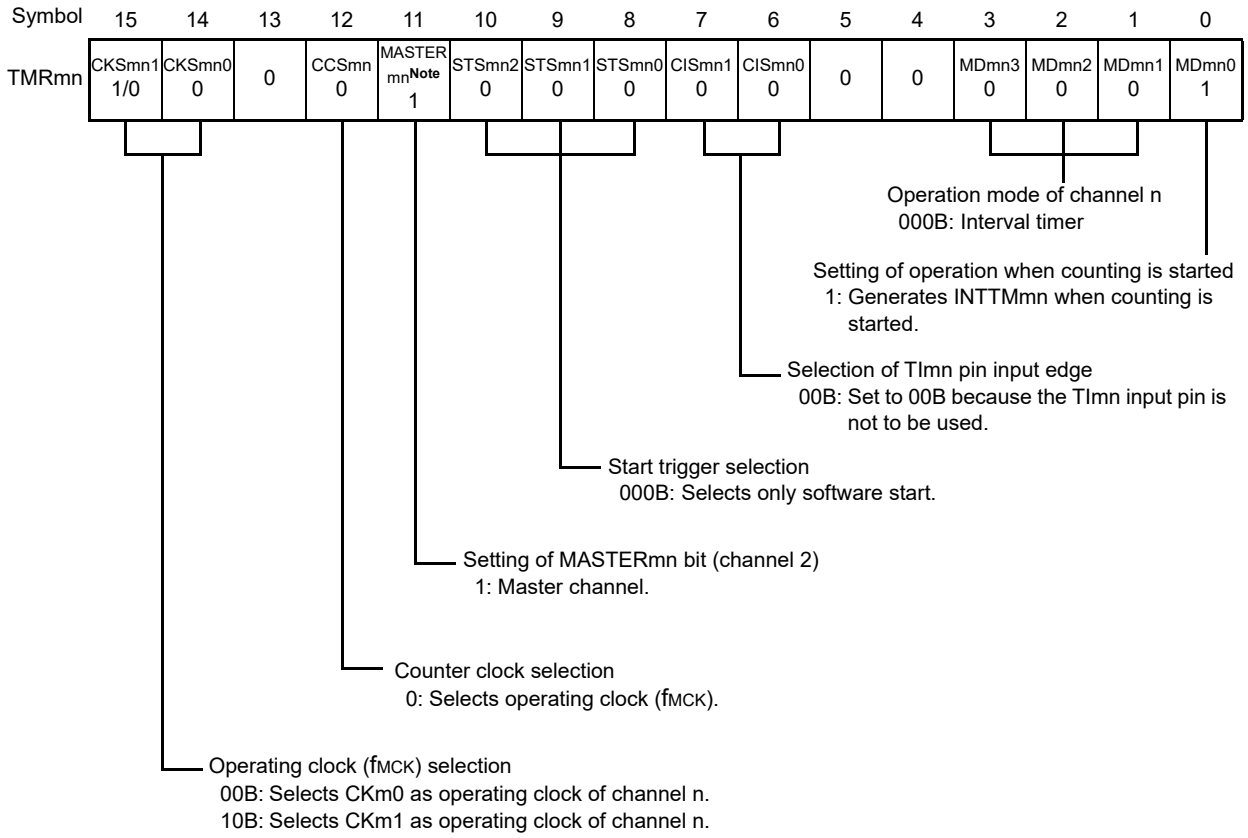


Remark 1. m: Unit number (m = 0), n: Master channel number (n = 0, 2)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 3 (Where p and q are integers)

Remark 2. TSmn, TSmp, TSmq: Bits n, p, and q of timer channel start register m (TSm)
TEmn, TEmp, TEmq: Bits n, p, and q of timer channel enable status register m (TEm)
TCRmn, TCRmp, TCRmq: Timer counter registers mn, mp, mq (TCRmn, TCRmp, TCRmq)
TDRmn, TDRmp, TDRmq: Timer data registers mn, mp, mq (TDRmn, TDRmp, TDRmq)
TOmn, TOmp, TOmq: Signals on the TOmn, TOmp, and TOmq output pins

Figure 10 - 78 Example of Register Settings for the Master Channel When the Multiple PWM Output Function Is to Be Used

a) Timer mode register mn (TMRmn)



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: Set this bit to 0 when TOMmn = 0 (master channel output mode).

e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn 0

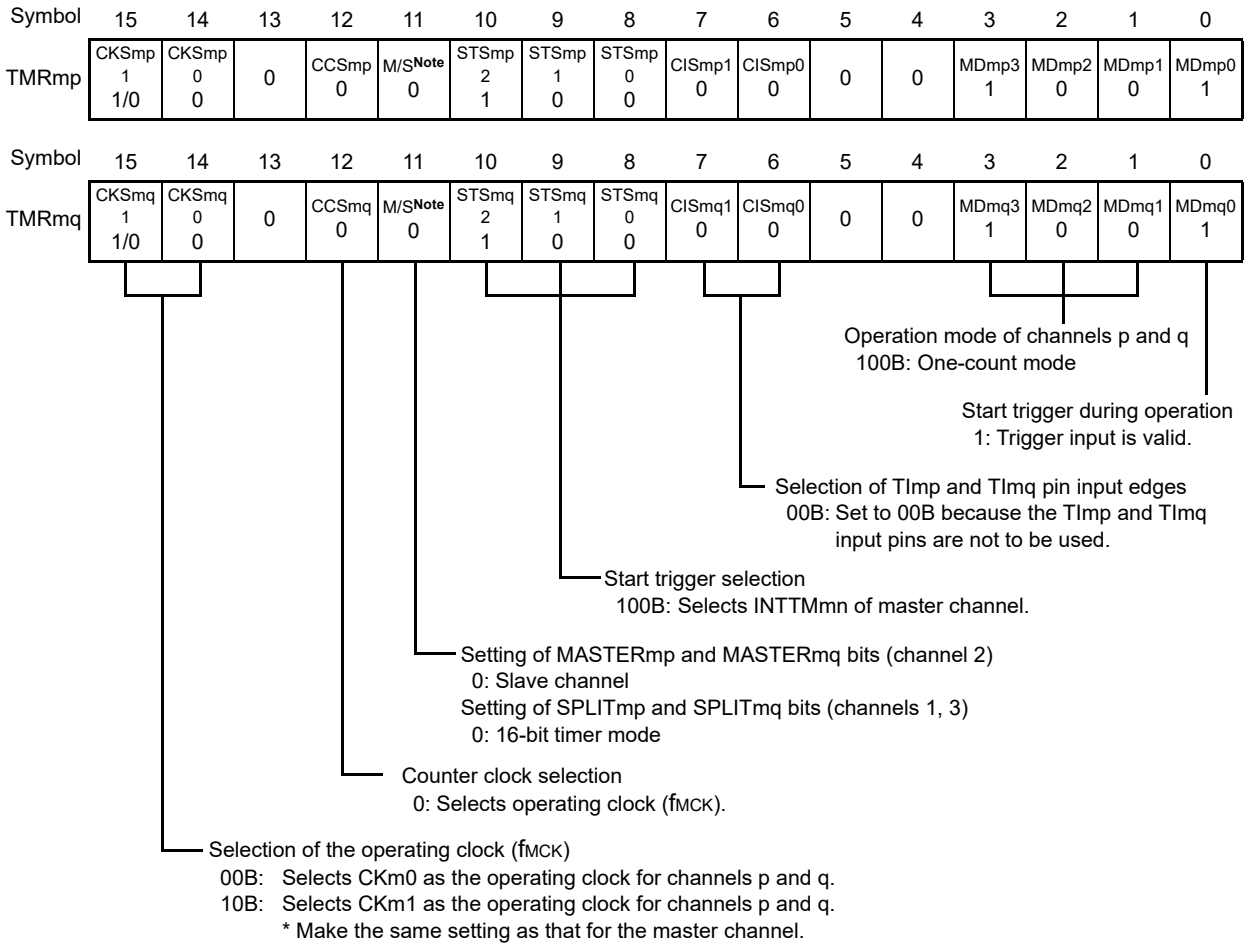
0: Sets master channel output mode.

Note TMRm2: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2)

Figure 10 - 79 Example of Register Settings for the Slave Channel When the Multiple PWM Output Function Is to Be Used (for Two Types of PWM Output)

a) Timer mode registers mp and mq (TMRmp and TMRmq)



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 and Remark are listed on the next page.)

Note TMRm2: MASTERmp and MASTERmq bits
TMRm1, TMRm3: SPLITmp and SPLITmq bits

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2)
p: Slave channel number, q: Slave channel number
$n < p < q \leq 3$ (Where p and q are integers)

Figure 10 - 80 Procedure for Operations When the Multiple PWM Output Function Is to Be Used (for Two Types of PWM Output) (1/2)

	Software Operation	Hardware State
TAU default setting		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on state. 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, 0q (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). Sets the TOLmp and TOLmq bits. Sets the TOmp and TOmq bits and determines default level of the TOmp and TOmq outputs. →	The TOmp and TOmq pins go into Hi-Z output state. 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.

Figure 10 - 80 Procedure for Operations When the Multiple PWM Output Function Is to Be Used (for Two Types of PWM Output) (2/2)

Operation is resumed.

	Software Operation	Hardware State
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 TSR0q registers are not used.	The counter of the master channel loads the TDRmn register value to timer counter 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 level of TOmp becomes active one cycle of the counter 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 level of TOmq becomes active one cycle of the counter 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 outputs are not initialized and retain their current states.
	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. → Set the TAUmRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	This stops supply of the input clock to timer array unit m. 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: Master channel number (n = 0, 2)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 3 (Where p and q are integers)

10.10 Cautions When Using Timer Array Unit

10.10.1 Cautions when using timer output

Pins may be assigned multiplexed timer output and other alternate functions. The assignment depends on the product.
If you intend to use a timer output, set the outputs from all other multiplexed pin functions to their initial values.
For details, see **7.5 Register Settings When Using Alternate Function**.

Section 11 Timer RJ

11.1 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 11 - 1 lists the timer RJ specifications. **Figure 11 - 1** shows the timer RJ block diagram.

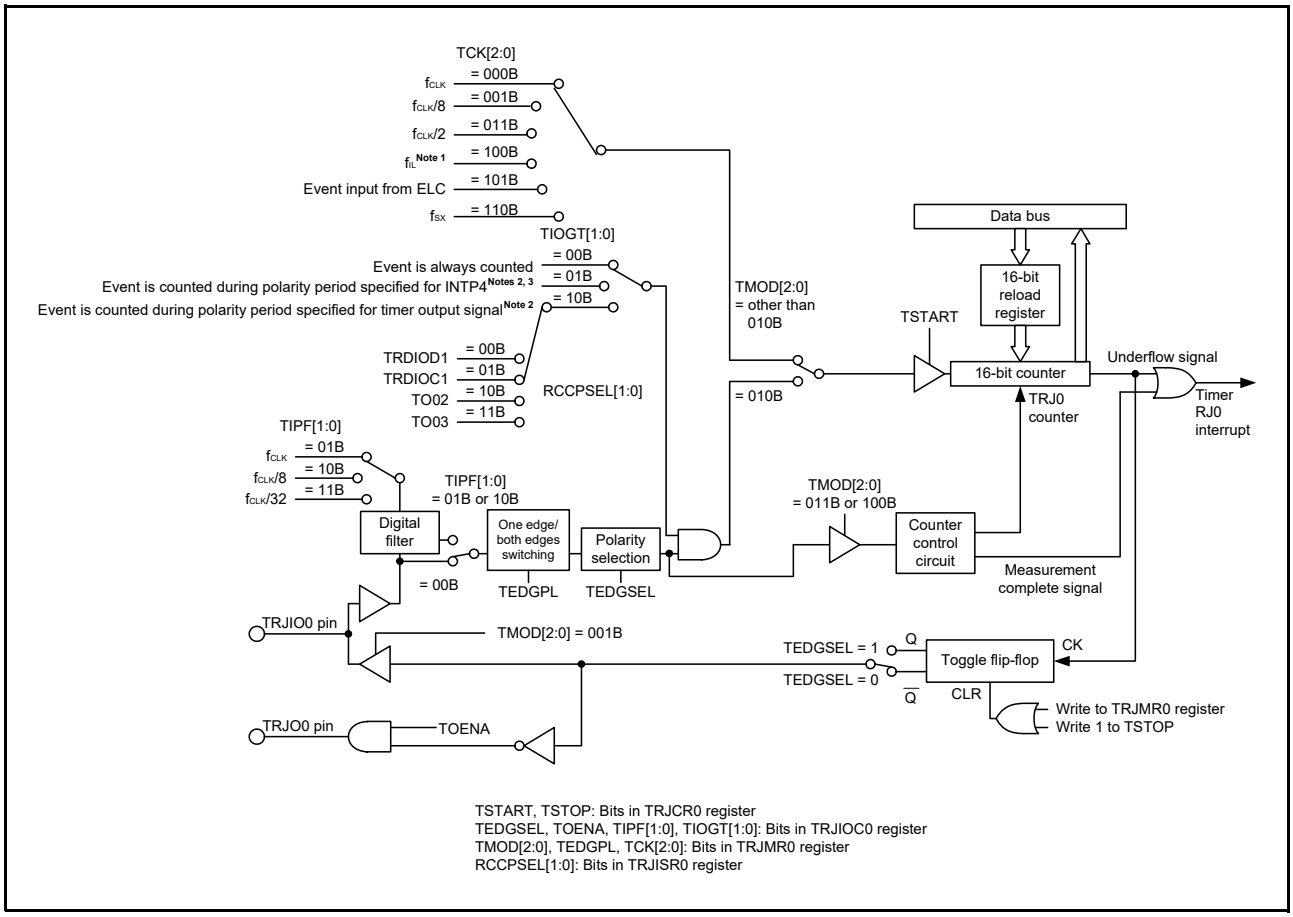
Table 11 - 1 Timer RJ Specifications

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}, f_{SX}, 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.

11.2 Configuration of Timer RJ

Figure 11 - 1 shows the timer RJ block diagram and **Table 11 - 2** lists the timer RJ pin configuration.

Figure 11 - 1 Timer RJ Block Diagram



Note 1. When selecting fIL as the count source, set the WUTMMCK0 bit in the subsystem clock supply mode control register (OSMC) to 1. However, fIL cannot be selected as the count source for timer RJ when fSX is selected as the count source for the realtime clock or the 32-bit interval timer.

Note 2. The polarity can be selected by the RCCPSEL2 bit in the TRJISR0 register.

Note 3. INTP4 is only usable for this purpose in the 30- to 64-pin products.

Table 11 - 2 Timer RJ Pin Configuration

Pin Name	I/O	Function
INTP4	Input	Event counter mode control for timer RJ
TRJIO0Note	Input/output	External event input and pulse output for timer RJ
TRJO0Note	Output	Pulse output for timer RJ

Note The assignment of the TRJO0 pin is selected by bits PIOR12 and PIOR13 in the PIOR1 register. The assignment of the TRJIO0 pin is selected by bits PIOR10 and PIOR11 in the PIOR1 register. Refer to **Section 7 Port Functions** for details.

11.3 Registers to Control Timer RJ

The following registers are used to control timer RJ.

- Peripheral enable register 2 (PER2)
- Peripheral reset control register 2 (PRR2)
- Subsystem clock supply mode control register (OSMC)
- Timer RJ counter register 0 (TRJ0)**Note**
- Timer RJ control register 0 (TRJCR0)
- Timer RJ I/O control register 0 (TRJIOC0)
- Timer RJ mode register 0 (TRJMR0)
- Timer RJ event pin select register 0 (TRJISR0)
- Port mode registers xx (PMxx) (xx = 0, 3 to 5)
- Port registers xx (Pxx) (xx = 0, 3 to 5)

Note In the case of access to the TRJ0 register, the CPU does not automatically proceed to processing of the next instruction but enters a state of waiting for the completion access before further CPU processing. This wait state will increase the number of clock cycles for instruction execution by the number of clock cycles of waiting. The number of clock cycles of waiting for access to the TRJ0 register is one for both writing and reading.

11.3.1 Peripheral enable register 2 (PER2)

The PER2 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. If timer RJ is to be used, be sure to set bit 0 (TRJ0EN) to 1. The PER2 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 11 - 2 Format of Peripheral Enable Register 2 (PER2)

Address: F00FCH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER2	FAAEN	MEMEN	TKBEN	TRGEN	TRD0EN	PWMOPEN	TRXEN	TRJ0EN

TRJ0EN	Control of supply of an input clock to timer RJ
0	Stops supply of an input clock. • The SFRs used by timer RJ cannot be written.[Note]
1	Enables supply of an input clock. • The SFRs used by timer RJ can be read and written.

Note That is, the SFRs return 00H or 0000H regardless of their actual settings when read.

Caution **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 the value read is 00H or 0000H (except for port mode registers 0 and 3 to 5 (PM0 and PM3 to PM5) and port registers 0 and 3 to 5 (P0 and P3 to P5)).**

11.3.2 Peripheral reset control register 2 (PRR2)

The PRR2 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. To place timer RJ in the reset state, set bit 0 (TRJ0RES) to 1. The PRR2 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 11 - 3 Format of Peripheral Reset Control Register 2 (PRR2)

Address: F00FDH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PRR2	FAARES	MEMRES	TKBRES	TRGRES	TRD0RES	PWMOPRES	TRXRES	TRJ0RES

TRJ0RES	Control resetting of timer RJ
0	Timer RJ is released from the reset state.
1	Timer RJ is in the reset state. • The SFRs for use with timer RJ are initialized.

11.3.3 Subsystem clock supply mode control register (OSMC)

The operating clock for timer RJ can be selected by the WUTMMCK0 bit of the OSMC register. In addition, by stopping clock functions that are unnecessary, the RTCLPC bit can be used to reduce power consumption. For details about setting the RTCLPC bit, see **Section 9 Clock Generator**. The OSMC register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is undefined.

Figure 11 - 4 Format of Subsystem Clock Supply Mode Control Register (OSMC)

Address: F00F3H
After reset: Undefined**Note 1**
R/W: R/W

Symbol	<7>	6	5	<4>	3	2	1	<0>
OSMC	RTCLPC	0	0	WUTMMCK0	×	×	0	HIPREC

WUTMMCK0	Selection of operation clock for timer RJ
0	Subsystem clock X (f_{SX})
1	Low-speed on-chip oscillator clock (f_{IL})**Notes 2, 3, 4**

Note 1. The RTCLPC and WUTMMCK bits have the value 0 and the HIPREC bit has the value 1 following a reset.

Note 2. Do not set the WUTMMCK0 bit to 1 while the subsystem clock X is oscillating.

Note 3. Switching between the subsystem clock X and the low-speed on-chip oscillator clock by using the WUTMMCK0 bit is only possible while operation of the realtime clock, 32-bit interval timer, clock output/buzzer output controller, and timer RJ are all stopped.

Note 4. When f_{SX} is selected as the count source for the realtime clock or 32-bit interval timer, f_{IL} cannot be selected as the count source for timer RJ.

11.3.4 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 **11.4.1 Reload register and counter rewrite operation**. The TRJ0 register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is FFFFH.

Figure 11 - 5 Format of Timer RJ Counter Register 0 (TRJ0)

Address: F0500H
After reset: FFFFH
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRJ0																

Bits 15 to 0	Function
—	16-bit counter Notes 1, 2

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 the TCK[2:0] bits 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 DTC and the ELC is generated only once immediately after counting 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 the TCK[2:0] bits, a request signal to the DTC and the ELC is generated only once immediately after counting 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 0001H or a higher value, a request signal is generated each time TRJ underflows.

Caution **In the case of access to the TRJ0 register, the CPU does not automatically proceed to processing of the next instruction but enters a state of waiting for the completion access before further CPU processing. This wait state will increase the number of clock cycles for instruction execution by the number of clock cycles of waiting. The number of clock cycles of waiting for access to the TRJ0 register is one for both writing and reading.**

11.3.5 Timer RJ control register 0 (TRJCR0)

The TRJCR0 register starts or stops counting by timer RJ and indicates the state of timer RJ. The TRJCR0 register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 11 - 6 Format of Timer RJ Control Register 0 (TRJCR0) (1/2)

Address: F0240H
After reset: 00H
R/W: 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 judgment 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 counter forced stop**Note 1**
Writing 1 to this bit forcibly stops counting. The read value is 0.	

TCSTF	Timer RJ counter status flag**Note 2**
0	Counting stops.
1	Counting is 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).	

Figure 11 - 6 Format of Timer RJ Control Register 0 (TRJCR0) (2/2)

TSTART	Timer RJ counter start[Note 2]
0	Counting stops.
1	Counting starts.
Writing 1 to the TSTART bit starts counting and writing 0 stops counting. When the TSTART bit is set to 1 (to start counting), the TCSTF flag is set to 1 (to indicate that counting is in progress) in synchronization with the count source. Also, after 0 is written to the TSTART bit, the TCSTF flag is set to 0 (counting is stopped) in synchronization with the count source. For details, see **11.5.1 Control over starting and stopping of the counter**.	

Note 1. Writing 1 to the TSTOP bit (to forcibly stop counting) initializes the TSTART bit and TCSTF flag at the same time. The pulse output level is also initialized.

Note 2. For notes on using the TSTART bit and TCSTF flag, see **11.5.1 Control over starting and stopping of the counter**.

11.3.6 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. The value of this register following a reset is 00H.

Figure 11 - 7 Format of Timer RJ I/O Control Register 0 (TRJIOC0)

Address: F0241H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TRJIOC0	TIOGT1	TIOGT0	TIPF1	TIPF0	0	TOENA	0	TEDGSEL

TIOGT1	TIOGT0	TRJIO counting control[Notes 1, 2, 3]
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	TRJO0 output enable
0	TRJO0 output disabled
1	TRJO0 output enabled

TEDGSEL	I/O polarity switch
Function varies depending on the operating mode (see **Tables 11 - 3** and **11 - 4**).	

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. The TIOGT[1:0] bits are enabled only in event counter mode.

Note 3. INTP4 is only usable for this purpose in the 30- to 64-pin products.

Table 11 - 3 TRJIO I/O Edge and Polarity Switching

Operating Mode	Function
Timer mode	Not used (I/O port)
Pulse output mode	0: Output starts at the high level (initial output level: high) 1: Output starts at the low level (initial output 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 the time from one rising edge to the next rising edge of the pulse for measurement. 1: Measure the time from one falling edge to the next falling edge of the pulse for measurement.

Table 11 - 4 TRJO0 Output Polarity Switching

Operating Mode	Function
All modes	0: Output starts at the low level (initial output level: low) 1: Output starts at the high level (initial output level: high)

11.3.7 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. The value of this register following a reset is 00H.

Figure 11 - 8 Format of Timer RJ Mode Register 0 (TRJMR0)

Address: F0242H
After reset: 00H
R/W: 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 3**
1	0	1	Event input from ELC
1	1	0	f_{SX}
Other than above			Setting prohibited

TEDGPL	TRJIO edge polarity select**Note 4**
0	One edge
1	Both edges

TMOD2	TMOD1	TMOD0	Timer RJ operating mode select**Note 5**
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 the TCK[2:0] bits.

Note 2. Do not switch count sources during counting. Count sources should be switched when both the TSTART bit and TCSTF flag in the TRJCR0 register are set to 0 (counting is stopped).

Note 3. When selecting f_{IL} as the count source, set the WUTMMCK0 bit in the subsystem clock supply mode control register (OSMC) to 1.
However, f_{IL} cannot be selected as the count source for timer RJ when f_{SX} is selected as the count source for the realtime clock or the 32-bit interval timer.

Note 4. The TEDGPL bit is enabled only in event counter mode.

Note 5. The operating mode can be changed only when counting is stopped while both the TSTART bit and TCSTF flag in the TRJCR0 register are set to 0 (counting is stopped). Do not change the operating mode during counting.

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 Tables 11 - 3 and 11 - 4.

11.3.8 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. The value of this register following a reset is 00H.

Figure 11 - 9 Format of Timer RJ Event Pin Select Register 0 (TRJISR0)

Address: F0243H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TRJISR0	0	0	0	0	0	RCCPSEL2	RCCPSEL1	RCCPSEL0

RCCPSEL2Note	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

RCCPSEL1Note	RCCPSEL0Note	Timer output signal selection
0	0	TRDIOD1
0	1	TRDIOC1
1	0	TO02
1	1	TO03

Note Bits RCCPSEL2 and RCCPSEL[1:0] are enabled only in event counter mode.

11.3.9 Registers for controlling the port functions multiplexed with the inputs and outputs of timer RJ

Set the following registers to control the port functions multiplexed with inputs and outputs of timer RJ.

- Port mode registers xx (PMxx)
- Port registers xx (Pxx)

For details, see **7.3.1 Port mode registers xx (PMxx)** and **7.3.2 Port registers xx (Pxx)**.

When the pins multiplexed with TRJO0 and TRJIO are to be used for timer outputs, set the corresponding bits in the given port mode registers xx (PMxx) and port registers xx (Pxx) 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 the pin multiplexed with TRJIO is to be used for timer input, set the corresponding bit in the given port mode register xx (PMxx) to 1. At this time, the bit in the corresponding port register xx (Pxx) 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.

Remark xx = 0, 3 to 5

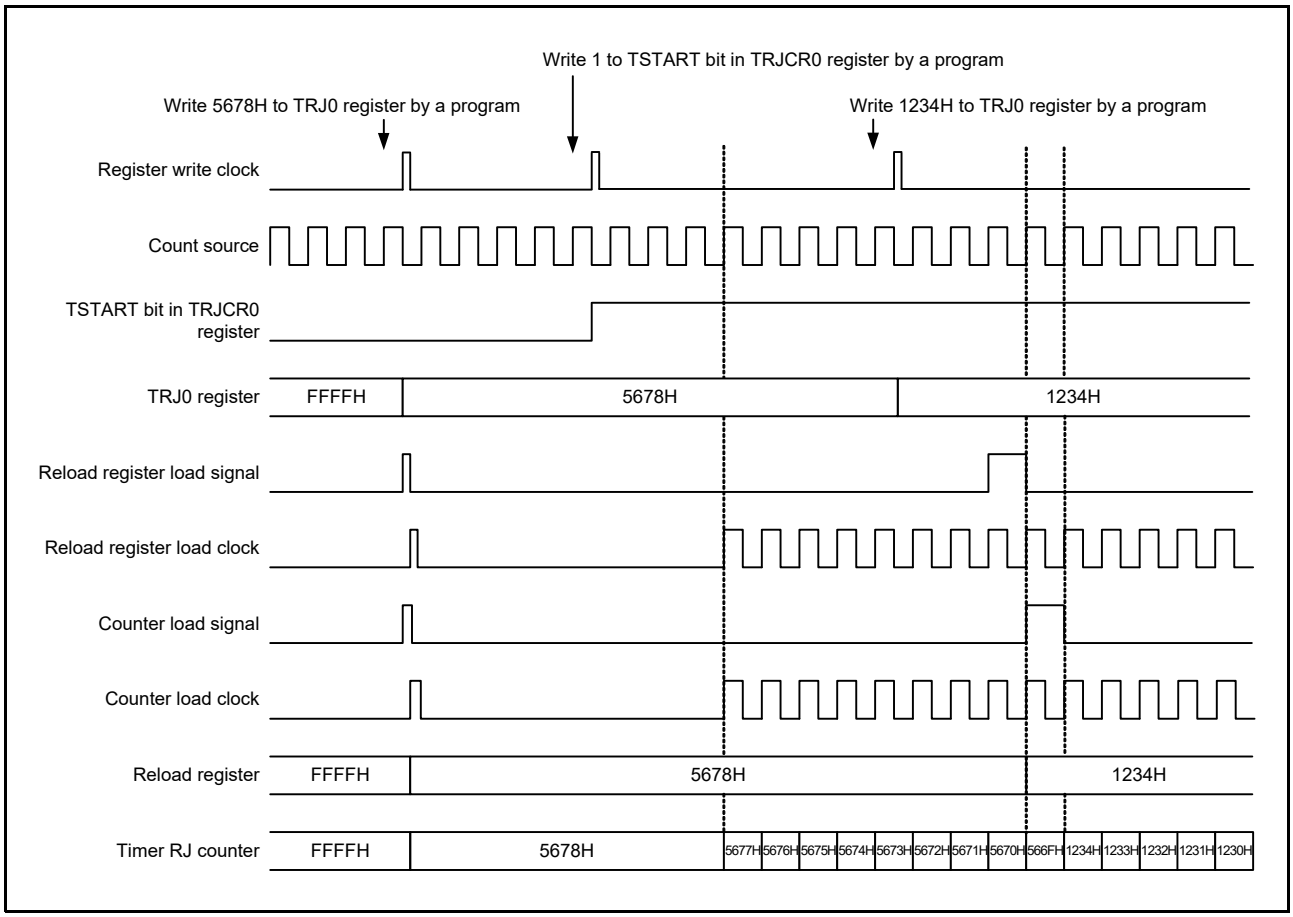
11.4 Timer RJ Operation

11.4.1 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 (counting is stopped), the count value is directly written to the reload register and the counter. When the TSTART bit is 1 (to start counting), 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 11 - 10 shows the timing of rewrite operation with TSTART bit value.

Figure 11 - 10 Timing of Rewrite Operation with TSTART Bit Value

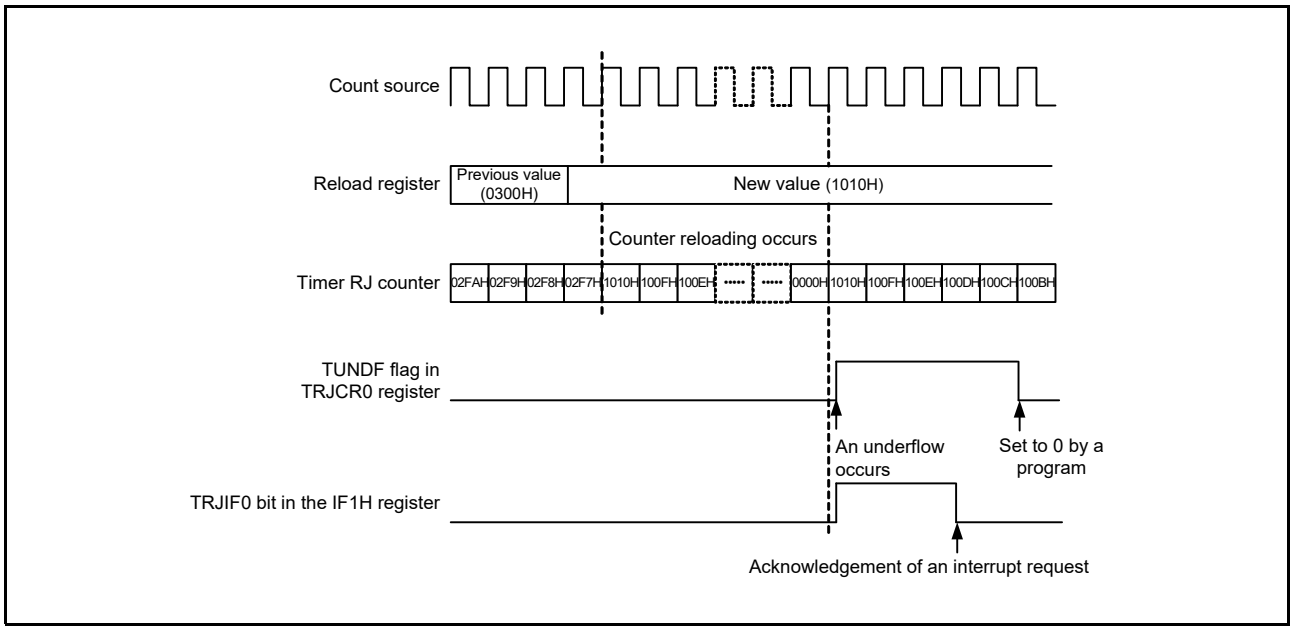


11.4.2 Timer mode

In this mode, the counter is decremented by the count source selected by the TCK[2:0] bits 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 11 - 11 shows the operation example in timer mode.

Figure 11 - 11 Operation Example in Timer Mode



11.4.3 Pulse output mode

In this mode, the counter is decremented by the count source selected by the TCK[2:0] bits in the TRJMR0 register, and the output level of pins TRJIO and TRJO pin is inverted each time an underflow occurs.

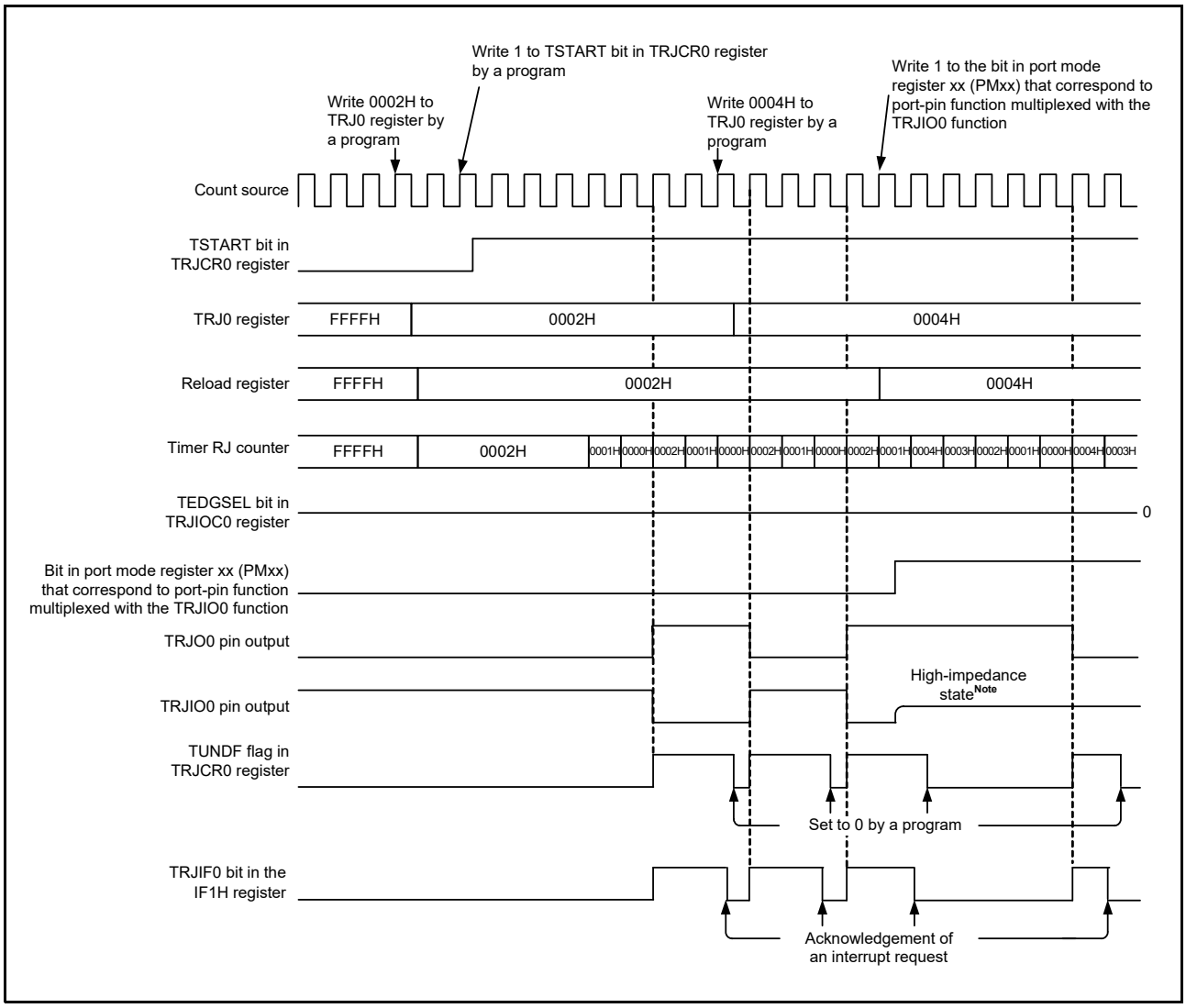
In pulse output mode, the count value is decremented by 1 each time the count source is input. When the count value reaches 0000H and the next count source is input, an underflow occurs and an interrupt request is generated.

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 initial output level can be selected by the TEDGSEL bit in the TRJIOC0 register.

Figure 11 - 12 shows the operation example in pulse output mode.

Figure 11 - 12 Operation Example in Pulse Output Mode



Note The TRJIO0 pin becomes high impedance by output enable control on the port selected as the TRJIO0 function.

11.4.4 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 the TIOGT[1:0] bits in the TRJIOC0 register and the TRJISR0 register. In addition, the filter function for the TRJIO0 input can be specified by the TIPF[1:0] bits 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 **11.5.5 Procedure for setting pins TRJO0 and TRJIO0**.
Figure 11 - 13 shows the operation example 1 in event counter mode.

Figure 11 - 13 Operation Example 1 in Event Counter Mode

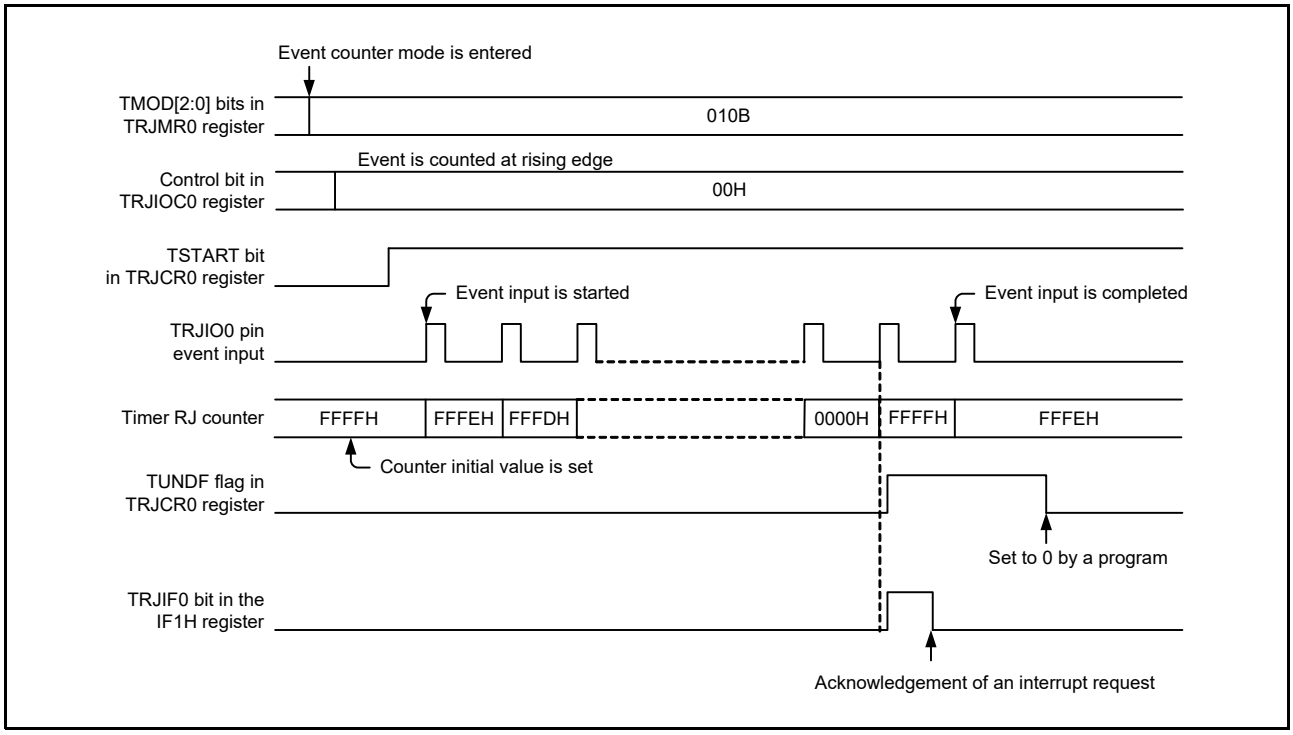
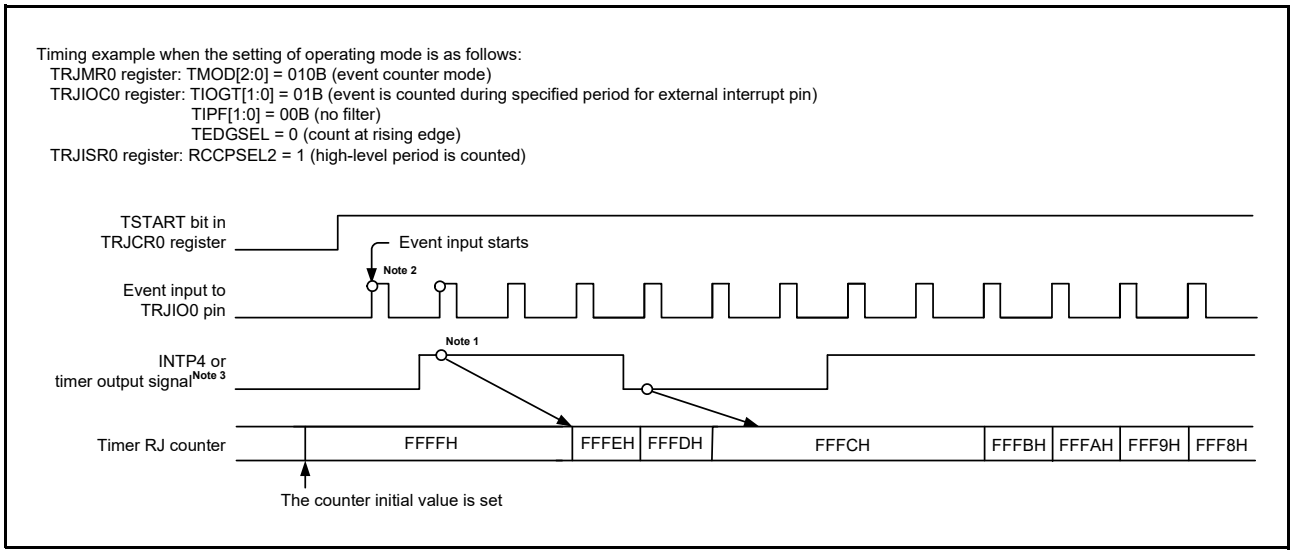


Figure 11 - 14 shows an operation example for counting during the specified period in event counter mode (bits TIOGT[1:0] in the TRJIOC0 register are set to 01B or 10B).

Figure 11 - 14 Operation Example 2 in Event Counter Mode



The following notes apply only when the TIOGT[1:0] bits in the TRJIOC0 register are 01B or 10B for the setting of operating mode in event counter mode.

Note 1. To control synchronization, there is a delay of two cycles of the count source until count operation is affected.

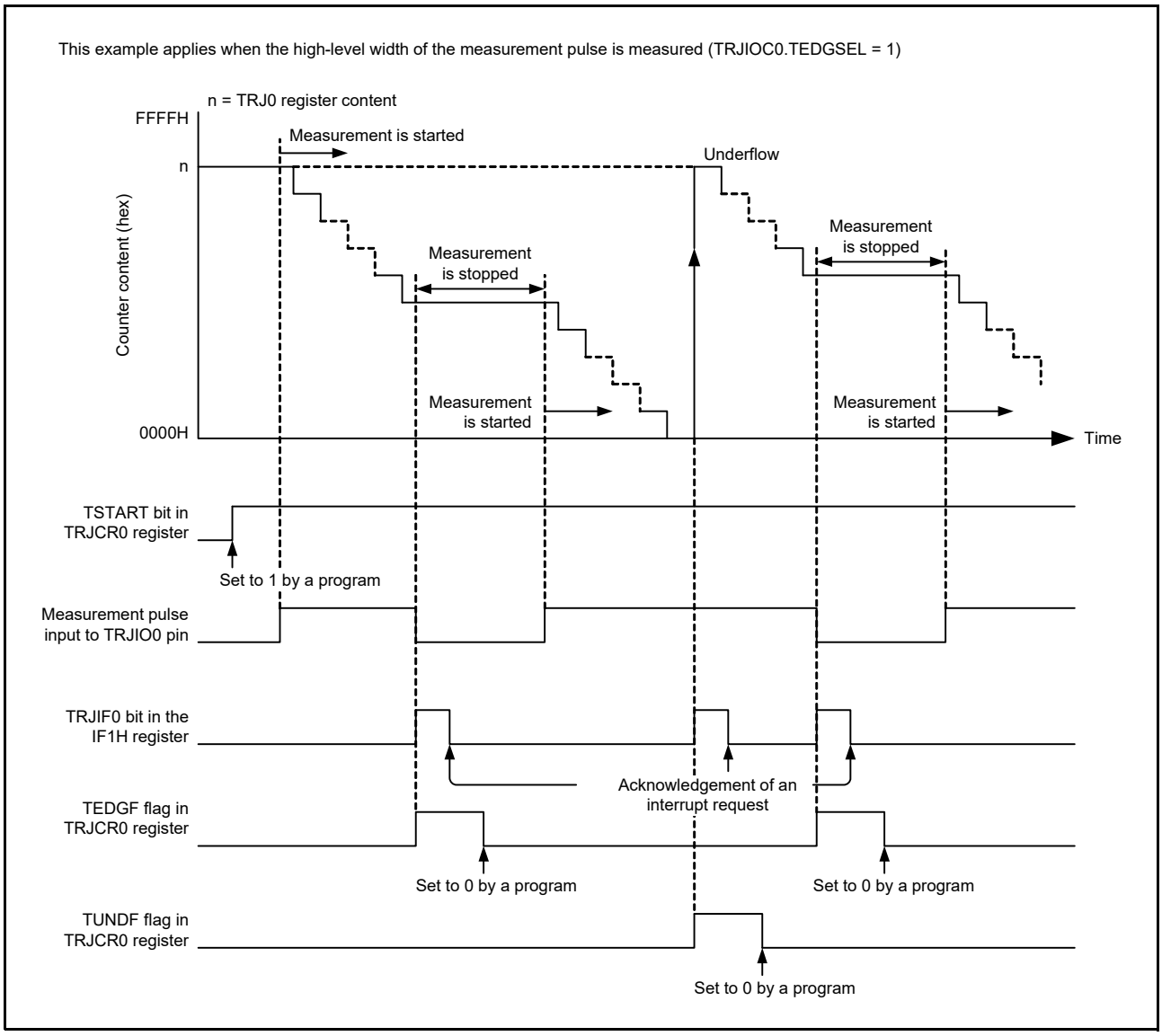
Note 2. Counting of two cycles of the source for counting may proceed immediately after counting is started, depending on the state before counting was previous stopped. To disable this counting of two cycles immediately after counting is started, write 1 to the TSTOP bit in the TRJCR0 register to initialize the internal circuit, and then make the settings for operation before starting the counter.

Note 3. For the timer output signal selected by the RCCPSEL[1:0] 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.

11.4.5 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 flag 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 flag in the TRJCR0 register is set to 1 (underflow) and an interrupt request is generated.
Figure 11 - 15 shows the operation example in pulse width measurement mode.
When accessing the TEDGF and TUNDF flags in the TRJCR0 register, see **11.5.2 Access to flags (flags TEDGF and TUNDF in TRJCR0 register)**.

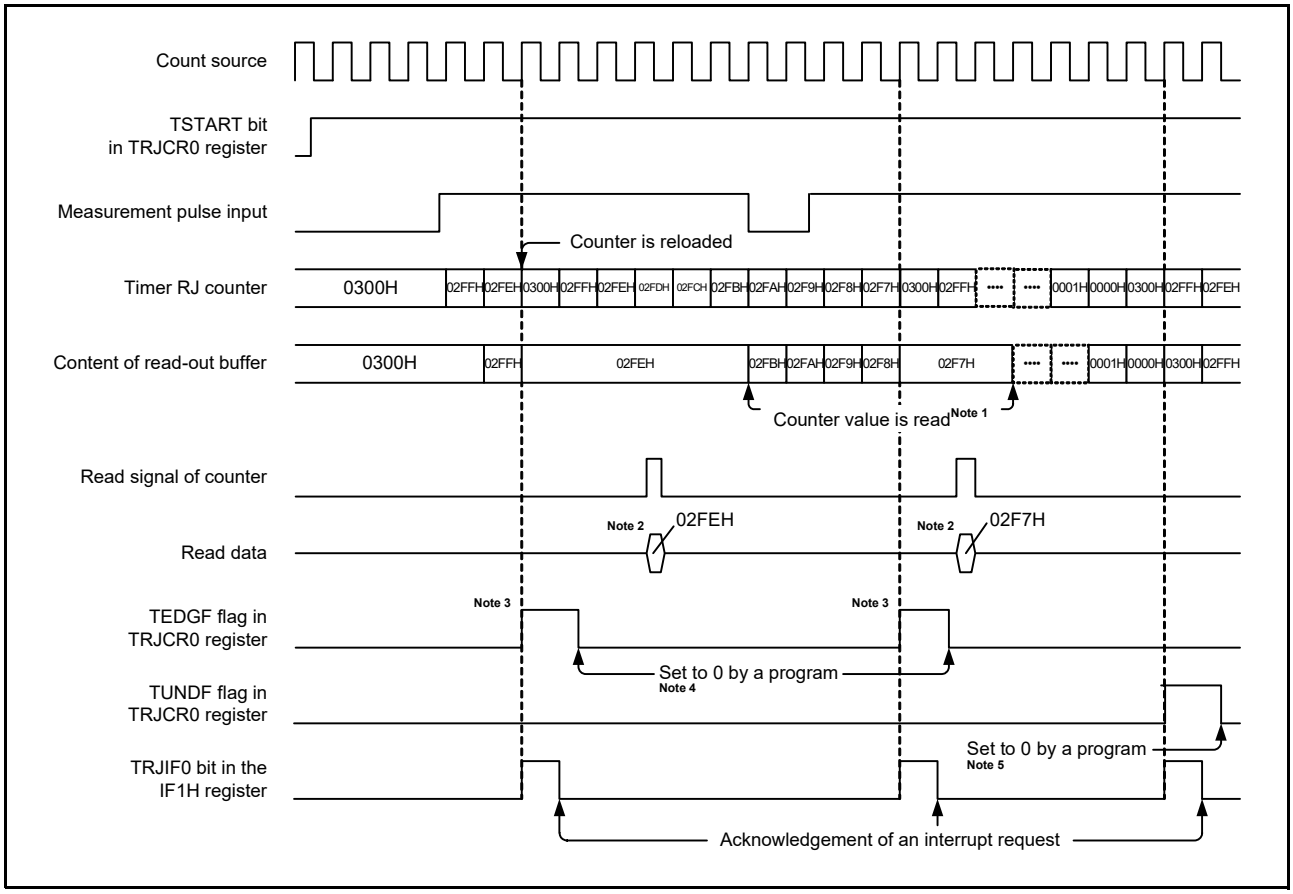
Figure 11 - 15 Operation Example in Pulse Width Measurement Mode



11.4.6 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 the TCK[2:0] bits 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 flag 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 flag in the TRJCR0 register is set to 1 (underflow) and an interrupt request is generated.
Figure 11 - 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 11 - 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 flag 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 flag 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 flag 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 flag in the TRJCR0 register using an 8-bit memory manipulation instruction.

11.4.7 Coordination with event link controller (ELC)

Through coordination with the ELC, event input from the ELC can be set to be the count source. Counting is of rising edges of the event input from the ELC when this is selected as the source for counting by the TCK[2:0] bits of the TRJMR0 register. However, ELC input does not function in event counter mode.

The ELC setting procedure is shown below:

- Procedure for starting operation
  - <1> Set event output destination select register n (ELSELRn) for the ELC.
  - <2> Set the operating mode for the event generation source.
  - <3> Set the mode for timer RJ.
  - <4> Start counting by 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 counting by timer RJ.
  - <3> Set event output destination select register n (ELSELRn) for the ELC to 0.

11.4.8 Output settings for each mode

Tables 11 - 5 and **11 - 6** list the settings of pins TRJO0 and TRJIO0 in each mode.

Table 11 - 5 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 11 - 6 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 This refers to the bits in port mode registers xx (PMxx) that correspond to port-pin functions multiplexed with the TRJIO0 function.

11.5 Cautions for Timer RJ

11.5.1 Control over starting and stopping of the counter

- When the ELC is selected as the source for counting in the event counter mode or any other desired mode.
  After 1 is written to the TSTART bit in the TRJCR0 register (to start counting) while counting is stopped, the TCSTF flag in the TRJCR0 register remains 0 (counting is stopped) for two cycles of the CPU clock. Do not access the registers associated with timer RJ**Note** other than the TCSTF flag until this flag is set to 1 (counting is in progress).
  After 0 is written to the TSTART bit (to stop counting) during counting, the TCSTF flag remains 1 for two cycles of the CPU clock. When the setting of the TCSTF flag becomes 0, counting has stopped. Do not access the registers associated with timer RJ**Note** other than the TCSTF flag until this flag is set to 0.
  Clear the interrupt register before changing the TSTART bit from 0 to 1. Refer to **Section 29 Interrupt Functions** for details.
  Note Registers associated with timer RJ: TRJ0, TRJCR0, TRJIOC0, TRJMR0, and TRJISR0

- When a setting other than that described above is made.
  After 1 is written to the TSTART bit in the TRJCR0 register (to start counting) while counting is stopped, the TCSTF flag in the TRJCR0 register remains 0 (counting is stopped) for three cycles of the count source. Do not access the registers associated with timer RJ**Note** other than the TCSTF flag until this flag is set to 1 (counting is in progress).
  After 0 is written to the TSTART bit (to stop counting) during counting, the TCSTF flag remains 1 for three cycles of the count source. When the setting of the TCSTF flag becomes 0, counting has stopped. Do not access the registers associated with timer RJ**Note** other than the TCSTF flag until this flag is set to 0.
  Clear the interrupt register before changing the TSTART bit from 0 to 1. Refer to **Section 29 Interrupt Functions** for details.
  Note Registers associated with timer RJ: TRJ0, TRJCR0, TRJIOC0, TRJMR0, and TRJISR0

11.5.2 Access to flags (flags TEDGF and TUNDF in TRJCR0 register)

Flags 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, flags TEDGF and TUNDF may be erroneously set to 0 depending on the timing, even when the TEDGF flag is set to 1 (active edge received) and the TUNDF flag is set to 1 (underflow) during execution of the instruction. Use an 8-bit memory manipulation instruction to access to the TRJCR0 register.

11.5.3 Access to counter register

When the TSTART bit in the TRJCR0 register is 1 (to have started counting) or the TCSTF flag in the TRJCR0 register is 1 (counting is in progress), allow at least three cycles of the source clock for counting between write operations in the case of successive writing to the TRJ0 register.

11.5.4 When changing mode

The registers associated with timer RJ operating mode (TRJIOC0, TRJMR0, and TRJISR0) can be changed only when counting is stopped with both the TSTART bit and TCSTF flag set to 0 (counting is stopped). Do not change these registers during counting.
When the registers associated with timer RJ operating mode are changed, the values of the TEDGF and TUNDF flags are undefined. Write 0 (no active edge received) to the TEDGF flag and 0 (no underflow) to the TUNDF flag before starting the counter.

11.5.5 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:
<1> Set the mode.
<2> Set the initial value/output enabled.
<3> Set the bits in port registers xx that correspond to the TRJO0 and TRJIO0 pin functions to 0.
<4> Set the bits in port mode registers xx that correspond to the TRJO0 and TRJIO0 pin functions to output mode. (Output is started from pins TRJO0 and TRJIO0)
<5> Start the counter (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 bits in port mode registers xx that correspond to the TRJIO0 pin function to input mode. (Input is started from the TRJIO0 pin)
<4> Start the counter (TSTART in TRJMR0 register = 1).
<5> Wait until the TCSTF flag in the TRJCR0 register is set to 1 (counting is 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)

11.5.6 When timer RJ is not used

When timer RJ is not used, set the TMOD[2:0] bits in the TRJMR0 register to 000B (timer mode) and set the TOENA bit in the TRJIOC0 register to 0 (TRJO output disabled).

11.5.7 When timer RJ operating clock is stopped

Supplying or stopping the timer RJ clock can be controlled by the TRJ0EN bit in the PER2 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.

11.5.8 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.

<1> Set the operating mode.
<2> Start the counter (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 counter (TSTART = 0, TCSTF = 0)

11.5.9 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.

11.5.10 When counting 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

11.5.11 Digital filter

When the digital filter is used, do not start timer operation for five cycles of the digital filter clock after setting the TIPF[1:0] bits.
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.

11.5.12 When selecting f_{IL} as count source

When selecting f_{IL} as the count source, set the WUTMMCK0 bit in the subsystem clock supply mode control register (OSMC) to 1. However, f_{IL} cannot be selected as the count source for timer RJ when f_{SX} is selected as the count source for the realtime clock or the 32-bit interval timer.

Section 12 Timer RD2

12.1 Functions of Timer RD2

Timer RD2 consists of two counters: timer RD2 counter 0 (timer RD20) and timer RD2 counter 1 (timer RD21), and has seven modes:

- Timer mode
  - Input capture function: Transfer the counter value to a register triggered by an external signal
  - 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 six modes use PWM functions.

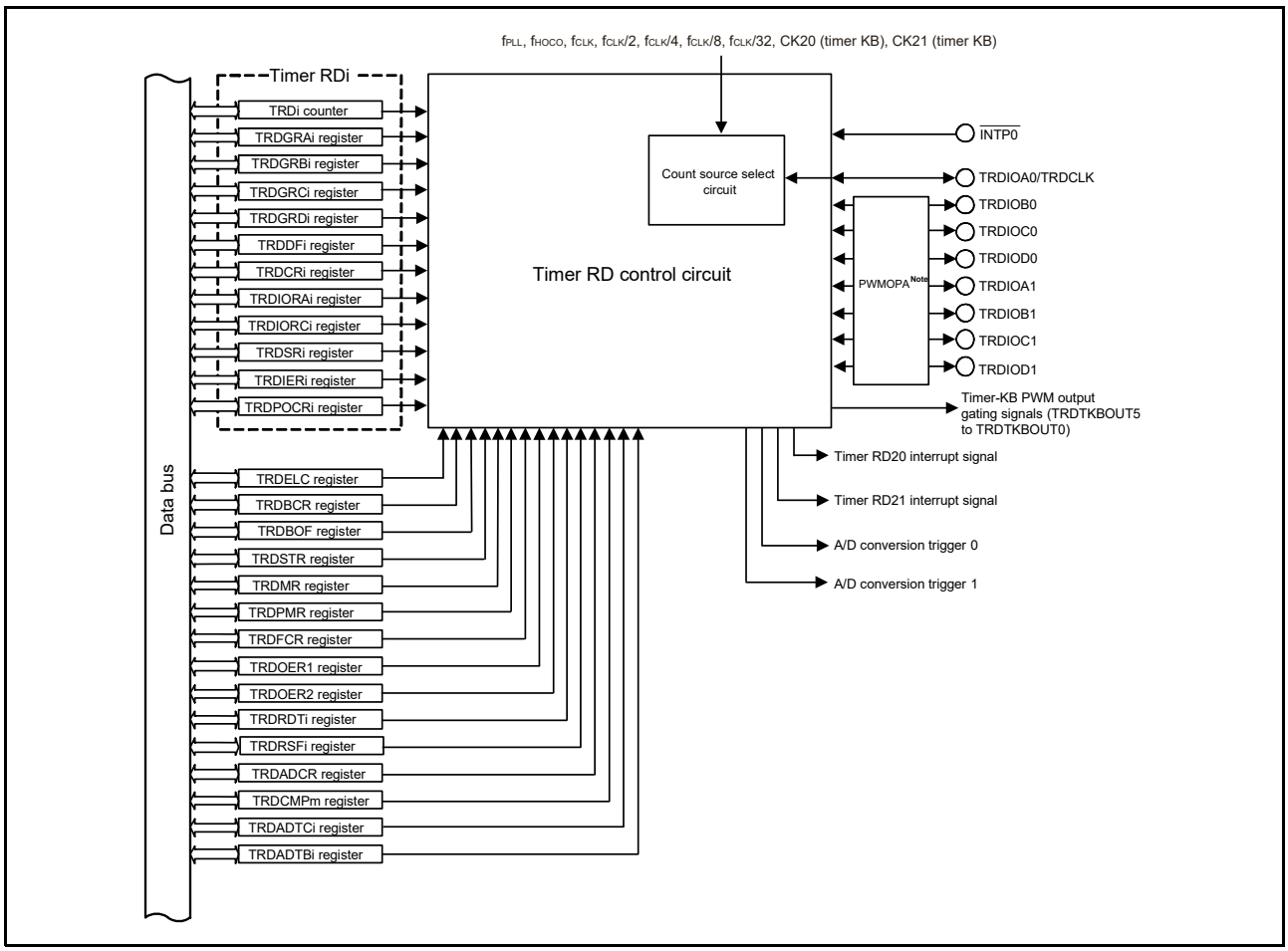
- Reset synchronous PWM mode: Output three-phase waveforms (six waveforms) with sawtooth wave modulation and without dead time.
- Complementary PWM mode: Output three-phase waveforms (six waveforms) with triangular wave modulation and dead time.
- PWM3 mode: Output PWM waveforms (two waveforms) with a fixed period.
- Extended PWM mode: Enable simultaneous update of compare registers and output up to four PWM waveforms (two waveforms per counter) with the same period.
- Extended complementary PWM mode: Output three-phase symmetric or asymmetric waveforms (six waveforms) with triangular wave modulation and dead time.
- Timer-KB PWM output gating mode: Output PWM waveforms for gating the output from 16-bit timers KB30, KB31, and KB32.

The input capture function, output compare function, and PWM function in the timer mode and extended PWM mode are equivalent between timer RD20 and timer RD21, and these functions can be selected individually for each pin. Also, a combination of these functions can be used in timer RD20 and timer RD21.
In reset synchronous PWM mode, complementary PWM mode, PWM3 mode, extended complementary PWM mode, and timer-KB PWM output gating mode, waveforms are output by using a combination of counters and registers in timer RD20 and timer RD21. Pin functions depend on the mode.
Timer RD2 has four I/O pins.
The operating clock for timer RD2 is f_{CLK}, f_{HOCO}, or f_{PLL}.
Timer RD2 operates in coordination with timer RX, timer KB30, timer KB31, and timer KB32.

12.2 Configuration of Timer RD2

Figure 12 - 1 shows a block diagram of timer RD2 and **Table 12 - 1** lists the input/output pins of timer RD2. For the PWMOPA functions, see **12.8 PWM Option Unit A (PWMOPA)**.

<R> Figure 12 - 1 Block Diagram of Timer RD2



Note Output signals can be cut off, while input signals cannot.

Remark i = 0 or 1
m = B0, D0, A1, B1, C1, D1

Table 12 - 1 Configuration of Timer RD2 Pins

Pin Name	Alternate Port Name	I/O	Function
TRDIOA0/TRDCLK	P17	Input/Output	Function varies depending on the mode. Refer to the 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	

12.3 Registers for Controlling Timer RD2

The following registers are used to control timer RD2.

- Peripheral enable register 2 (PER2)
- Peripheral reset control register 2 (PRR2)
- Timer RD ELC register (TRDELC)
- Timer RD timer-KB PWM output gating mode control register (TRDBCR)
- Timer RD timer KB PWM output monitor register (TRDBOF)
- Timer RD start register (TRDSTR)
- Timer RD mode register (TRDMR)
- Timer RD PWM function select register (TRDPMR)
- Timer RD function control register (TRDFCR)
- Timer RD output master enable register 1 (TRDOER1)
- Timer RD output master enable register 2 (TRDOER2)
- Timer RD output control register (TRDOCR)
- Timer RD digital filter function select registers 0 and 1 (TRDDF0 and TRDDF1)
- Timer RD control registers 0 and 1 (TRDCR0 and TRDCR1)
- Timer RD I/O control registers A0 and A1 (TRDIORA0 and TRDIORA1)
- Timer RD I/O control registers C0 and C1 (TRDIORC0 and TRDIORC1)
- Timer RD status register 0 (TRDSR0)
- Timer RD status register 1 (TRDSR1)
- Timer RD interrupt enable registers 0 and 1 (TRDIER0 and TRDIER1)
- Timer RD PWM output level control register 0 (TRDPOCR0)
- Timer RD PWM output level control register 1 (TRDPOCR1)
- Timer RD counters 0 and 1 (TRD0 and TRD1)
- Timer RD general registers A0, A1, B0, B1, C0, C1, D0, and D1 (TRDGRA0, TRDGRA1, TRDGRB0, TRDGRB1, TRDGRC0, TRDGRC1, TRDGRD0, and TRDGRD1)
- Timer RD extended compare registers B0, D0, A1, B1, C1, and D1 (TRDCMPB0, TRDCMPD0, TRDCMPA1, TRDCMPB1, TRDCMPC1, and TRDCMPD1)
- Timer RD A/D conversion trigger compare register 0/timer-KB PWM output gating mode compare register (TRDADTC0/TRDCMPE1)
- Timer RD A/D conversion trigger buffer register 0/timer-KB PWM output gating mode buffer register (TRDADTB0/ TRDCMPF1)
- Timer RD A/D conversion trigger compare register 1 (TRDADTC1)
- Timer RD A/D conversion trigger buffer register 1 (TRDADTB1)
- Timer RD simultaneous update trigger register 0 (TRDRDT0)
- Timer RD simultaneous update trigger register 1 (TRDRDT1)
- Timer RD simultaneous update flag register 0 (TRDRSF0)
- Timer RD simultaneous update flag register 1 (TRDRSF1)
- Timer RD A/D conversion trigger control register (TRDADCR)
- Timer RD skipping control register (TRDTCTL)
- Timer RD skipping count setting register (TRDTCMP)
- Timer RD output port mask enable register (TRDPOE)
- Port mode registers xx (PMxx) (xx = 1, 3, 5, 7)
- Port registers xx (Pxx) (xx = 1, 3, 5, 7)
- Port mode control A registers xx (PMCAxx) (xx = 1, 3, 5, 7)

12.3.1 Peripheral enable register 2 (PER2)

The PER2 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. Be sure to set bit 3 (TRD0EN) of this register to 1 to use timer RD2. The PER2 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 12 - 2 Format of Peripheral Enable Register 2 (PER2)

Address: F00FCH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER2	FAAEN	MEMEN	TKBEN	TRGEN	TRD0EN	PWMOPEN	TRXEN	TRJ0EN

TRD0EN	Control of timer RD2 input clock supply
0	Stops supply of an input clock. • The SFRs used by timer RD2 cannot be written.
1	Enables supply of an input clock. • The SFRs used by timer RD2 can be read and written.

PWMOPEN	Control of PWMOPA input clock supply
0	Stops supply of an input clock. • The SFRs used by PWMOPA cannot be written.
1	Enables supply of an input clock. • The SFRs used by PWMOPA can be read and written.

Caution 1. When setting timer RD2, be sure to set the TRD0EN bit to 1 first. If TRD0EN = 0, writing to a control register of timer RD2 is ignored, and the read value is 00H or 0000H (except for port mode register 1 (PM1) and port register 1 (P1)).

Caution 2. When selecting f_{HOCO} as the count source for timer RD2, set f_{CLK} to f_{IH} before setting bit 3 (TRD0EN) in peripheral enable register 2 (PER2). When changing f_{CLK} to a clock other than f_{IH}, clear bit 3 (TRD0EN) in peripheral enable register 2 (PER2) before changing.

Caution 3. When setting PWMOPA, be sure to set the PWMOPEN bit to 1 first. If PWMOPEN = 0, writing to a control register of PWMOPA is ignored, and read value is 00H or 0000H. For PWMOPA, see 12.8 PWM Option Unit A (PWMOPA).

12.3.2 Peripheral reset control register 2 (PRR2)

The PRR2 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. Set bit 3 (TRD0RES) of this register to 1 to place timer RD2 in the reset state. The PRR2 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 12 - 3 Format of Peripheral Reset Control Register 2 (PRR2)

Address: F00FDH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PRR2	FAARES	MEMRES	TKBRES	TRGRES	TRD0RES	PWMOP RES	TRXRES	TRJ0RES

TRD0RES	Control resetting of timer RD2
0	Timer RD2 is released from the reset state.
1	Timer RD2 is in the reset state. • The SFRs used by timer RD2 are initialized.

PWMOP RES	Control resetting of PWMOPA
0	PWMOPA is released from the reset state.
1	PWMOPA is in the reset state. • The SFRs used by PWMOPA are initialized.

12.3.3 Timer RD ELC register (TRDELC)

Figure 12 - 4 Format of Timer RD ELC Register (TRDELC)

Address: F0390H
After reset: 00H[Note]
R/W: 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 RD2 pulse output forced cutoff
0	Forced cutoff is disabled
1	Forced cutoff is enabled

ELCICE1	ELC event input 1 select for timer RD2 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 RD2 pulse output forced cutoff
0	Forced cutoff is disabled
1	Forced cutoff is enabled

ELCICE0	ELC event input 0 select for timer RD2 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 the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

12.3.4 Timer RD timer-KB PWM output gating mode control register (TRDBCR)

Figure 12 - 5 Format of Timer RD Timer-KB PWM Output Gating Mode Control Register (TRDBCR)

Address: F0391H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TRDBCR	GCE	0	0	GCKS	0	0	0	GMD

GCE	Enabling timer-KB PWM output gating mode
0	Timer-KB PWM output gating mode is disabled.
1	Timer-KB PWM output gating mode is enabled.

GCKS	Selecting the count source for timer-KB PWM output gating mode
<R> 0	The CK20 operating clock for timer KB3 is selected.
<R> 1	The CK21 operating clock for timer KB3 is selected.

GMD	Selecting the operation in timer-KB PWM output gating mode
0	Standalone mode
1	Timer KB3 interlocked mode

12.3.5 Timer RD timer KB PWM output monitor register (TRDBOF)

Figure 12 - 6 Format of Timer RD Timer KB PWM Output Monitor Register (TRDBOF)

Address: F0392H
After reset: 00H
R/W: R

Symbol	7	6	5	4	3	2	1	0
TRDBOF	0	0	BOF5	BOF4	BOF3	BOF2	BOF1	BOF0

BOF5	Timer KB PWM output monitor flag 5
0	A low level is output through TRDTKBOUT5.
1	A high level is output through TRDTKBOUT5.

BOF4	Timer KB PWM output monitor flag 4
0	A low level is output through TRDTKBOUT4.
1	A high level is output through TRDTKBOUT4.

BOF3	Timer KB PWM output monitor flag 3
0	A low level is output through TRDTKBOUT3.
1	A high level is output through TRDTKBOUT3.

BOF2	Timer KB PWM output monitor flag 2
0	A low level is output through TRDTKBOUT2.
1	A high level is output through TRDTKBOUT2.

BOF1	Timer KB PWM output monitor flag 1
0	A low level is output through TRDTKBOUT1.
1	A high level is output through TRDTKBOUT1.

BOF0	Timer KB PWM output monitor flag 0
0	A low level is output through TRDTKBOUT0.
1	A high level is output through TRDTKBOUT0.

12.3.6 Timer RD start register (TRDSTR)

The TRDSTR register can be set by an 8-bit memory manipulation instruction. See **12.7.1 SFR read/write access**, **1. TRDSTR register**.

Figure 12 - 7 Format of Timer RD Start Register (TRDSTR)

Address: F0393H
After reset: 0CH**Note 1**
R/W: 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 the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. 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).

12.3.7 Timer RD mode register (TRDMR)

Figure 12 - 8 Format of Timer RD Mode Register (TRDMR)

Address: F0394H
After reset: 00H[Note 1]
R/W: 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[Notes 2, 3]
0	General register
1	Buffer register for TRDGRB1 register

TRDBFC1	TRDGRC1 register function select[Notes 2, 3]
0	General register
1	Buffer register for TRDGRA1 register

TRDBFD0	TRDGRD0 register function select[Notes 2, 3]
0	General register
1	Buffer register for TRDGRB0 register

TRDBFC0	TRDGRC0 register function select[Notes 2, 3, 4]
0	General register
1	Buffer register for TRDGRA0 register

TRDSYNC	Timer RD2 synchronization[Note 5]
0	TRD0 and TRD1 operate independently
1	TRD0 and TRD1 operate synchronously

Note 1. The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. 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. This bit is not used in extended PWM mode, extended complementary PWM mode, or timer-KB PWM output gating mode.

Note 4. Set to 0 (general register) in complementary PWM mode.

Note 5. Set to 0 (TRD0 and TRD1 operate independently) in reset synchronous PWM mode, complementary PWM mode, PWM3 mode, and extended complementary PWM mode.

12.3.8 Timer RD PWM function select register (TRDPMR)

Figure 12 - 9 Format of Timer RD PWM Function Select Register (TRDPMR) [Timer Mode]

Address: F0395H
After reset: 00H[Note 1]
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
TRDPMR	TRDPWMA1	TRDPWMD1	TRDPWMC1	TRDPWMB1	0	TRDPWMD0	TRDPWMC0	TRDPWMB0

TRDPWMA1[Note 2]	This bit is not used in timer mode. Set to 0.

TRDPWMD1	Selecting PWM function of TRDIOD1
0	Input capture function or output compare function
1	PWM function

TRDPWMC1	Selecting PWM function of TRDIOC1
0	Input capture function or output compare function
1	PWM function

TRDPWMB1	Selecting PWM function of TRDIOB1
0	Input capture function or output compare function
1	PWM function

TRDPWMD0	Selecting PWM function of TRDIOD0
0	Input capture function or output compare function
1	PWM function

TRDPWMC0	Selecting PWM function of TRDIOC0
0	Input capture function or output compare function
1	PWM function

TRDPWMB0	Selecting PWM function of TRDIOB0
0	Input capture function or output compare function
1	PWM function

Note 1. The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. Set this bit to 0 in timer mode.

Figure 12 - 10 Format of Timer RD PWM Function Select Register (TRDPMR) [Extended PWM Mode]

Address: F0395H
After reset: 00H[Note 1]
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
TRDPMR	TRDPWMA1	TRDPWMD1	TRDPWMC1	TRDPWMB1	0	TRDPWMD0	TRDPWMC0	TRDPWMB0

TRDPWMA1[Note 2]	This bit is not used in extended PWM mode. Set to 0.

TRDPWMD1	Selecting PWM function of TRDIOD1
0	Input capture function or output compare function
1	Extended PWM function

TRDPWMC1	This bit is not used in extended PWM mode. Set to 0.

TRDPWMB1	Selecting PWM function of TRDIOB1
0	Input capture function or output compare function
1	Extended PWM function

TRDPWMD0	Selecting PWM function of TRDIOD0
0	Input capture function or output compare function
1	Extended PWM function

TRDPWMC0	This bit is not used in extended PWM mode. Set to 0.

TRDPWMB0	Selecting PWM function of TRDIOB0
0	Input capture function or output compare function
1	Extended PWM function

Note 1. The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. Set this bit to 0 in extended PWM mode.

Figure 12 - 11 Format of Timer RD PWM Function Select Register (TRDPMR) [Timer-KB PWM Output Gating Mode]

Address: F0395H
After reset: 00H[Note]
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
TRDPMR	TRDPWMA1	TRDPWMD1	TRDPWMC1	TRDPWMB1	0	TRDPWMD0	TRDPWMC0	TRDPWMB0

TRDPWMA1	Enabling TRDTKBOUT3 output
0	TRDTKBOUT3 output is disabled.
1	TRDTKBOUT3 output is enabled.

TRDPWMD1	Enabling TRDTKBOUT2 output
0	TRDTKBOUT2 output is disabled.
1	TRDTKBOUT2 output is enabled.

TRDPWMC1	Enabling TRDTKBOUT1 output
0	TRDTKBOUT1 output is disabled.
1	TRDTKBOUT1 output is enabled.

TRDPWMB1	Enabling TRDTKBOUT0 output
0	TRDTKBOUT0 output is disabled.
1	TRDTKBOUT0 output is enabled.

TRDPWMD0	Enabling TRDTKBOUT5 output
0	TRDTKBOUT5 output is disabled.
1	TRDTKBOUT5 output is enabled.

TRDPWMC0	This bit is not used in timer-KB PWM output gating mode. Set to 0.

TRDPWMB0	Enabling TRDTKBOUT4 output
0	TRDTKBOUT4 output is disabled.
1	TRDTKBOUT4 output is enabled.

Note The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Caution When the output of a gating signal (TRDTKBOUT) is disabled through a bit in this register in timer-KB PWM output gating mode, the initial output level selected by the TRDOCR register is output through the corresponding external pin (TRDIO).

12.3.9 Timer RD function control register (TRDFCR)

Figure 12 - 12 Format of Timer RD Function Control Register (TRDFCR) (1/2)

Address: F0396H
After reset: 80H[Note 1]
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TRDFCR	PWM3	STCLK	EPWM	CPSS	OLS1	OLS0	CMD1	CMD0

PWM3	PWM3 mode select[Note 2]
• In timer mode or extended PWM mode, set to 1 (other than PWM3 mode). • In PWM3 mode, set to 0 (PWM3 mode). • The setting of this bit is invalid in reset synchronous PWM mode, complementary PWM mode, extended complementary PWM mode, or timer-KB PWM output gating mode.	

STCLK	External clock input select
• In timer mode, reset synchronous PWM mode, complementary PWM mode, extended PWM mode, or extended complementary PWM mode: 0: External clock input disabled 1: External clock input enabled • In PWM3 mode or timer-KB PWM output gating mode, set to 0 (external clock input disabled).	

EPWM	Extended mode select
• In extended PWM mode or extended complementary PWM mode, set to 1. • In the other modes, set to 0.	

CPSS	Operation select in extended complementary PWM mode
• To output an asymmetric triangular PWM waveform, set to 1. • To output a symmetric triangular PWM waveform, set to 0.	

OLS1	Counter-phase output level select (in reset synchronous PWM mode, complementary PWM mode, or extended complementary PWM mode)[Note 3]
• In reset synchronous PWM mode or complementary PWM mode: 0: High initial output and low active level 1: Low initial output and high active level • In extended complementary PWM mode: 0: Low active level 1: High active level • The setting of this bit is invalid in timer mode, PWM3 mode, extended PWM mode, or timer-KB PWM output gating mode.	

OLS0	Normal-phase output level select (in reset synchronous PWM mode, complementary PWM mode, or extended complementary PWM mode)[Note 3]
• In reset synchronous PWM mode or complementary PWM mode: 0: High initial output and low active level 1: Low initial output and high active level • In extended complementary PWM mode: 0: Low active level 1: High active level • The setting of this bit is invalid in timer mode, PWM3 mode, extended PWM mode, or timer-KB PWM output gating mode.	

Figure 12 - 12 Format of Timer RD Function Control Register (TRDFCR) (2/2)

CMD1	CMD0	Combination mode select[Notes 4, 5]
• In timer mode, PWM3 mode, extended PWM mode, or timer-KB PWM output gating mode, set to 00B (timer mode, PWM3 mode, extended PWM mode, or timer-KB PWM output gating mode). • In extended complementary PWM mode, set to 10B (extended complementary PWM 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 the TRD1 counter underflows) 1 0: Complementary PWM mode (transfer from the buffer register to the general register at compare match between the TRD0 counter and TRDGRA0 register) Other than the above: Setting prohibited.		

Note 1. The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. 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, PWM3 mode, extended PWM mode, or timer-KB PWM output gating mode), the setting of the PWM3 bit is valid.

Note 3. Use the TRDOCR register to select the initial output level in extended complementary PWM mode.

Note 4. Set bits CMD1 and CMD0 while the TSTART1 and TSTART0 bits in the TRDSTR register are 00B (count stops).

Note 5. When bits CMD1 and CMD0 are set to 01B, 10B, or 11B, timer RD2 enters reset synchronous PWM mode, complementary PWM mode, or extended complementary PWM mode regardless of the settings of the TRDPMR register.

12.3.10 Timer RD output master enable register 1 (TRDOER1)

Figure 12 - 13 Format of Timer RD Output Master Enable Register 1 (TRDOER1) [Output Compare Function, PWM Function, Reset Synchronous PWM Mode, Complementary PWM Mode, PWM3 Mode, Extended PWM Mode, Extended Complementary PWM Mode, and Timer-KB PWM Output Gating Mode]

Address: F0397H
After reset: FFH**Note 1**
R/W: 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**Notes 2, 3**
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, 4**
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**Notes 2, 3, 5**
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, 5, 6, 7**
0	Output enabled
1	Output disabled (TRDIOA0 pin functions as an I/O port)

Note 1. The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. Set to 1 in PWM3 mode.

(Notes and Caution are listed on the next page.)

Note 3. Set to 1 in extended PWM mode.
Note 4. Set to 1 in PWM function.
Note 5. Set to 1 in timer-KB PWM output gating mode.
Note 6. Set to 1 in reset synchronous PWM mode or complementary PWM mode.
Note 7. Set to 1 in extended complementary PWM mode.

Caution When the HAZAD_SET bit in the OPCTL0 register is set to 1 (hazard measure enabled), the TRDOER1 register can be changed during count operation of timer RD2. (The TRDIO function and port function are switched during operation of timer RD2.)

12.3.11 Timer RD output master enable register 2 (TRDOER2)

Figure 12 - 14 Format of Timer RD Output Master Enable Register 2 (TRDOER2)
[PWM Function, Reset Synchronous PWM Mode, Complementary PWM Mode, PWM3 Mode, Extended PWM Mode, Extended Complementary PWM Mode, and Timer-KB PWM Output Gating Mode]

Address: F0398H
After reset: 00H**Note 1**
R/W: R/W

Symbol	<7>	6	5	4	3	2	1	<0>
TRDOER2	TRDPTO	0	0	0	0	0	0	TRDSHUTS

TRDPTO	Enabling the input of pulse output forced cutoff signal through the INTP0 pin**Note 2**
0	Pulse output forced cutoff input disabled
1	Pulse output forced cutoff input enabled (The TRDSHUTS flag 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 pin 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 flag in an enabled mode.	

Note 1. The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. See **12.4.4 Pulse output forced cutoff**.

12.3.12 Timer RD output control register (TRDOCR)

Write to the TRDOCR register when bits TSTART1 and TSTART0 in the TRDSTR register are 00B (count stops).

Figure 12 - 15 Format of Timer RD Output Control Register (TRDOCR) [Output Compare Function]

Address: F0399H
After reset: 00H[Note 1]
R/W: 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[Note 2]
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[Note 2]
0	Low initial output
1	High initial output

Note 1. The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. 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 12 - 16 Format of Timer RD Output Control Register (TRDOCR) [PWM Function]

Address: F0399H
After reset: 00H[Note 1]
R/W: 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	Output is initially at the inactive level.
1	Output is initially at the active level.

TOC1	TRDIOC1 initial output level select[Note 2]
0	Output is initially at the inactive level.
1	Output is initially at the active level.

TOB1	TRDIOB1 initial output level select[Note 2]
0	Output is initially at the inactive level.
1	Output is initially at the active level.

TOA1	TRDIOA1 initial output level select[Note 2]
Set to 0.	

TOD0	TRDIOD0 initial output level select[Note 2]
0	Output is initially at the inactive level.
1	Output is initially at the active level.

TOC0	TRDIOC0 initial output level select[Note 2]
0	Output is initially at the inactive level.
1	Output is initially at the active level.

TOB0	TRDIOB0 initial output level select[Note 2]
0	Output is initially at the inactive level.
1	Output is initially at the active level.

TOA0	TRDIOA0 initial output level select[Note 2]
Set to 0.	

Note 1. The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. 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 12 - 17 Format of Timer RD Output Control Register (TRDOCR) [Reset Synchronous PWM Mode and Complementary PWM Mode]

Address: F0399H
After reset: 00H[Note 1]
R/W: 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	The setting of these bits 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	Low initial output	In reset synchronous PWM mode, the output is inverted every PWM period. In complementary PWM mode, the output is inverted every 1/2 PWM period.
1	High initial output	

Note 1. The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H) and the TRD0EN bit is 0 in the PER2 register. 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 12 - 18 Format of Timer RD Output Control Register (TRDOCR) [PWM3 Mode]

Address: F0399H
After reset: 00H[Note 1]
R/W: 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 and high active level A high level is output at TRDGRB1 compare match. A low level is output at TRDGRB0 compare match.
1	High initial output and low active level A low level is output at TRDGRB1 compare match. A high level is output at TRDGRB0 compare match.

TOA0	TRDIOA0 initial output level select[Note 2]
0	Low initial output and high active level A high level is output at TRDGRA1 compare match. A low level is output at TRDGRA0 compare match.
1	High initial output and low active level A low level is output at TRDGRA1 compare match. A high level is output at TRDGRA0 compare match.

Note 1. The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. 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.

12.3.13 Timer RD digital filter function select registers 0 and 1 (TRDDF0 and TRDDF1)

Figure 12 - 19 Format of Timer RD Digital Filter Function Select Register i (TRDDFi) [Input Capture Function]

Address: F039AH (TRDDF0), F039BH (TRDDF1)
After reset: 00H[Note 1]
R/W: 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 TCK2 to TCK0 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 the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. Set bits DFCK1 and DFCK0 before starting count operation.

Note 3. When the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H), $f_{CLK}/32$, $f_{CLK}/8$, and f_{CLK} are set to $f_{HOCO}/32$, $f_{HOCO}/8$, and f_{HOCO}, respectively. When the CKSELR bit is 1 in the main clock control register (MCKC), $f_{CLK}/32$, $f_{CLK}/8$, and f_{CLK} are set to $f_{PLL}/32$, $f_{PLL}/8$, and f_{PLL}, respectively.

Figure 12 - 20 Format of Timer RD Digital Filter Function Select Register i (TRDDFi) [PWM Function, Reset Synchronous PWM Mode, Complementary PWM Mode, PWM3 Mode, Extended PWM Mode, Extended Complementary PWM Mode, and Timer-KB PWM Output Gating Mode]

Address: F039AH (TRDDF0), F039BH (TRDDF1)
After reset: 00H[Note]
R/W: 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 RD2 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 RD2 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 RD2 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 RD2 output port in these modes. Also, set these bits while the count is stopped.		

Note The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

12.3.14 Timer RD control registers 0 and 1 (TRDCR0 and TRDCR1)

The TRDCR1 register is not used in reset synchronous PWM mode or PWM3 mode. Set the TRDCR0 and TRDCR1 registers to 00H in timer-KB PWM output gating mode.

Figure 12 - 21 Format of Timer RD Control Register i (TRDCRi) [Input Capture Function and Output Compare Function]

Address: F03A0H (TRDCR0), F03B0H (TRDCR1)
After reset: 00H**Note 1**
R/W: 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 RD2i 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**, f_{PLL}**Note 7**
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 the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. This setting is valid when the TRDSYNC bit in the TRDMR register is 1 (TRD0 and TRD1 operate synchronously).

Note 3. The setting of these bits is 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 the FRQSEL4 bit is 0 in the user option byte (000C2H/010C2H) and f_{HOCO} is selected when the FRQSEL4 bit is 1. When selecting f_{HOCO} as the count source for timer RD2, set f_{CLK} to f_{IH} before setting bit 3 (TRD0EN) in peripheral enable register 2 (PER2). When changing f_{CLK} to a clock other than f_{IH}, clear bit 3 (TRD0EN) in peripheral enable register 2 (PER2) before changing.

Note 5. Do not set this value when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) or the CKSELR bit is 1 in the main clock control register (MCKC).

Note 6. This value is valid when the STCLK bit in the TRDFCR register is set to 1 (external clock input enabled).

Note 7. f_{PLL} is supplied when the CKSELR bit is 1 in the main clock control register (MCKC).

Figure 12 - 22 Format of Timer RD Control Register i (TRDCRi) [PWM Function]

Address: F03A0H (TRDCR0), F03B0H (TRDCR1)
After reset: 00H[Note 1]
R/W: 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 counter 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], f_{PLL}[Note 6]
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 the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. The setting of these bits is 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. CLK is selected when the FRQSEL4 bit is 0 in the user option byte (000C2H/010C2H) and f_{HOCO} is selected when the FRQSEL4 bit is 1. When selecting f_{HOCO} as the count source for timer RD2, set f_{CLK} to f_{IH} before setting bit 3 (TRD0EN) in peripheral enable register 2 (PER2). When changing f_{CLK} to a clock other than f_{IH}, clear bit 3 (TRD0EN) in peripheral enable register 2 (PER2) before changing.

Note 4. Do not set this value when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) or the CKSELR bit is 1 in the main clock control register (MCKC).

Note 5. This value is valid when the STCLK bit in the TRDFCR register is set to 1 (external clock input enabled).

Note 6. f_{PLL} is supplied when the CKSELR bit is 1 in the main clock control register (MCKC).

Figure 12 - 23 Format of Timer RD Control Register 0 (TRDCR0) [Reset Synchronous PWM Mode]

Address: F03A0H
After reset: 00H[Note 1]
R/W: 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 counter 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], f_{PLL}[Note 6]
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 the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. The setting of these bits is 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 the FRQSEL4 bit is 0 in the user option byte (000C2H/010C2H) and f_{HOCO} is selected when the FRQSEL4 bit is 1. When selecting f_{HOCO} as the count source for timer RD2, set f_{CLK} to f_{IH} before setting bit 3 (TRD0EN) in peripheral enable register 2 (PER2). When changing f_{CLK} to a clock other than f_{IH}, clear bit 3 (TRD0EN) in peripheral enable register 2 (PER2) before changing.

Note 4. Do not set this value when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) or the CKSELR bit is 1 in the main clock control register (MCKC).

Note 5. This value is valid when the STCLK bit in the TRDFCR register is set to 1 (external clock input enabled).

Note 6. f_{PLL} is supplied when the CKSELR bit is 1 in the main clock control register (MCKC).

Figure 12 - 24 Format of Timer RD Control Register i (TRDCRi) [Complementary PWM Mode and Extended Complementary PWM Mode]

Address: F03A0H (TRDCR0), F03B0H (TRDCR1)
After reset: 00H**Note 1**
R/W: 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**, f_{PLL}**Note 7**
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 the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. The setting of these bits is 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 TCK2 to TCK0, CKEG1, and CKEG0 in registers TRDCR0 and TRDCR1.

Note 4. f_{CLK} is selected when the FRQSEL4 bit is 0 in the user option byte (000C2H/010C2H) and f_{HOCO} is selected when the FRQSEL4 bit is 1. When selecting f_{HOCO} as the count source for timer RD2, set f_{CLK} to f_{IH} before setting bit 3 (TRD0EN) in peripheral enable register 2 (PER2). When changing f_{CLK} to a clock other than f_{IH}, clear bit 3 (TRD0EN) in peripheral enable register 2 (PER2) before changing.

Note 5. Do not set this value when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) or the CKSELR bit is 1 in the main clock control register (MCKC).

Note 6. This value is valid when the STCLK bit in the TRDFCR register is set to 1 (external clock input enabled).

Note 7. f_{PLL} is supplied when the CKSELR bit is 1 in the main clock control register (MCKC).

Figure 12 - 25 Format of Timer RD Control Register 0 (TRDCR0) [PWM3 Mode]

Address: F03A0H (TRDCR0)
After reset: 00H[Note 1]
R/W: 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 counter 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], f_{PLL}[Note 4]
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 the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. f_{CLK} is selected when the FRQSEL4 bit is 0 in the user option byte (000C2H/010C2H) and f_{HOCO} is selected when the FRQSEL4 bit is 1. 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 the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) or the CKSELR bit is 1 in the main clock control register (MCKC).

Note 4. f_{PLL} is supplied when the CKSELR bit is 1 in the main clock control register (MCKC).

12.3.15 Timer RD I/O control registers A0 and A1 (TRDIORA0 and TRDIORA1)

Figure 12 - 26 Format of Timer RD I/O Control Register Ai (TRDIORAi) [Input Capture Function]

Address: F03A1H (TRDIORA0), F03B1H (TRDIORA1)
After reset: 00H[Note 1]
R/W: 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 the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. 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 12 - 27 Format of Timer RD I/O Control Register Ai (TRDIORAi) [Output Compare Function]

Address: F03A1H (TRDIORA0), F03B1H (TRDIORA1)
After reset: 00H[Note 1]
R/W: 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 the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. 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.

12.3.16 Timer RD I/O control registers C0 and C1 (TRDIORC0 and TRDIORC1)

Figure 12 - 28 Format of Timer RD I/O Control Register Ci (TRDIORCi) [Input Capture Function]

Address: F03A2H (TRDIORC0), F03B2H (TRDIORC1)
After reset: 88H[Note 1]
R/W: 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 the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. 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 12 - 29 Format of Timer RD I/O Control Register Ci (TRDIORCi) [Output Compare Function]

Address: F03A2H (TRDIORC0), F03B2H (TRDIORC1)
After reset: 88H[Note 1]
R/W: 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 **12.5.2 Output compare function**, **2. Changing output pins for registers TRDGRCi and TRDGRDi (i = 0 or 1)**.)
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 **12.5.2 Output compare function**, **2. Changing output pins for registers TRDGRCi and TRDGRDi (i = 0 or 1)**.)
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 the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. 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.

12.3.17 Timer RD status register 0 (TRDSR0)

Figure 12 - 30 Format of Timer RD Status Register 0 (TRDSR0) [Input Capture Function]

Address: F03A3H
After reset: 00H[Note 1]
R/W: 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 counter overflows.	

IMFD	Input capture/compare match flag D[Note 4]
[Source for setting to 0] Write 0 after reading.[Note 3] [Source for setting to 1] Input edge of TRDIOD0 pin[Note 5]	

IMFC	Input capture/compare match flag C[Note 4]
[Source for setting to 0] Write 0 after reading.[Note 3] [Source for setting to 1] Input edge of TRDIOC0 pin[Note 5]	

IMFB	Input capture/compare match flag B[Note 4]
[Source for setting to 0] Write 0 after reading.[Note 3] [Source for setting to 1] Input edge of TRDIOB0 pin[Note 6]	

IMFA	Input capture/compare match flag A[Note 4]
[Source for setting to 0] Write 0 after reading.[Note 3] [Source for setting to 1] Input edge of TRDIOA0 pin[Note 6]	

Note 1. The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. When the counter value of timer RD20 changes from FFFFH to 0000H, the overflow flag is set to 1. Also, if the counter value of timer RD20 changes from FFFFH to 0000H due to an input capture/compare match during operation according to the settings of bits CCLR2 to CCLR0 in the TRDCR0 register, the overflow flag is set to 1.

(Notes and Remark are listed on the next page.)

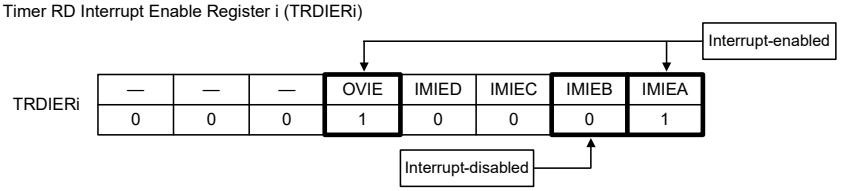
Note 3. The writing results are as follows:

- 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 RD2 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).

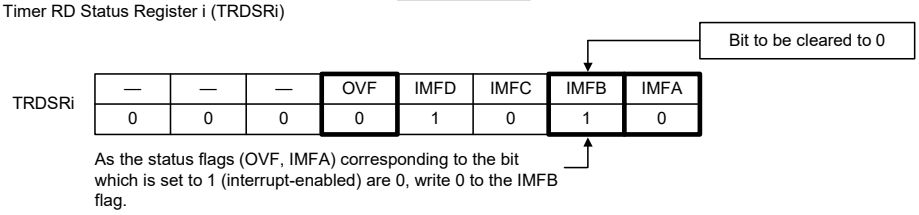
a) Set 00H (all interrupts disabled) to timer RD interrupt enable register i (TRDIERi) and write 0 to applicable status flags.

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 flag 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)

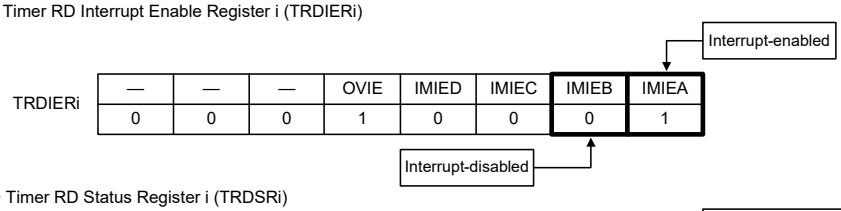


• Timer RD Status Register i (TRDSRi)

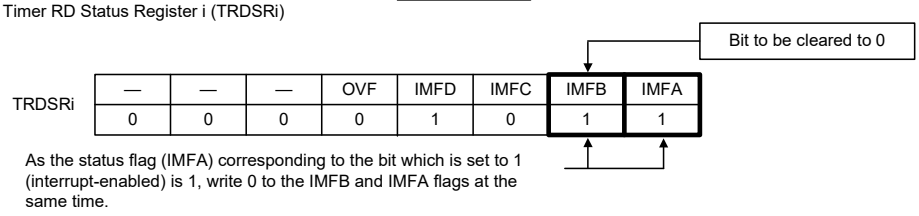


c) When there are bits set to 1 (interrupt-enabled) in timer RD interrupt enable register i (TRDIERi) and status flags of interrupt sources related to their bits are 1, write 0 to these status flags and applicable status flags at the same time.
Example: To clear the IMFB flag to 0 when the IMIEA bit 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)



Note 4. When the DTC is used, the IMFA, IMFB, IMFC, and IMFD flags are set to 1 after DTC transfer is completed.

Note 5. 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 6. Edge selected by bits IOj1 and IOj0 (j = A or B) in the TRDIORA0 register.

Remark i = 0

Figure 12 - 31 Format of Timer RD Status Register 0 (TRDSR0) [Functions Other Than Input Capture Function] (1/2)

Address: F03A3H
After reset: 00H[Note 1]
R/W: 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 counter overflows. Do not use this flag in extended complementary PWM mode or timer-KB PWM output gating mode.	

IMFD	Input capture/compare match flag D[Note 4]
In extended PWM mode [Source for setting to 0] Write 0 after reading.[Note 3] [Source for setting to 1] When the values of TRD0 and TRDCMPB0 match. In the other modes [Source for setting to 0] Write 0 after reading.[Note 3] [Source for setting to 1] When the values of TRD0 and TRDGRD0 match.[Note 5] Do not use this flag in extended complementary PWM mode or timer-KB PWM output gating mode.	

IMFC	Input capture/compare match flag C[Note 4]
[Source for setting to 0] Write 0 after reading.[Note 3] [Source for setting to 1] When the values of TRD0 and TRDGRC0 match.[Note 5] Do not use this flag in extended complementary PWM mode or timer-KB PWM output gating mode.	

IMFB	Input capture/compare match flag B[Note 4]
[Source for setting to 0] Write 0 after reading.[Note 3] [Source for setting to 1] When the values of TRD0 and TRDGRB0 match. Do not use this flag in extended complementary PWM mode or timer-KB PWM output gating mode.	

Figure 12 - 31 Format of Timer RD Status Register 0 (TRDSR0) [Functions Other Than Input Capture Function] (2/2)

IMFA	Input capture/compare match flag A[Note 4]
[Source for setting to 0] Write 0 after reading.[Note 3] [Source for setting to 1] When the values of TRD0 and TRDGRA0 match. Do not use this flag in extended complementary PWM mode.	

Note 1. The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

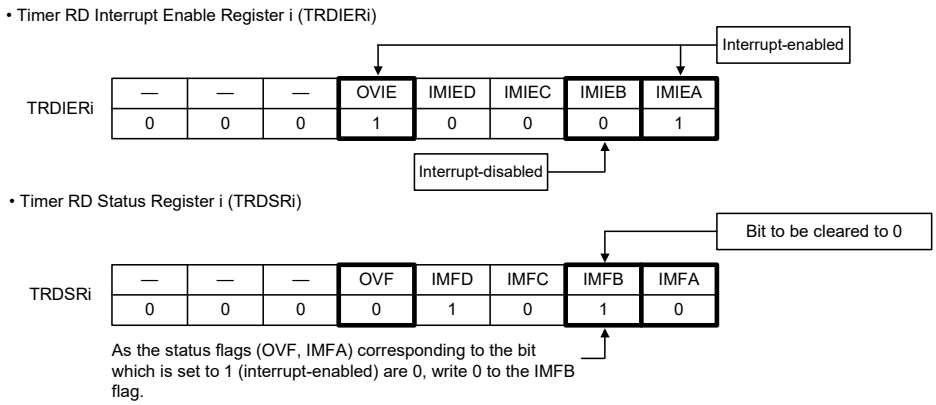
Note 2. When the counter value of timer RD20 changes from FFFFH to 0000H, the overflow flag is set to 1. Also, if the counter value of timer RD20 changes from FFFFH to 0000H due to an input capture/compare match during operation according to the settings of bits CCLR2 to CCLR0 in the TRDCR0 register, 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 RD2 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.

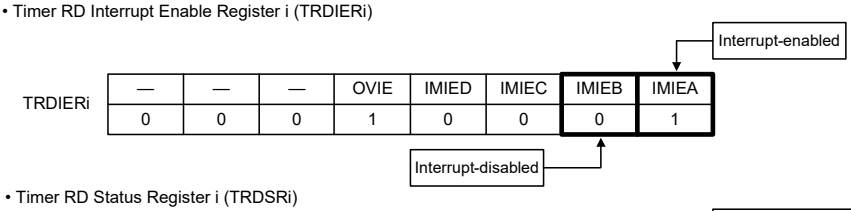
b) When there are bits set to 1 (interrupt-enabled) in timer RD interrupt enable register i (TRDIERi) and status flags of interrupt sources related to their bits are 0, write 0 to applicable status flags.
Example: To clear the IMFB flag to 0 when bits IMIEA and OVIE are set to 1 (interrupt-enabled) and the IMIEB bit is set to 0 (interrupt-disabled).



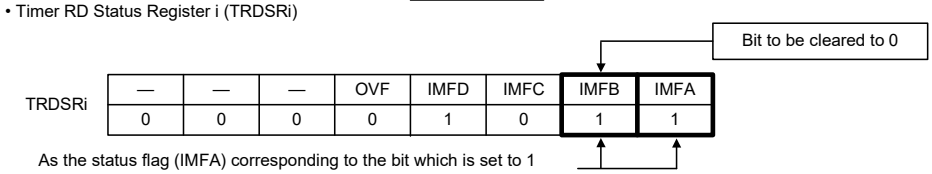
c) When there are bits set to 1 (interrupt-enabled) in timer RD interrupt enable register i (TRDIERi) and status flags of interrupt sources related to their bits are 1, write 0 to these status flags and applicable status flags at the same time.

Example: To clear the IMFB flag 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 the IMFB and IMFA flags at the same time.

Note 4. When the DTC is used, the IMFA, IMFB, IMFC, and IMFD flags are set to 1 after DTC transfer is completed.

Note 5. Including when the TRDBFk0 bit (k = C or D) in the TRDMR register is set to 1 (TRDGRK0 is buffer register).

Remark i = 0

12.3.18 Timer RD status register 1 (TRDSR1)

Figure 12 - 32 Format of Timer RD Status Register 1 (TRDSR1) [Input Capture Function]

Address: F03B3H
After reset: 40H Note 1
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TRDSR1	0	UDS	UDF	OVF	IMFD	IMFC	IMFB	IMFA

UDS	Counter operation status flag
Disabled in the input capture function.	

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 counter overflows.	

IMFD	Input capture/compare match flag D Note 4
[Source for setting to 0] Write 0 after reading. Note 3 [Source for setting to 1] Input edge of TRDIOD1 pin Note 5	

IMFC	Input capture/compare match flag C Note 4
[Source for setting to 0] Write 0 after reading. Note 3 [Source for setting to 1] Input edge of TRDIOC1 pin Note 5	

IMFB	Input capture/compare match flag B Note 4
[Source for setting to 0] Write 0 after reading. Note 3 [Source for setting to 1] Input edge of TRDIOB1 pin Note 6	

IMFA	Input capture/compare match flag A Note 4
[Source for setting to 0] Write 0 after reading. Note 3 [Source for setting to 1] Input edge of TRDIOA1 pin Note 6	

Note 1. The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

(Notes and Remark are listed on the next page.)

Note 2. When the counter value of timer RD21 changes from FFFFH to 0000H, the overflow flag is set to 1. Also, if the counter value of timer RD21 changes from FFFFH to 0000H due to an input capture/compare match during operation according to the settings of bits CCLR2 to CCLR0 in the TRDCR1 register, the overflow flag is set to 1.

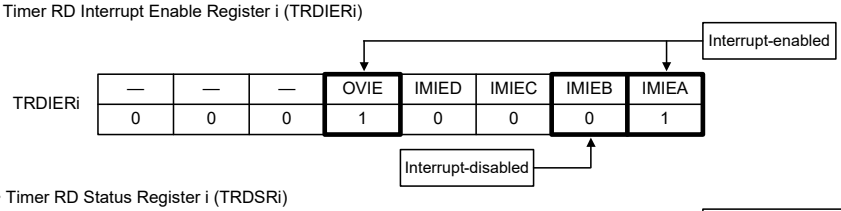
Note 3. The writing results are as follows:

- 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 RD2 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).

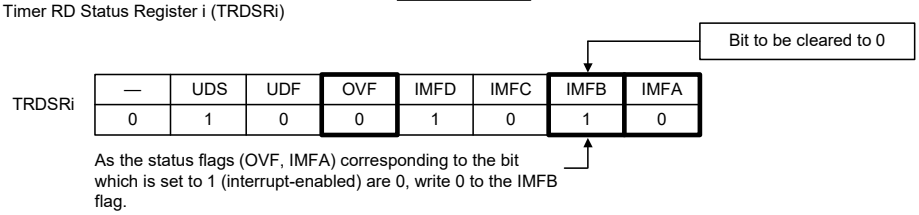
a) Set 00H (all interrupts disabled) to timer RD interrupt enable register i (TRDIERi) and write 0 to applicable status flags.

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 flag 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)

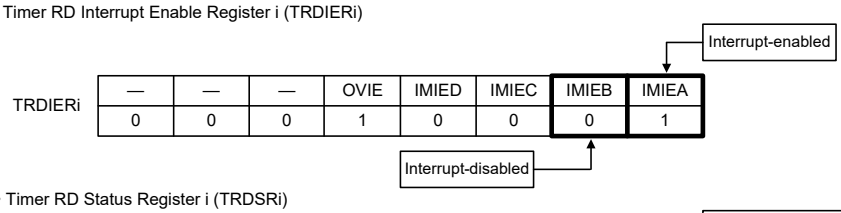


• Timer RD Status Register i (TRDSRi)

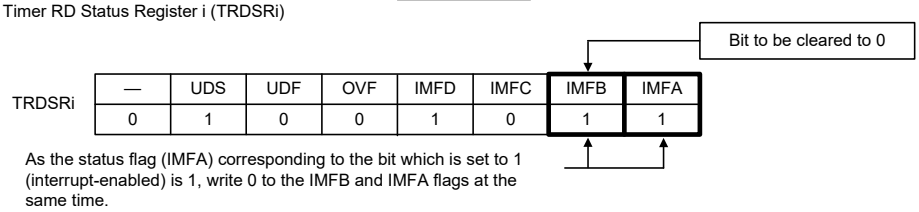


c) When there are bits set to 1 (interrupt-enabled) in timer RD interrupt enable register i (TRDIERi) and status flags of interrupt sources related to their bits are 1, write 0 to these status flags and applicable status flags at the same time.
Example: To clear the IMFB flag to 0 when the IMIEA bit 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)



Note 4. When the DTC is used, the IMFA, IMFB, IMFC, and IMFD flags are set to 1 after DTC transfer is completed.

Note 5. 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 6. Edge selected by bits IOj1 and IOj0 (j = A or B) in the TRDIORA1 register.

Remark i = 1

Figure 12 - 33 Format of Timer RD Status Register 1 (TRDSR1) [Functions Other Than Input Capture Function] (1/2)

Address: F03B3H
After reset: 40H[Note 1]
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TRDSR1	0	UDS	UDF	OVF	IMFD	IMFC	IMFB	IMFA

UDS	Counter operation status flag

In extended complementary PWM mode
[Source for setting to 0]
The TRD1 counter is decrementing.
[Source for setting to 1]
The TRD1 counter is incrementing.

Enabled only in extended complementary PWM mode.

UDF	Underflow flag

In complementary PWM mode or extended complementary PWM mode
[Source for setting to 0]
Write 0 after reading.[Note 2]
[Source for setting to 1]
When the TRD1 counter underflows.

Enabled only in complementary PWM mode and extended complementary PWM mode.

OVF	Overflow flag[Note 3]

In extended complementary PWM mode
[Source for setting to 0]
Write 0 after reading.[Note 2]
[Source for setting to 1]
When the values of TRD0 and TRDGRA0 match.

In the other modes
[Source for setting to 0]
Write 0 after reading.[Note 2]
[Source for setting to 1]
When the TRD1 counter overflows.

Do not use this flag in timer-KB PWM output gating mode.

Figure 12 - 33 Format of Timer RD Status Register 1 (TRDSR1) [Functions Other Than Input Capture Function] (2/2)

IMFD	Input capture/compare match flag D[Note 4]
In extended PWM mode [Source for setting to 0] Write 0 after reading.[Note 2] [Source for setting to 1] When the values of TRD1 and TRDCMPB1 match. In the other modes [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 5] Do not use this flag in extended complementary PWM mode or timer-KB PWM output gating mode.	

IMFC	Input capture/compare match flag C[Note 4]
[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 5] Do not use this flag in extended complementary PWM mode or timer-KB PWM output gating mode.	

IMFB	Input capture/compare match flag B[Note 4]
[Source for setting to 0] Write 0 after reading.[Note 2] [Source for setting to 1] When the values of TRD1 and TRDGRB1 match. Do not use this flag in extended complementary PWM mode or timer-KB PWM output gating mode.	

IMFA	Input capture/compare match flag A[Note 4]
[Source for setting to 0] Write 0 after reading.[Note 2] [Source for setting to 1] When the values of TRD1 and TRDGRA1 match. Do not use this flag in extended complementary PWM mode.	

Note 1. The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

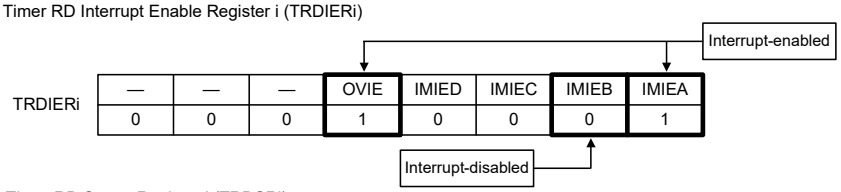
Note 2. The writing results are as follows:

- 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 RD2 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).

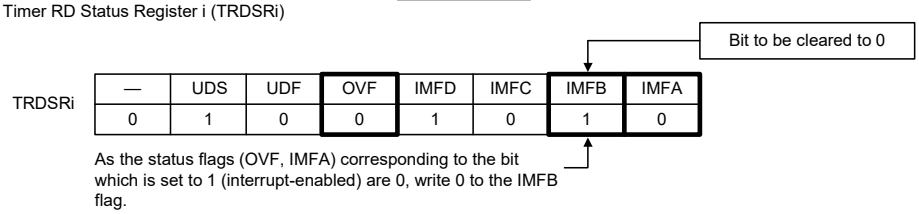
  a) Set 00H (all interrupts disabled) to timer RD interrupt enable register i (TRDIERi) and write 0 to applicable status flags.

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 flag 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)

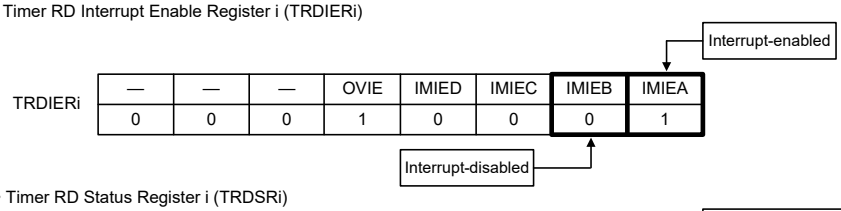


• Timer RD Status Register i (TRDSRi)

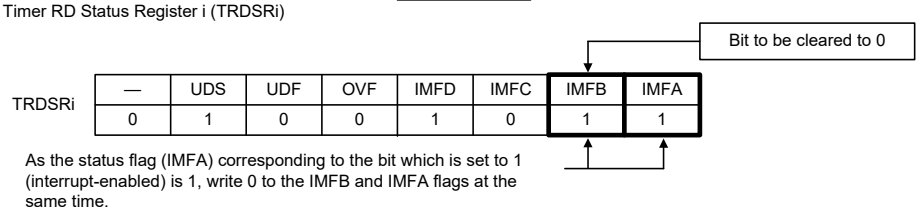


c) When there are bits set to 1 (interrupt-enabled) in timer RD interrupt enable register i (TRDIERi) and status flags of interrupt sources related to their bits are 1, write 0 to these status flags and applicable status flags at the same time.
Example: To clear the IMFB flag to 0 when the IMIEA bit 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)



Note 3. When the counter value of timer RD21 changes from FFFFH to 0000H, the overflow flag is set to 1. Also, if the counter value of timer RD21 changes from FFFFH to 0000H due to an input capture/compare match during operation according to the settings of bits CCLR2 to CCLR0 in the TRDCR1 register, the overflow flag is set to 1.

Note 4. When the DTC is used, the IMFA, IMFB, IMFC, and IMFD flags are set to 1 after DTC transfer is completed.

Note 5. Including when the TRDBFk1 bit (k = C or D) in the TRDMR register is set to 1 (TRDGRK1 is buffer register).

Remark i = 1

12.3.19 Timer RD interrupt enable registers 0 and 1 (TRDIER0 and TRDIER1)

Figure 12 - 34 Format of Timer RD Interrupt Enable Register i (TRDIERi)

Address: F03A4H (TRDIER0), F03B4H (TRDIER1)
After reset: 00H[Note 1]
R/W: 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 the OVF or UDF flag is disabled.
1	Interrupt (OVI) by the OVF or UDF flag is enabled.

IMIED	Input capture/compare match interrupt enable D
0	Interrupt by the IMFD flag is disabled.
1	Interrupt by the IMFD flag is enabled.

IMIEC[Note 2]	Input capture/compare match interrupt enable C
0	Interrupt by the IMFC flag is disabled.
1	Interrupt by the IMFC flag is enabled.

IMIEB	Input capture/compare match interrupt enable B
0	Interrupt by the IMFB flag is disabled.
1	Interrupt by the IMFB flag is enabled.

IMIEA	Input capture/compare match interrupt enable A
0	Interrupt by the IMFA flag is disabled.
1	Interrupt by the IMFA flag is enabled.

Note 1. The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. Set this bit to 0 in extended PWM mode.

12.3.20 Timer RD PWM output level control register 0 (TRDPOCR0)

Settings to the TRDPOCR0 register are only effective in timer mode (with the PWM function in use), extended PWM mode, and timer-KB PWM output gating mode.

Figure 12 - 35 Format of Timer RD PWM Output Level Control Register 0 (TRDPOCR0) [PWM Function and Extended PWM Mode]

Address: F03A5H (TRDPOCR0), F03B5H (TRDPOCR1)
After reset: 00H[Note]
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TRDPOCR0	0	0	0	0	0	POLD	POLC	POLB

POLD	Output level control D with the PWM function in use or in extended PWM mode
0	The output signals on the TRDIODi pins are active low.
1	The output signals on the TRDIODi pins are active high.

POLC	Output level control C with the PWM function in use or in extended PWM mode
0	The output signals on the TRDIOCi pins are active low.
1	The output signals on the TRDIOCi pins are active high.
Set this bit to 0 in extended PWM mode.	

POLB	Output level control B with the PWM function in use or in extended PWM mode
0	The output signals on the TRDIOBi pins are active low.
1	The output signals on the TRDIOBi pins are active high.

Note The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Figure 12 - 36 Format of Timer RD PWM Output Level Control Register 0 (TRDPOCR0) [Timer-KB PWM Output Gating Mode]

Address: F03A5H (TRDPOCR0), F03B5H (TRDPOCR1)
After reset: 00H[Note]
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TRDPOCR0	0	0	0	0	0	POLD	POLC	POLB

POLD	Output level control D in timer-KB PWM output gating mode
0	The output signal on the TRDTKBOUT5 or TRDIOD0 pin is active low.
1	The output signal on the TRDTKBOUT5 or TRDIOD0 pin is active high.

POLC	Output level control C in timer-KB PWM output gating mode
Set this bit to 0 in timer-KB PWM output gating mode.	

POLB	Output level control B in timer-KB PWM output gating mode
0	The output signal on the TRDTKBOUT4 or TRDIOB0 pin is active low.
1	The output signal on the TRDTKBOUT4 or TRDIOB0 pin is active high.

Note The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

12.3.21 Timer RD PWM output level control register 1 (TRDPOCR1)

Settings to the TRDPOCR1 register are only effective in timer mode (with the PWM function in use), extended PWM mode, and timer-KB PWM output gating mode.

Figure 12 - 37 Format of Timer RD PWM Output Level Control Register 1 (TRDPOCR1) [PWM Function and Extended PWM Mode]

Address: F03A5H (TRDPOCR0), F03B5H (TRDPOCR1)
After reset: 00H**Note**
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TRDPOCR1	0	0	0	0	POLA	POLD	POLC	POLB

POLA	Output level control A with the PWM function in use or in extended PWM mode
Set this bit to 0 with the PWM function in use or in extended PWM mode.	

POLD	Output level control D with the PWM function in use or in extended PWM mode
0	The output signals on the TRDIODi pins are active low.
1	The output signals on the TRDIODi pins are active high.

POLC	Output level control C with the PWM function in use or in extended PWM mode
0	The output signals on the TRDIOCi pins are active low.
1	The output signals on the TRDIOCi pins are active high.
Set this bit to 0 in extended PWM mode.	

POLB	Output level control B with the PWM function in use or in extended PWM mode
0	The output signals on the TRDIOBi pins are active low.
1	The output signals on the TRDIOBi pins are active high.

Note The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Figure 12 - 38 Format of Timer RD PWM Output Level Control Register 1 (TRDPOCR1) [Timer-KB PWM Output Gating Mode]

Address: F03A5H (TRDPOCR0), F03B5H (TRDPOCR1)
After reset: 00H[Note]
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TRDPOCR1	0	0	0	0	POLA	POLD	POLC	POLB

POLA	Output level control A in timer-KB PWM output gating mode
0	The output signal on the TRDTKBOUT3 or TRDIOA1 pin is active low.
1	The output signal on the TRDTKBOUT3 or TRDIOA1 pin is active high.

POLD	Output level control D in timer-KB PWM output gating mode
0	The output signal on the TRDTKBOUT2 or TRDIOD1 pin is active low.
1	The output signal on the TRDTKBOUT2 or TRDIOD1 pin is active high.

POLC	Output level control C in timer-KB PWM output gating mode
0	The output signal on the TRDTKBOUT1 or TRDIOC1 pin is active low.
1	The output signal on the TRDTKBOUT1 or TRDIOC1 pin is active high.

POLB	Output level control B in timer-KB PWM output gating mode
0	The output signal on the TRDTKBOUT0 or TRDIOB1 pin is active low.
1	The output signal on the TRDTKBOUT0 or TRDIOB1 pin is active high.

Note The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

12.3.22 Timer RD counters 0 and 1 (TRD0 and TRD1)

[Timer Mode]

Access the TRDi counter in 16-bit units. Do not access it in 8-bit units.

[Reset Synchronous PWM Mode and PWM3 Mode]

Access the TRD0 counter in 16-bit units. Do not access it in 8-bit units. The TRD1 counter is not used in reset synchronous PWM mode and PWM3 mode.

[Complementary PWM Mode and Extended Complementary PWM Mode (TRD0)]

Access the TRD0 counter in 16-bit units. Do not access it in 8-bit units.

[Complementary PWM Mode and Extended Complementary PWM Mode (TRD1)]

Access the TRD1 counter in 16-bit units. Do not access it in 8-bit units.

[Extended PWM Mode and Timer-KB PWM Output Gating Mode]

Access the TRDi counters in 16-bit units. Do not access them in 8-bit units.

Figure 12 - 39 Format of Timer RD Counter i (TRDi) [Timer Mode]

Address: F03A6H (TRD0), F03B6H (TRD1)
After reset: 0000H**Note**
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDi																

—	Function
Bits 15 to 0	Count the count source. Count operation is fixed to increment. When an overflow occurs, the OVF flag in the TRDSRi register is set to 1.

Note The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Figure 12 - 40 Format of Timer RD Counter 0 (TRD0) [Reset Synchronous PWM Mode and PWM3 Mode]

Address: F03A6H (TRD0)
After reset: 0000H**Note**
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRD0																

—	Function
Bits 15 to 0	Count the count source. Count operation is fixed to increment. When an overflow occurs, the OVF flag in the TRDSR0 register is set to 1.

Note The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Figure 12 - 41 Format of Timer RD Counter 0 (TRD0) [Complementary PWM Mode and Extended Complementary PWM Mode (TRD0)]

Address: F03A6H (TRD0)
After reset: 0000H[Note 1]
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRD0																

—	Function
Bits 15 to 0	Set the dead time. The specifiable range depends on the mode. In complementary PWM mode: 0001H to 7FFFH In extended complementary PWM mode: 0002H to 7FFFH[Note 2] Count the count source. Count operation is increment or decrement.

Note 1. The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Note 2. Set the value in extended complementary PWM mode so that the dead time is shorter than a quarter of the complementary PWM period.

Figure 12 - 42 Format of Timer RD Counter 1 (TRD1) [Complementary PWM Mode and Extended Complementary PWM Mode (TRD1)]

Address: F03B6H (TRD1)
After reset: 0000H[Note]
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRD1																

—	Function
Bits 15 to 0	Set to 0000H. Count the count source. Count operation is increment or decrement. When an underflow occurs, the UDF flag in the TRDSR1 register is set to 1.

Note The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Figure 12 - 43 Format of Timer RD Counter 1 (TRD1) [Extended PWM Mode and Timer-KB PWM Output Gating Mode]

Address: F03B6H (TRD1)
After reset: 0000H**Note**
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRD1																

—	Function
Bits 15 to 0	Set to 0000H. Count the count source. Count operation is increment. The value of the IMFA bit of the TRDSRi register becomes 1 in response to a match with the specified period value.

Note The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

12.3.23 Timer RD general registers A0, A1, B0, B1, C0, C1, D0, and D1 (TRDGRA0, TRDGRA1, TRDGRB0, TRDGRB1, TRDGRC0, TRDGRC1, TRDGRD0, and TRDGRD1)

[Input Capture Function]

Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.
Set the pulse width of the input capture signal applied to the TRDIOji pin to three or more cycles of the timer RD2 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.

[PWM Function]

Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.

[Reset Synchronous PWM Mode]

Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.

[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.

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 can 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.
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 can be set to 1 (buffer register).

[Extended PWM Mode]

Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.

[Extended Complementary PWM Mode]

Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.
In extended complementary PWM mode, rewriting the TRDGRA0 and TRDGRC0 registers (controlling the PWM period) during operation is not possible.

[Timer-KB PWM Output Gating Mode]

Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.

Figure 12 - 44 Format of Timer RD General Registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi, TRDGRCi, and TRDGRDi) [Input Capture Function]

Address: F03A8H (TRDGRA0), F03AAH (TRDGRB0), FFF6CH (TRDGRC0), FFF6EH (TRDGRD0), F03B8H (TRDGRA1), F03BAH (TRDGRB1), FFF70H (TRDGRC1), FFF72H (TRDGRD1)

After reset: FFFFH[Note]

R/W: 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 12 - 2 TRDGRji Register Functions in Input Capture Function**.

Note The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Table 12 - 2 TRDGRji Register Functions in Input Capture Function

Register	Setting	Register Function	Input-capture Input Pin
TRDGRAi	—	General register. The value of the TRDi counter can be read at input capture	TRDIOAi
TRDGRBi			TRDIOBi
TRDGRCi	TRDBFCi = 0	General register. The value of the TRDi counter can be read at input capture.	TRDIOCi
TRDGRDi	TRDBFDi = 0		TRDIODi
TRDGRCi	TRDBFCi = 1	Buffer register. The value of the TRDi counter can be read at input capture (see **12.4.2 Buffer operation**).	TRDIOAi
TRDGRDi	TRDBFDi = 1		TRDIOBi

Remark i = 0 or 1, j = A to D
TRDBFCi, TRDBFDi: Bits in TRDMR register

Figure 12 - 45 Format of Timer RD General Registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi, TRDGRCi, and TRDGRDi) [Output Compare Function]

Address: F03A8H (TRDGRA0), F03AAH (TRDGRB0), FFF6CH (TRDGRC0), FFF6EH (TRDGRD0), F03B8H (TRDGRA1), F03BAH (TRDGRB1), FFF70H (TRDGRC1), FFF72H (TRDGRD1)

After reset: FFFFH**Note**

R/W: 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 12 - 3 TRDGRji Register Functions in Output Compare Function**.

Note The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Table 12 - 3 TRDGRji Register Functions in Output Compare Function

Register	Setting		Register Function		Output-compare Output Pin
	TRDBFji	IOj3			
TRDGRAi	—	—	General register. Write the compare value.		TRDIOAi
TRDGRBi					TRDIOBi
TRDGRCi	0	1	General register. Write the compare value.		TRDIOCi
TRDGRDi					TRDIODi
TRDGRCi	1	1	Buffer register. Write the next compare value (see **12.4.2 Buffer operation**).		TRDIOAi
TRDGRDi					TRDIOBi
TRDGRCi	0	0	TRDIOAi output control	(See **12.5.2 Output compare function**, **2. Changing output pins for registers TRDGRCi and TRDGRDi (i = 0 or 1)**.)	TRDIOAi
TRDGRDi			TRDIOBi output control		TRDIOBi

Caution **When the setting of bits TCK2 to TCK0 in the TRDCRi register is 000B (f_{CLK}, f_{HOCO}, f_{PLL}) and the compare value is set to 0000H, a request signal to the DTC and 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 to D
TRDBFji: Bit in TRDMR register, IOj3: Bit in TRDIORCi register

Figure 12 - 46 Format of Timer RD General Registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi, TRDGRCi, and TRDGRDi) [PWM Function]

Address: F03A8H (TRDGRA0), F03AAH (TRDGRB0), FFF6CH (TRDGRC0), FFF6EH (TRDGRD0), F03B8H (TRDGRA1), F03BAH (TRDGRB1), FFF70H (TRDGRC1), FFF72H (TRDGRD1)

After reset: FFFFH[Note]

R/W: 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 12 - 4 TRDGRji Register Functions in PWM Function**.

Note The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Table 12 - 4 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 timing of toggling the PWM output level.	TRDIOBi
TRDGRCi	TRDBFCi = 0	General register. Set the timing of toggling the PWM output level.	TRDIOCi
TRDGRDi	TRDBFDi = 0		TRDIODi
TRDGRCi	TRDBFCi = 1	Buffer register. Set the next PWM period (see **12.4.2 Buffer operation**).	—
TRDGRDi	TRDBFDi = 1	Buffer register. Set the timing of toggling the PWM output level in the next PWM cycle (see **12.4.2 Buffer operation**).	TRDIOBi

Caution **When the setting of bits TCK2 to TCK0 in the TRDCRi register is 000B (f_{CLK}, f_{HOCO}, f_{PLL}) and the compare value is set to 0000H, a request signal to the DTC and 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 to D
TRDBFCi, TRDBFDi: Bits in TRDMR register

Figure 12 - 47 Format of Timer RD General Registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi, TRDGRCi, and TRDGRDi) [Reset Synchronous PWM Mode]

Address: F03A8H (TRDGRA0), F03AAH (TRDGRB0), FFF6CH (TRDGRC0), FFF6EH (TRDGRD0), F03B8H (TRDGRA1), F03BAH (TRDGRB1), FFF70H (TRDGRC1), FFF72H (TRDGRD1)

After reset: FFFFH[Note]

R/W: 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 12 - 5 TRDGRji Register Functions in Reset Synchronous PWM Mode**.

Note The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Table 12 - 5 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 timing of toggling the PWM1 output level.	TRDIOB0 TRDIOD0
TRDGRC0	TRDBFC0 = 0	(Not used in reset synchronous PWM mode.)	—
TRDGRD0	TRDBFD0 = 0		
TRDGRA1	—	General register. Set the timing of toggling the PWM2 output level.	TRDIOA1 TRDIOC1
TRDGRB1	—	General register. Set the timing of toggling the PWM3 output level.	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 **12.4.2 Buffer operation**).	TRDIOC0 (output inverted every PWM period)
TRDGRD0	TRDBFD0 = 1	Buffer register. Set the timing of toggling the PWM1 output level in the next PWM cycle (see **12.4.2 Buffer operation**).	TRDIOB0 TRDIOD0
TRDGRC1	TRDBFC1 = 1	Buffer register. Set the timing of toggling the PWM2 output level in the next PWM cycle (see **12.4.2 Buffer operation**).	TRDIOA1 TRDIOC1
TRDGRD1	TRDBFD1 = 1	Buffer register. Set the timing of toggling the PWM3 output level in the next PWM cycle (see **12.4.2 Buffer operation**).	TRDIOB1 TRDIOD1

Caution **When the setting of bits TCK2 to TCK0 in the TRDCR0 register is 000B (f_{CLK}, f_{HOCO}, f_{PLL}) and the compare value is set to 0000H, a request signal to the DTC and 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 to D
TRDBFC0, TRDBFD0, TRDBFC1, TRDBFD1: Bits in TRDMR register

Figure 12 - 48 Format of Timer RD General Registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi, TRDGRCi, and TRDGRDi) [Complementary PWM Mode]

Address: F03A8H (TRDGRA0), F03AAH (TRDGRB0), FFF6CH (TRDGRC0), FFF6EH (TRDGRD0), F03B8H (TRDGRA1), F03BAH (TRDGRB1), FFF70H (TRDGRC1), FFF72H (TRDGRD1)

After reset: FFFFH[Note]

R/W: 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 12 - 6 TRDGRji Register Functions in Complementary PWM Mode**.

Note The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Table 12 - 6 TRDGRji Register Functions in Complementary PWM Mode (1/2)

Register	Setting	Register Function	PWM Output Pin
TRDGRA0	—	General register. Set the PWM period at initialization. Setting range: ≥ Value set in TRD0 counter (initial count value) + 1 ≤ FFFFH - value set in TRD0 counter Do not write to this register when both the TSTART0 and TSTART1 bits in the TRDSTR register are set to 1 (count starts).	TRDIOC0 (output inverted every half period)
TRDGRB0	—	General register. Set the timing of toggling the PWM1 output level at initialization. Setting range: ≥ Value set in TRD0 counter (initial count value) + 1 ≤ Value set in TRDGRA0 register - value set in TRD0 counter - 1 Do not write to this register when both the TSTART0 and TSTART1 bits in the TRDSTR register are set to 1 (count starts).	TRDIOB0 TRDIOD0
TRDGRA1	—	General register. Set the timing of toggling the PWM2 output level at initialization. Setting range: ≥ Value set in TRD0 counter (initial count value) + 1 ≤ Value set in TRDGRA0 register - value set in TRD0 counter - 1 Do not write to this register when both the TSTART0 and TSTART1 bits in the TRDSTR register are set to 1 (count starts).	TRDIOA1 TRDIOC1
TRDGRB1	—	General register. Set the timing of toggling the PWM3 output level at initialization. Setting range: ≥ Value set in TRD0 counter (initial count value) + 1 ≤ Value set in TRDGRA0 register - value set in TRD0 counter - 1 Do not write to this register when both the TSTART0 and TSTART1 bits in the TRDSTR register are set to 1 (count starts).	TRDIOB1 TRDIOD1
TRDGRC0	—	Not used	—
TRDGRD0	TRDBFD0 = 1	Buffer register. Set the timing of toggling the PWM1 output level in the next PWM cycle (see **12.4.2 Buffer operation**). Setting range: ≥ Value set in TRD0 counter (initial count value) + 1 ≤ Value set in TRDGRA0 register - value set in TRD0 counter - 1 Set this register to the same value as the TRDGRB0 register at initialization.	TRDIOB0 TRDIOD0

Table 12 - 6 TRDGRji Register Functions in Complementary PWM Mode (2/2)

Register	Setting	Register Function	PWM Output Pin
TRDGRC1	TRDBFC1 = 1	Buffer register. Set the timing of toggling the PWM2 output level in the next PWM cycle (see **12.4.2 Buffer operation**). Setting range: ≥ Value set in TRD0 counter (initial count value) ≤ Value set in TRDGRA0 register - value set in TRD0 counter Set this register to the same value as the TRDGRA1 register at initialization.	TRDIOA1 TRDIOC1
TRDGRD1	TRDBFD1 = 1	Buffer register. Set the timing of toggling the PWM3 output level in the next PWM cycle (see **12.4.2 Buffer operation**). Setting range: ≥ Value set in TRD0 counter (initial count value) ≤ Value set in TRDGRA0 register - value set in TRD0 counter Set this register to the same value as the TRDGRB1 register at initialization.	TRDIOB1 TRDIOD1

Caution **When the setting of bits TCK2 to TCK0 in the TRDCRi register is 000B (f_{CLK}, f_{HOCO}, f_{PLL}) and the compare value is set to 0000H, a request signal to the DTC and 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 to D
TRDBFD0, TRDBFC1, TRDBFD1: Bits in TRDMR register

Figure 12 - 49 Format of Timer RD General Registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi, TRDGRCi, and TRDGRDi) [PWM3 Mode]

Address: F03A8H (TRDGRA0), F03AAH (TRDGRB0), FFF6CH (TRDGRC0), FFF6EH (TRDGRD0), F03B8H (TRDGRA1), F03BAH (TRDGRB1), FFF70H (TRDGRC1), FFF72H (TRDGRD1)

After reset: FFFFH[Note]

R/W: 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 12 - 7 TRDGRji Register Functions in PWM3 Mode**.

Note The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Table 12 - 7 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 timing of toggling the PWM output level (timing of driving the output to the active level). Setting range: ≤ Value set in TRDGRA0 register	
TRDGRB0		General register. Set the timing of toggling the PWM output level (timing of returning to the initial output level). Setting range: ≥ Value set in TRDGRB1 register and ≤ Value set in TRDGRA0 register	TRDIOB0
TRDGRB1		General register. Set the timing of toggling the PWM output level (timing of driving the output to the active level). 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 **12.4.2 Buffer operation**). Setting range: ≥ Value set in TRDGRC1 register	TRDIOA0
TRDGRC1	TRDBFC1 = 1	Buffer register. Set the timing of toggling the PWM output level in the next PWM cycle (see **12.4.2 Buffer operation**). Setting range: ≤ Value set in TRDGRC0 register	
TRDGRD0	TRDBFD0 = 1	Buffer register. Set the timing of toggling the PWM output level in the next PWM cycle (see **12.4.2 Buffer operation**). Setting range: ≥ Value set in TRDGRD1 register and ≤ Value set in TRDGRC0 register	TRDIOB0
TRDGRD1	TRDBFD1 = 1	Buffer register. Set the timing of toggling the PWM output level in the next PWM cycle (see **12.4.2 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}, f_{PLL}) and the compare value is set to 0000H, a request signal to the DTC and 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 is listed on the next page.)

Remark i = 0 or 1, j = A to D
TRDBFC0, TRDBFD0, TRDBFC1, TRDBFD1: Bits in TRDMR register

Figure 12 - 50 Format of Timer RD General Registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi, TRDGRCi, and TRDGRDi) [Extended PWM Mode]

Address: F03A8H (TRDGRA0), F03AAH (TRDGRB0), FFF6CH (TRDGRC0), FFF6EH (TRDGRD0), F03B8H (TRDGRA1), F03BAH (TRDGRB1), FFF70H (TRDGRC1), FFF72H (TRDGRD1)

After reset: FFFFH[Note]

R/W: 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 12 - 8 TRDGRji Register Functions in Extended PWM Mode**.

Note The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Table 12 - 8 TRDGRji Register Functions in Extended PWM Mode

Register	Register Function	PWM Output Pin
TRDGRAi	General register. Set the PWM period.	—
TRDGRBi	General register. Set the timing of toggling the PWM output level.	TRDIOBi
TRDGRCi	Buffer register. Set the next PWM period.	—
TRDGRDi	Buffer register. Set the timing of toggling the PWM output level in the next PWM cycle.	TRDIOBi

Caution **The simultaneous update function is used to transfer data from the buffer registers to the compare registers. After modifying the necessary buffer registers, set the RDT bit in the TRDRDTi register to 1. The values to be used in the next PWM cycle are transferred from the buffer registers to the compare registers when the timing of simultaneous update comes while the RSF flag in the TRDRSFi register is 1. While the RSF flag is 0, values are not transferred from the buffer registers to the compare registers.**

Remark i = 0 or 1, j = A to D

Figure 12 - 51 Format of Timer RD General Registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi, TRDGRCi, and TRDGRDi) [Extended Complementary PWM Mode]

Address: F03A8H (TRDGRA0), F03AAH (TRDGRB0), FFF6CH (TRDGRC0), FFF6EH (TRDGRD0), F03B8H (TRDGRA1), F03BAH (TRDGRB1), FFF70H (TRDGRC1), FFF72H (TRDGRD1)

After reset: FFFFH[Note]

R/W: 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 12 - 9 TRDGRji Register Functions in Extended Complementary PWM Mode**.

Note The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Table 12 - 9 TRDGRji Register Functions in Extended Complementary PWM Mode

Register	Setting	Register Function	PWM Output Pin
TRDGRA0	—	General register. Set the PWM period.	TRDIOC0 (output inverted every half period)
TRDGRB0	CPSS = 0	General register. The timing of toggling the PWM1 output level is stored.	TRDIOB0 TRDIOD0
	CPSS = 1	General register. The timing of toggling the PWM1 output level during increment operation is stored.	
TRDGRA1	CPSS = 0	General register. The timing of toggling the PWM2 output level is stored.	TRDIOA1 TRDIOC1
	CPSS = 1	General register. The timing of toggling the PWM2 output level during increment operation is stored.	
TRDGRB1	CPSS = 1	General register. The timing of toggling the PWM3 output level is stored.	TRDIOB1 TRDIOD1
	CPSS = 1	General register. The timing of toggling the PWM3 output level during increment operation is stored.	
TRDGRC0	—	Buffer register. Set the PWM period at initialization. Setting range: > Value set in TRD0 counter × 2 ≤ FFFFH - value set in TRD0 counter	TRDIOC0 (output inverted every half period)
TRDGRD0	CPSS = 0	Buffer register. Set the timing of toggling the PWM1 output level (see **12.4.2 Buffer operation**).	TRDIOB0 TRDIOD0
	CPSS = 1	Buffer register. Set the timing of toggling the PWM1 output level during increment operation (see **12.4.2 Buffer operation**).	
TRDGRC1	CPSS = 0	Buffer register. Set the timing of toggling the PWM2 output level (see **12.4.2 Buffer operation**).	TRDIOA1 TRDIOC1
	CPSS = 1	Buffer register. Set the timing of toggling the PWM2 output level during increment operation.	
TRDGRD1	CPSS = 0	Buffer register. Set the timing of toggling the PWM3 output level (see **12.4.2 Buffer operation**).	TRDIOB1 TRDIOD1
	CPSS = 1	Buffer register. Set the timing of toggling the PWM3 output level during increment operation (see **12.4.2 Buffer operation**).	

(Caution and Remark are listed on the next page.)

Caution **To generate asymmetric waveforms, also set the registers described in 12.3.24 Timer RD extended compare registers B0, D0, A1, B1, C1, and D1 (TRDCMPB0, TRDCMPD0, TRDCMPA1, TRDCMPB1, TRDCMPC1, and TRDCMPD1).**
To modify the timing of toggling the PWM output level, specify a value through the TRDGRD0, TRDGRC1, or TRDGRD1 register. The PWM period value cannot be rewritten.
The simultaneous update function is used to transfer data from the buffer registers to the compare registers. After modifying the necessary buffer registers, set the RDT bit in the TRDRDT1 register to 1. The values to be used in the next PWM cycle are transferred from the buffer registers to the compare registers when the timing of simultaneous update comes while the RSF flag in the TRDRSF1 register is 1. While the RSF flag is 0, values are not transferred from the buffer registers to the compare registers.
Set the compare register to a value greater than the dead time and smaller than TRDGRA0 (PWM period) - dead time except when the output duty cycle is set to 0% or 100%.

Remark i = 0 or 1, j = A to D

Figure 12 - 52 Format of Timer RD General Registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi, TRDGRCi, and TRDGRDi) [Timer-KB PWM Output Gating Mode]

Address: F03A8H (TRDGRA0), F03AAH (TRDGRB0), FFF6CH (TRDGRC0), FFF6EH (TRDGRD0), F03B8H (TRDGRA1), F03BAH (TRDGRB1), FFF70H (TRDGRC1), FFF72H (TRDGRD1)

After reset: FFFFH[Note]

R/W: 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 12 - 10 TRDGRji Register Functions in Timer-KB PWM Output Gating Mode**.

Note The value after reset is undefined when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H) and the TRD0EN bit is 0 in the PER2 register. To read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1 before reading.

Table 12 - 10 TRDGRji Register Functions in Timer-KB PWM Output Gating Mode

Register	Register Function	PWM Output Pin
TRDGRA0	General register. The PWM period for TRD0 is stored.	—
TRDGRB0	General register. The timing of toggling the PWM output level is stored.	TRDIOB0 TRDTKBOUT4
TRDGRA1	General register. The PWM period for TRD1 is stored.	—
TRDGRB1	General register. The timing of toggling the PWM output level is stored.	TRDIOB1 TRDTKBOUT0
TRDGRC0	Buffer register. Set the PWM period for TRD0.	—
TRDGRD0	Buffer register. Set the timing of toggling the PWM output level.	TRDIOB0 TRDTKBOUT4
TRDGRC1	Buffer register. Set the PWM period for TRD1.	—
TRDGRD1	Buffer register. Set the timing of toggling the PWM output level.	TRDIOB1 TRDTKBOUT0

Caution **To modify the timing of toggling the PWM output level, specify a value through the TRDGRD0, TRDGRC1, or TRDGRD1 register. The PWM period value cannot be rewritten.**
The simultaneous update function is used to transfer data from the buffer registers to the compare registers. After modifying the necessary buffer registers, set the RDT bit in the TRDRDTi register to 1. The values to be used in the next PWM cycle are transferred from the buffer registers to the compare registers when the timing of simultaneous update comes while the RSF flag in the TRDRSFi register is 1. While the RSF flag is 0, values are not transferred from the buffer registers to the compare registers.
For further details other than those above on the settings of the TRDTKBOUT pins, see 12.3.24 Timer RD extended compare registers B0, D0, A1, B1, C1, and D1 (TRDCMPB0, TRDCMPD0, TRDCMPA1, TRDCMPB1, TRDCMPC1, and TRDCMPD1).

Remark i = 0 or 1, j = A to D

12.3.24 Timer RD extended compare registers B0, D0, A1, B1, C1, and D1 (TRDCMPB0, TRDCMPD0, TRDCMPA1, TRDCMPB1, TRDCMPC1, and TRDCMPD1)

[Extended PWM Mode]

Access the TRDCMPm register in 16-bit units. Do not access it in 8-bit units.

[Extended Complementary PWM Mode]

Access the TRDCMPm register in 16-bit units. Do not access it in 8-bit units.

[Timer-KB PWM Output Gating Mode]

Access the TRDCMPm register in 16-bit units. Do not access it in 8-bit units.

Figure 12 - 53 Format of Timer RD Extended Compare Register (TRDCMPm) [Extended PWM Mode]

Address: F03C0H (TRDCMPB0), FFF74H (TRDCMPD0)
F03C4H (TRDCMPA1), FFF76H (TRDCMPC1)
F03C8H (TRDCMPB1), FFF78H (TRDCMPD1)

After reset: FFFFH

R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDCMPm																

—	Function
Bits 15 to 0	See **Table 12 - 11 TRDCMPm Register Functions in Extended PWM Mode**.

Table 12 - 11 TRDCMPm Register Functions in Extended PWM Mode

Register	Register Function	PWM Output Pin
TRDCMPB0	General register. Set the timing of toggling the PWM output level.	TRDIOD0
TRDCMPD0	Buffer register. Set the timing of toggling the PWM output level in the next PWM cycle.	
TRDCMPA1	These registers are not used in extended PWM mode.	—
TRDCMPC1		
TRDCMPB1	General register. Set the timing of toggling the PWM output level.	TRDIOD1
TRDCMPD1	Buffer register. Set the timing of toggling the PWM output level in the next PWM cycle.	

Caution 1. To modify the timing of toggling the PWM output level, specify a value through the TRDCMPD0 or TRDCMPD1 register.

Caution 2. The simultaneous update function is used to transfer data from the buffer registers to the compare registers. After modifying the necessary buffer registers, set the RDT bit in the TRDRDTi register to 1. The values to be used in the next PWM cycle are transferred from the buffer registers to the compare registers when the timing of simultaneous update comes while the RSF flag in the TRDRSFi register is 1. While the RSF flag is 0, values are not transferred from the buffer registers to the compare registers.

Remark m = B0, D0, A1, B1, C1, or D1

Figure 12 - 54 Format of Timer RD Extended Compare Register (TRDCMPm) [Extended Complementary PWM Mode]

Address: F03C0H (TRDCMPB0), FFF74H (TRDCMPD0)
F03C4H (TRDCMPA1), FFF76H (TRDCMPC1)
F03C8H (TRDCMPB1), FFF78H (TRDCMPD1)

After reset: FFFFH

R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDCMPm																

—	Function
Bits 15 to 0	See **Table 12 - 12 TRDCMPm Register Functions in Extended Complementary PWM Mode**.

Table 12 - 12 TRDCMPm Register Functions in Extended Complementary PWM Mode

Register	Setting	Register Function	PWM Output Pin
TRDCMPB0	CPSS = 1	General register. The timing of toggling the PWM1 output level during decrement operation is stored.	TRDIOB0 TRDIOD0
TRDCMPD0		Buffer register. Set the timing of toggling the PWM1 output level during decrement operation.	
TRDCMPA1		General register. The timing of toggling the PWM2 output level during decrement operation is stored.	TRDIOA1 TRDIOC1
TRDCMPC1		Buffer register. Set the timing of toggling the PWM2 output level during decrement operation.	
TRDCMPB1		General register. The timing of toggling the PWM3 output level during decrement operation is stored.	TRDIOB1 TRDIOD1
TRDCMPD1		Buffer register. Set the timing of toggling the PWM3 output level during decrement operation.	

Caution 1. To modify the timing of toggling the PWM output level, specify a value through the TRDCMPD0, TRMCMPC1, or TRDCMPD1 register.

Caution 2. Writing to the TRDCMPm register is ignored while the CPSS bit is 0.

Caution 3. To output a symmetric waveform by setting the CPSS bit to 1, set the TRDCMPm register to the same value as the TRDGRji register.

Caution 4. The simultaneous update function is used to transfer data from the buffer registers to the compare registers. After modifying the necessary buffer registers, set the RDT bit in the TRDRDTi register to 1. The values to be used in the next PWM cycle are transferred from the buffer registers to the compare registers when the timing of simultaneous update comes while the RSF flag in the TRDRSFi register is 1. While the RSF flag is 0, values are not transferred from the buffer registers to the compare registers.

Caution 5. Set the compare register to a value greater than the dead time and smaller than TRDGRA0 (PWM period) - dead time except when the output duty cycle is set to 0% or 100%.

Remark m = B0, D0, A1, B1, C1, or D1
CPSS: Bit in the TRDFCR register

Figure 12 - 55 Format of Timer RD Extended Compare Register (TRDCMPm) [Timer-KB PWM Output Gating Mode]

Address: F03C0H (TRDCMPB0), FFF74H (TRDCMPD0)
F03C4H (TRDCMPA1), FFF76H (TRDCMPC1)
F03C8H (TRDCMPB1), FFF78H (TRDCMPD1)

After reset: FFFFH

R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDCMPm																

—	Function
Bits 15 to 0	See **Table 12 - 13 TRDCMPm Register Functions in Timer-KB PWM Output Gating Mode**.

Table 12 - 13 TRDCMPm Register Functions in Timer-KB PWM Output Gating Mode

Register	Register Function	PWM Output Pin
TRDCMPB0	General register. The timing of toggling the PWM output level is stored.	TRDIOD0 TRDTKBOUT5
TRDCMPD0	Buffer register. Set the timing of toggling the PWM output level.	
TRDCMPA1	General register. The timing of toggling the PWM output level is stored.	TRDIOA1 TRDTKBOUT3
TRDCMPC1	Buffer register. Set the timing of toggling the PWM output level.	
TRDCMPB1	General register. The timing of toggling the PWM output level is stored.	TRDIOD1 TRDTKBOUT2
TRDCMPD1	Buffer register. Set the timing of toggling the PWM output level.	

Caution 1. To modify the timing of toggling the PWM output level, specify a value through the TRDCMPD0, TRMCMPC1, or TRDCMPD1 register.

Caution 2. The simultaneous update function is used to transfer data from the buffer registers to the compare registers. After modifying the necessary buffer registers, set the RDT bit in the TRDRDTi register to 1. The values to be used in the next PWM cycle are transferred from the buffer registers to the compare registers when the timing of simultaneous update comes while the RSF flag in the TRDRSFi register is 1. While the RSF flag is 0, values are not transferred from the buffer registers to the compare registers.

Remark m = B0, D0, A1, B1, C1, or D1

12.3.25 Timer RD A/D conversion trigger compare register 0/timer-KB PWM output gating mode compare register (TRDADTC0/TRDCMPE1)

[Extended Complementary PWM Mode]

Access the TRDADTC0 register in 16-bit units. Do not access it in 8-bit units.

[Timer-KB PWM Output Gating Mode]

Access the TRDCMPE1 register in 16-bit units. Do not access it in 8-bit units.

Figure 12 - 56 Format of Timer RD A/D Conversion Trigger Compare Register 0 (TRDADTC0) [Extended Complementary PWM Mode]

Address: F03CCH (TRDADTC0)
After reset: FFFFH
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDADTC0																

—	Function
Bits 15 to 0	• The timing of generating A/D conversion trigger 0 is stored. • Setting range (when the TRDADCR.ADMD0 bit is 1): 0000H ≤ (TRDGRA0 register setting – TRD0 counter setting + 1) • Setting range (when the TRDADCR.ADMD0 bit is 0): 0000H ≤ (TRDGRA0 register setting – TRD0 counter setting)

Caution 1. The TRDADTC0 register setting is valid when the ADE0 bit in the TRDADCR register is 1.
Caution 2. To set up the TRDADTC0 register after counting begins, specify a value through the TRDADTB0 register.

Figure 12 - 57 Format of Timer-KB PWM Output Gating Mode Compare Register (TRDCMPE1) [Timer-KB PWM Output Gating Mode]

Address: F03CCH (TRDCMPE1)
After reset: FFFFH
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDCMPE1																

—	Function	PWM Output Pin
Bits 15 to 0	General register. The timing of toggling the PWM output level is stored.	TRDIOC1 TRDTKBOUT1

Caution To modify the timing of toggling the PWM output level, specify a value through the TRDCMPF1 register.

12.3.26 Timer RD A/D conversion trigger buffer register 0/timer-KB PWM output gating mode buffer register (TRDADTB0/TRDCMPF1)

[Extended Complementary PWM Mode]

Access the TRDADTB0 register in 16-bit units. Do not access it in 8-bit units.

[Timer-KB PWM Output Gating Mode]

Access the TRDCMPF1 register in 16-bit units. Do not access it in 8-bit units.

Figure 12 - 58 Format of Timer RD A/D Conversion Trigger Buffer Register 0 (TRDADTB0) [Extended Complementary PWM Mode]

Address: FFF7AH (TRDADTB0)
After reset: FFFFH
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDADTB0																

—	Function
Bits 15 to 0	• A/D conversion trigger buffer register 0 • Setting range (when the TRDADCR.ADMD0 bit is 1): 0000H ≤ (TRDGRA0 register setting – TRD0 counter setting + 1) • Setting range (when the TRDADCR.ADMD0 bit is 0): 0000H ≤ (TRDGRA0 register setting – TRD0 counter setting)

Caution 1. The simultaneous update function is used to transfer data from the buffer registers to the compare registers. After modifying the necessary buffer registers, set the RDT bit in the TRDRDT1 register to 1. The values to be used in the next PWM cycle are transferred from the buffer registers to the compare registers when the timing of simultaneous update comes while the RSF flag in the TRDRSF1 register is 1. While the RSF flag is 0, values are not transferred from the buffer registers to the compare registers.

Caution 2. Output at the same time as interrupt request 0 requires setting the ADMD0 bit to 0 and this register to the result of TRDGRA0 register setting – TRD0 counter setting.
Output at the same time as interrupt request 1 requires setting the ADMD0 bit to 1 and this register to 0000H.

Figure 12 - 59 Format of Timer-KB PWM Output Gating Mode Buffer Register (TRDCMPF1) [Timer-KB PWM Output Gating Mode]

Address: FFF7AH (TRDCMPF1)
After reset: FFFFH
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDCMPF1																

—	Function	PWM Output Pin
Bits 15 to 0	Buffer register. Set the timing of toggling the PWM output level.	TRDIOC1 TRDTKBOUT3

Caution 1. To modify the timing of toggling the PWM output level in the TRDCMPE1 register, specify a value through the TRDCMPF1 register.

Caution 2. The simultaneous update function is used to transfer data from the buffer registers to the compare registers. After modifying the necessary buffer registers, set the RDT bit in the TRDRDTi register to 1. The values to be used in the next PWM cycle are transferred from the buffer registers to the compare registers when the timing of simultaneous update comes while the RSF flag in the TRDRSFi register is 1. While the RSF flag is 0, values are not transferred from the buffer registers to the compare registers.

12.3.27 Timer RD A/D conversion trigger compare register 1 (TRDADTC1)

[Extended Complementary PWM Mode]

Access the TRDADTC1 register in 16-bit units. Do not access it in 8-bit units.

Figure 12 - 60 Format of Timer RD A/D Conversion Trigger Compare Register 1(TRDADTC1) [Extended Complementary PWM Mode]

Address: F03D0H
After reset: FFFFH
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDADTC1																

—	Function
Bits 15 to 0	• The timing of generating A/D conversion trigger 1 is stored. • Setting range (when the TRDADCR.ADMD1 bit is 1): 0000H ≤ (TRDGRA0 register setting – TRD0 counter setting + 1) • Setting range (when the TRDADCR.ADMD1 bit is 0): 0000H ≤ (TRDGRA0 register setting – TRD0 counter setting)

Caution 1. The TRDADTC1 register setting is valid when the ADE1 bit in the TRDADCR register is 1.

Caution 2. To set up the TRDADTC1 register after counting begins, specify a value through the TRDADTB1 register.

12.3.28 Timer RD A/D conversion trigger buffer register 1 (TRDADTB1)

[Extended Complementary PWM Mode]

Access the TRDADTB1 register in 16-bit units. Do not access it in 8-bit units.

Figure 12 - 61 Format of Timer RD A/D Conversion Trigger Buffer Register 1 (TRDADTB1) [Extended Complementary PWM Mode]

Address: FFF7CH
After reset: FFFFH
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDADTB1																

—	Function
Bits 15 to 0	• A/D conversion trigger buffer register 1 • Setting range (when the TRDADCR.ADMD1 bit is 1): 0000H ≤ (TRDGRA0 register setting – TRD0 counter setting + 1) • Setting range (when the TRDADCR.ADMD1 bit is 0): 0000H ≤ (TRDGRA0 register setting – TRD0 counter setting)

Caution 1. The simultaneous update function is used to transfer data from the buffer registers to the compare registers. After modifying the necessary buffer registers, set the RDT bit in the TRDRDT1 register to 1. The values to be used in the next PWM cycle are transferred from the buffer registers to the compare registers when the timing of simultaneous update comes while the RSF flag in the TRDRSF1 register is 1. While the RSF flag is 0, values are not transferred from the buffer registers to the compare registers.

Caution 2. Output at the same time as interrupt request 0 requires setting the ADMD1 bit to 0 and this register to the result of TRDGRA0 register setting – TRD0 counter setting.
Output at the same time as interrupt request 1 requires setting the ADMD1 bit to 1 and this register to 0000H.

12.3.29 Timer RD simultaneous update trigger register 0 (TRDRDT0)

The TRDRDT0 register can be set by an 8-bit memory manipulation instruction. Access to this register by a 16-bit memory manipulation instruction is also possible when it is in use in combination with the TRDRDT1 register as the TRDRDT register.

Figure 12 - 62 Format of Timer RD Simultaneous Update Trigger Register 0 (TRDRDT0) [Extended PWM Mode and Timer-KB PWM Output Gating Mode]

Address: FFF7EH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TRDRDT0	0	0	0	0	0	0	0	RDT

RDT	Simultaneous update trigger
0	Writing to this bit is ignored.Note 1
1	Simultaneous update is enabled.Note 2

Note 1. This bit is always read as 0.

Note 2. In extended PWM mode, TRDGRA0, TRDGRB0, and TRDCMPB0 are updated simultaneously.
In timer-KB PWM output gating mode, the compare registers for the TRD0 counter — that is, TRDGRA0, TRDGRB0, and TRDCMPB0 — are updated simultaneously.

12.3.30 Timer RD simultaneous update trigger register 1 (TRDRDT1)

The TRDRDT1 register can be set by an 8-bit memory manipulation instruction. Access to this register by a 16-bit memory manipulation instruction is also possible when it is in use in combination with the TRDRDT0 register as the TRDRDT register.

Figure 12 - 63 Format of Timer RD Simultaneous Update Trigger Register 1 (TRDRDT1) [Extended PWM Mode, Extended Complementary PWM Mode, and Timer-KB PWM Output Gating Mode]

Address: FFF7FH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TRDRDT1	0	0	0	0	0	0	0	RDT

RDT	Simultaneous update trigger
0	Writing to this bit is ignored.Note 1
1	Simultaneous update is enabled.Note 2

Note 1. This bit is always read as 0.

Note 2. In extended complementary PWM mode, all compare registers are updated simultaneously.
In extended PWM mode, the TRDGRA1, TRDGRB1, and TRDCMPB1 registers are updated simultaneously.
In timer-KB PWM output gating mode, the compare registers for the TRD1 counter — that is, TRDGRA1, TRDGRB1, TRDCMPB1, TRDCMPA1, and TRDCMPE1 — are updated simultaneously.

12.3.31 Timer RD simultaneous update flag register 0 (TRDRSF0)

The TRDRSF0 register can be set by an 8-bit memory manipulation instruction. Access to this register by a 16-bit memory manipulation instruction is also possible when it is in use in combination with the TRDRSF1 register as the TRDRSF register.

Figure 12 - 64 Format of Timer RD Simultaneous Update Flag Register 0 (TRDRSF0) [Extended PWM Mode and Timer-KB PWM Output Gating Mode]

Address: F03D6H
After reset: 00H
R/W: R

Symbol	7	6	5	4	3	2	1	0
TRDRSF0	0	0	0	0	0	0	0	RSF

RSF	Simultaneous update waiting flag
0	Simultaneous update is enabled. [Source for setting to 0] The timing of simultaneous update or the timing of starting counting comes.
1	Timer RD2 is waiting for the end of simultaneous update.**Note** [Source for setting to 1] When the RDT bit in the TRDRDT0 register is set to 1.

Note In extended PWM mode, TRDGRA0, TRDGRB0, and TRDCMPB0 are updated simultaneously.
In timer-KB PWM output gating mode, the compare registers for the TRD0 counter — that is, TRDGRA0, TRDGRB0, and TRDCMPB0 — are updated simultaneously.

12.3.32 Timer RD simultaneous update flag register 1 (TRDRSF1)

The TRDRSF1 register can be set by an 8-bit memory manipulation instruction. Access to this register by a 16-bit memory manipulation instruction is also possible when it is in use in combination with the TRDRSF0 register as the TRDRSF register.

Figure 12 - 65 Format of Timer RD Simultaneous Update Flag Register 1 (TRDRSF1) [Extended PWM Mode, Extended Complementary PWM Mode, and Timer-KB PWM Output Gating Mode]

Address: F03D7H
After reset: 00H
R/W: R

Symbol	7	6	5	4	3	2	1	0
TRDRSF1	0	0	0	0	0	0	0	RSF

RSF	Simultaneous update waiting flag
0	Simultaneous update is enabled. [Source for setting to 0] The timing of simultaneous update or the timing of starting counting comes.
1	Timer RD2 is waiting for the end of simultaneous update.**Note** [Source for setting to 1] When the RDT bit in the TRDRDT1 register is set to 1.

Note In extended PWM mode, the TRDGRA1, TRDGRB1, and TRDCMPB1 registers are updated simultaneously.
In extended complementary PWM mode, all compare registers are updated simultaneously.
In timer-KB PWM output gating mode, the compare registers for the TRD1 counter — that is, TRDGRA1, TRDGRB1, TRDCMPB1, TRDCMPA1, and TRDCMPE1 — are updated simultaneously.

12.3.33 Timer RD A/D conversion trigger control register (TRDADCR)

Figure 12 - 66 Format of Timer RD A/D Conversion Trigger Control Register (TRDADCR) [Extended Complementary PWM Mode]

Address: F03D8H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TRDADCR	0	0	ADMD1	ADE1	0	0	ADMD0	ADE0

ADMD1	Bit 1 for selecting the A/D conversion trigger output mode
0	Values are compared during increment operation of the counter.
1	Values are compared during decrement operation of the counter.

ADE1	Bit 1 for enabling the output of A/D conversion triggers
0	Output of A/D conversion triggers is disabled.
1	Output of A/D conversion triggers is enabled.

ADMD0	Bit 0 for selecting the A/D conversion trigger output mode
0	Values are compared during increment operation of the counter.
1	Values are compared during decrement operation of the counter.

ADE0	Bit 0 for enabling the output of A/D conversion triggers
0	Output of A/D conversion triggers is disabled.
1	Output of A/D conversion triggers is enabled.

12.3.34 Timer RD skipping control register (TRDTCTL)

The TRDTCTL register is used to control the skipping of interrupt or A/D trigger signals output from timer RD2. The TRDTCTL register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 12 - 67 Format of Timer RD Skipping Control Register (TRDTCTL)

Address: F0498H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TRDTCTL	0	0	0	TRDTMD	ADTEN1	ADTEN0	INTEN1	INTEN0

TRDTMD	Enabling the output in the first cycle after timer RD2 begins operation**Note**
0	Output is disabled.
1	Output is enabled.

ADTEN1	Enabling the skipping of A/D conversion trigger 1**Note**
0	Skipping is disabled.
1	Skipping is enabled.

ADTEN0	Enabling the skipping of A/D conversion trigger 0**Note**
0	Skipping is disabled.
1	Skipping is enabled.

INTEN1	Enabling the skipping of interrupt requests from timer RD21**Note**
0	Skipping is disabled.
1	Skipping is enabled.

INTEN0	Enabling the skipping of interrupt requests from timer RD20**Note**
0	Skipping is disabled.
1	Skipping is enabled.

Note This bit can only be used in extended complementary PWM mode.
Clear the bit to 0 (disabled) when the timer is not set to extended complementary PWM mode.

Caution 1. Do not modify the TRDTCTL register while timer RD2 is operating.
Caution 2. Be sure to clear bits 7 to 5 to 0.

12.3.35 Timer RD skipping count setting register (TRDTCMP)

The TRDTCMP register is used to control the skipping of interrupt or A/D trigger signals output form timer RD2. The TRDTCMP register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 12 - 68 Format of Timer RD Skipping Count Setting Register (TRDTCMP)

Address: F0499H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TRDTCMP	0	0	0	TRDTCMP[4:0]				

TRDTCMP[4:0]					Number of times the output is to be skipped
0	0	0	0	0	None
0	0	0	0	1	One time (Interrupt or A/D conversion trigger signals are not output in one cycle after the previous interrupt or A/D trigger signal is output.)
⋮					⋮
1	1	1	1	0	30 times (Interrupt or A/D conversion trigger signals are not output in 30 cycles after the previous interrupt or A/D trigger signal is output.)
1	1	1	1	1	31 times (Interrupt or A/D conversion trigger signals are not output in 31 cycles after the previous interrupt or A/D trigger signal is output.)

Caution 1. Do not modify the TRDTCMP register while timer RD2 is operating.
Caution 2. Be sure to clear bits 7 to 5 to 0.

12.3.36 Timer RD output port mask enable register (TRDPOE)

The TRDPOE register is used to control the output through the TRDIOC0 pin of timer RD2. To use a pin function multiplexed with TRDIOC0 instead of outputting the timer signal through the pin in complementary PWM mode or extended complementary PWM mode, write 1 to the EC0 bit in the TRDPOE register. The TRDPOE register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 12 - 69 Format of Timer RD Output Port Mask Enable Register (TRDPOE)

Address: F02ACH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TRDPOE	0	0	0	0	0	0	0	EC0

EC0	Masking the output through the TRDIOC0 pin
0	Output is enabled.
1	Output is disabled. (The TRDIOC0 pin functions as an I/O port.)

Remark For the pin functions multiplexed with the TRDIOC0 signal, see **2.1 Functions of Port Pins**.

12.3.37 Registers for controlling the port functions multiplexed with the inputs and outputs of timer RD2

Set the following registers to control the port functions multiplexed with the inputs and outputs of timer RD2.

- Port mode registers xx (PMxx)
- Port registers xx (Pxx)
- Port mode control A registers xx (PMCAxx)

For details, see the following sections.

- **7.3.1 Port mode registers xx (PMxx)**
- **7.3.2 Port registers xx (Pxx)**
- **7.3.7 Port mode control A registers xx (PMCAxx)**

When the pins multiplexed with TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1, TRDIOC0, TRDIOC1, TRDIOD0, and TRDIOD1 are to be used for outputs of the timer, set the following register bits corresponding to each port to 0.

- Port mode registers xx (PMxx)
- Port registers xx (Pxx)
- Port mode control A registers xx (PMCAxx)

Example: When using P10/TRDIOD1 for timer output, make the following settings.

- Set the PMCA10 bit of port mode control A register 1 to 0.
- Set the PM10 bit of port mode register 1 to 0.
- Set the P10 bit of port register 1 to 0.

When the pins multiplexed with TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1, TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1, and TRDCLK are to be used for inputs of the timer, set the port mode register xx (PMxx) bit corresponding to each port to 1. Furthermore, set the port mode control A register xx (PMCAxx) bit corresponding to each port to 0. The corresponding bit in the port register xx (Pxx) can be 0 or 1.

Example: When using P10/TRDIOD1 for timer input, make the following settings.

- Set the PMCA10 bit of port mode control A register 1 to 0.
- Set the PM10 bit of port mode register 1 to 1.
- Set the P10 bit of port register 1 to 0.

Remark xx = 1, 3, 5, 7

12.4 Items Common to Multiple Modes

12.4.1 Count sources

The count source selection method is the same in all modes except timer-KB PWM output gating mode. However, the external clock cannot be selected in PWM3 mode.

Table 12 - 14 Count Source Selection

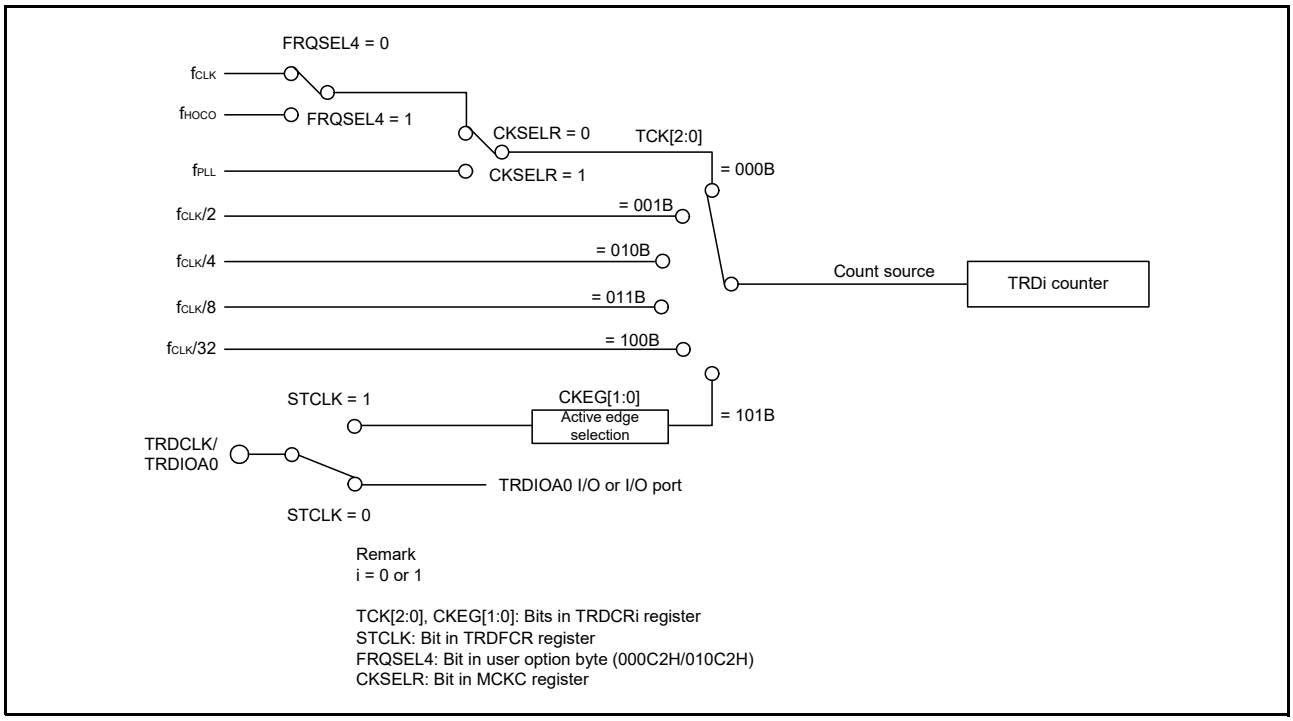
Count Source	Selection
f_{CLK}, f_{HOCO}**Note 1**, f_{PLL}**Note 2**, $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, $f_{CLK}/32$	The count source is selected by bits TCK2 to TCK0 in the TRDCRi register.
External signal input to TRDCLK pin	The STCLK bit in the TRDFCR register is set to 1 (external clock input 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).

Note 1. f_{CLK} is selected when the FRQSEL4 bit is 0 and f_{HOCO} is selected when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H). When selecting f_{HOCO} as the count source for timer RD2, set f_{CLK} to f_{IH} before setting bit 3 (TRD0EN) in peripheral enable register 2 (PER2). When changing f_{CLK} to a clock other than f_{IH}, clear bit 3 (TRD0EN) in peripheral enable register 2 (PER2) before changing.

Note 2. f_{PLL} is supplied when the CKSELR bit is 1 in the main clock control register (MCKC).

Remark i = 0 or 1

Figure 12 - 70 Count Source Block Diagram



Set the pulse width of the external clock applied to the TRDCLK pin to three or more cycles of the timer RD2 operating clock (f_{CLK}).

12.4.2 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 12 - 15** shows the buffer operation in each mode.

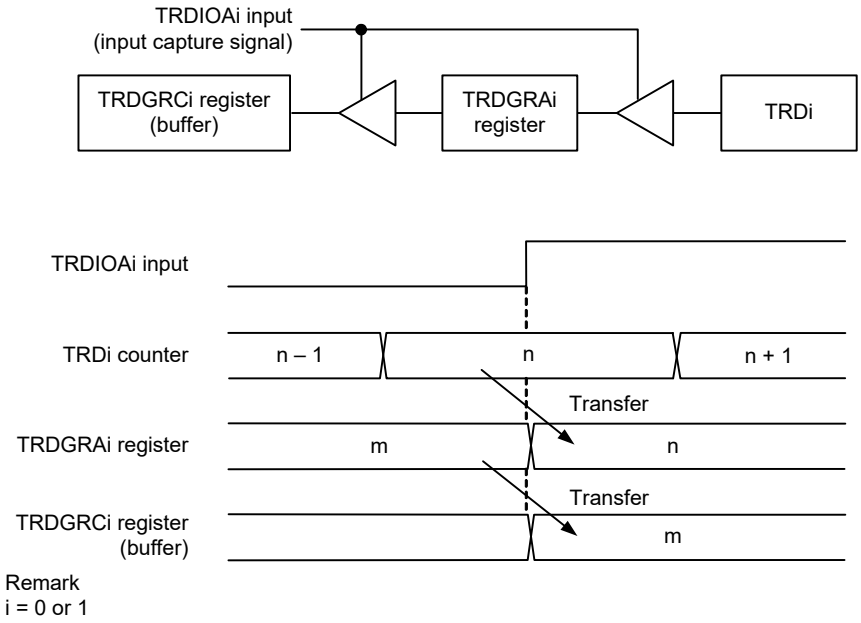
Table 12 - 15 Buffer Operation in Each Mode

Function and Mode		Transfer Timing	Transfer Registers
Timer mode	Input capture function	TRDIOAi input signal (Input capture signal input)	Transfer the content of the TRDGRAi register to the TRDGRCi register (buffer register).
		TRDIOBi input signal (Input capture signal input)	Transfer the content of the TRDGRBi register to the TRDGRDi register (buffer register).
	Output compare function	Compare match between the TRDi counter and TRDGRAi register	Transfer the content of the TRDGRCi register (buffer register) to the TRDGRAi register.
		Compare match between the TRDi counter and TRDGRBi register	Transfer the content of the TRDGRDi register (buffer register) to the TRDGRBi register.
	PWM function	Compare match between the TRDi counter and TRDGRAi register	Transfer the content of the TRDGRCi register (buffer register) to the TRDGRAi register.
		Compare match between the TRDi counter and TRDGRBi register	Transfer the content of the TRDGRDi register (buffer register) to the TRDGRBi register.
Reset synchronous PWM mode		Compare match between the TRD0 counter and TRDGRA0 register	Transfer the content of the TRDGRCi register (buffer register) to the TRDGRAi register. Transfer the content of the TRDGRDi register (buffer register) to the TRDGRBi register.
Complementary PWM mode		• Underflow of the TRD1 counter when the CMD1 and CMD0 bits in the TRDFCR register are 11B • Compare match between the TRD0 counter and TRDGRA0 register when the CMD1 and CMD0 bits in the TRDFCR register are 10B	Transfer the content of the TRDGRC1 register (buffer register) to the TRDGRA1 register. Transfer the content of the TRDGRDi register (buffer register) to the TRDGRBi register.
PWM3 mode		Compare match between the TRD0 counter and TRDGRA0 register	Transfer the content of the TRDGRCi register (buffer register) to the TRDGRAi register. Transfer the content of the TRDGRDi register (buffer register) to the TRDGRBi register.
Extended PWM mode		Compare match between the TRDi counter and TRDGRAi register after the RDT bit in the TRDRDTi register is set to 1	Transfer the contents of the buffer registers to the TRDGRAi, TRDGRBi, and TRDCMPBi registers.
Extended complementary PWM mode		Underflow of the TRD1 counter (earlier than the actual underflow timing by the amount of dead time) after the RDT bit in the TRDRDTi register is set to 1	Transfer the contents of the buffer registers to the TRDGRB0, TRDGRA1, and TRDGRB1 registers.
		Underflow of the TRD1 counter after the RDT bit in the TRDRDTi register is set to 1	Transfer the contents of the buffer registers to the TRDCMPB0, TRDCMPA1, and TRDCMPB1 registers.**Note**
		Underflow of the TRD1 counter after the RDT bit in the TRDRDTi register is set to 1	Transfer the contents of the buffer registers to the TRDADTC0 and TRDADTC1 registers.
Timer-KB PWM output gating mode		Compare match between the TRDi counter and TRDGRAi register after the RDT bit in the TRDRDTi register is set to 1	Transfer the contents of the buffer registers to the TRDGRAj, TRDGRBj, TRDCMPB0, TRDCMPA1, TRDCMPB1, TRDCMPE1 registers.

Note Values are transferred to the registers only when TRDFCR.CPSS = 1.

Remark i = 0 or 1

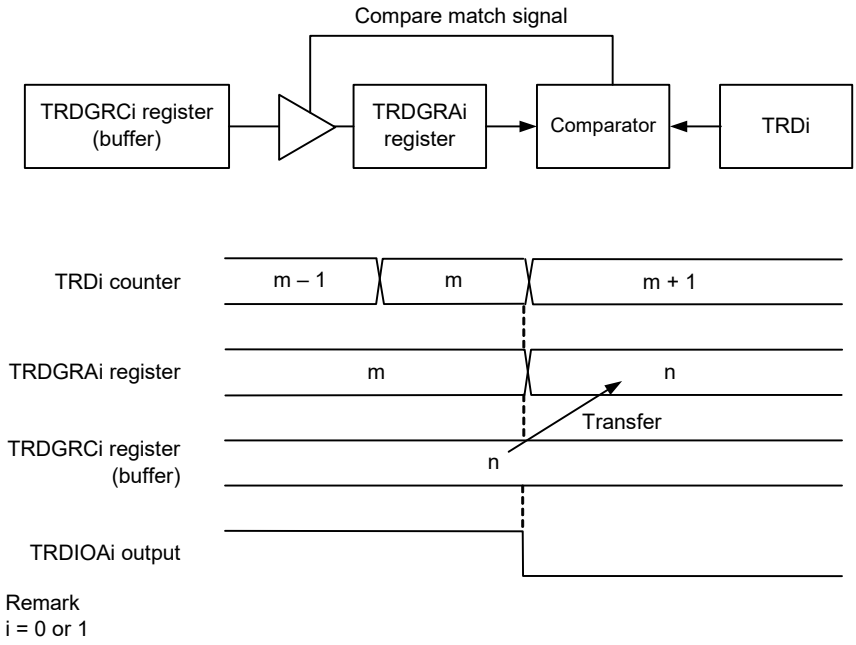
Figure 12 - 71 Buffer Operation in Input Capture Function



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 IOA[2:0] in the TRDIORAi register are set to 100B (input capture at the rising edge).

Figure 12 - 72 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 IOA[2:0] in the TRDIORAi register are set to 001B (low-level output by compare match).

Perform the following in 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 or IMFD flag in the TRDSRi register is set to 1 at the input edge of the TRDIOCi pin or TRDIODi pin.

In the output compare function, PWM function, reset synchronous PWM mode, complementary PWM mode, and PWM3 mode, when the TRDGRCi or TRDGRDi register is used as a buffer register, the IMFC or IMFD flag in the TRDSRi register is set to 1 by a compare match with the TRDi counter.

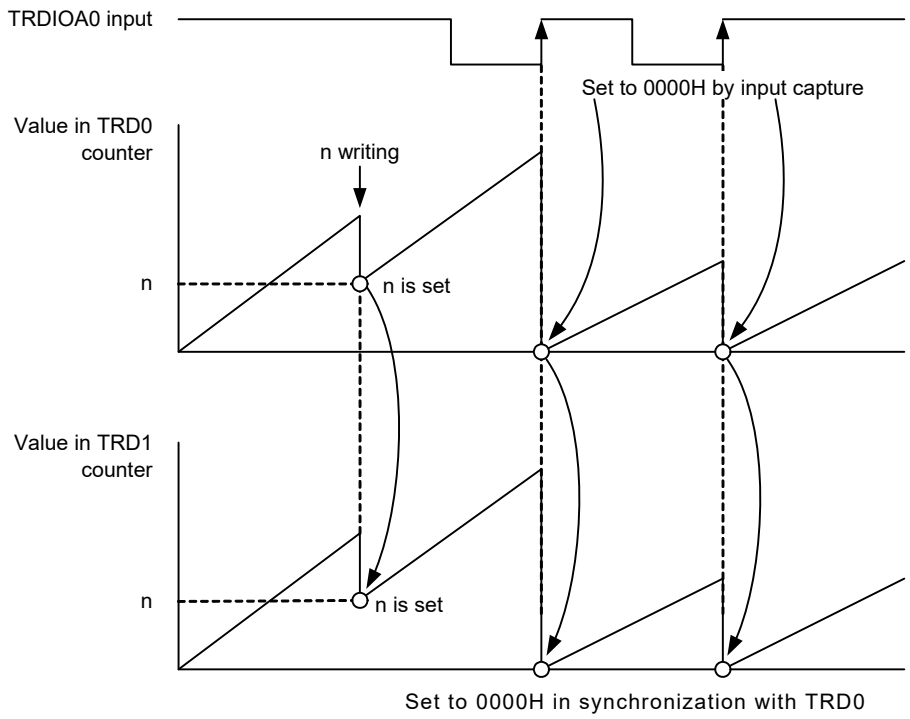
In the extended PWM mode, extended complementary PWM mode, and timer-KB PWM output gating mode, the simultaneous update function is used for buffer operation. For the simultaneous update function, see **12.4.7 Simultaneous update of compare registers**.

12.4.3 Synchronous operation

The TRD1 counter is synchronized with the TRD0 counter.

- 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 counters after writing to the TRDi counter.
- Synchronous clear
  When the TRDSYNC bit is 1 and bits CCLR2 to CCLR0 in the TRDCR0 register are 011B (synchronous clear), the TRD0 counter is set to 0000H at the same time as the TRD1 counter is set to 0000H.
  Likewise, when the TRDSYNC bit is 1 and bits CCLR2 to CCLR0 in the TRDCR1 register are 011B (synchronous clear), the TRD1 counter is set to 0000H at the same time as the TRD0 counter is set to 0000H.

Figure 12 - 73 Synchronous Operation



The above diagram applies under the following conditions:

- The TRDSYNC bit in the TRDMR register is set to 1 (synchronous operation).
- Bits CCLR[2:0] in the TRDCR0 register are set to 001B (TRD0 is set to 0000H by input capture).
  Bits CCLR[2:0] in the TRDCR1 register are set to 011B (TRD1 is set to 0000H in synchronization with TRD0).
- Bits IOA[2:0] in the TRDIORA0 register are set to 100B.
- Bits CMD[1:0] 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)

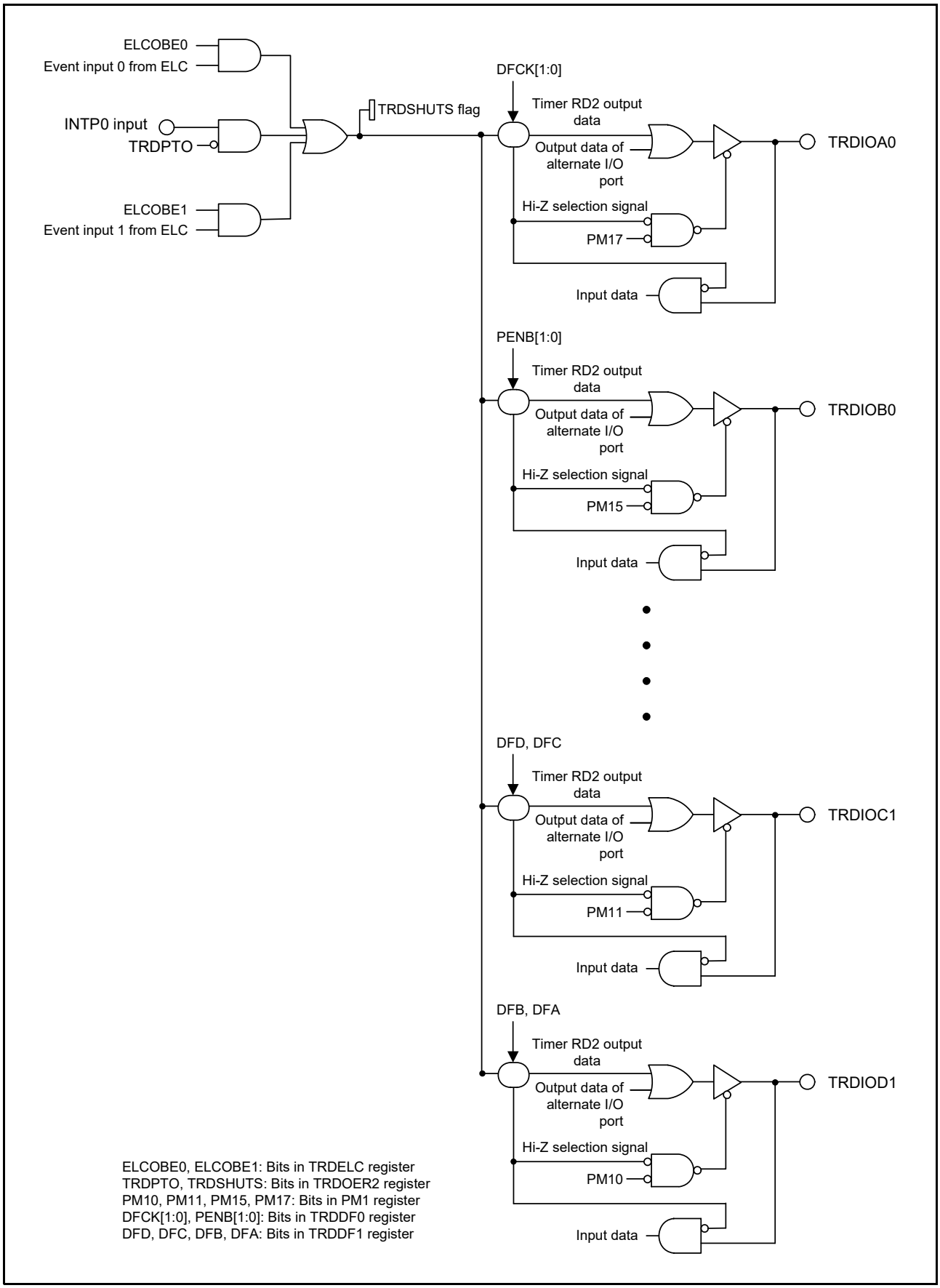
12.4.4 Pulse output forced cutoff

In the PWM function, reset synchronous PWM mode, complementary PWM mode, PWM3 mode, extended PWM mode, extended complementary PWM mode, and timer-KB PWM output gating mode, the pulse output from the TRDIOji output pin (i = 0 or 1, j = A to D) can be cut off by the INTP0 pin input. The pins used for output in these functions or modes can function as the output pin of timer RD2 when the corresponding bit in the TRDOER1 register is set to 0 (timer RD2 output enabled). When the TRDPTO bit in the TRDOER2 register is 1 (the input of the pulse output forced cutoff signal to the INTP0 pin is enabled), the output pin used as a timer RD2 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:

- Use the TRDDFi register to set the pin state (high impedance, low output, or high output) when the pulse output is forcibly cut off.
- Refer to **12.4.5 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 flag in the TRDOER2 register is set to 1. To suspend the forced cutoff of the pulse output, set the TRDSHUTS flag to 0 while the count is stopped (TSTARTi = 0).
- Set the TRDPTO bit in the TRDOER2 register to 1 (the input of the pulse output forced cutoff signal to the INTP0 pin is enabled).

Figure 12 - 74 Pulse Output Forced Cutoff



12.4.5 Event input from event link controller (ELC)

Timer RD2 performs two operations by event input from the ELC.

a) TRDIOD0/TRDIOD1 input capture

Timer RD2 captures the TRDIOD0/TRDIOD1 input when an event is input from the ELC. The IMFD flag 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 the other modes (the output compare function in timer mode, PWM function, reset synchronous PWM mode, complementary PWM mode, PWM3 mode, extended PWM mode, extended complementary PWM mode, and timer-KB PWM output gating mode).

b) Pulse output forced cutoff operation**Note**

The pulse output is forcibly cut off by event input from the ELC. To use this function, select a mode for outputting pulses (PWM function, reset synchronous PWM mode, complementary PWM mode, PWM3 mode, extended PWM mode, extended complementary PWM mode, or timer-KB PWM output gating mode) and set the ELCOBE0 or ELCOBE1 bit to 1. This function is disabled for the input capture function in timer mode.

Note The forced cutoff function using the input from the INTP0 pin cuts output pulses off while the low level is input through the pin, but the forced cutoff function using ELC events cuts a single output pulse off per event input from the ELC.

[Setting Procedure]

<1> Set timer RD2 as the ELC event link destination.

<2> Set bits ELCICEi and ELCOBEi (i = 0 or 1) to 1 in the TRDELC register.

12.4.6 Event output to event link controller (ELC) and data transfer controller (DTC)

Table 12 - 16 shows the events output to the ELC and DTC in individual timer RD2 modes.
Events are not output to the ELC or DTC in extended complementary PWM mode or timer-KB PWM output gating mode.

Table 12 - 16 Events Output to ELC and DTC in Individual Timer RD2 Modes

Mode	Output Source	ELC	DTC
Input capture function	TRDIOA0 edge detection set by bits IOA1 and IOA0 in the TRDIORA0 register	Output	Output
	TRDIOB0 edge detection set by bits IOB1 and IOB0 in the TRDIORA0 register	Output	Output
	TRDIOC0 edge detection set by bits IOC1 and IOC0 in the TRDIORC0 register	—	Output
	TRDIOD0 edge detection set by bits IOD1 and IOD0 in the TRDIORC0 register	—	Output
	TRDIOA1 edge detection set by bits IOA1 and IOA0 in the TRDIORA1 register	Output	Output
	TRDIOB1 edge detection set by bits IOB1 and IOB0 in the TRDIORA1 register	Output	Output
	TRDIOC1 edge detection set by bits IOC1 and IOC0 in the TRDIORC1 register	—	Output
	TRDIOD1 edge detection set by bits IOD1 and IOD0 in the TRDIORC1 register	—	Output
Output compare function, PWM function, reset synchronous PWM mode, complementary PWM mode, and PWM3 mode	Compare match between the TRD0 counter and TRDGRA0 register	Output	Output
	Compare match between the TRD0 counter and TRDGRB0 register	Output	Output
	Compare match between the TRD0 counter and TRDGRC0 register	—	Output
	Compare match between the TRD0 counter and TRDGRD0 register	—	Output
	Compare match between the TRD1 counter and TRDGRA1 register	Output	Output
	Compare match between the TRD1 counter and TRDGRB1 register	Output	Output
	Compare match between the TRD1 counter and TRDGRC1 register	—	Output
	Compare match between the TRD1 counter and TRDGRD1 register	—	Output
Extended PWM mode	Compare match between the TRD0 counter and TRDGRA0 register	Output	Output
	Compare match between the TRD0 counter and TRDGRB0 register	Output	Output
	Compare match between the TRD0 counter and TRDCMPB0 register	—	Output
	Compare match between the TRD1 counter and TRDGRA1 register	Output	Output
	Compare match between the TRD1 counter and TRDGRB1 register	Output	Output
	Compare match between the TRD1 counter and TRDCMPB1 register	—	Output
Complementary PWM mode	TRD1 counter underflow	Output	—

12.4.7 Simultaneous update of compare registers

The compare registers are updated simultaneously in extended PWM mode, extended complementary PWM mode, or timer-KB PWM output gating mode.
In extended complementary PWM mode, the compare registers are updated when the TRD1 counter underflows (or earlier than the underflow timing by the amount of dead time). In extended PWM mode or timer-KB PWM output gating mode, the compare registers are updated when a match with the specified period value is detected. Be sure to check that the RSF flag in the TRDRSFi register is 0 before modifying a buffer register. After completing the modification of the necessary buffer registers, set the RDT bit in the TRDRDTi register to 1. Writing 1 to this bit sets the RSF flag in the TRDRSFi register to 1.
When the timing of simultaneous update comes while the RSF flag in the TRDRSFi register is 1, the values to be used in the next PWM cycle are transferred from the buffer registers to the compare registers. While the RSF flag is 0, values are not transferred. For the timing of buffer operation, see **12.4.2 Buffer operation**.
Note that values are always transferred from the buffer registers to the compare registers regardless of the value of the RSF flag in the TRDRSFi register at the beginning of timer operation in extended complementary PWM mode or timer-KB PWM output gating mode.

Use the following procedure to modify the compare registers through the simultaneous update function.

1. Check that the RSF flag in the TRDRSFi register is 0.
2. Specify desired values in the target buffer registers for simultaneous updating.
3. Write 1 to the RDT bit in the TRDRDTi register.
4. The RSF flag in the TRDRSFi register is set to 1. After simultaneous updating is completed, the RSF flag is automatically cleared.

Table 12 - 17 lists the registers to be updated simultaneously.
Figures 12 - 75 to **12 - 77** show operation examples of the simultaneous update function in extended PWM mode, extended complementary PWM mode, and timer-KB PWM output gating mode, respectively.

Remark i = 0 or 1

Table 12 - 17 Registers to Be Updated Simultaneously

Mode	TRDRDT0.RDT	TRDRDT1.RDT
Extended PWM mode	TRDGRA0 TRDGRB0 TRDCMPB0	TRDGRA1 TRDGRB1 TRDCMPB1
Extended complementary PWM mode	—	TRDGRB0 TRDCMPB0 TRDGRA1 TRDCMPA1 TRDGRB1 TRDCMPB1 TRDADTC0 TRDADTC1
Timer-KB PWM output gating mode	TRDGRA0 TRDGRB0 TRDCMPB0	TRDGRA1 TRDGRB1 TRDCMPA1 TRDCMPB1 TRDCMPE1

Figure 12 - 75 Operation Example of Simultaneous Updating in Extended PWM Mode

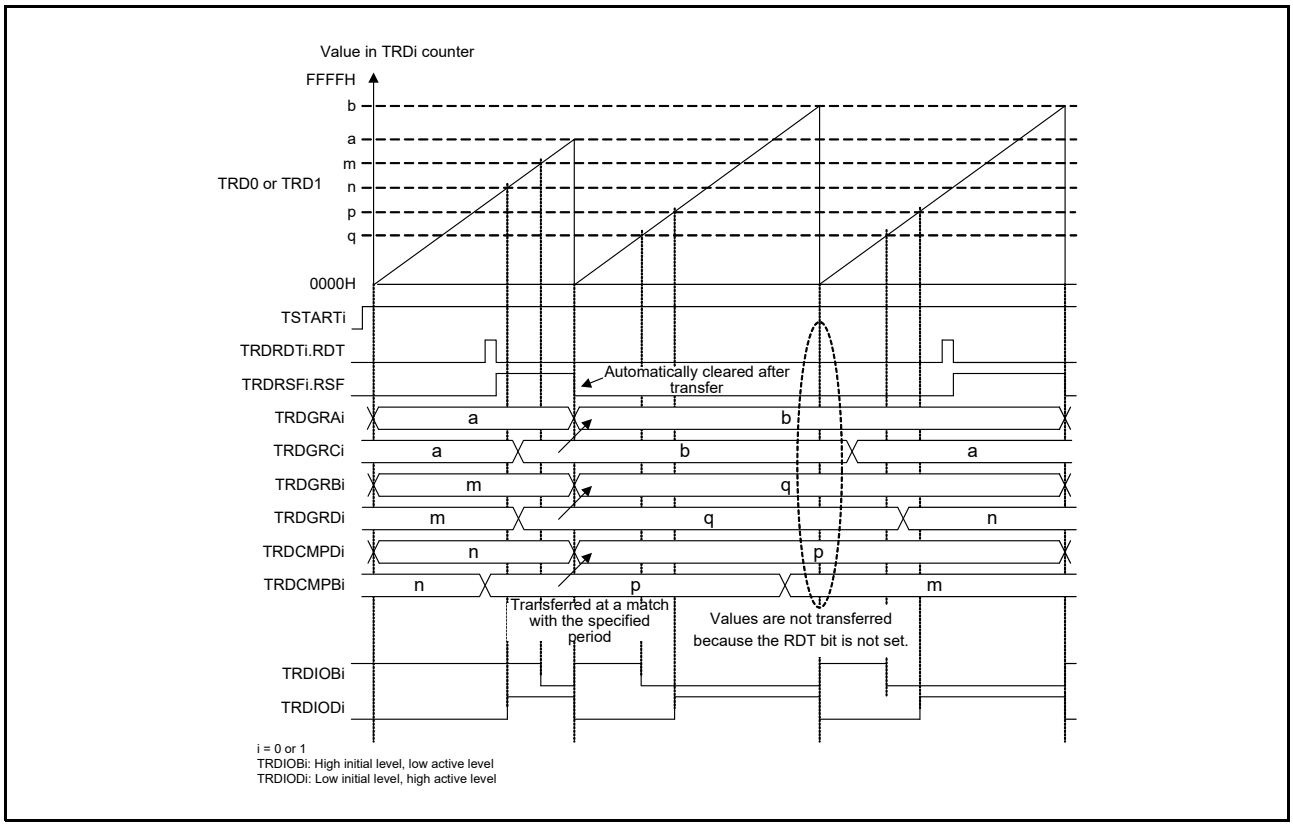


Figure 12 - 76 Operation Example of Simultaneous Updating in Extended Complementary PWM Mode

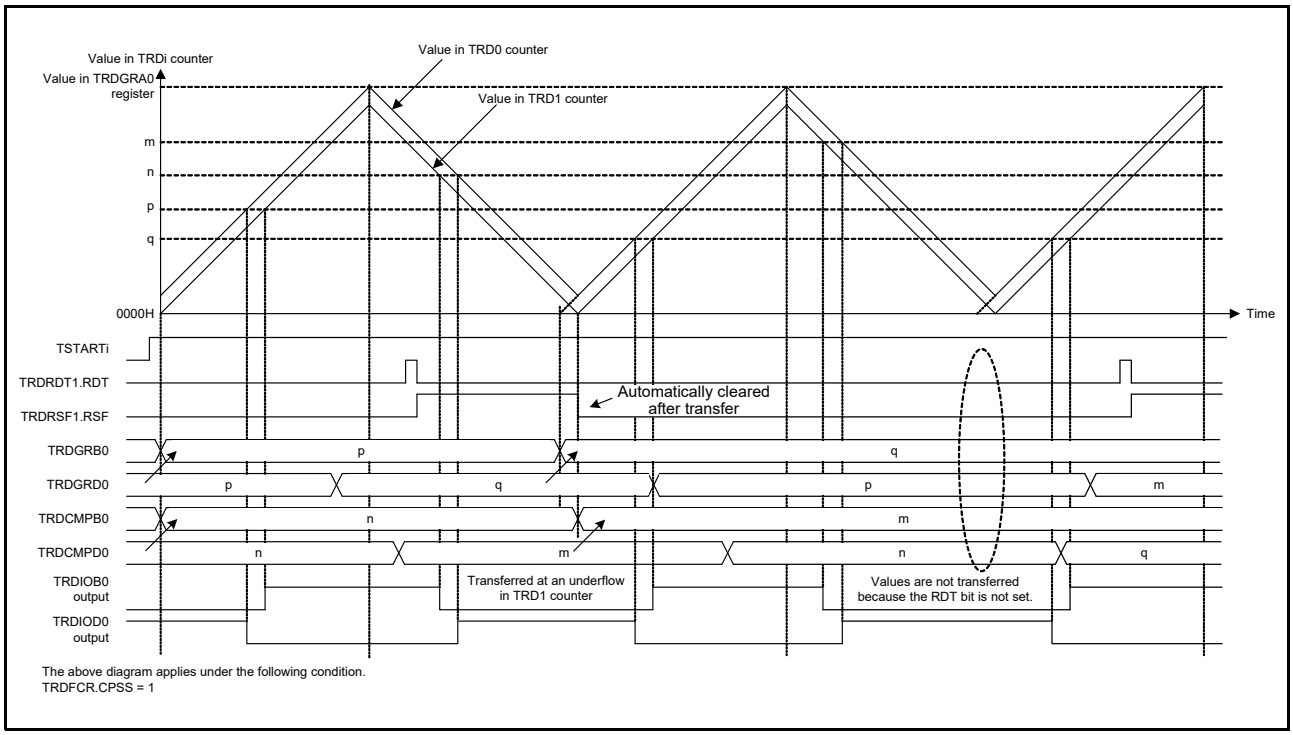
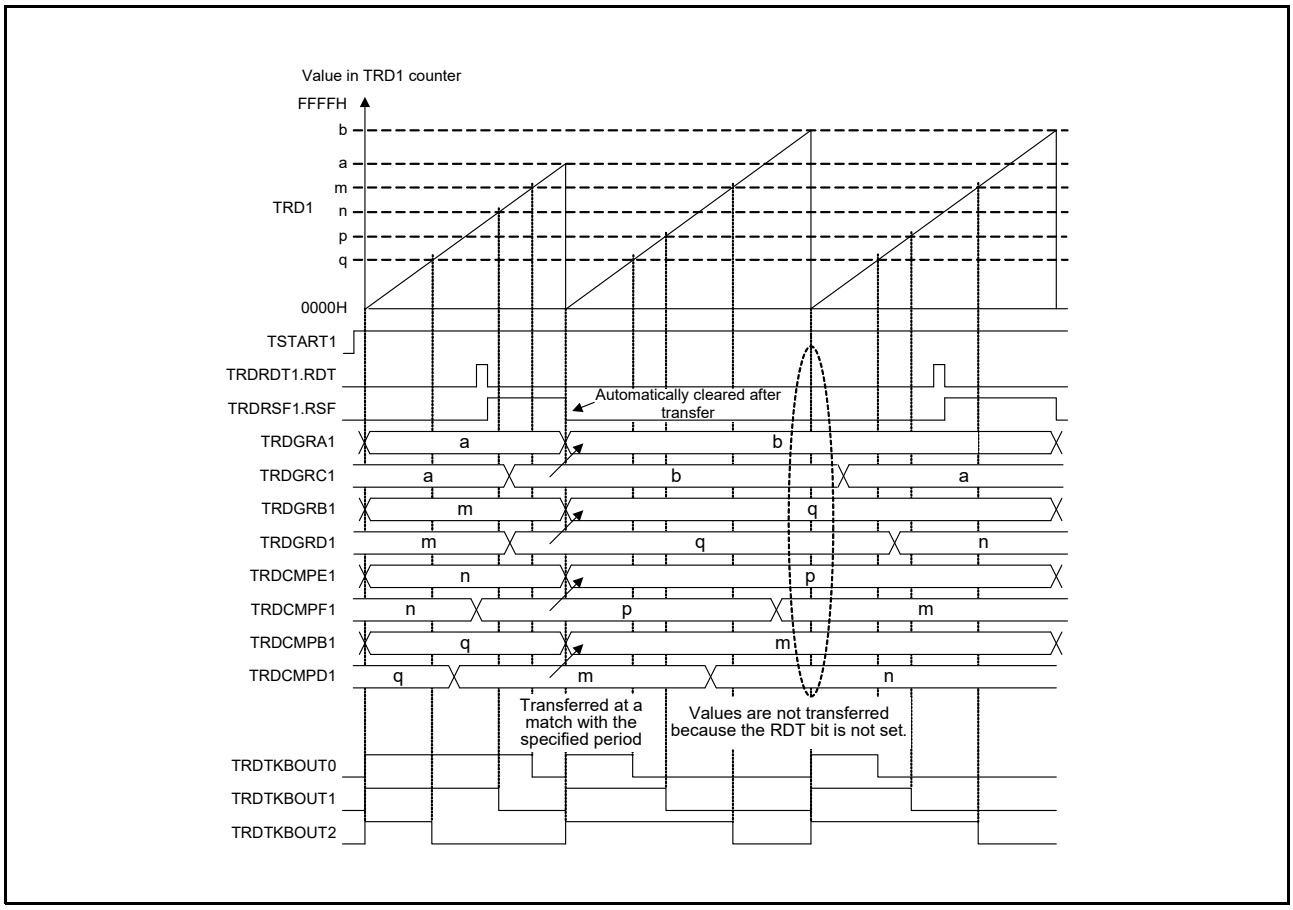


Figure 12 - 77 Operation Example of Simultaneous Updating in Timer-KB PWM Output Gating Mode



12.4.8 Skipping of interrupt requests and A/D conversion triggers

Output of interrupt request and A/D conversion trigger signals can be skipped in extended complementary PWM mode. Whether to enable or disable the output of interrupt request and A/D conversion trigger signals in the first cycle after the start of timer RD2 operation can be specified by the TRDTMD bit in the TRDTCTL register. When the TRDTMD bit is set to 1, the interrupt and trigger signals are output in the first cycle.

Figure 12 - 78 shows an operation example when the output of interrupt request and A/D conversion trigger signals is disabled in the first cycle after the start of timer RD2 operation (TRDTCTL.TRDTMD = 0) in extended complementary PWM mode and **Figure 12 - 79** shows an operation example when the output is enabled (TRDTCTL.TRDTMD = 1).

Figure 12 - 78 Operation Example When Output Is Disabled in the First Cycle after the Start of Timer RD2 Operation (TRDTCTL.TRDTMD = 0) in Extended Complementary PWM Mode

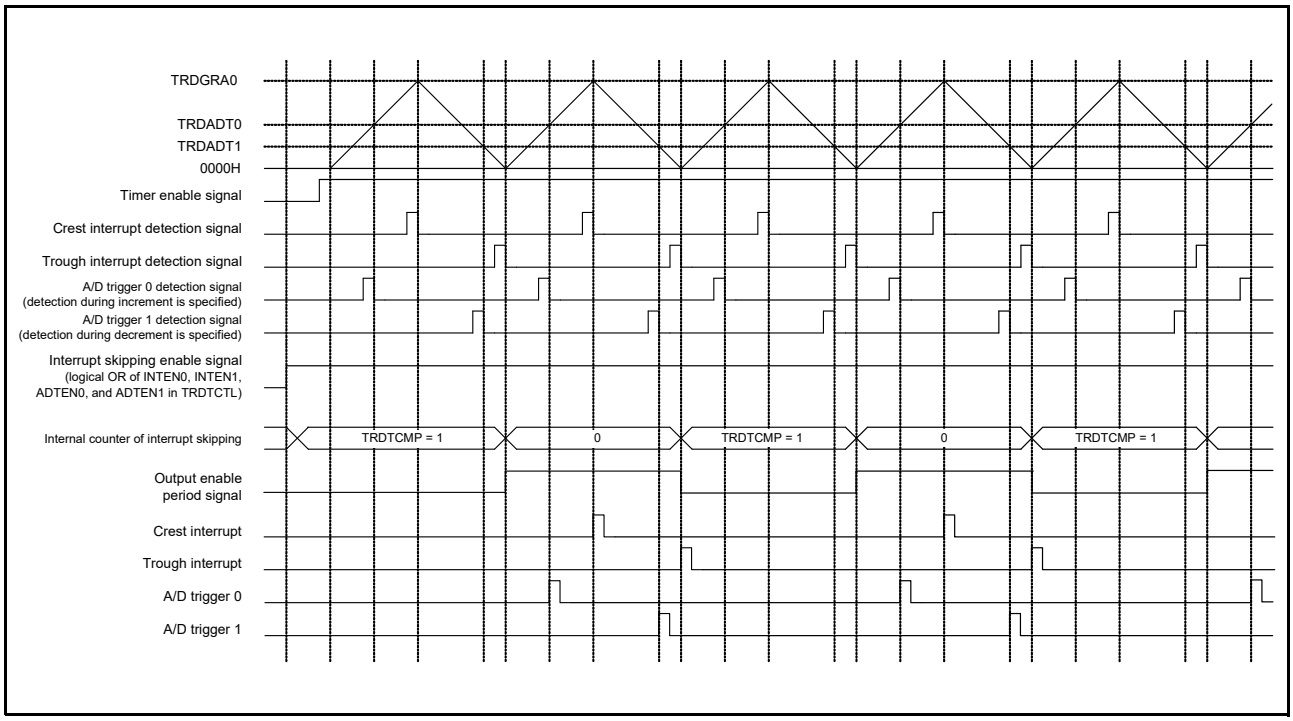
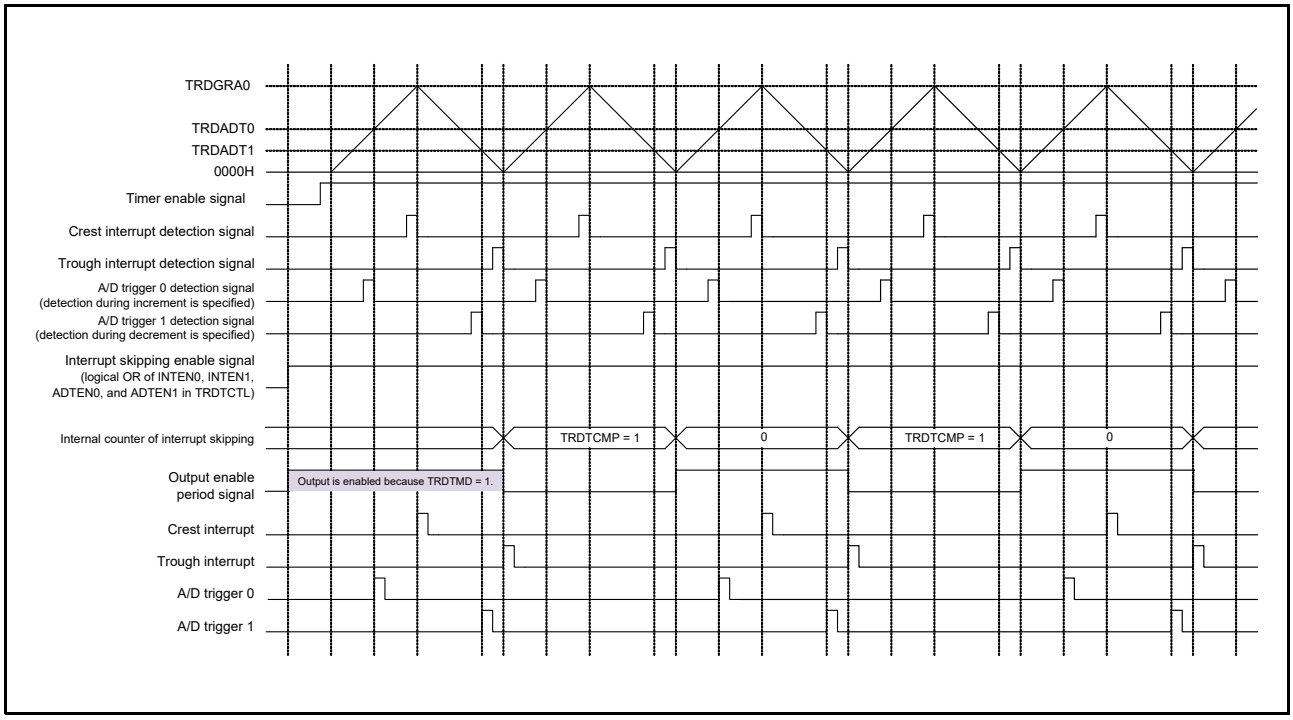


Figure 12 - 79 Operation Example When Output Is Enabled in the First Cycle after the Start of Timer RD2 Operation (TRDTCTL.TRDTMD = 1) in Extended Complementary PWM Mode



12.5 Timer RD2 Operation

For timer RX and interlinked operation, and timer RD2 forced cutoff control (PWMOPA) and interlinked operation, refer to **Section 14 Timer RX** and **12.8 PWM Option Unit A (PWMOPA)**.
For interlinked operation with timers KB30, KB31, and KB32, refer to **12.5.9 Timer-KB PWM output gating mode** and **Section 15 16-bit Timers KB30, KB31, and KB32**.

12.5.1 Input capture function

The input capture function measures the external signal width and period. The content of the TRDi counter is transferred to the TRDGRji register (input capture), triggered by an external signal input through the TRDIOji pin (i = 0 or 1, j = A to D). Since this function is enabled with a combination of the TRDIOji pin and TRDGRji register, whether to use the input capture function or any other mode or function can be selected for each individual pin.
Figure 12 - 80 is a block diagram of the input capture function (for timer RD20), **Table 12 - 18** lists the specifications of the input capture function, and **Figure 12 - 81** shows an operation example of the input capture function.

Figure 12 - 80 Block Diagram of Input Capture Function (for Timer RD20)

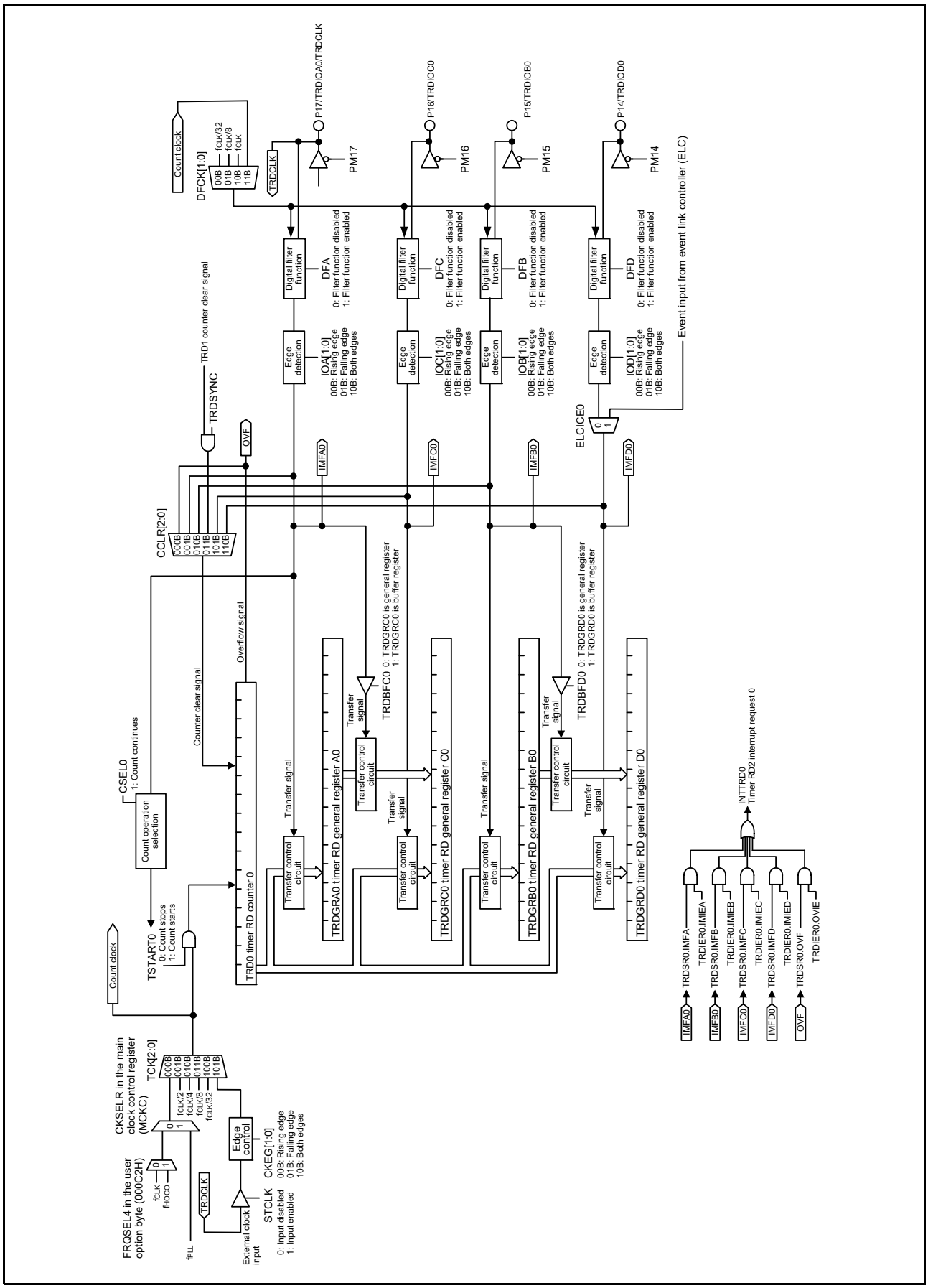


Table 12 - 18 Specifications of the Input Capture Function

Item	Specification
Count sources	f_{HOCO}**Note 1**, f_{PLL}**Note 2**, 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). $1/f_K \times 65536$ f_K: 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 counter overflow
TRDIOA0 pin function	I/O port, input-capture input, or TRDCLK (external clock) input
TRDIOB0, TRDIOC0, TRDIOD0, and 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 counter.
Write to timer	• When the TRDSYNC bit in the TRDMR register is 0 (timer RD20 and timer RD21 operate independently) Data can be written to the TRDi counter. • When the TRDSYNC bit in the TRDMR register is 1 (timer RD20 and timer RD21 operate synchronously) Data can be written to both the TRD0 and TRD1 counters by writing to the TRDi counter.
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 counter to 0000H At overflow or input capture • Buffer operation (see **12.4.2 Buffer operation**) • Synchronous operation (see **12.4.3 Synchronous operation**) • Digital filter The TRDIOji input is sampled, and when the sampled input level matches three times, that level is determined. • Input capture operation by event input from ELC

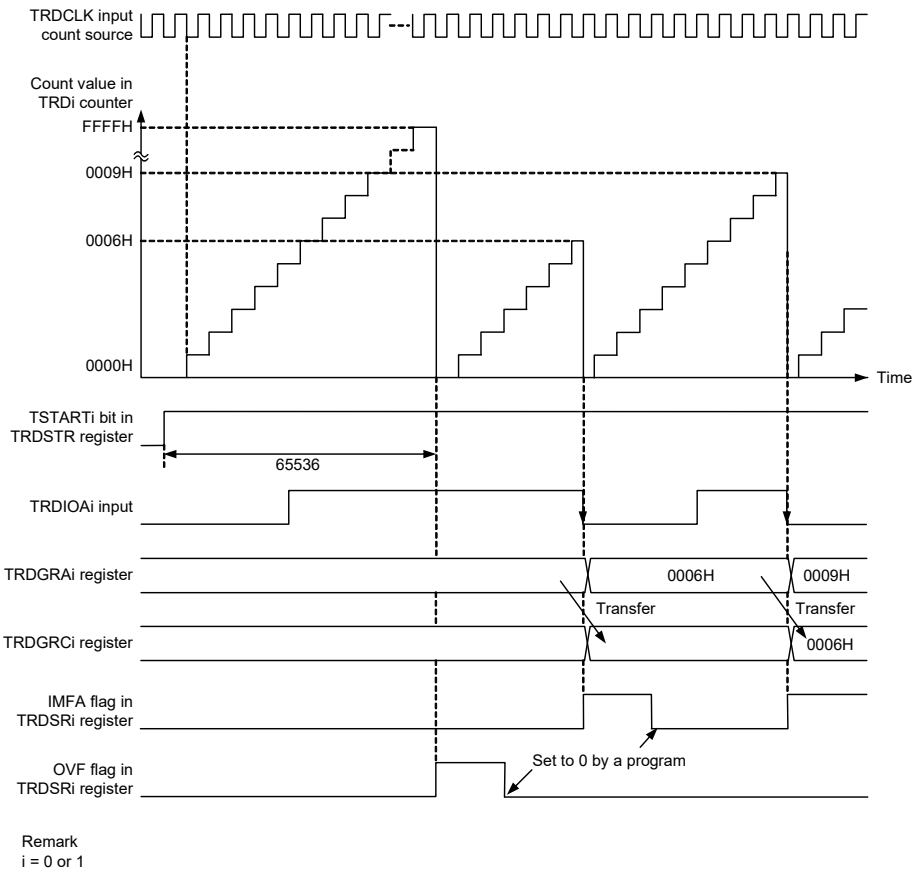
Note 1. f_{HOCO} is selected only when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H). When selecting f_{HOCO} as the count source for timer RD2, set f_{CLK} to f_{IH} before setting bit 3 (TRD0EN) in peripheral enable register 2 (PER2). When changing f_{CLK} to a clock other than f_{IH}, clear bit 3 (TRD0EN) in peripheral enable register 2 (PER2) before changing.

Note 2. f_{PLL} is supplied when the CKSELR bit is 1 in the main clock control register (MCKC).

Remark i = 0 or 1, j = A to D

1. Operation example
   By setting bits CCLR2 to CCLR0 in the TRDCRi register (i = 0 or 1), the timer RD2i counter value is reset by an input capture or a compare match. **Figure 12 - 81** 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, interrupt flags IMFA to IMFD and OVF in the TRDSRi register may be set to 1 simultaneously depending on the timing of the count source and input capture operation.

Figure 12 - 81 Operation Example of Input Capture Function



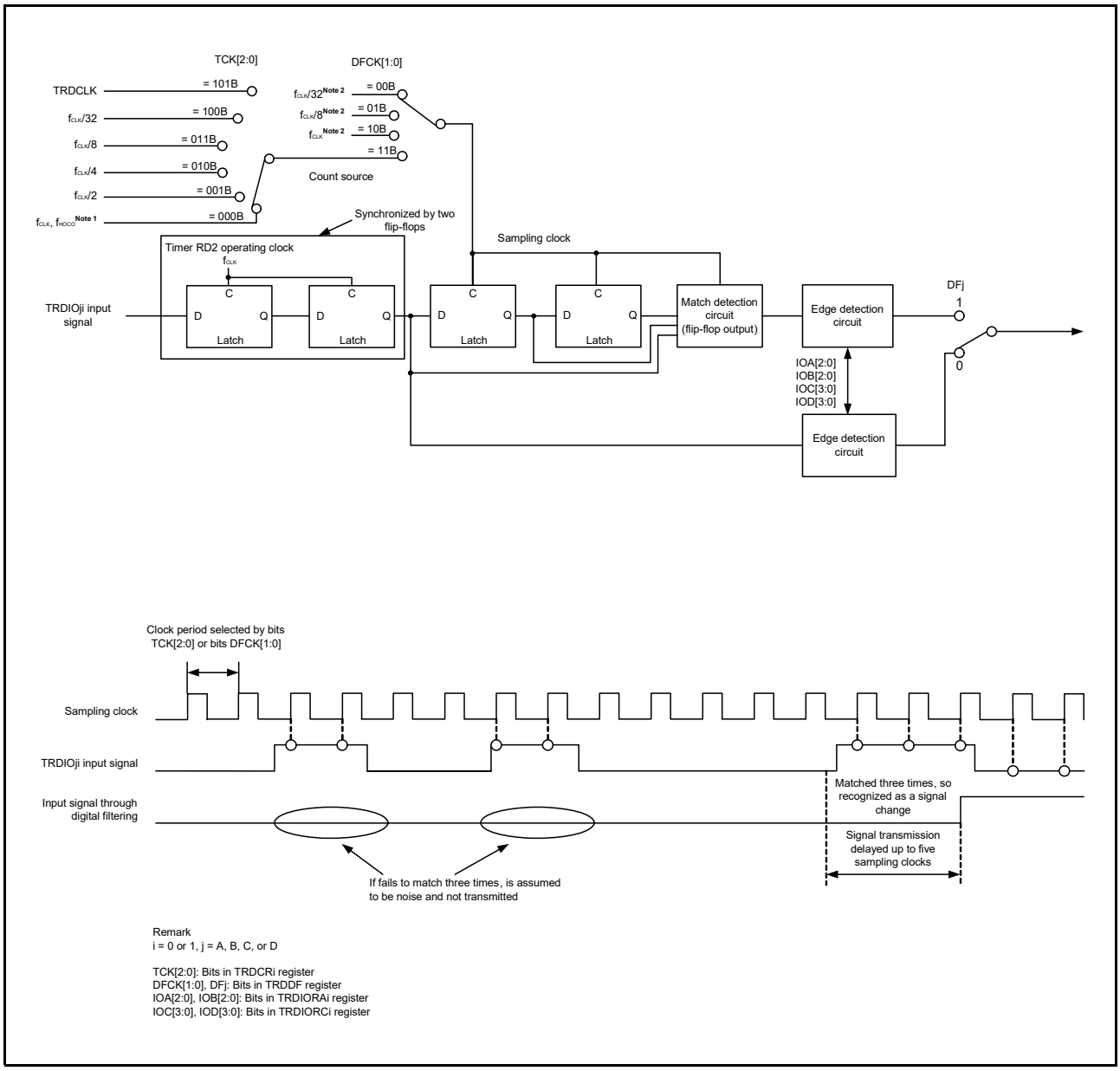
Remark
i = 0 or 1

The above diagram applies under the following conditions:
Bits CCLR[2:0] in the TRDCRi register are set to 001B (TRDi counter is set to 0000H by TRDGRAi register input capture).
Bits TCK[2:0] in the TRDCRi register are set to 101B (TRDCLK input for the count source).
Bits CKEG[1:0] in the TRDCRi register are set to 01B (count at the falling edge for the count source).
Bits IOA[2:0] 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 to 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 12 - 82** is a block diagram of the digital filter.

Figure 12 - 82 Block Diagram of Digital Filter



Note 1. fCLK is selected when the FRQSEL4 bit is 0 and fHOCO is selected when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H).

Note 2. When the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H), fCLK/32, fCLK/8, and fCLK are set to fHOCO/32, fHOCO/8, and fHOCO, respectively.
When the CKSELR bit is 1 in the main clock control register (MCKC), fCLK/32, fCLK/8, and fCLK are set to fPLL/32, fPLL/8, and fPLL, respectively.

12.5.2 Output compare function

This function detects matches (compare match) between the content of the TRDGRji register (j = A to D) and the content of the TRDi counter (i = 0 or 1). When the contents match, a user-specified level is output from the TRDIOji pin. Since this function is enabled with a combination of the TRDIOji pin and TRDGRji register, whether to use the output compare function or any other mode or function can be selected for each individual pin.

Figure 12 - 83 is a block diagram of the output compare function (for timer RD20), **Table 12 - 19** lists the specifications of the output compare function, and **Figure 12 - 84** shows an operation example of the output compare function.

Figure 12 - 83 Block Diagram of Output Compare Function (for Timer RD20)

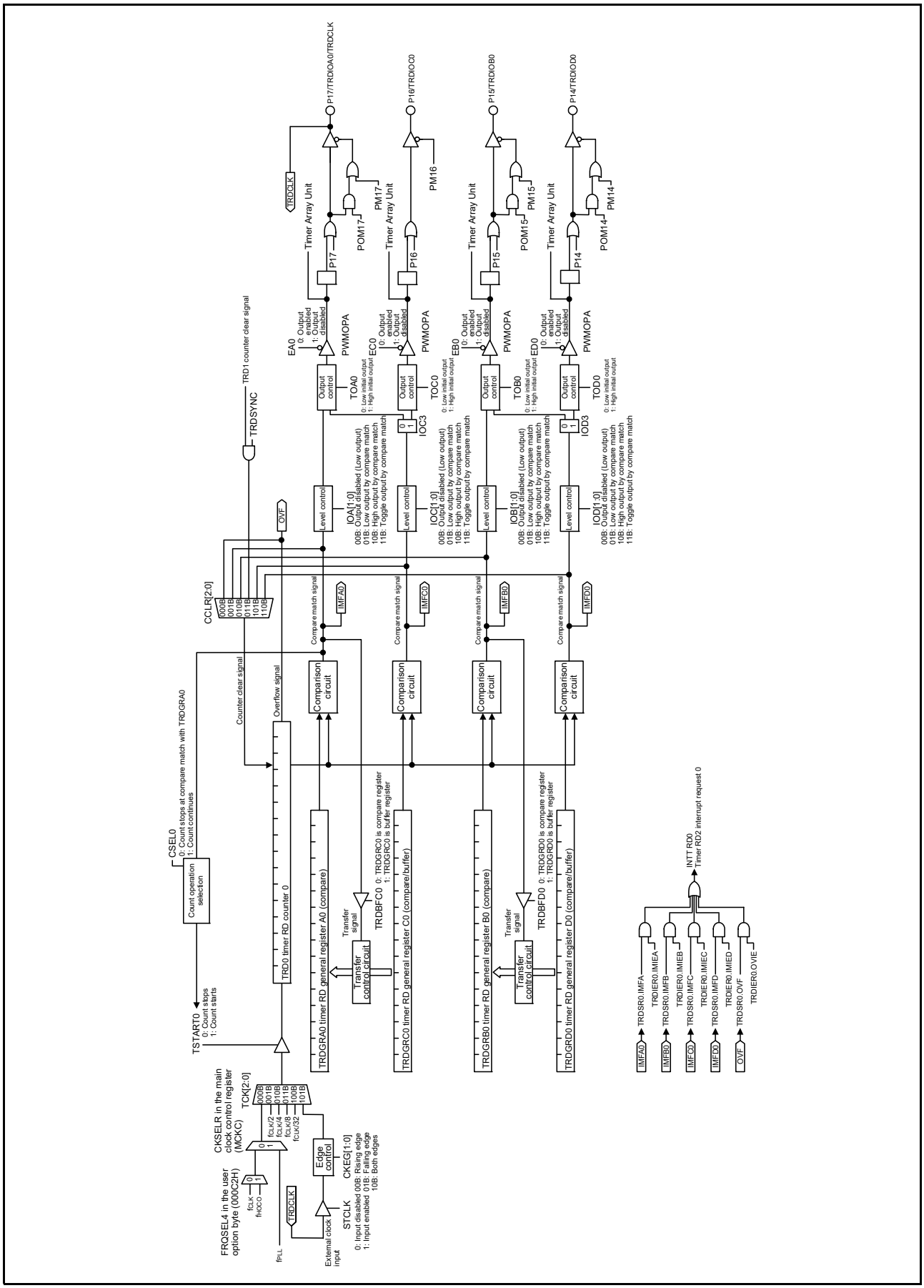


Table 12 - 19 Specifications of the Output Compare Function

Item	Specification
Count sources	f_{HOCO}**Note 1**, f_{PLL}**Note 2**, 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) $1/f_K \times 65536$ f_K: Frequency of count source • When bits CCLR1 and CCLR0 in the TRDCRi register are set to 01B or 10B (TRDi counter is set to 0000H at compare match with TRDGRji register) $1/f_K \times (n + 1)$ n: Value set in the TRDGRji register
Waveform output timing	Compare match (the value of the TRDi counter matches that of 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 output compare output pins hold the output level before the count stops. • When the CSELi bit in the TRDSTR register is set to 0, the count stops at a compare match with the TRDGRAi register. The output compare output pins hold the level after output change by a compare match.
Interrupt request generation timing	• Compare match (the value of the TRDi counter matches that of the TRDGRji register) • TRDi counter overflow
TRDIOA0 pin function	I/O port, output-compare output, or TRDCLK (external clock) input
TRDIOB0, TRDIOC0, TRDIOD0, and 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 counter.
Write to timer	• When the TRDSYNC bit in the TRDMR register is set to 0 (timer RD20 and timer RD21 operate independently) Data can be written to the TRDi counter. • When the TRDSYNC bit in the TRDMR register is set to 1 (timer RD20 and timer RD21 operate synchronously) Data can be written to both the TRD0 and TRD1 counters by writing to the TRDi counter.
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 counter to 0000H Overflow or compare match in the TRDGRAi register • Buffer operation (see **12.4.2 Buffer operation**) • Synchronous operation (see **12.4.3 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 RD2 can be used as the internal timer without output.

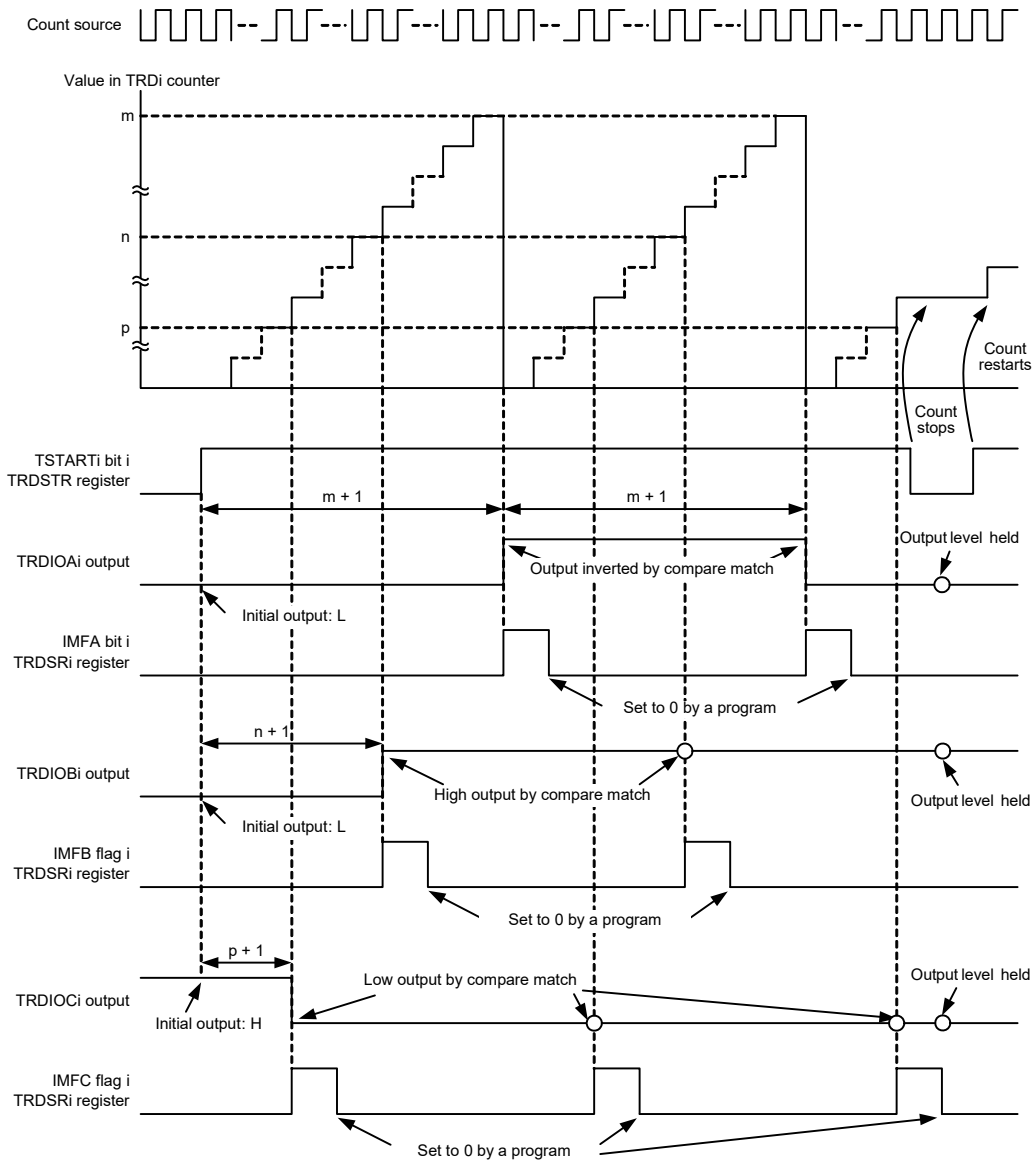
Note 1. f_{HOCO} is selected only when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H). When selecting f_{HOCO} as the count source for timer RD2, set f_{CLK} to f_{IH} before setting bit 3 (TRD0EN) in peripheral enable register 2 (PER2). When changing f_{CLK} to a clock other than f_{IH}, clear bit 3 (TRD0EN) in peripheral enable register 2 (PER2) before changing.

Note 2. f_{PLL} is supplied when the CKSELR bit is 1 in the main clock control register (MCKC).

Remark i = 0 or 1, j = A, to D

1. Operation example
   By setting bits CCLR2 to CCLR0 in the TRDCRi register (i = 0 or 1), the timer RD2i counter value is reset by an input capture or a compare match. If the expected compare value is FFFFH at this time, the counter value changes from FFFFH to 0000H in the same way as the overflow operation, and the overflow flag is set to 1.

Figure 12 - 84 Operation Example of Output Compare Function

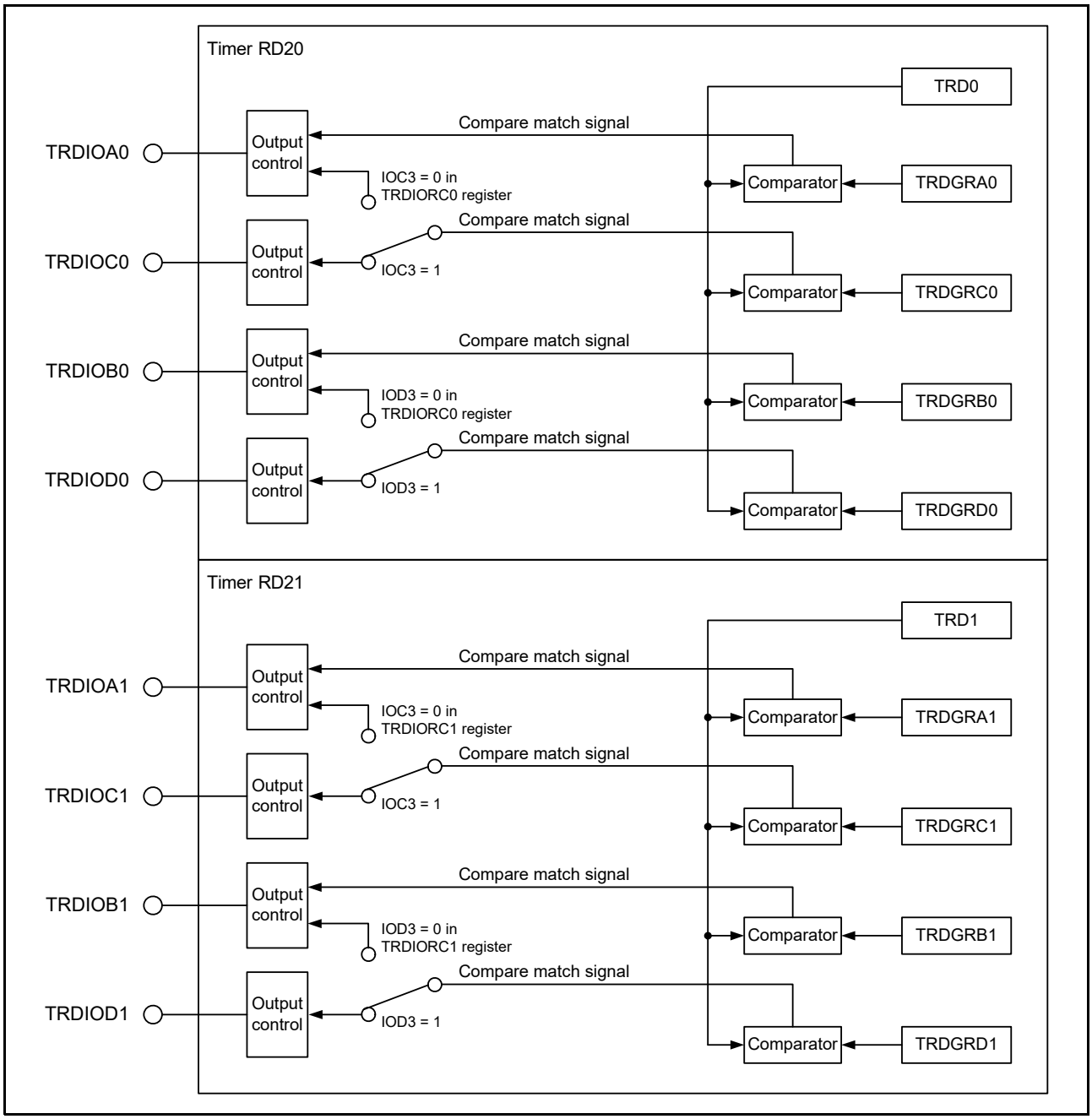


Remark
i = 0 or 1
M: Value set in TRDGRAi register
n: Value set in TRDGRBi register
p: Value set in TRDGRCi register

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 CCLR[2:0] 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 IOA[2:0] in the TRDIORAi register are set to 011B (TRDIOAi output inverted at TRDGRAi compare match).
Bits IOB[2:0] in the TRDIORAi register are set to 010B (TRDIOBi high output at TRDGRBi compare match).
Bits IOC[2:0] in the TRDIORCi register are set to 1001B (TRDIOCi low output at TRDGRCi register compare match).
Bits IOD[2:0] 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 for registers TRDGRCi and TRDGRDi (i = 0 or 1)
   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 12 - 85 Changing Output Pins for Registers TRDGRCi and TRDGRDi

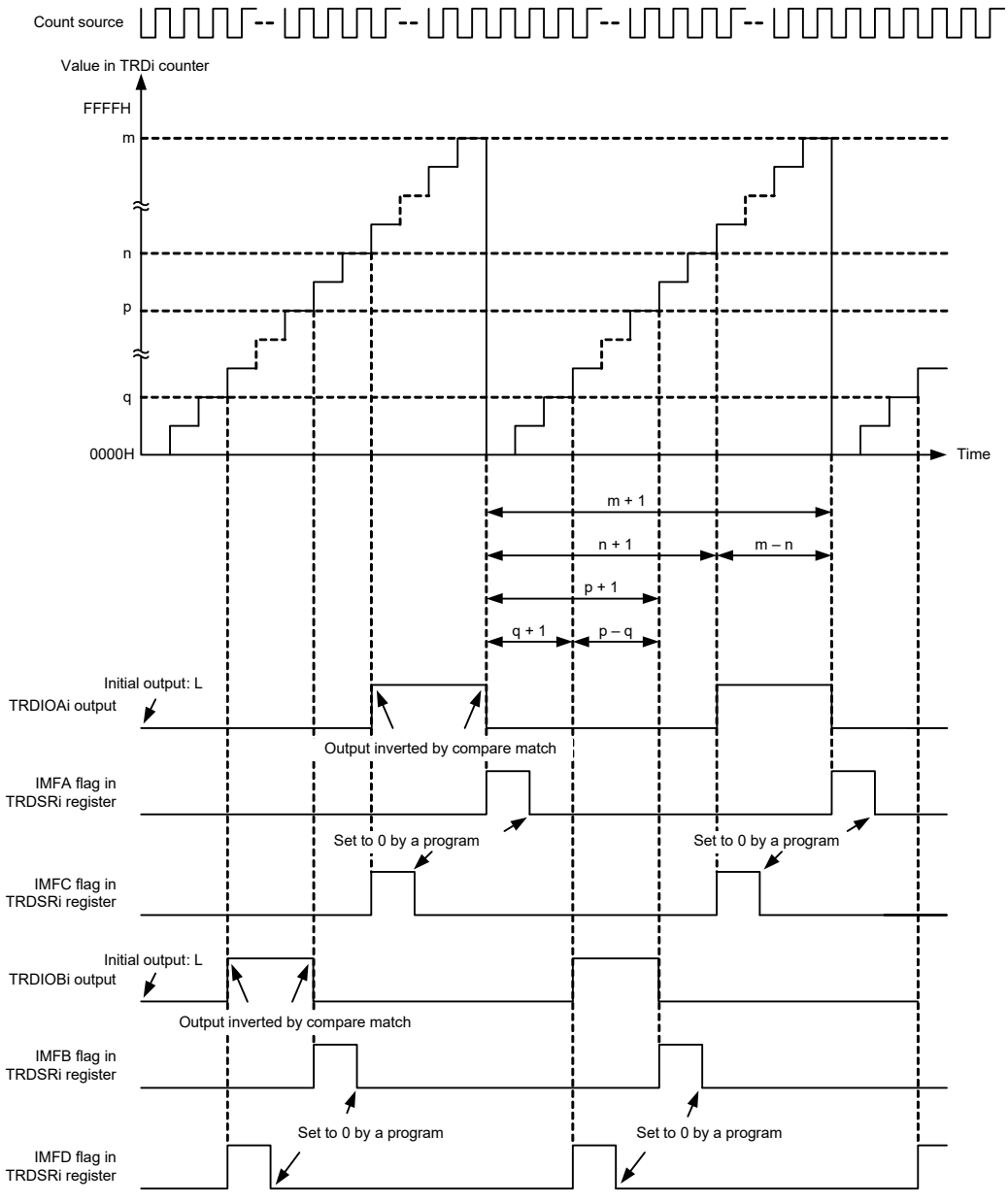


Change output pins for registers TRDGRCi and TRDGRDi as follows:

- 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 12 - 86 shows an operation example when the TRDGRCi register is used for output control of the TRDIOAi pin and the TRDGRDi register is used for output control of the TRDIOBi pin.

Figure 12 - 86 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 counter 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 CCLR[2:0] 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 until compare match : L).
Bits IOA[2:0] in the TRDIORAi register are set to 011B (TRDIOAi output inverted at TRDGRAi compare match).
Bits IOB[2:0] in the TRDIORAi register are set to 011B (TRDIOBi output inverted at TRDGRBi compare match).
Bits IOC[2:0] in the TRDIORCi register are set to 0011B (TRDIOAi output inverted at TRDGRCi compare match).
Bits IOD[2:0] in the TRDIORCi register are set to 0011B (TRDIOBi output inverted at TRDGRDi compare match).

12.5.3 PWM function

This function outputs PWM waveforms Up to three PWM waveforms with the same period can be output by timer RD2i (i = 0 or 1), or up to six PWM waveforms with the same period can be output by synchronizing timer RD20 and timer RD21.

Since this function works by a combination of the TRDIOji pin (i = 0 or 1, j = B to D) and TRDGRji register, whether to use the PWM function or any other mode or function can be selected for each individual pin. (However, since the TRDGRAi register is always used for the PWM function when any pin is set to the PWM function, the TRDGRAi register cannot be used for other modes.)

Figure 12 - 87 is a block diagram of the PWM function (for timer RD20), **Table 12 - 20** lists the specifications of the PWM function, **Figure 12 - 88** shows an operation example of the PWM function, and **Figure 12 - 89** shows operation examples when the duty cycle is 0% and 100%.

Figure 12 - 87 Block Diagram of PWM Function (for Timer RD20)

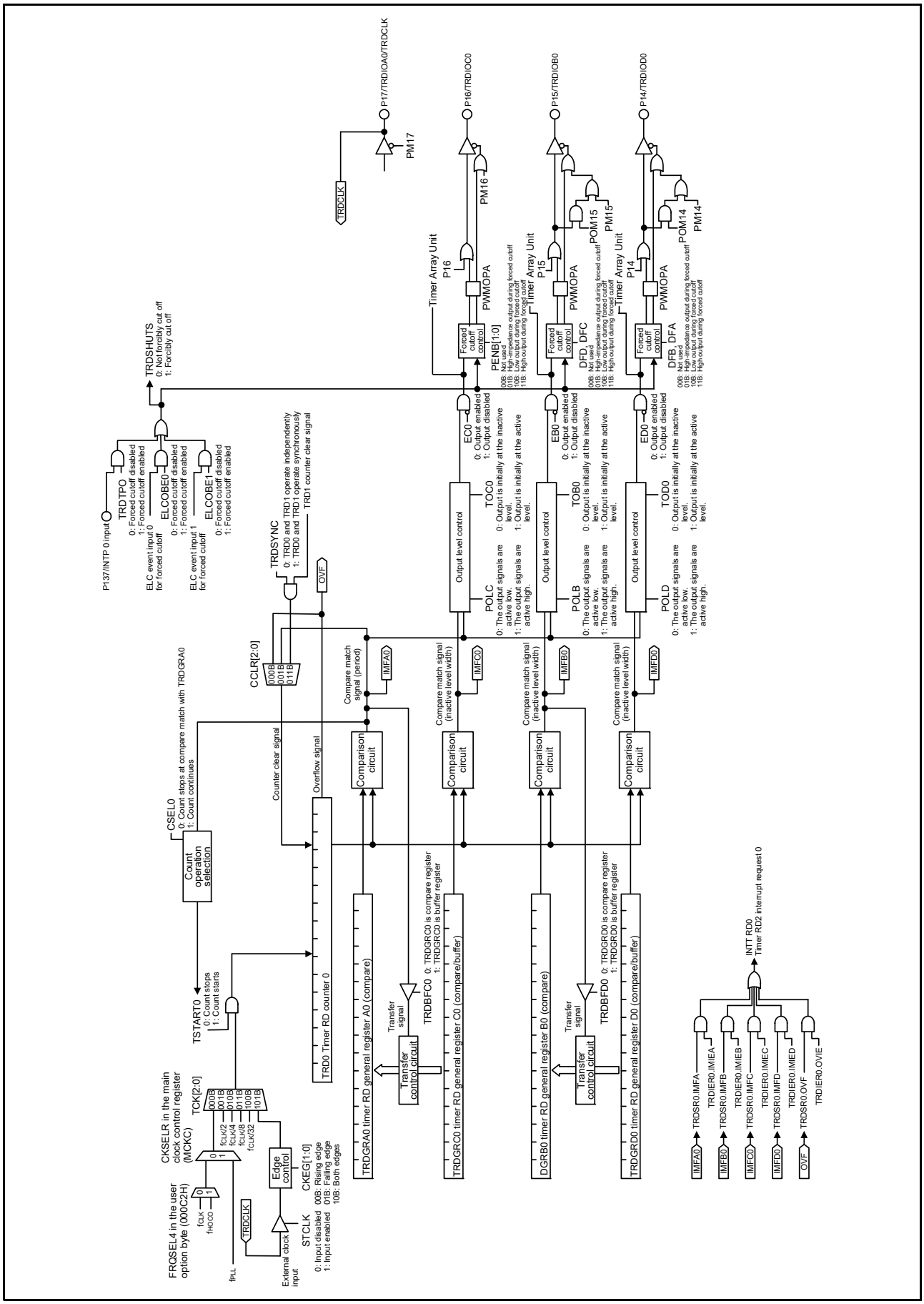
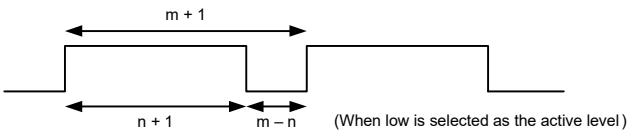


Table 12 - 20 Specifications of the PWM Function

Item	Specification
Count sources	f_{HOCO}**Note 1**, f_{PLL}**Note 2**, 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: $1/f_K \times (m + 1)$ Active level width: $1/f_K \times (m - n)$ Inactive level width: $1/f_K \times (n + 1)$ f_K: 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 pins hold the output level before the count stops. • When the CSELi bit in the TRDSTR register is set to 0, the count stops at a compare match with the TRDGRAi register. The PWM output pins hold the level after output change by compare match.
Interrupt request generation timing	• Compare match (the value of the TRDi counter matches that of the TRDGRhi register) • TRDi counter overflow
TRDIOA0 pin function	I/O port or TRDCLK (external clock) input
TRDIOA1 pin function	I/O port
TRDIOB0, TRDIOC0, TRDIOD0, TRDIOB1, TRDIOC1, and TRDIOD1 pin function	I/O port or pulse output (selectable for each pin)
INTP0 pin function	Input of pulse output forced cutoff signal (input-only port or INTP0 interrupt input)
Read from timer	The count value can be read by reading the TRDi counter.
Write to timer	A value can be written to the TRDi counter.
Selectable functions	• One to three PWM output pins selectable with timer RD2i 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 **12.4.3 Synchronous operation**) • Buffer operation (see **12.4.2 Buffer operation**) • Input of pulse output forced cutoff signal (see **12.4.4 Pulse output forced cutoff**)

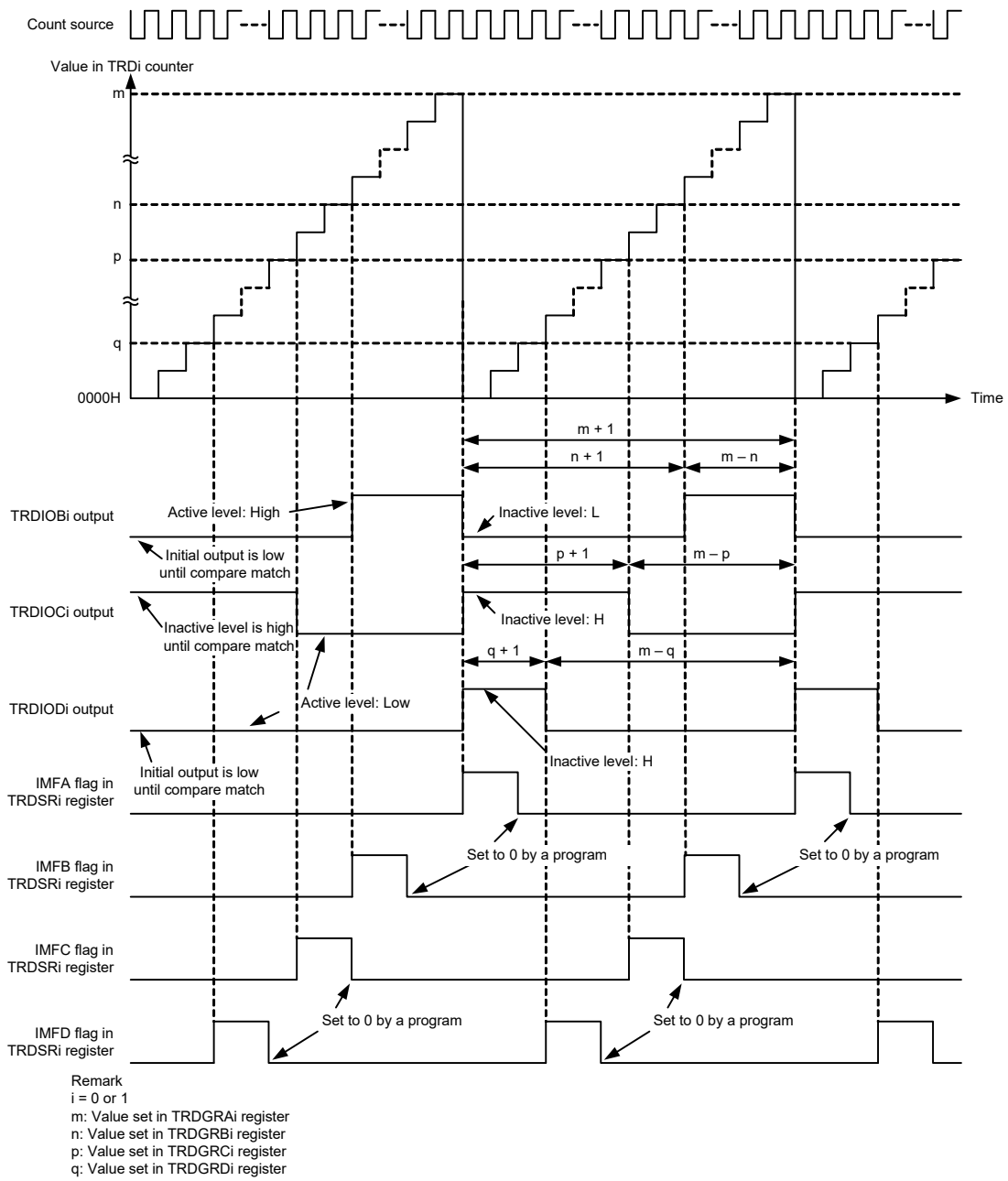
Note 1. f_{HOCO} is selected only when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H). When selecting f_{HOCO} as the count source for timer RD2, set f_{CLK} to f_{IH} before setting bit 3 (TRD0EN) in peripheral enable register 2 (PER2). When changing f_{CLK} to a clock other than f_{IH}, clear bit 3 (TRD0EN) in peripheral enable register 2 (PER2) before changing.

Note 2. f_{PLL} is supplied when the CKSELR bit is 1 in the main clock control register (MCKC).

Remark i = 0 or 1, j = B to D, h = A to D

1. Operation examples

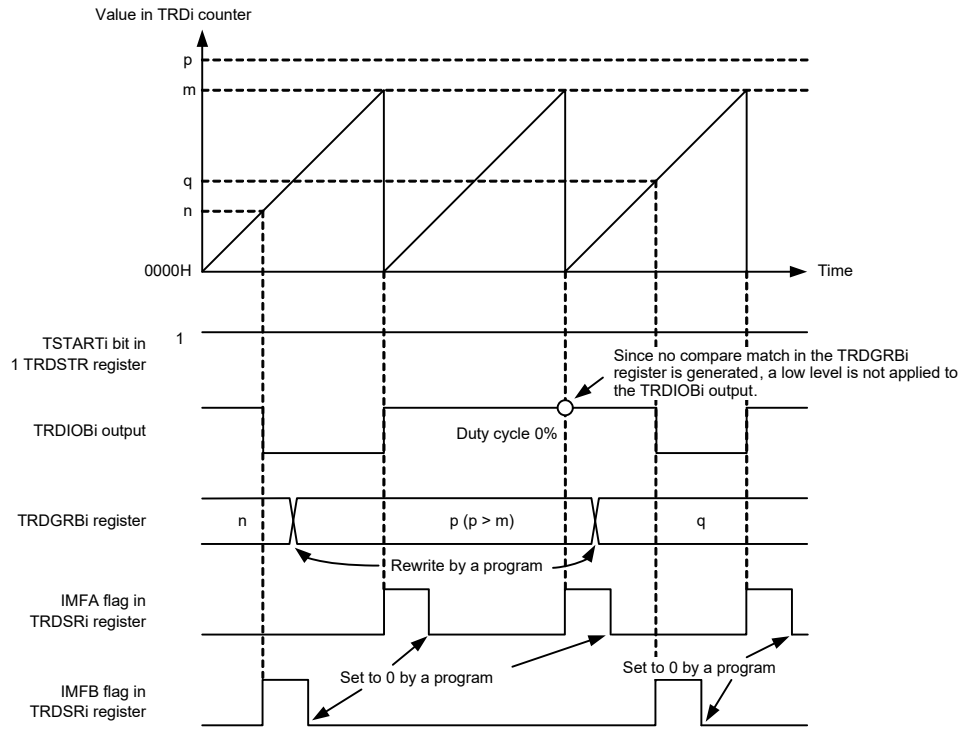
Figure 12 - 88 Operation Example of PWM Function

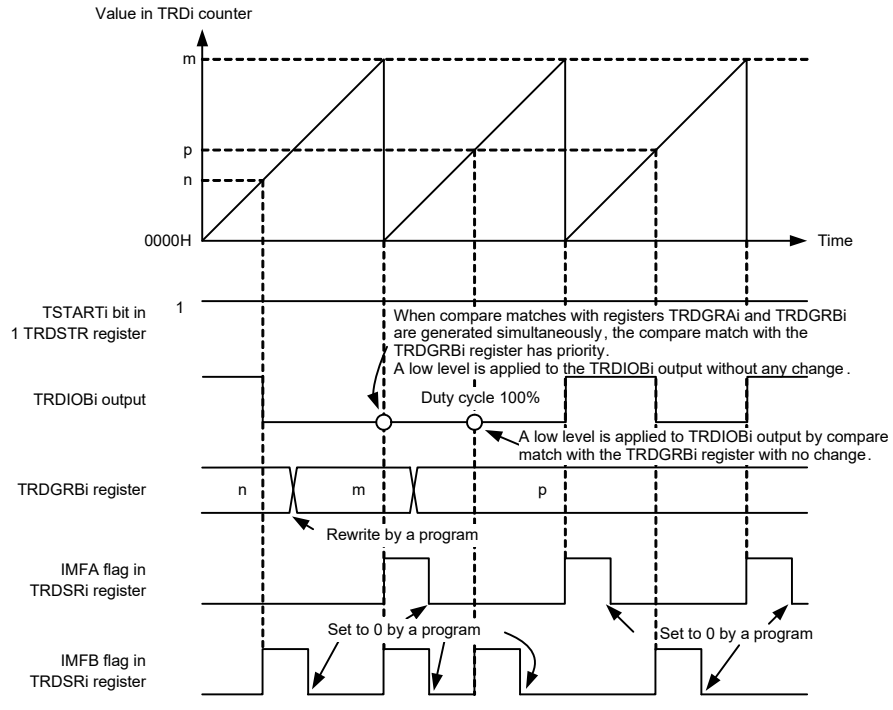


Remark
i = 0 or 1
m: Value set in TRDGRAi register
n: Value set in TRDGRBi register
p: Value set in TRDGRCi register
q: Value set in TRDGRDi register

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 12 - 89 Operation Examples of 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: L).

12.5.4 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 12 - 90 is a block diagram of reset synchronous PWM mode (for timer RD20), **Table 12 - 21** lists the specifications of reset synchronous PWM mode, and **Figure 12 - 91** shows an operation example in reset synchronous PWM mode.

See **Figure 12 - 89 Operation Examples of PWM Function (Duty Cycle 0%, Duty Cycle 100%)** for an operation example in PWM mode with duty cycle 0% and duty cycle 100%.

Figure 12 - 90 Block Diagram of Reset Synchronous PWM Mode (For Timer RD20)

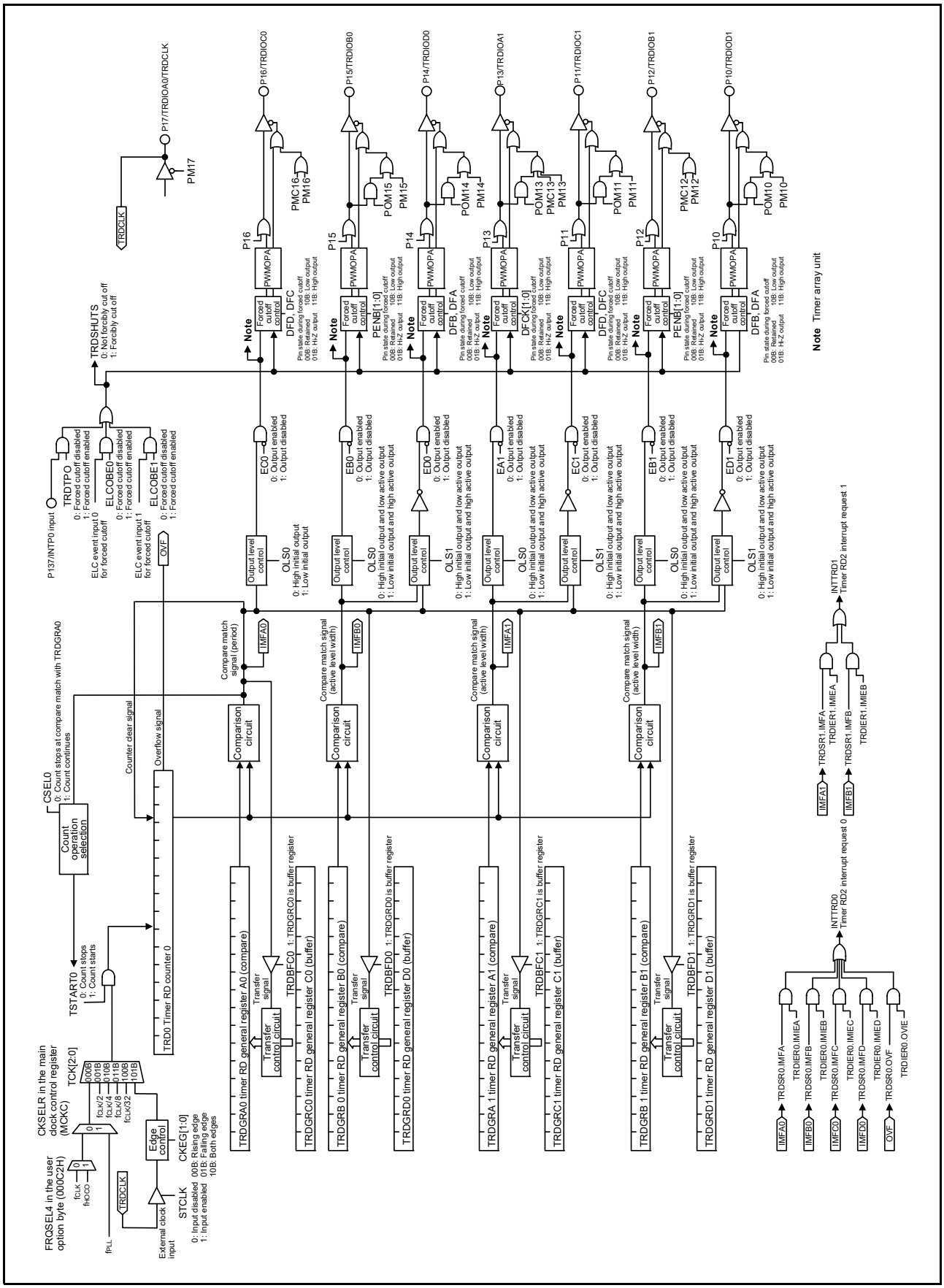
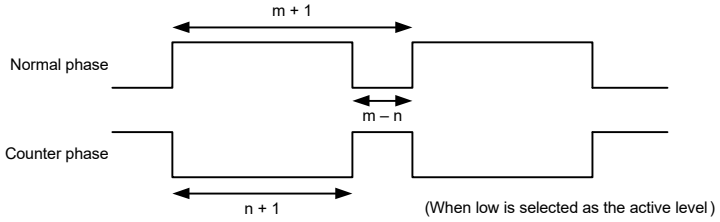


Table 12 - 21 Specifications of Reset Synchronous PWM Mode

Item	Specification
Count sources	f_{HOCO} **Note 1**, f_{PLL} **Note 2**, 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	The TRD0 counter is incremented (the TRD1 counter is not used).
PWM waveform	PWM period: $1/f_K \times (m + 1)$ Active level width of normal phase: $1/f_K \times (m - n)$ Active level width of counter phase: $1/f_K \times (n + 1)$ f_K: 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 pins output 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 a compare match with the TRDGRA0 register. The PWM output pins output the initial output level selected by bits OLS0 and OLS1 in the TRDFCR register.
Interrupt request generation timing	• Compare match (the value of the TRD0 counter matches that of registers TRDGRj0, TRDGRA1, and TRDGRB1) • TRD0 counter 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	Input of pulse output forced cutoff signal (input-only port or INTP0 interrupt input)
Read from timer	The count value can be read by reading the TRD0 counter.
Write to timer	A value can be written to the TRD0 counter.
Selectable functions	• The normal-phase and counter-phase active level and initial output level are selected individually. • Buffer operation (see **12.4.2 Buffer operation**) • Input of pulse output forced cutoff signal (see **12.4.4 Pulse output forced cutoff**)

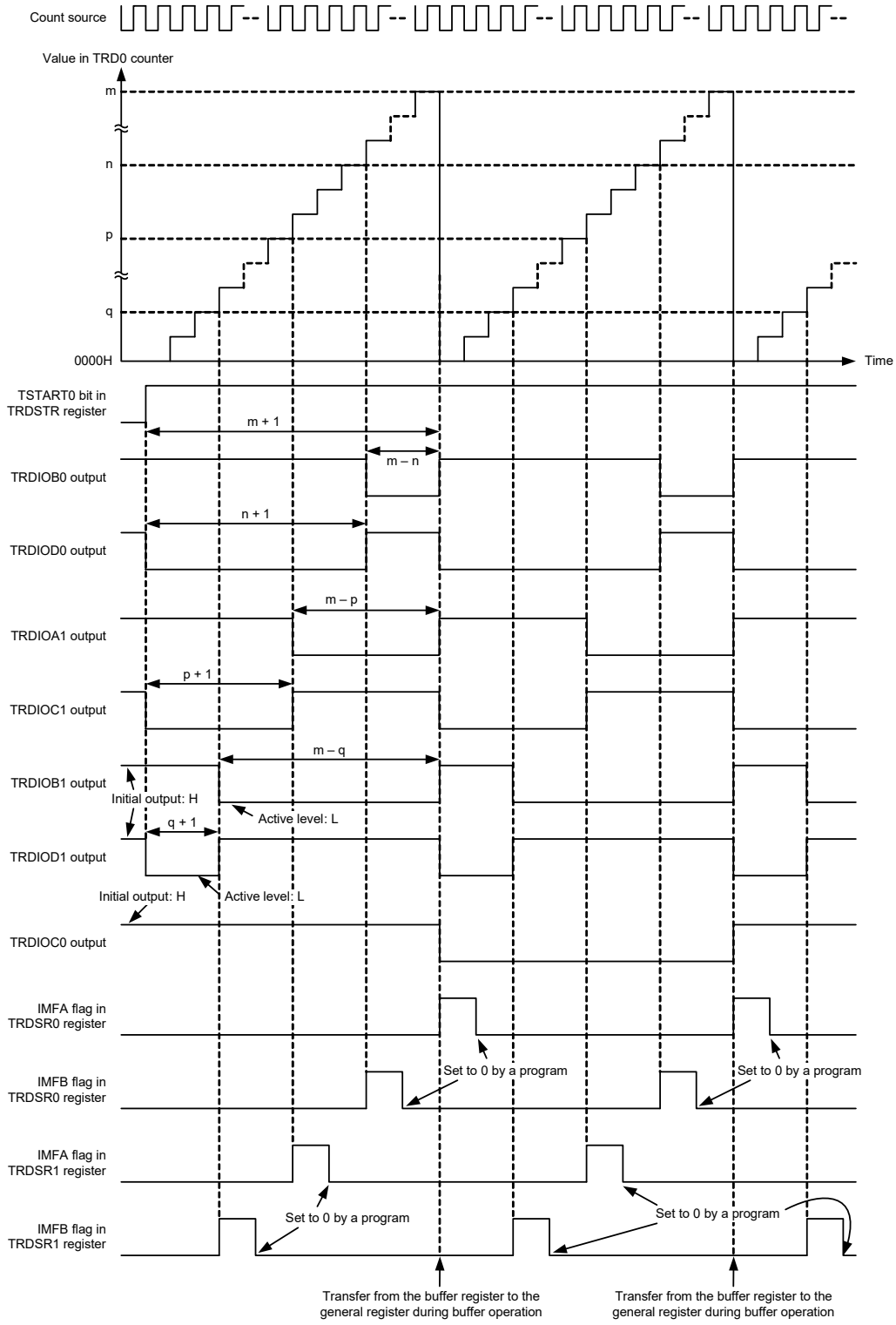
Note 1. f_{HOCO} is selected only when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H). When selecting f_{HOCO} as the count source for timer RD2, set f_{CLK} to f_{IH} before setting bit 3 (TRD0EN) in peripheral enable register 2 (PER2). When changing f_{CLK} to a clock other than f_{IH}, clear bit 3 (TRD0EN) in peripheral enable register 2 (PER2) before changing.

Note 2. f_{PLL} is supplied when the CKSELR bit is 1 in the main clock control register (MCKC).

Remark j = A to D

1. Operation example

Figure 12 - 91 Operation Example in Reset Synchronous PWM Mode



Remark
i = 0, 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: H, active level: L).

12.5.5 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 12 - 92 is a block diagram of complementary PWM mode (for timer RD20), **Table 12 - 22** lists the specifications of complementary PWM mode, **Figure 12 - 93** shows the output model of complementary PWM mode, and **Figure 12 - 94** shows an operation example in complementary PWM mode.

Figure 12 - 92 Block Diagram of Complementary PWM Mode (For Timer RD20)

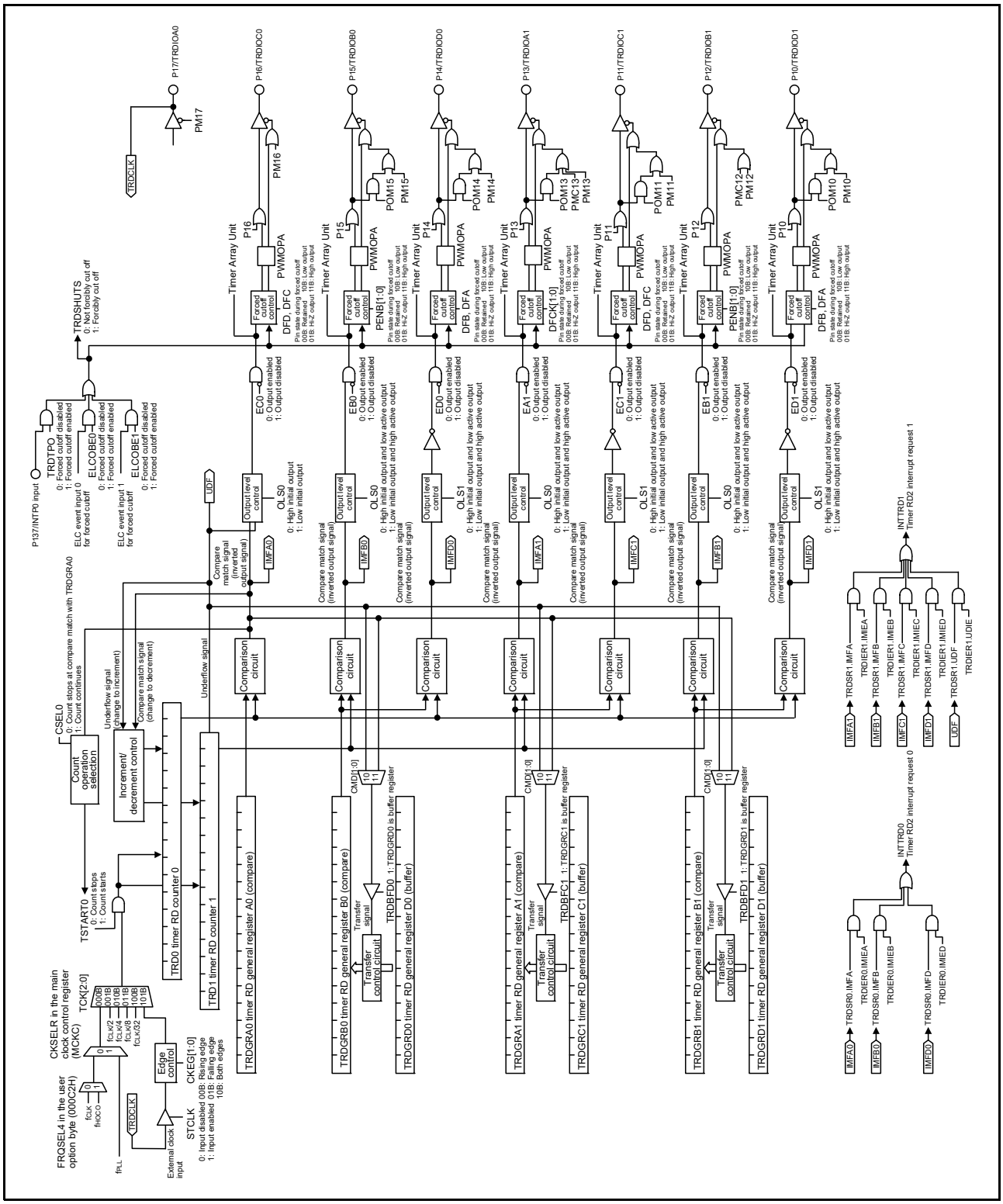
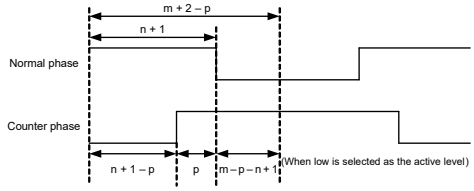


Table 12 - 22 Specifications of Complementary PWM Mode

Item	Specification
Count sources	f_{HOCO}**Note 1**, f_{PLL}**Note 2**, 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 that specified in bits TCK2 to TCK0 in the TRDCR0 register.
Count operations	Increment or decrement. When a compare match between the TRD0 counter and TRDGRA0 register is detected during increment operation, both the TRD0 and TRD1 counters are decremented. When the TRD1 counter value changes from 0000H to FFFFH during decrement operation, both the TRD0 and TRD1 counters are incremented.
PWM waveform	PWM period: $1/f_K \times (m + 2 - p) \times 2$**Note 3** Dead time: p Active level width of normal phase: $1/f_K \times (m - n - p + 1) \times 2$ Active level width of counter phase: $1/f_K \times (n + 1 - p) \times 2$ f_K: 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 counter 
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 pins output the initial output level selected by bits OLS0 and OLS1 in the TRDFCR register.)
Interrupt request generation timing	• Compare match (the value of the TRDi counter matches that of the TRDGRji register) • TRD1 counter 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	Input of pulse output forced cutoff signal (input-only port or INTP0 interrupt input)
Read from timer	The count value can be read by reading the TRDi counter.
Write to timer	A value can be written to the TRDi counter.
Selectable functions	Input of pulse output forced cutoff signal (see **12.4.4 Pulse output forced cutoff**) The normal-phase and counter-phase active level and initial output level are selected individually. Timing of transfer from the buffer register selection

Note 1. f_{HOCO} is selected only when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H). When selecting f_{HOCO} as the count source for timer RD2, set f_{CLK} to f_{IH} before setting bit 3 (TRD0EN) in peripheral enable register 2 (PER2). When changing f_{CLK} to a clock other than f_{IH}, clear bit 3 (TRD0EN) in peripheral enable register 2 (PER2) before changing.

Note 2. f_{PLL} is supplied when the CKSELR bit is 1 in the main clock control register (MCKC).

(Note and Remark are listed on the next page.)

Note 3. After counting starts, the PWM period is fixed.

Remark i = 0 or 1, j = A to D

1. Operation examples

Figure 12 - 93 Output Model of Complementary PWM Mode

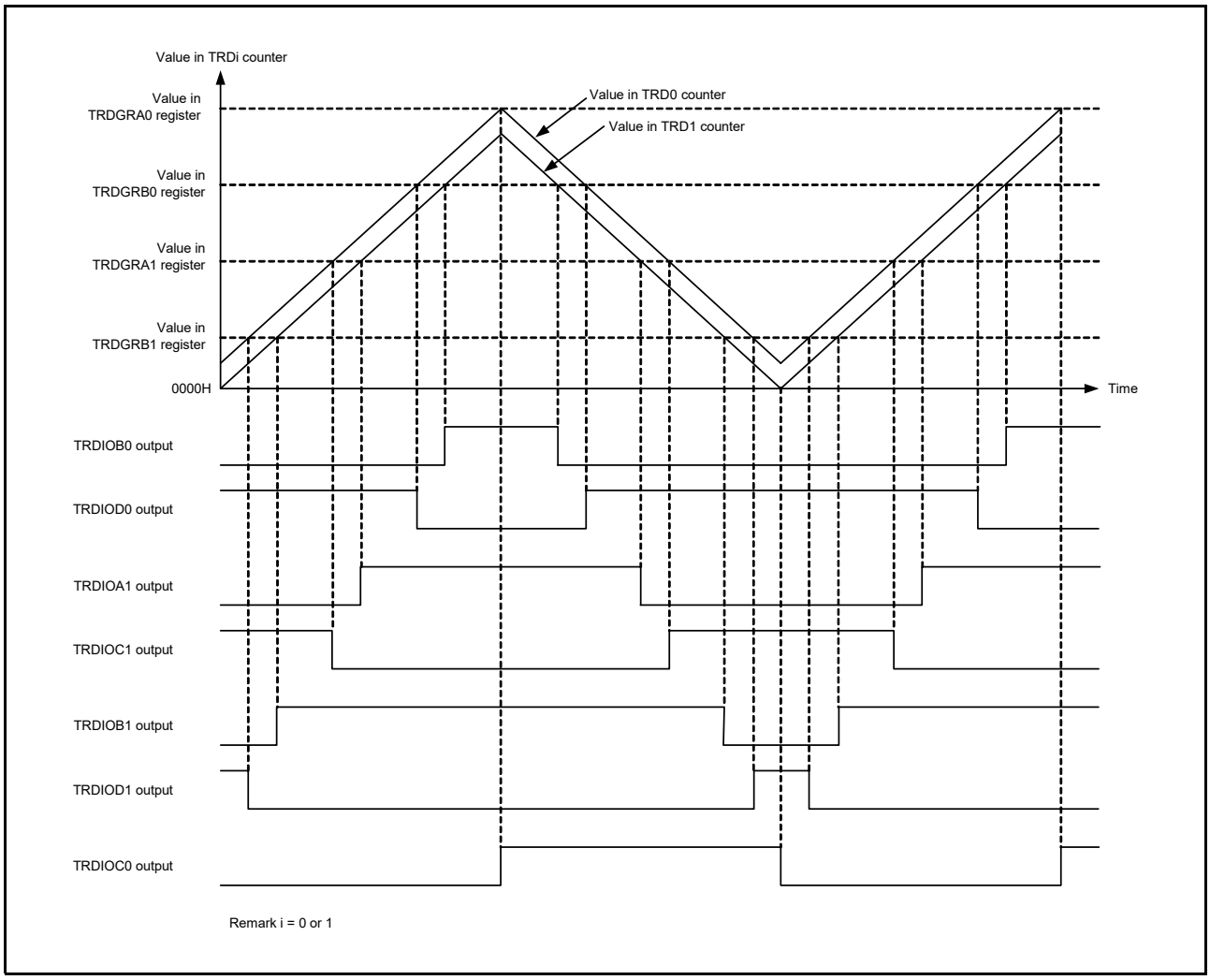
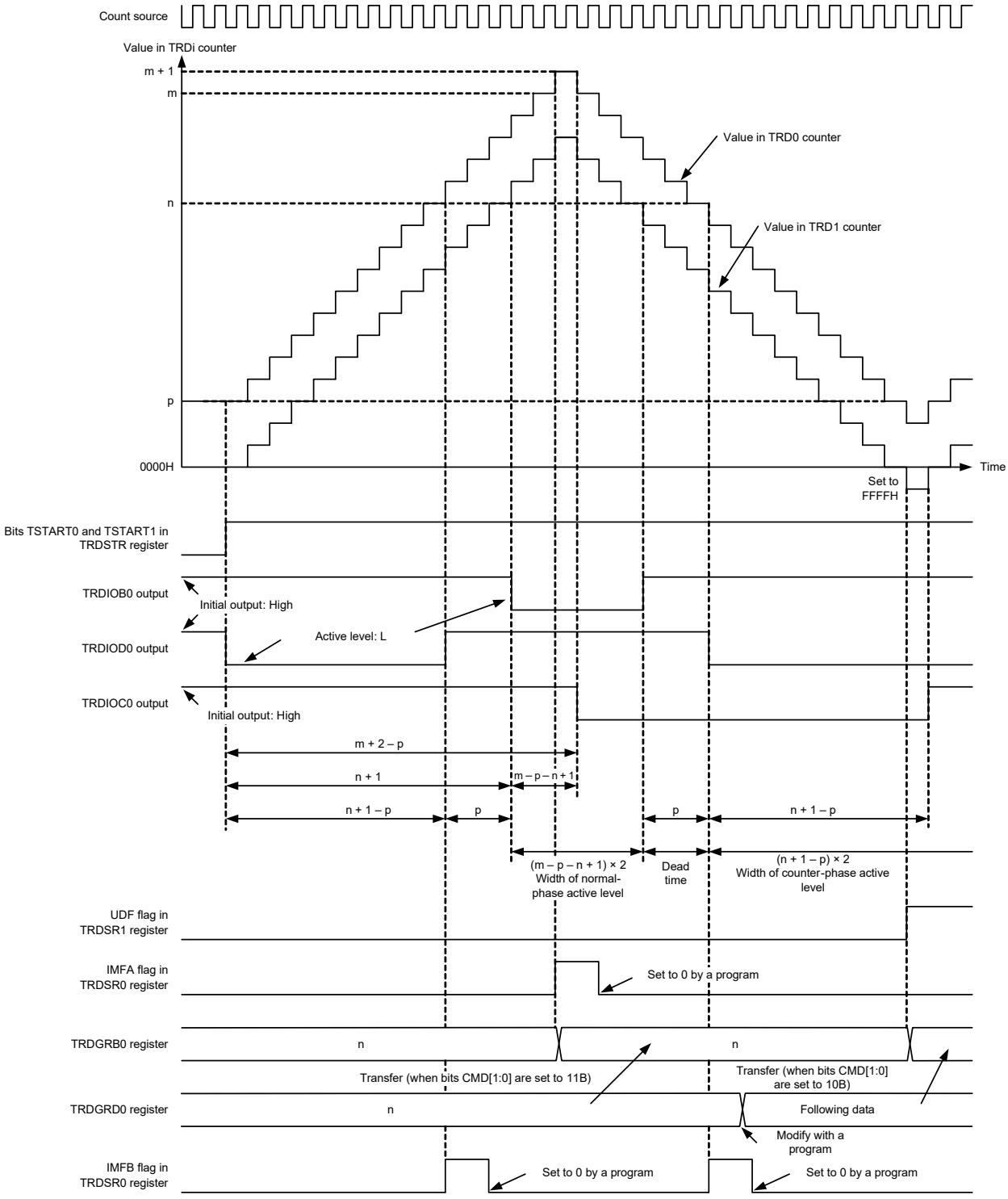


Figure 12 - 94 Operation Example in Complementary PWM Mode



Remark
CMD[1:0]: 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 counter

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. Timing of transfer 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 counter underflows.
     When bits CMD1 and CMD0 are set to 11B, the content is transferred at compare match between the TRD0 counter and TRDGRA0 register.

12.5.6 PWM3 mode

In this mode, two PWM waveforms are output with the same period.
Figure 12 - 95 is a block diagram of PWM3 mode (for timer RD20), **Table 12 - 23** lists the specifications of PWM3 mode, and **Figure 12 - 96** shows an operation example in PWM3 mode.

Figure 12 - 95 Block Diagram of PWM3 Mode (For Timer RD20)

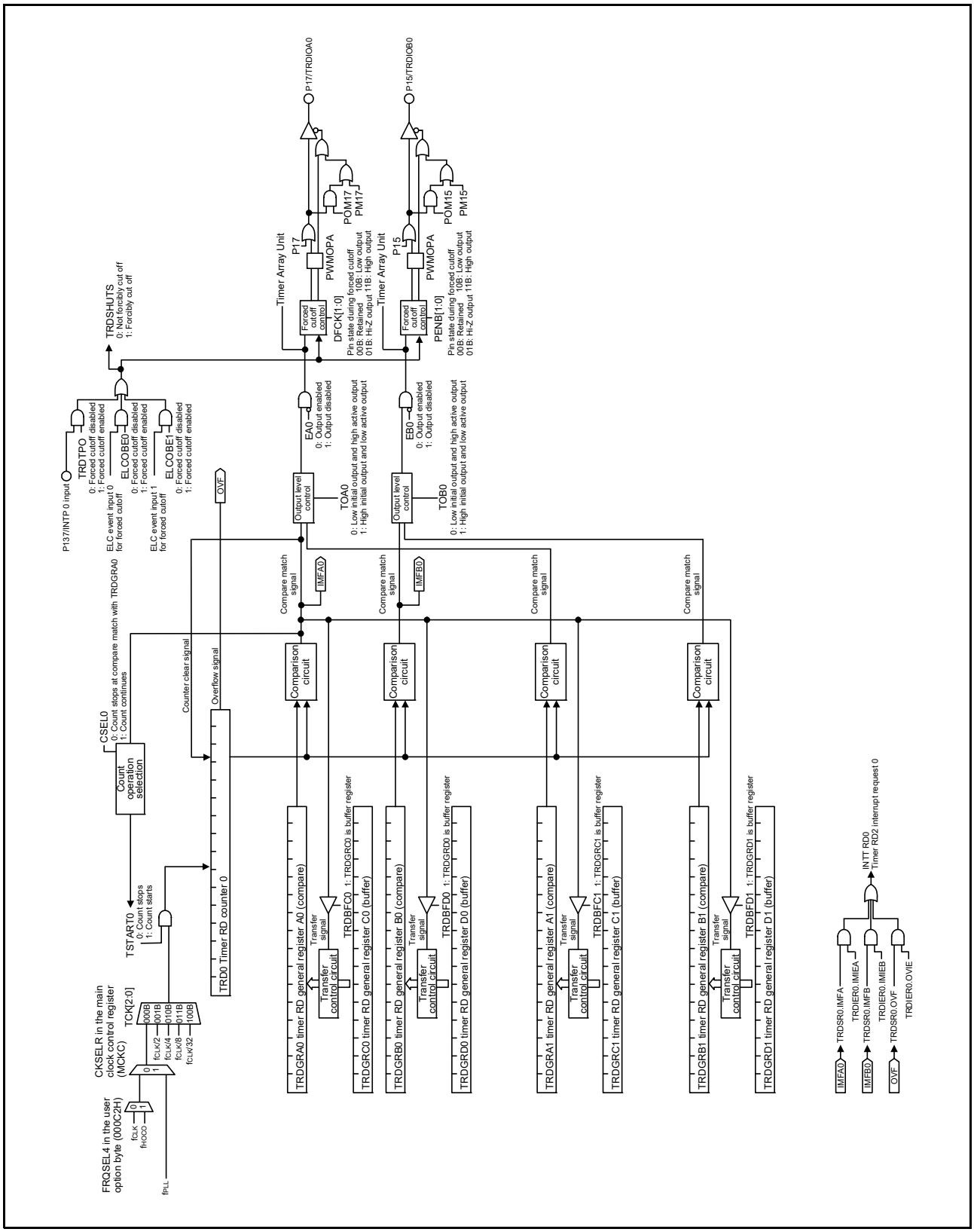
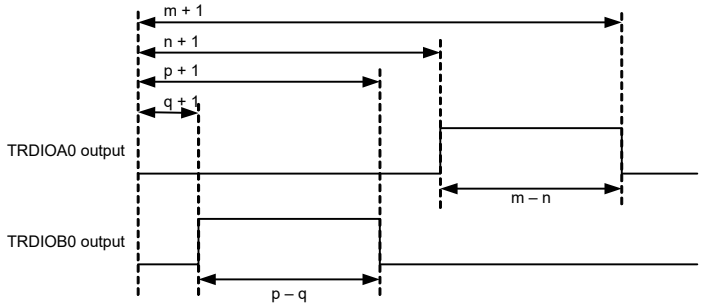


Table 12 - 23 Specifications of PWM3 Mode

Item	Specification
Count sources	f_{HOCO} **Note 1**, f_{PLL} **Note 2**, f_{CLK}, $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, $f_{CLK}/32$
Count operations	The TRD0 counter is incremented (the TRD1 counter is not used).
PWM waveform	PWM period: $1/f_K \times (m + 1)$ Active level width of TRDIOA0 output: $1/f_K \times (m - n)$ Active level width of TRDIOB0 output: $1/f_K \times (p - q)$ f_K: 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  (When high is selected as the active level)
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 pins hold the output level before the count stops. • When the CSEL0 bit in the TRDSTR register is set to 0, the count stops at a compare match with the TRDGRA0 register. The PWM output pins hold the level after output change by compare match.
Interrupt request generation timing	• Compare match (the value of the TRDi counter matches that of the TRDGRji register) • TRD0 counter overflow
TRDIOA0 and TRDIOB0 pin function	PWM output
TRDIOA0, TRDIOD0, and TRDIOA1 to TRDIOD1 pin function	I/O port
INTP0 pin function	Input of pulse output forced cutoff signal (input-only port or INTP0 interrupt input)
Read from timer	The count value can be read by reading the TRD0 counter.
Write to timer	A value can be written to the TRD0 counter.
Selectable functions	• Input of pulse output forced cutoff signal (see **12.4.4 Pulse output forced cutoff**) • Active level selectable for each pin • Buffer operation (see **12.4.2 Buffer operation**)

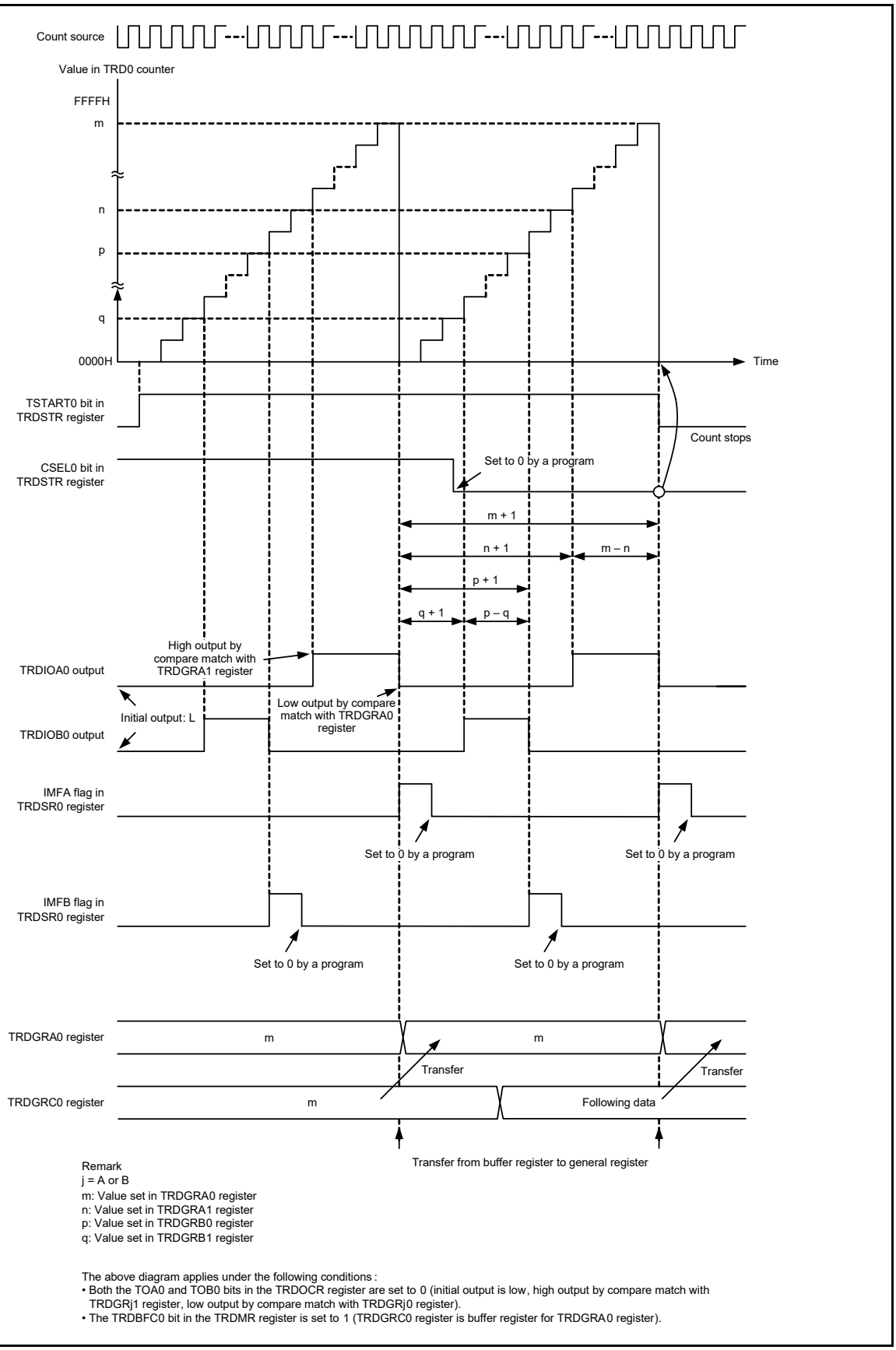
Note 1. f_{HOCO} is selected only when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H). When selecting f_{HOCO} as the count source for timer RD2, set f_{CLK} to f_{IH} before setting bit 3 (TRD0EN) in peripheral enable register 2 (PER2). When changing f_{CLK} to a clock other than f_{IH}, clear bit 3 (TRD0EN) in peripheral enable register 2 (PER2) before changing.

Note 2. f_{PLL} is supplied when the CKSELR bit is 1 in the main clock control register (MCKC).

Remark i = 0 or 1, j = A to D

1. Operation example

Figure 12 - 96 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).

12.5.7 Extended PWM mode

In this mode, two PWM waveforms can be generated from the TRD0 counter and output through the TRDIOB0 and TRDIOD0 pins, and two PWM waveforms can be generated from the TRD1 counter and output through the TRDIOB1 and TRDIOD1 pins. Alternatively, the TRD0 and TRD1 counters can be synchronized to output up to four PWM waveforms with the same period. The simultaneous update function is used to modify the settings of the PWM period and duty cycle. For the simultaneous update function, see **12.4.7 Simultaneous update of compare registers**. **Figure 12 - 97** is a block diagram of extended PWM mode, **Table 12 - 24** lists the specifications of extended PWM mode, **Figure 12 - 98** shows an operation example in extended PWM mode, and **Figure 12 - 99** shows operation examples in extended PWM mode when the duty cycle is 0% and 100%.

Figure 12 - 97 Block Diagram of Extended PWM Mode

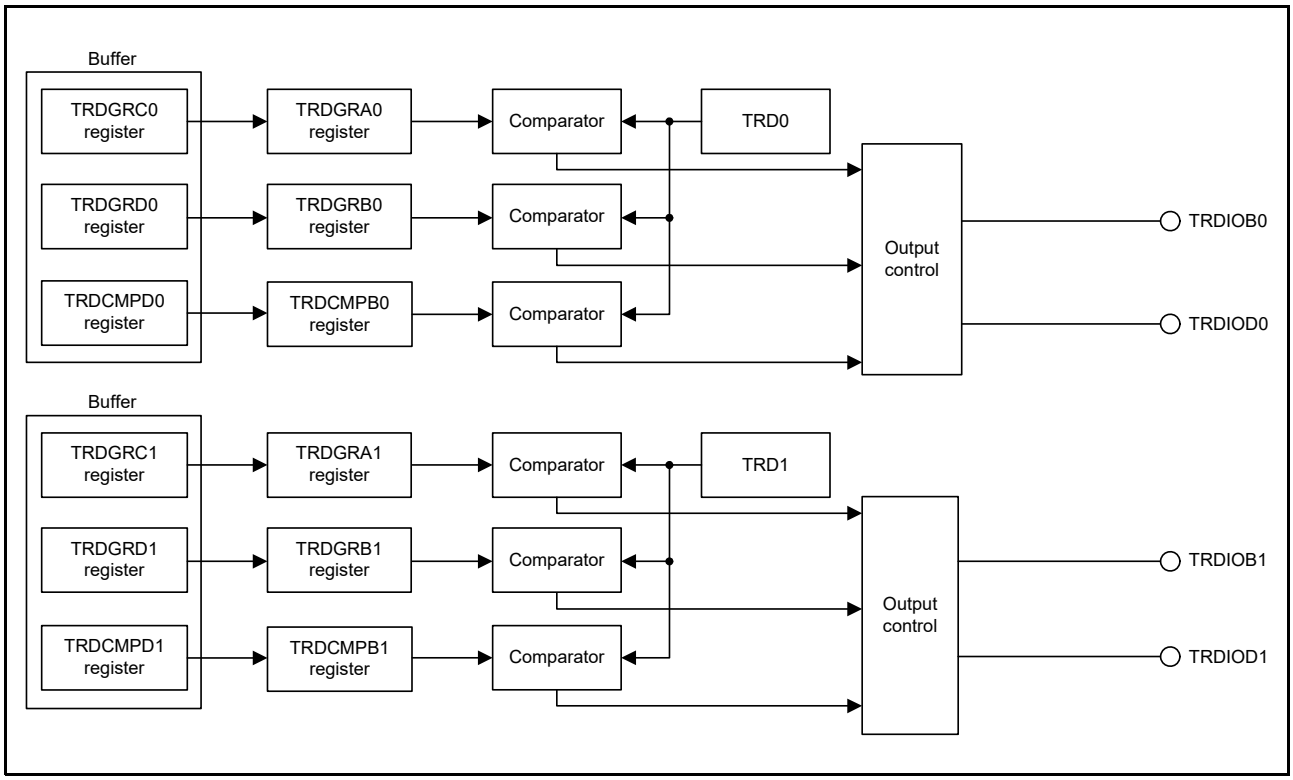
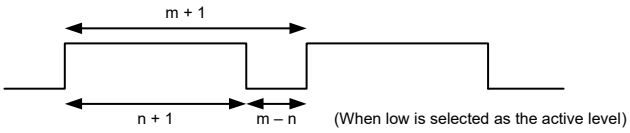


Table 12 - 24 Specifications of Extended PWM Mode

Item	Specification
Count sources	f_{HOCO} **Note 1**, f_{PLL} **Note 2**, 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: $1/f_K \times (m + 1)$ Active level width: $1/f_K \times (m - n)$ Inactive level width: $1/f_K \times (n + 1)$ f_K: 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 pins hold the output level before the count stops. • When the CSELi bit in the TRDSTR register is set to 0, the count stops at a compare match with the TRDGRAi register. The PWM output pins hold the level after the output changes by a compare match.
Interrupt request generation timing	• Compare match (the value of the TRDi counter matches that of the TRDGRhi register) • TRDi counter overflow
TRDIOA0 pin function	I/O port or TRDCLK (external clock) input
TRDIOA1, TRDIOC0, and TRDIOC1 pin function	I/O port
TRDIOB0, TRDIOD0, TRDIOB1, and TRDIOD1 pin function	I/O port or PWM output (selectable for each pin)
INTP0 pin function	Input of pulse output forced cutoff signal (input-only port or INTP0 interrupt input)
Read from timer	The count value can be read by reading the TRDi counter.
Write to timer	A value can be written to the TRDi counter.
Selectable functions	• One or two PWM output pins selectable with timer RD2i Either one pin or both pins of TRDIOBi and TRDIODi • Active level selectable for each pin • Initial output level selectable for each pin • Synchronous operation (see **12.4.3 Synchronous operation**) • Input of pulse output forced cutoff signal (see **12.4.4 Pulse output forced cutoff**)

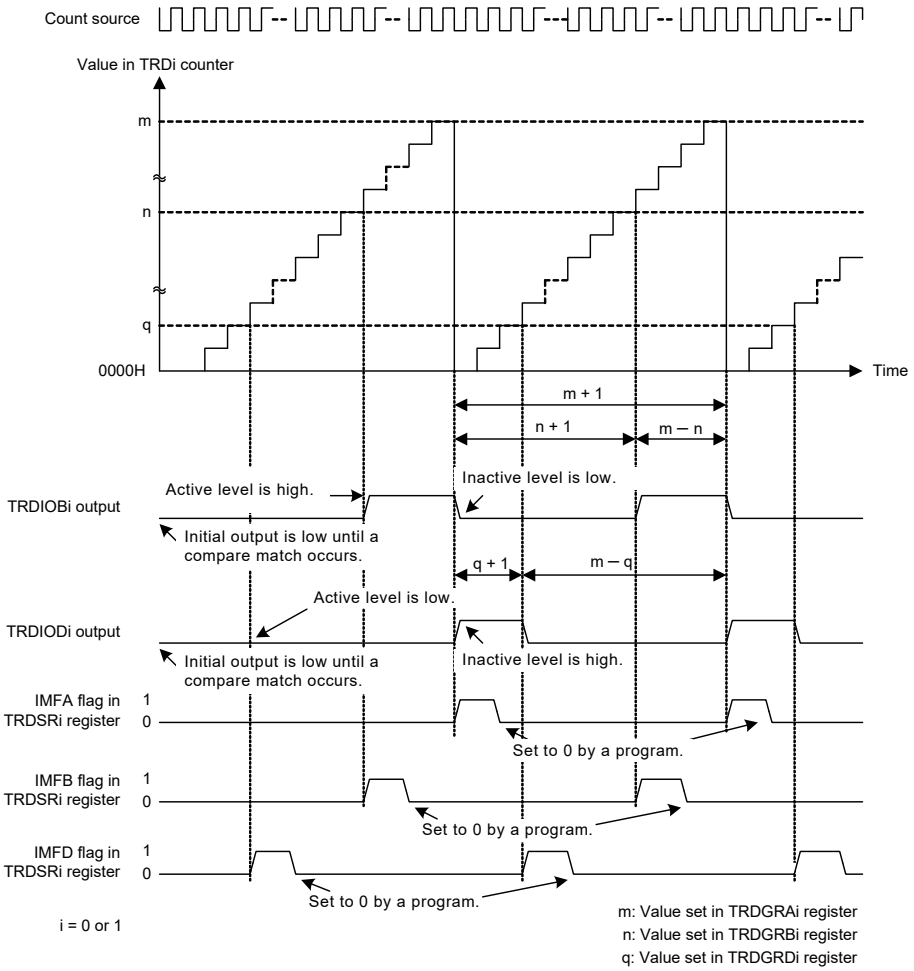
Note 1. f_{HOCO} is selected only when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H). When selecting f_{HOCO} as the count source for timer RD2, set f_{CLK} to f_{IH} before setting bit 3 (TRD0EN) in peripheral enable register 2 (PER2). When changing f_{CLK} to a clock other than f_{IH}, clear bit 3 (TRD0EN) in peripheral enable register 2 (PER2) before changing.

Note 2. f_{PLL} is supplied when the CKSELR bit is 1 in the main clock control register (MCKC).

Remark i = 0 or 1, j = B to D, h = A to D

1. Operation examples

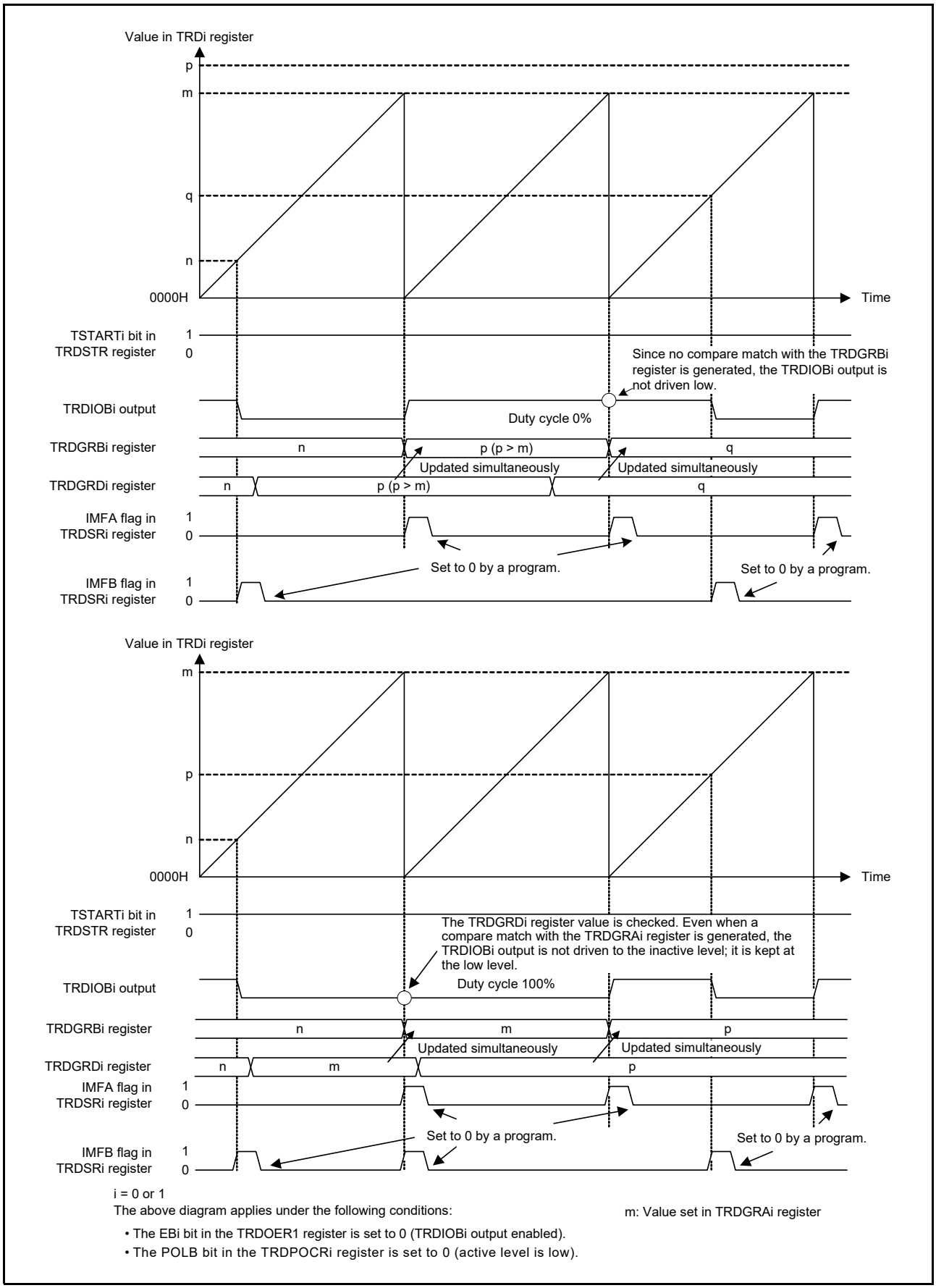
Figure 12 - 98 Operation Example in Extended PWM Mode



The above diagram applies under the following conditions:

- The EBi and EDi bits in the TRDOER1 register are set to 0 (TRDIOBi and TRDIODi output enabled).
- The TOBi bit in the TRDOCR register is set to 0 (inactive level) and the TODi bit is set to 1 (active level).
- The POLB bit in the TRDPOCRi register is set to 1 (active level is high) and the POLD bit is set to 0 (active level is low).

Figure 12 - 99 Operation Examples in Extended PWM Mode (Duty Cycle 0% and Duty Cycle 100%)



i = 0 or 1

The above diagram applies under the following conditions:

m: Value set in TRDGRAi register

- 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).

12.5.8 Extended complementary PWM mode

In this mode, three normal-phase signals and three counter-phase signals of symmetric or asymmetric PWM waveforms (six signals in total) are output with the same period (three-phase, triangular wave modulation, and with dead time). **Figure 12 - 100** is a block diagram of extended complementary PWM mode, **Table 12 - 25** lists the specifications of extended complementary PWM mode, **Figure 12 - 101** shows an example of asymmetric output waveforms in extended complementary PWM mode, **Figure 12 - 102** shows an example of symmetric output waveforms in extended complementary PWM mode, and **Figure 12 - 103** shows an operation example in extended complementary PWM mode.

Figure 12 - 100 Block Diagram of Extended Complementary PWM Mode

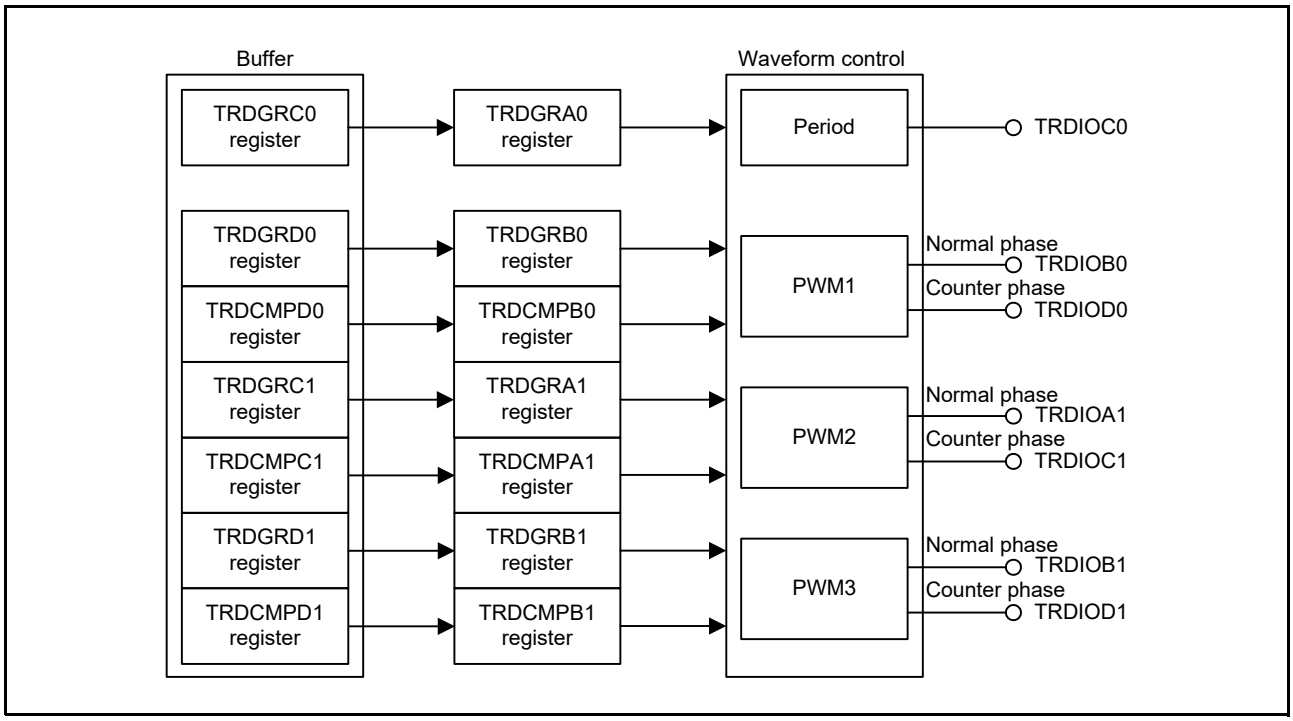


Table 12 - 25 Specifications of Extended Complementary PWM Mode (1/2)

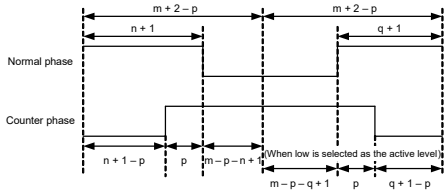
Item	Specification
Count sources	f_{HOCO} **Note 1**, f_{PLL} **Note 2**, f_{CLK}, $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, $f_{CLK}/32$ External signal input to the TRDCLK pin (active edge selected by a program) Set bits TCK2 to TCK0 in the TRDCR1 register to the same value (same count source) as that specified in bits TCK2 to TCK0 in the TRDCR0 register.
Count operations	Increment or decrement. When a compare match between the TRD0 counter and TRDGRA0 register is detected during increment operation, both the TRD0 and TRD1 counters are decremented. When the TRD1 counter value changes from 0000H to FFFFH during decrement operation, both the TRD0 and TRD1 counters are incremented.
PWM waveform	PWM period: $1/f_K \times (m + 2 - p) \times 2$ **Note 3** Dead time: p Active level width of normal phase: $1/f_K \times \{(m - n - p + 1) + (m - p - q + 1)\}$ Active level width of counter phase: $1/f_K \times \{(n + 1 - p) + (q + 1 - p)\}$ Output with 100% duty cycle: Compare register setting = 0000H Output with 0% duty cycle: Compare register setting ≥ TRDGRA0 setting f_K: 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) q: Value set in the TRDCMPB0 register (PWM1 output) Value set in the TRDCMPA1 register (PWM2 output) Value set in the TRDCMPB1 register (PWM3 output) p: Value set in the TRD0 counter 
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 pins output the initial output level selected by the TRDOCR register.)
Interrupt request generation timing	• Compare match between the TRD0 counter and TRDGRA0 register (interrupt request 0) • TRD1 counter underflow (interrupt request 1)
Timing of updating buffers **Note 4**	The buffers are updated simultaneously when the TRD1 counter underflows (or earlier than the actual underflow timing by the amount of dead time) while the RSF flag in the TRDRSF1 register is 1.
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 of the PWM period to the PWMOPA (output inverted every half PWM period or earlier than that by the amount of dead time)
INTP0 pin function	Input of pulse output forced cutoff signal (input-only port or INTP0 interrupt input)
Read from timer	The count value can be read by reading the TRDi counter.
Write to timer	A value can be written to the TRDi counter.

Table 12 - 25 Specifications of Extended Complementary PWM Mode (2/2)

Item	Specification
Selectable functions	• Input of pulse output forced cutoff signal (see **12.4.4 Pulse output forced cutoff**) • The normal-phase and counter-phase active levels and initial output levels**Note 5** are selected individually. • Symmetric or asymmetric output waveforms are selected. • A/D conversion trigger output

Note 1. f_{HOCO} is selected only when the FRQSEL4 bit is 1 in the user option byte (000C2H/010C2H). When selecting f_{HOCO} as the count source for timer RD2, set f_{CLK} to f_{IH} before setting bit 3 (TRD0EN) in peripheral enable register 2 (PER2). When changing f_{CLK} to a clock other than f_{IH}, clear bit 3 (TRD0EN) in peripheral enable register 2 (PER2) before changing.

Note 2. f_{PLL} is supplied when the CKSELR bit is 1 in the main clock control register (MCKC).

Note 3. After counting starts, the PWM period is fixed.

Note 4. Timing of transfer from the buffer registers:
Values are transferred from the TRDGRD0, TRDGRC1, TRDGRD1, TRDCMPD0, TRDCMPC1, and TRDCMPD1 registers to the TRDGRB0, TRDGRA1, TRDGRB1, TRDCMPB0, TRDCMPA1, and TRDCMPB1 registers when the TRD1 counter underflows (earlier than the actual underflow timing by the amount of dead time). The simultaneous update function is used to modify the compare registers in this mode. After setting values in the buffer registers, set the RDT bit in the TRDRDT1 register to 1. For details of the simultaneous update function, see **12.4.7 Simultaneous update of compare registers**.

Note 5. Use the TRDOCR register to specify the initial level to be output from each pin in extended complementary PWM mode. After counting begins, the active levels selected by the OLS1 and OLS0 bits in the TRDFCR register become valid. In the TRDOCR register, select the inactive and active levels for the normal-phase and counter-phase outputs, respectively. However, to output a waveform with 100% duty cycle in the first cycle, select the inactive level for counter-phase output.

Remark i = 0 or 1, j = A to D

1. Operation examples

Figure 12 - 101 Example of Asymmetric Output Waveforms in Extended Complementary PWM Mode

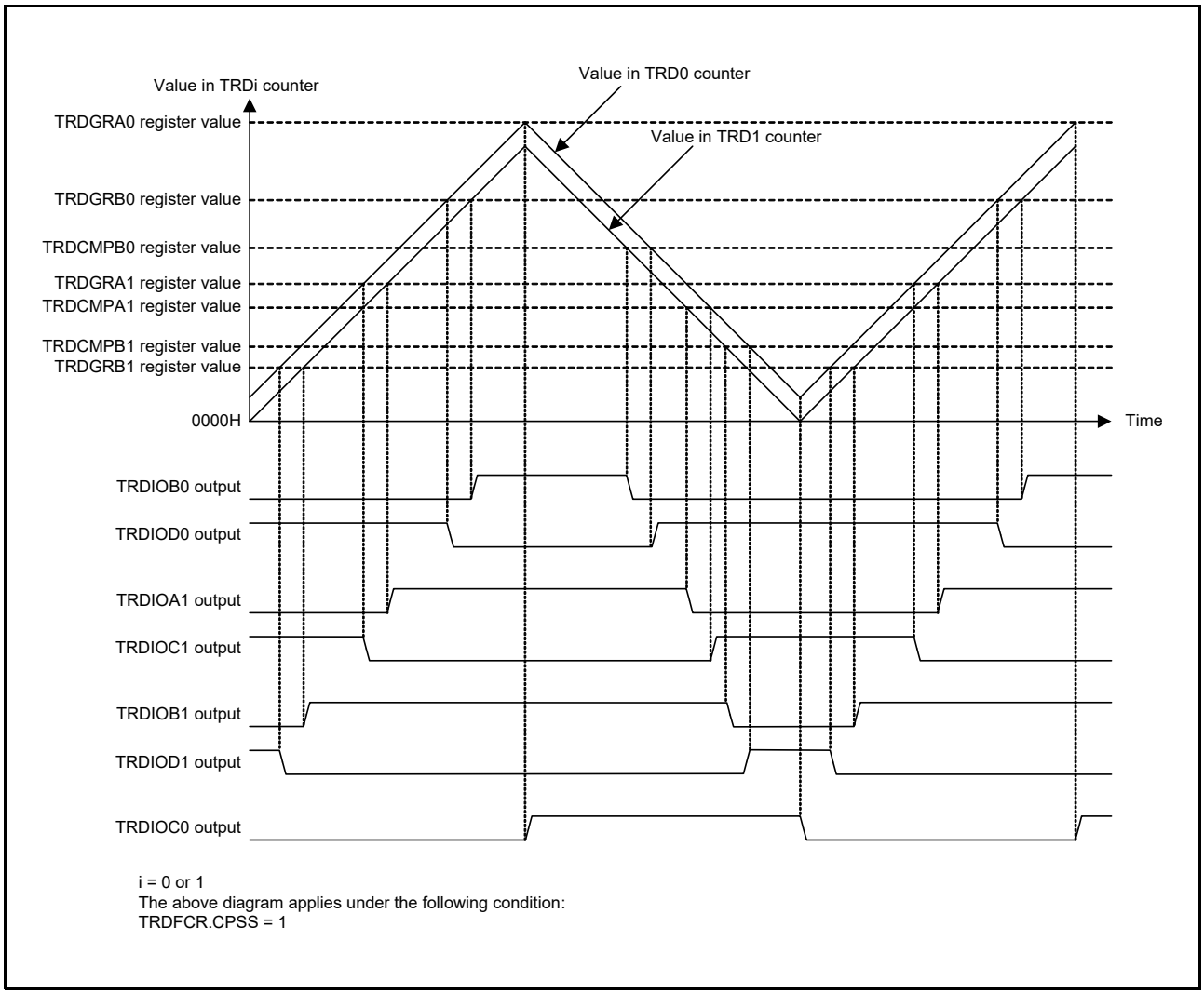


Figure 12 - 102 Example of Symmetric Output Waveforms in Extended Complementary PWM Mode

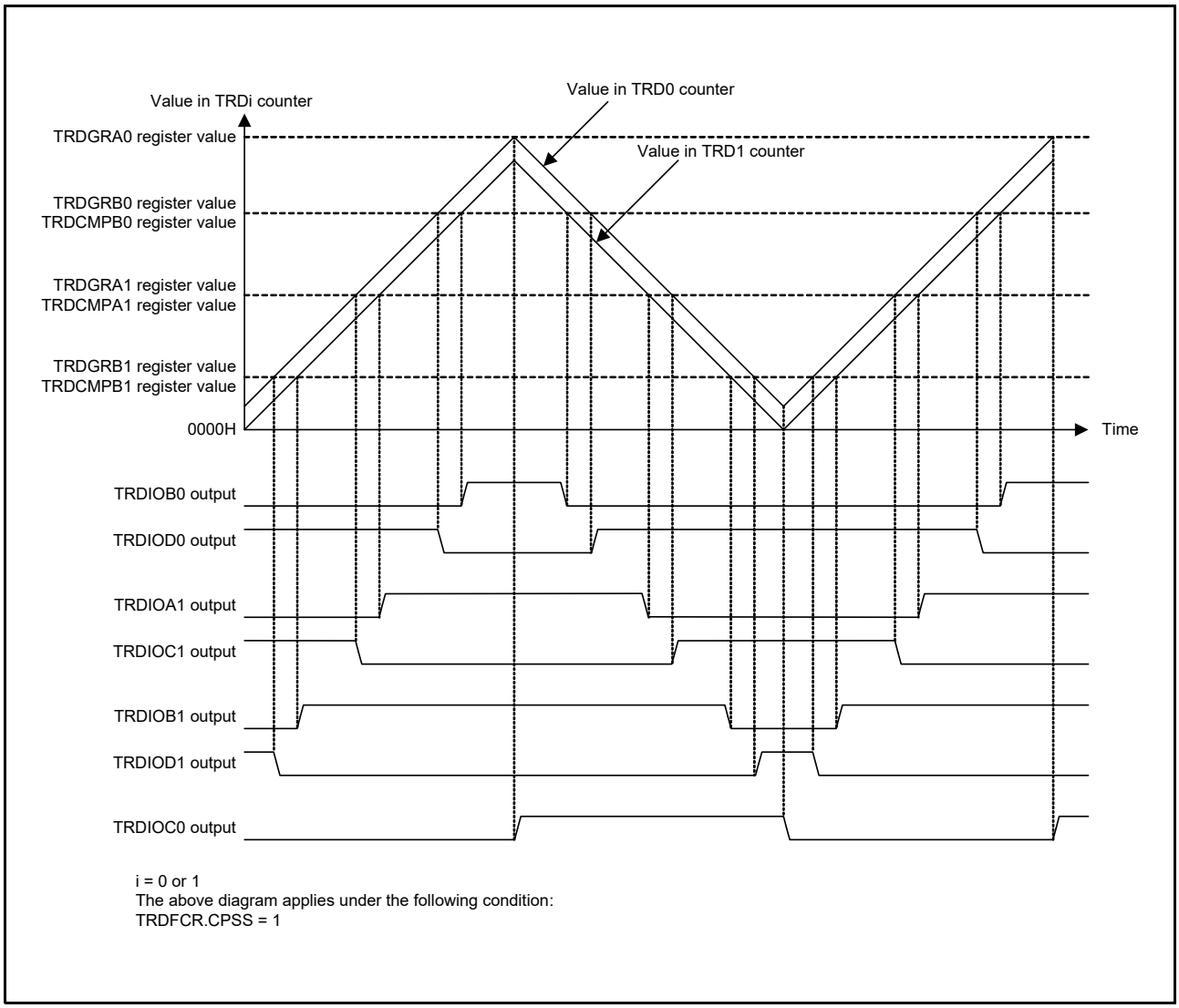
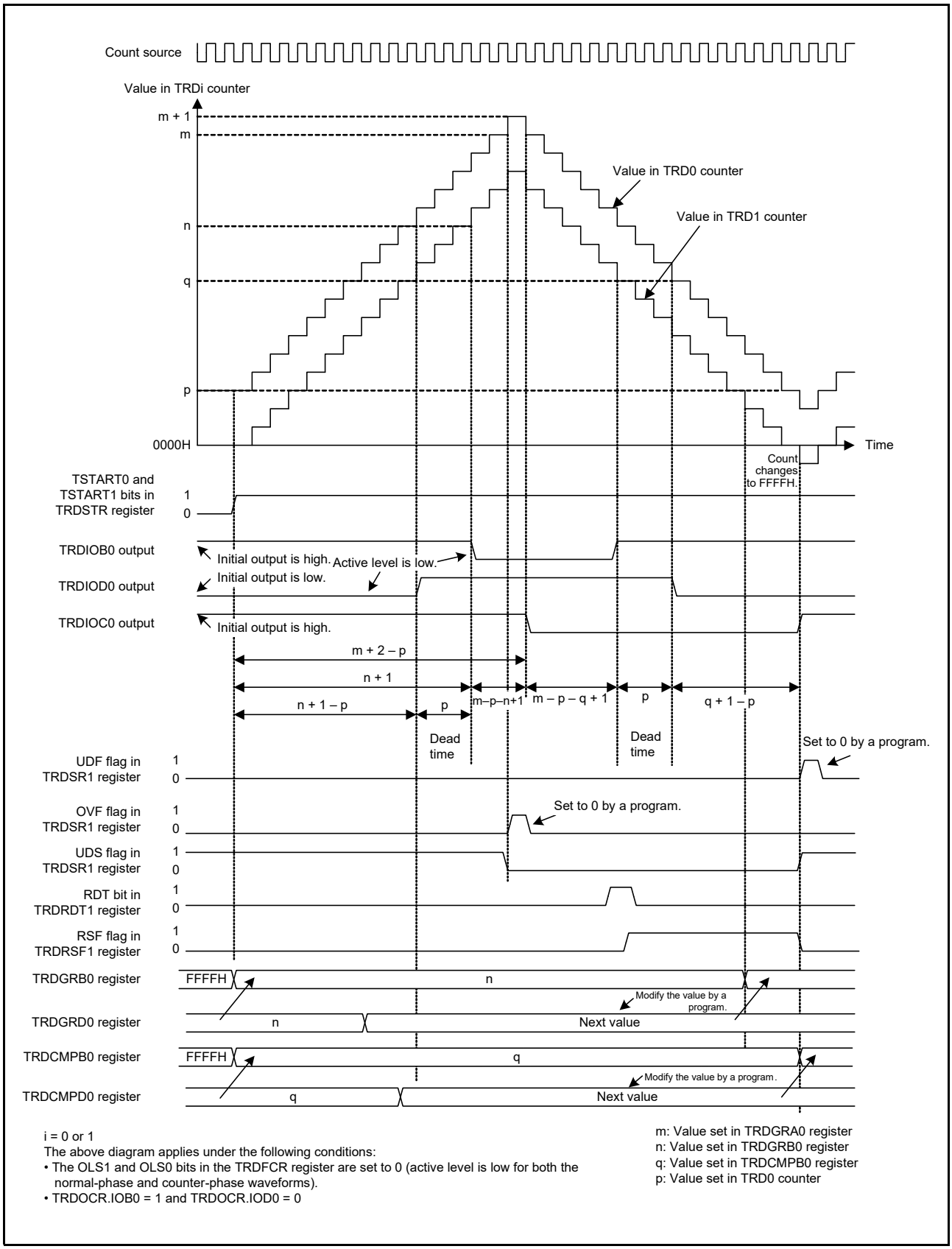


Figure 12 - 103 Operation Example in Extended Complementary PWM Mode



i = 0 or 1
The above diagram applies under the following conditions:
- The OLS1 and OLS0 bits in the TRDFCR register are set to 0 (active level is low for both the normal-phase and counter-phase waveforms).
- TRDOCR.IOB0 = 1 and TRDOCR.IOD0 = 0

m: Value set in TRDGRA0 register
n: Value set in TRDGRB0 register
q: Value set in TRDCMPB0 register
p: Value set in TRD0 counter

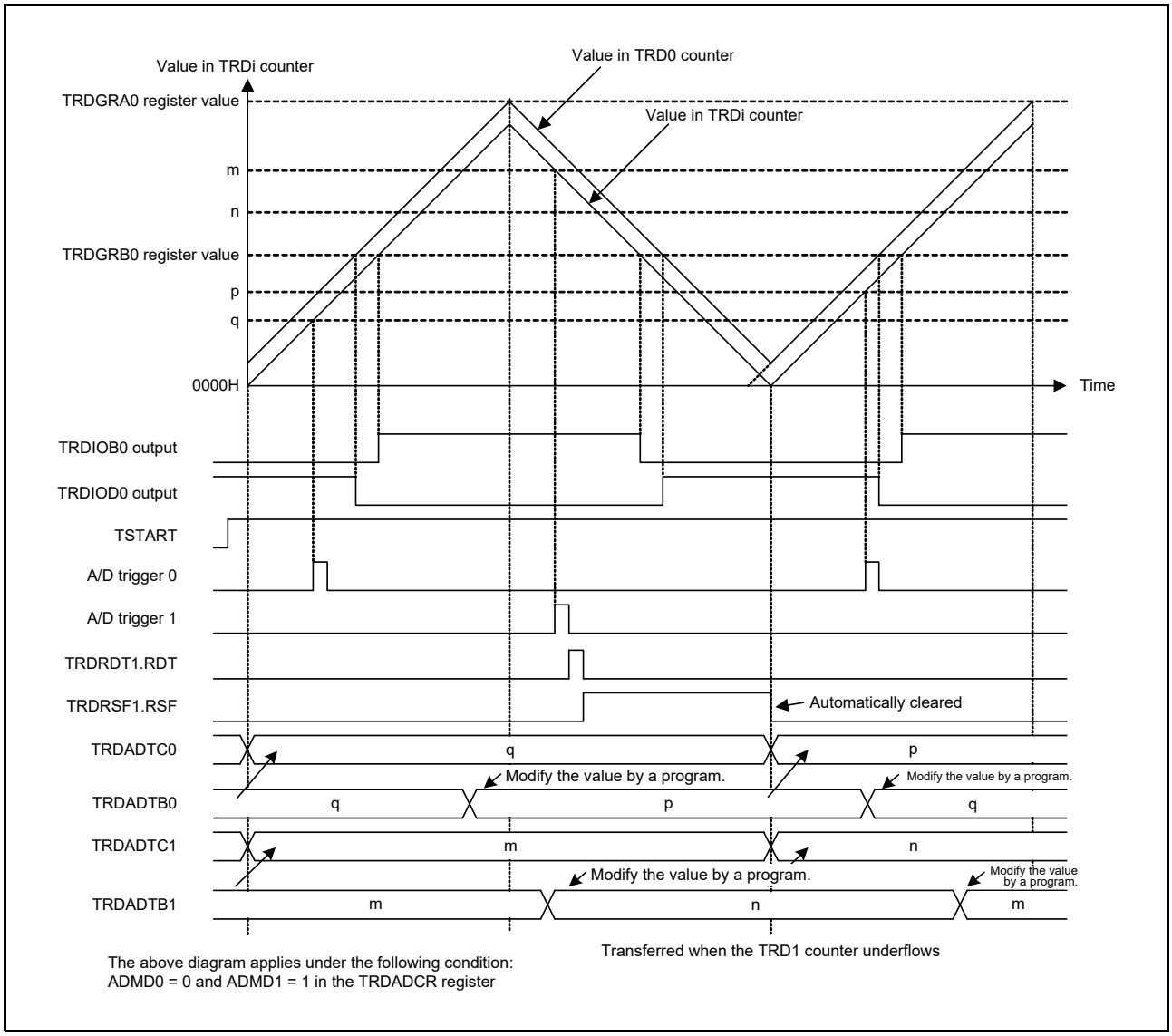
2. Generating A/D conversion triggers

Two A/D conversion triggers can be generated at desired timing. Each trigger is linked with a buffer register and a compare register. To generate a trigger, set the ADEi bit in the TRDADCR register to 1 (i = 0 or 1). Triggers are output during either increment or decrement operation, which can be selected by the ADMDi bit in the TRDADCR register (i = 0 or 1). An A/D conversion trigger is output at a compare match with the TRD1 counter. This function uses the simultaneous update function for the compare registers to transfer data from the TRDADTBi (i = 0 or 1) buffer registers to the TRDADTCi (i = 0 or 1) compare registers. After modifying the necessary buffer registers, set the RDT bit in the TRDRDT1 register to 1.

For details on the simultaneous update function, see **12.4.7 Simultaneous update of compare registers**.

Figure 12 - 104 shows the timing to output A/D conversion triggers.

Figure 12 - 104 Timing to Output A/D Conversion Triggers



<R> 12.5.9 Timer-KB PWM output gating mode

This mode implements the PWM output gating function interlocked with timer KB3. Up to six PWM waveforms can be output to correspond to the signals output from timer KB3. The following two modes using the TRD0 and TRD1 counters in timer RD2 are available.

Standalone mode: Timer RD2 runs or stops as a stand-alone unit; that is, its operation is not interlocked with timer KB3.
Timer KB3 interlocked mode: Operation of timer RD2 is interlocked with that of timer KB3. That is, timer RD2 runs and stops at the same times as timer KB3 (TKBCE0 = 1 or 0).

In timer-KB PWM output gating mode, the TRDTKBOUT5 to TRDTKBOUT0 and TRDIOji signals are output. They are to be sent to timer KB3. TRDTKBOUT3 and TRDIOA1, and TRDTKBOUT5 and TRDIOD0 are output in accord with the value of the TRD0 counter and the PWM period specified in TRDGRA0. TRDTKBOUT0 and TRDIOB1, TRDTKBOUT1 and TRDIOC1, TRDTKBOUT2 and TRDIOD1, and TRDTKBOUT3 and TRDIOA1 are output in accord with the value of the TRD1 counter and the PWM period specified in TRDGRA1. PWM waveforms with the same period can be output by synchronizing the TRD0 and TRD1 counters. The TRDIOji output signals indicate the values of the TRDTKBOUT5 to TRDTKBOUT0 signals that were output one clock cycle earlier. The TRDTKBOUT5 to TRDTKBOUT0 outputs are enabled or disabled through the setting of the TRDPMR register, and the TRDIOji outputs are enabled or disabled through the setting of the TRDOER1 register. While counting is stopped (TRDSTR.TSTARTi = 0) or the output of a signal is disabled, the initial output level specified in the TRDOCR register is output instead. When the output from a pin is enabled at the start of counting (TRDSTR.TSTARTi = 1), a PWM waveform according to the setting of the TRDPOCRi register is output from the pin. Even when the initial output level and active level are set to the same value, a PWM waveform can be output from the first cycle after counting begins.

1. Standalone mode

Figure 12 - 105 is a block diagram of timer-KB PWM output gating mode (standalone mode), **Table 12 - 26** shows the specifications of timer-KB PWM output gating mode (standalone mode), **Figure 12 - 106** shows an operation example in timer-KB PWM output gating mode (standalone mode), and **Figure 12 - 107** shows an operation example in timer-KB PWM output gating mode (when operation is stopped in standalone mode).

Figure 12 - 105 Block Diagram of Timer-KB PWM Output Gating Mode (Standalone Mode)

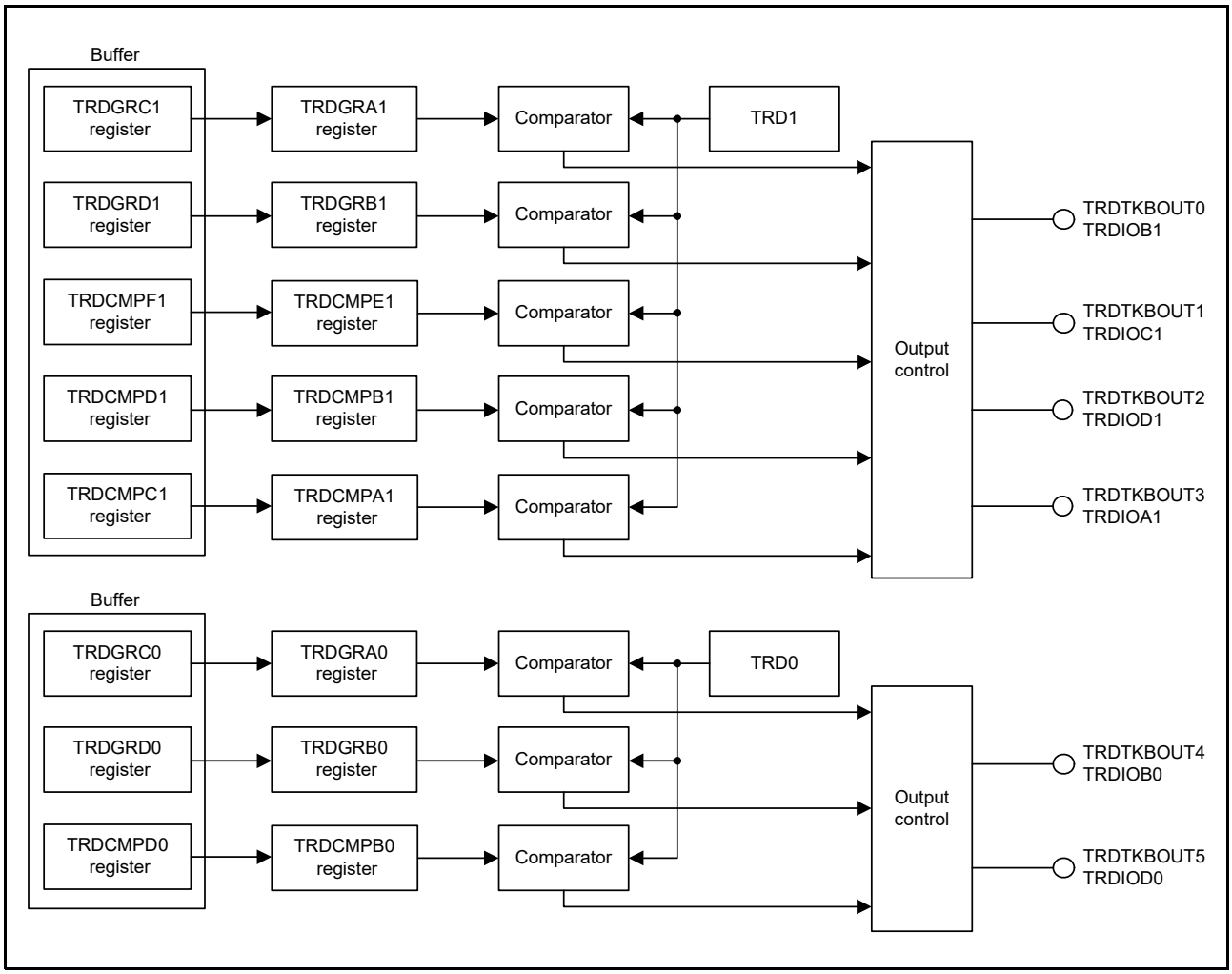
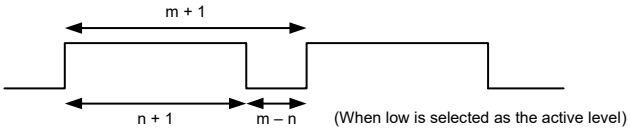


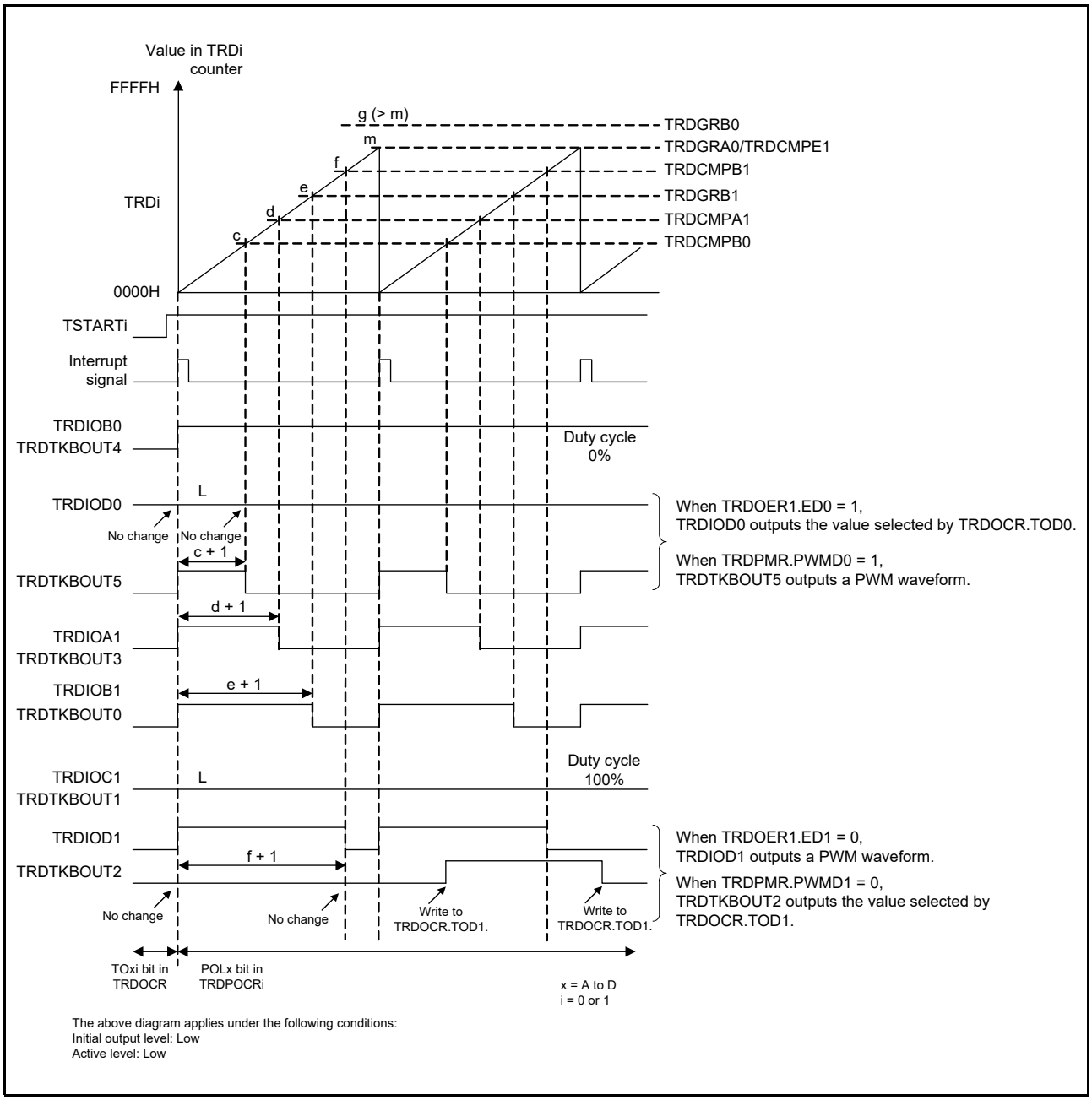
Table 12 - 26 Specifications of Timer-KB PWM Output Gating Mode (Standalone Mode)

Item	Specification
Count sources	TKBTCK0, TKBTCK1
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 or TRDCMPm register 
Count start condition	TRD0: 1 (count starts) is written to the TSTART0 bit in the TRDSTR register. TRD1: 1 (count starts) is written to the TSTART1 bit in the TRDSTR register.
Count stop condition	• When the CSELi bit in the TRDSTR register is set to 1 TRD0: 0 (count stops) is written to the TSTART0 bit. TRD1: 0 (count stops) is written to the TSTART1 bit (The PWM output pins output the initial output level selected by the TRDOCR register.)
Interrupt request generation timing	• Match with the specified period value (the value of the TRDi counter matches that of the TRDGRAi register) • Counter operation begins.
Timing of updating buffers	The buffers are updated simultaneously when a match with the specified period value is detected (the value of the TRDi counter matches that of the TRDGRAi register) while the RSF flag in the TRDRSF1 register is 1.
TRDIOA0 pin function	I/O port
TRDIOC0 pin function	I/O port
TRDIOB0 pin function	PWM output, timer-KB PWM output gating signal (TRDTKBOUT4)
TRDIOD0 pin function	PWM output, timer-KB PWM output gating signal (TRDTKBOUT5)
TRDIOA1 pin function	PWM output, timer-KB PWM output gating signal (TRDTKBOUT3)
TRDIOB1 pin function	PWM output, timer-KB PWM output gating signal (TRDTKBOUT0)
TRDIOC1 pin function	PWM output, timer-KB PWM output gating signal (TRDTKBOUT1)
TRDIOD1 pin function	PWM output, timer-KB PWM output gating signal (TRDTKBOUT2)
INTP0 pin function**Note**	Input of pulse output forced cutoff signal (input-only port or INTP0 interrupt input)
Read from timer	The count value can be read by reading the TRDi counter.
Write to timer	A value can be written to the TRDi counter.
Selectable functions	• The number of timer-KB PWM output gating signals to be output is selected (TRDPMR). Either one signal or multiple signals of TRDTKBOUT5 to TRDTKBOUT0 • Synchronous operation (see **12.4.3 Synchronous operation**) • PWM output control period is selected. Six PWM waveforms with the same period or two and four PWM waveforms with two different periods can be selected. To output six PWM waveforms with the same period, specify the same period in the TRDGRA0 and TRDGRA1 registers and write 1 to the TSTART0 and TSTART1 bits in the TRDSTR register to start operation. To stop the operation, write 0 to the TSTART0 and TSTART1 bits. To update the compare registers simultaneously, execute 16-bit access to the TRDRDT0 and TRDRDT1 registers to set the RDT bit in both registers at the same time.

Note The pulse output forced cutoff function using the INTP0 input is only applied to the TRDIOji signals; the TRDTKBOUT signals are not cut off.

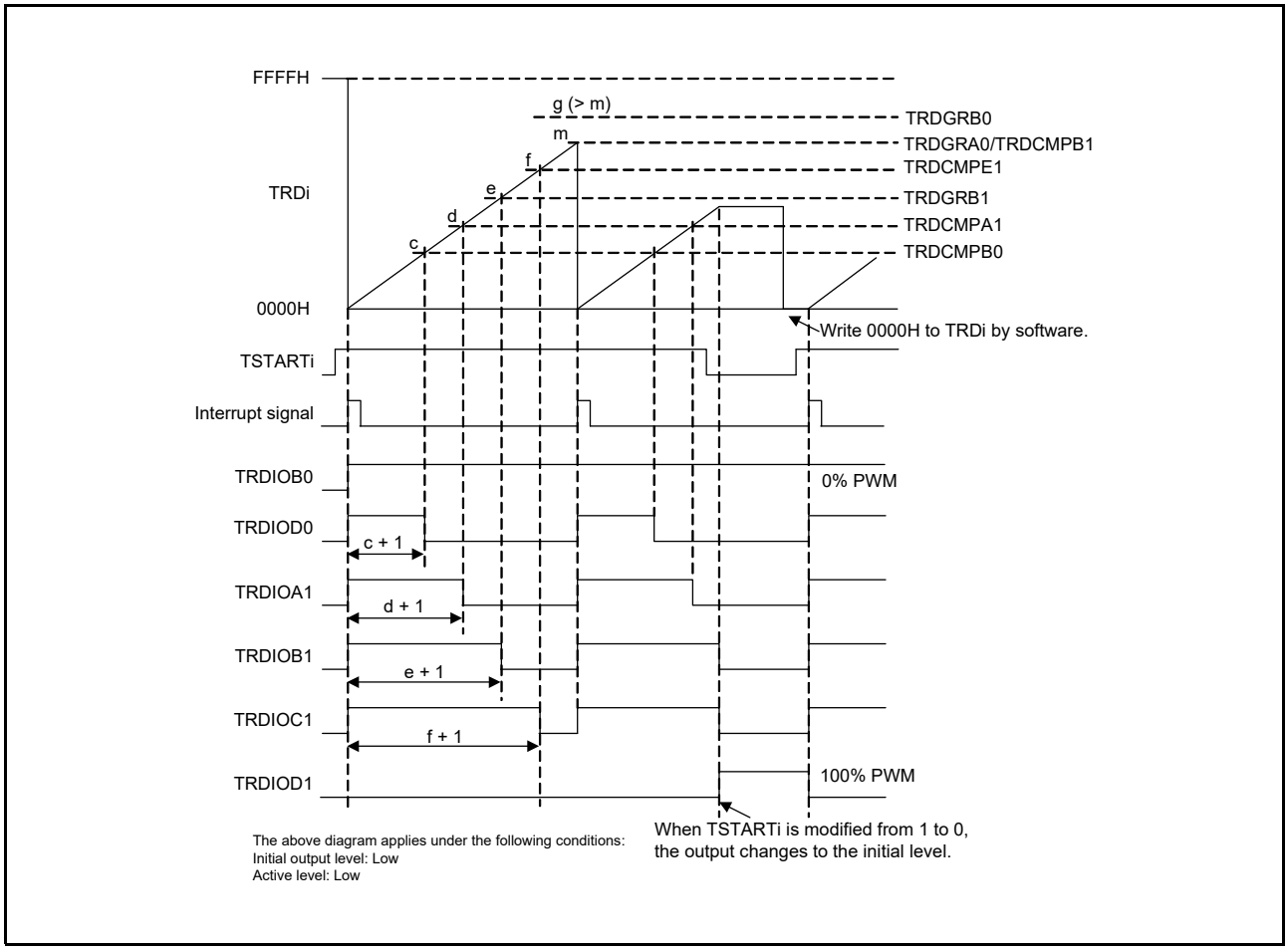
Remark i = 0 or 1, j = B to D

Figure 12 - 106 Operation Example in Timer-KB PWM Output Gating Mode (Standalone Mode)



The above diagram applies under the following conditions:
Initial output level: Low
Active level: Low

Figure 12 - 107 Operation Example in Timer-KB PWM Output Gating Mode (When Operation Is Stopped in Standalone Mode)



2. Timer KB3 interlocked mode

Table 12 - 27 shows the specifications of timer-KB PWM output gating mode (interlocked mode) and **Figure 12 - 108** shows an operation example in timer-KB PWM output gating mode (interlocked mode). For a block diagram of timer-KB PWM output gating mode, see **Figure 12 - 105**.

Table 12 - 27 Specifications of Timer-KB PWM Output Gating Mode (Interlocked Mode) (1/2)

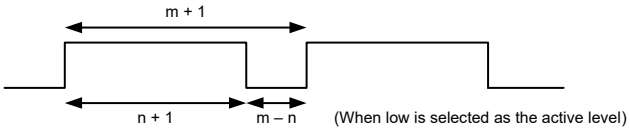
Item	Specification
Count sources	TKBTCK0, TKBTCK1
Count operations	Increment
PWM waveform	PWM period: 1/fK × (m + 1) Active level width: 1/fK × (m – n) Inactive level width: 1/fK × (n + 1) fK: Frequency of count source m: Value set in the TRDGRAi register n: Value set in the TRDGRji register or TRDCMPm register 
Count start condition	1 is written to the TKBCE0 bit of the TKBCTL01 register in timer KB3.
Count stop condition	0 is written to the TKBCE0 bit of the TKBCTL01 register in timer KB3. (The PWM output pins output the initial output level selected by the TRDOCR register.)
Interrupt request generation timing	• Match with the specified period value (the value of the TRDi counter matches that of the TRDGRAi register) • Counter operation begins.
Timing of updating buffers	The buffers are updated simultaneously when a match with the specified period value is detected (the value of the TRDi counter matches that of the TRDGRAi register) while the RSF flag in the TRDRSF1 register is 1.
TRDIOA0 pin function	I/O port
TRDIOC0 pin function	I/O port
TRDIOB0 pin function	PWM output, timer-KB PWM output gating signal (TRDTKBOUT4)
TRDIOD0 pin function	PWM output, timer-KB PWM output gating signal (TRDTKBOUT5)
TRDIOA1 pin function	PWM output, timer-KB PWM output gating signal (TRDTKBOUT3)
TRDIOB1 pin function	PWM output, timer-KB PWM output gating signal (TRDTKBOUT0)
TRDIOC1 pin function	PWM output, timer-KB PWM output gating signal (TRDTKBOUT1)
TRDIOD1 pin function	PWM output, timer-KB PWM output gating signal (TRDTKBOUT2)
INTP0 pin function**Note**	Input of pulse output forced cutoff signal (input-only port or INTP0 interrupt input)
Read from timer	The count value can be read by reading the TRDi counter.
Write to timer	A value can be written to the TRDi counter.

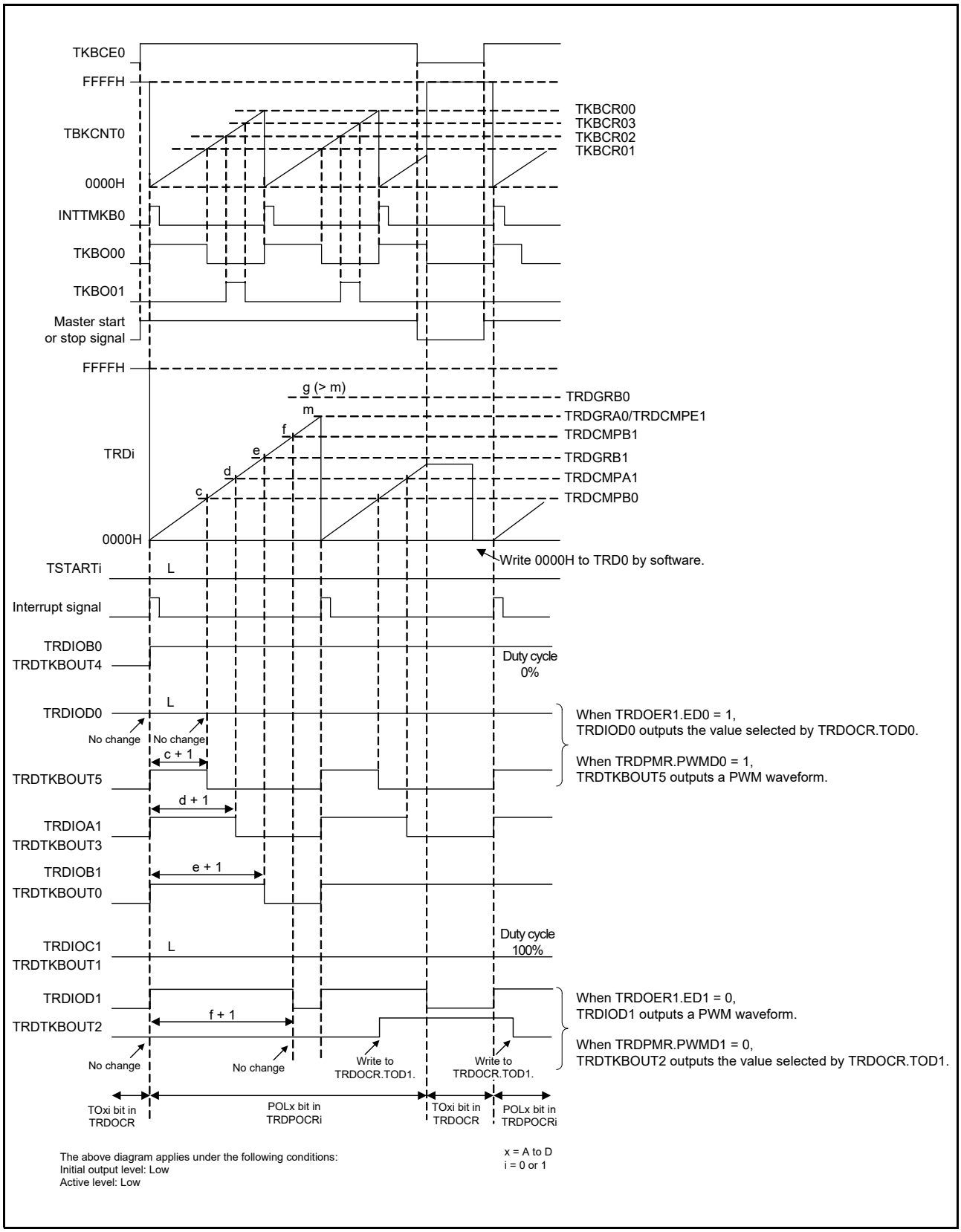
Table 12 - 27 Specifications of Timer-KB PWM Output Gating Mode (Interlocked Mode) (2/2)

Item	Specification
Selectable functions	• The number of timer-KB PWM output gating signals to be output is selected (TRDPMR). Either one signal or multiple signals of TRDTKBOUT5 to TRDTKBOUT0 • Synchronous operation (see **12.4.3 Synchronous operation**) • PWM output control period is selected. Six PWM waveforms with the same period or two and four PWM waveforms with two different periods can be selected. To output six PWM waveforms with the same period, specify the same period in the TRDGRA0 and TRDGRA1 registers and write 1 to the TKBCE0 bit in the TKBCTL01 register to start operation. To stop the operation, write 0 to the TKBCE0 bit. To update the compare registers simultaneously, execute 16-bit access to the TRDRDT0 and TRDRDT1 registers to set the RDT bit in both registers at the same time.

Note The pulse output forced cutoff function using the INTP0 input is only applied to the TRDIOji signals; the TRDTKBOUT signals are not cut off.

Remark i = 0 or 1, j = B to D

Figure 12 - 108 Operation Example in Timer-KB PWM Output Gating Mode (Interlocked Mode)



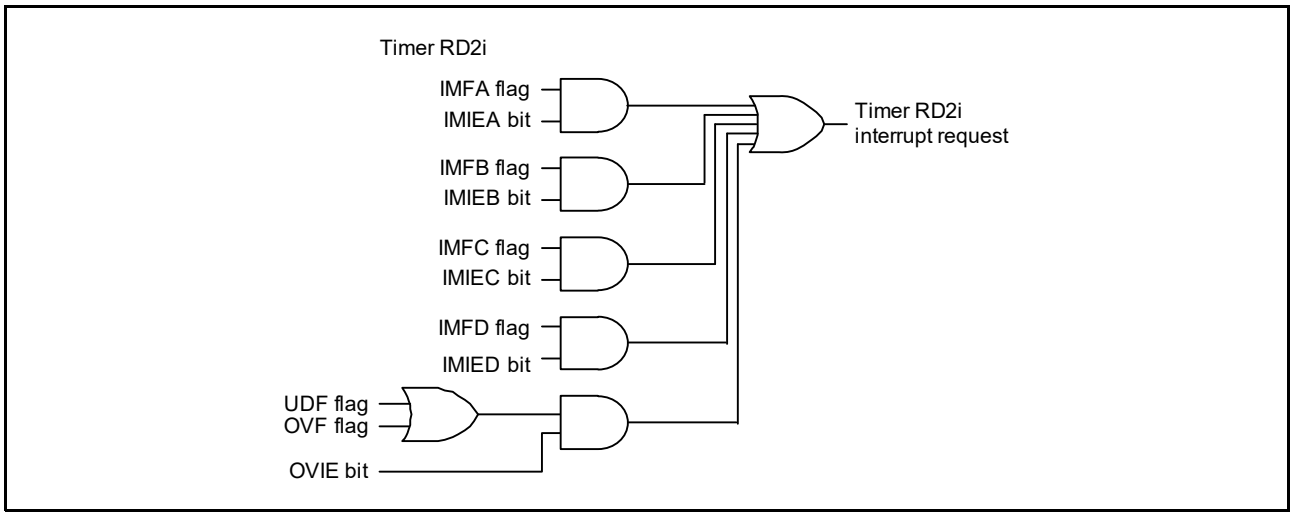
12.6 Timer RD2 Interrupts

When timer RD2 is not set to extended complementary PWM mode or timer-KB PWM output gating mode, timer RD2 generates a timer RD2i (i = 0 or 1) interrupt request from six sources each for timer RD20 and timer RD21. **Table 12 - 28** lists the registers associated with timer RD2 interrupts and **Figure 12 - 109** shows a block diagram of the timer RD2 interrupt section.

<R> Table 12 - 28 Registers Associated with Timer RD2 Interrupts

	Timer RD Status Register	Timer RD Interrupt Enable Register	Interrupt Request Flag (Register)	Interrupt Mask Flag (Register)	Priority Specification Flag (Register)
Timer RD20	TRDSR0	TRDIER0	TRDIF0 (IF1H)	TRDMK0 (MK1H)	TRDPR00 (PR01H) TRDPR10 (PR11H)
Timer RD21	TRDSR1	TRDIER1	TRDIF1 (IF1H)	TRDMK1 (MK1H)	TRDPR01 (PR01H) TRDPR11 (PR11H)

Figure 12 - 109 Block Diagram of Timer RD2 Interrupt Section



Remark i = 0 or 1
IMFA, IMFB, IMFC, IMFD, OVF, UDF: TRDSRi register flags
IMIEA, IMIEB, IMIEC, IMIED, OVIE: TRDIERi register bits

Since an interrupt request (timer RD2 interrupt) is generated by a combination of multiple interrupt request sources for timer RD2, the following differences from other maskable interrupts except timer RG interrupt apply:

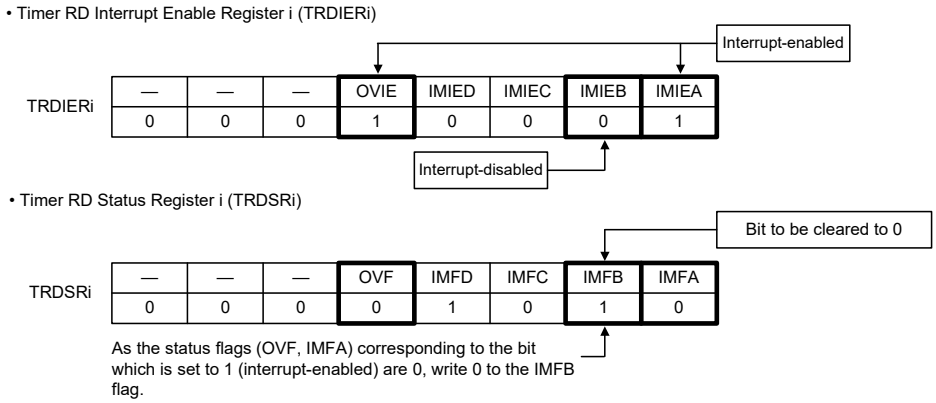
- When a bit in the TRDSRi register is 1 and the corresponding bit in the TRDIERi register is 1 (interrupt enabled), the TRDIFi bit in the IF2H register is set to 1 (interrupt requested).
- If multiple bits in the TRDIERi register are set to 1, use the TRDSRi register to determine the source of the interrupt request.
- Since the bits in the TRDSRi register are not automatically cleared to 0 even if the interrupt is acknowledged, clear them to 0 in the interrupt routine.

- When status flags of interrupt sources (applicable status flags) of timer RD2 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.

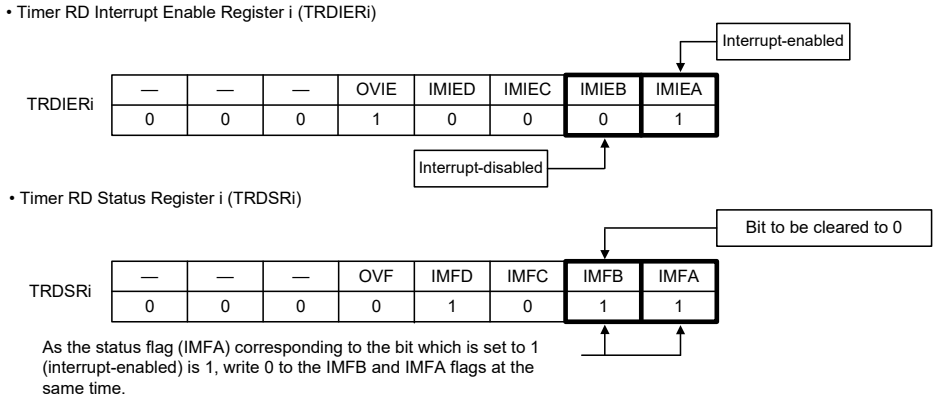
  b) When there are bits set to 1 (enabled) in timer RD interrupt enable register i (TRDIERi) and status flags of interrupt sources related to their bits are 0, write 0 to applicable status flags.

  Example: To clear the IMFB flag to 0 when bits IMIEA and OVIE are set to 1 (interrupt-enabled) and the IMIEB bit is set to 0 (interrupt-disabled).



  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 flag to 0 when the IMIEA is set to 1 (interrupt-enabled) and the IMIEB is set to 0 (interrupt-disabled).



When timer RD2 is set to extended complementary PWM mode or timer-KB PWM output gating mode, a timer RDi interrupt request is generated at the time an interrupt request of the operating mode is generated regardless of the settings of the timer RD interrupt enable registers i (TRDIERi).

In extended complementary PWM mode, control of skipping interrupt request signals can be applied. For details on skipping control, see **12.4.8 Skipping of interrupt requests and A/D conversion triggers**.

12.7 Cautions for Timer RD2

12.7.1 SFR read/write access

When setting timer RD2, set the TRD0EN bit in the PER2 register to 1 first. If the TRD0EN bit is 0, writes to the control registers of timer RD2 are ignored and 00H or 0000H is always read from the control registers (except for the port registers and the port mode registers).
The following registers must not be rewritten during count operation:
TRDELC, TRDBCR, TRDMR, TRDPMR, TRDFCR, TRDOER1, TRDPTO bit in TRDOER2, TRDOCR, TRDDFi, TRDCRi, TRDIORAi, TRDIORCi, TRDPOCRi, TRDADCR, TRDTCTL

Remark i = 0 or 1

1. TRDSTR register
   - The TRDSTR register can be set by an 8-bit memory manipulation instruction.
   - When the CSELi bit in the TRDSTR register is set to 0 (count stops at compare match between the TRDi counter and TRDGRAi register), 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, write 0 to 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 12 - 29** shows the TRDIOji pin output level when count stops while using the TRDIOji pin for timer RD2 output.

Table 12 - 29 TRDIOji 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 RD2 complementary and reset synchronous PWM modes or the initial output level selected by the TRDOCR register in extended complementary PWM mode.)
When the CSELi bit is set to 0, the count stops at compare match with the TRDi counter and TRDGRAi register.	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 RD2 complementary and reset synchronous PWM modes or the initial output level selected by the TRDOCR register in extended complementary PWM mode.)

2. TRDDFi register (i = 0 or 1)
   Set bits DFCK1 and DFCK0 in the TRDDFi register before starting count operation.

3. TRDi counter (i = 0 or 1)
   If the TRDi counter is set to 0000H and a value is written to the TRDi counter at the same time, the value written to the counter has priority.

Remark i = 0 or 1, j = A to D

12.7.2 Mode switching

- Stop the count operation (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 **Section 29 Interrupt Functions** for details.

12.7.3 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 TCK2 to TCK0 in the TRDCRi register.
     When selecting f_{HOCO} (64 MHz or 48 MHz) as the count source for timer RD2, set f_{CLK} to f_{IH} before setting bit 3 (TRD0EN) in peripheral enable register 2 (PER2). When changing f_{CLK} to a clock other than f_{IH}, clear bit 3 (TRD0EN) in peripheral enable register 2 (PER2) before changing.

12.7.4 Input capture function

- Set the pulse width of the input capture signal to three or more cycles of the timer RD2 operating clock.
- The value of the TRDi counter is transferred to the TRDGRji register two to three cycles of the timer RD2 operating clock (f_{CLK}) after the input capture signal is applied to the TRDIOji pin (i = 0 or 1, j = A to 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 if the edge selected by bits TRDIOj0 and TRDIOj1 in the TRDIORji register is input to the TRDIOji pin (i = 0 or 1, j = A to D) even while the TSTARTi bit in the TRDSTR register is 0 (count stops).

12.7.5 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 bits of port registers m that correspond to pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi to 0.
  4. Set the bits of port mode registers m that correspond 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 bit TSTART0 or TSTART1 to 1).

- To change the bits of port mode registers m that correspond to pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi from output mode to input mode, use the following setting procedure:
  [Changing procedure]
  1. Set the bits of port mode registers m that correspond 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 bit TSTART0 or 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.

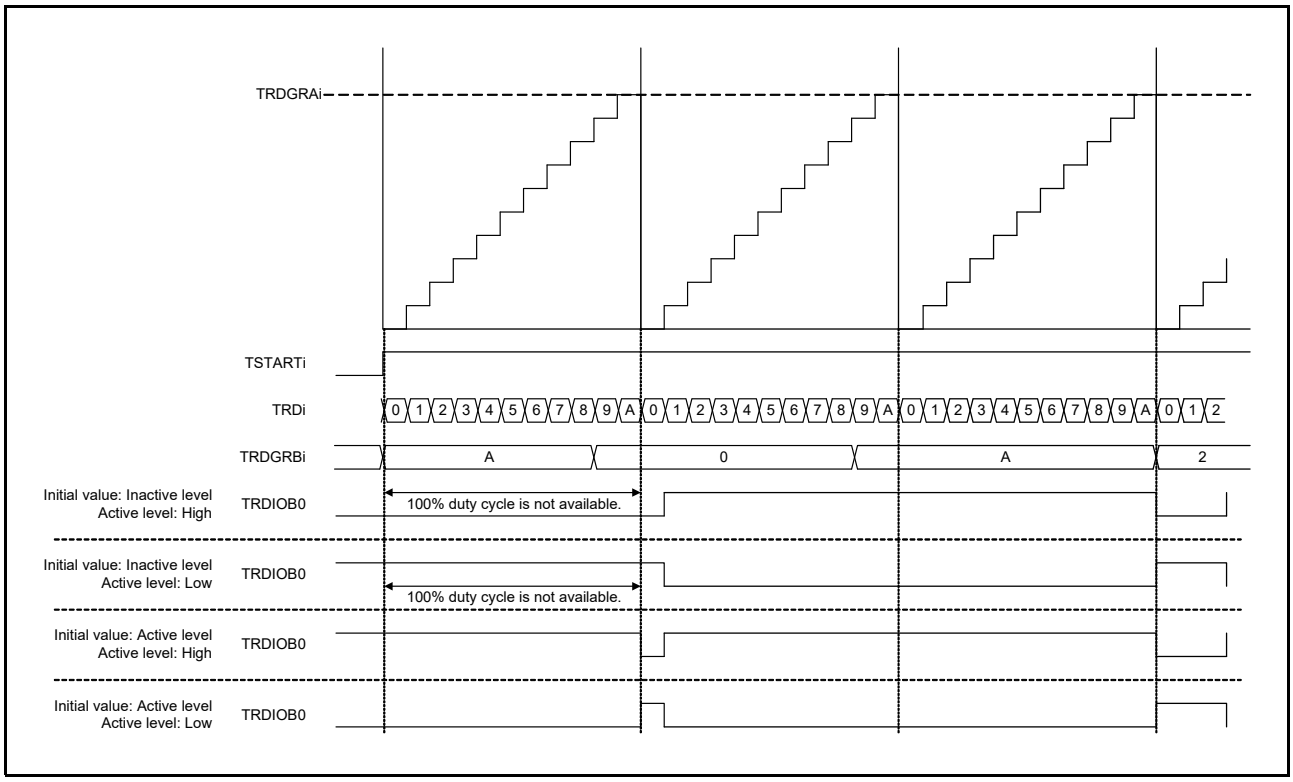
12.7.6 External clock TRDCLK

Set the pulse width of the external clock applied to the TRDCLK pin to three or more cycles of the timer RD2 operating clock.

12.7.7 PWM function in timer mode

The initial value specified in a general register used to set a PWM output waveform must not be the same value as the PWM period. When the initial output level of a pin is set to the inactive level, a waveform with 100% duty cycle cannot be output in the first cycle after counting begins.

Figure 12 - 110 PWM Function in Timer Mode (Restriction on Output with 100% Duty Cycle after Counting Begins)



12.7.8 Reset synchronous PWM mode

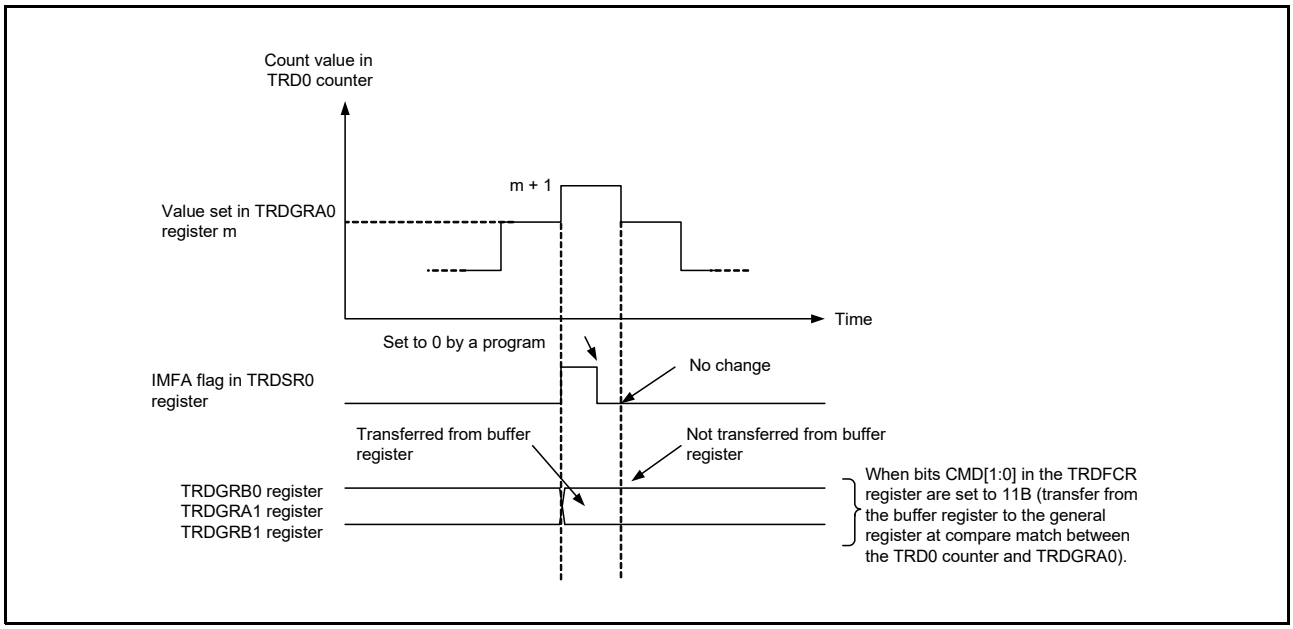
- When reset synchronous PWM mode is used for motor control, make sure OLS0 = OLS1.
- Set to reset synchronous PWM mode by the following procedure:
  [Changing procedure]
  1. Set the TSTART0 bit in the TRDSTR register to 0 (count stops).
  2. Set bits CMD1 and CMD0 in the TRDFCR register to 00B (timer 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 RD2 again.

12.7.9 Complementary PWM mode

- When complementary PWM mode is used for motor control, make sure OLS0 = OLS1.
- Change bits CMD1 and CMD0 in the TRDFCR register in the following procedure.
  [Changing procedure: When setting to complementary PWM mode (including re-set), or changing the timing of transfer 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 RD2 again.

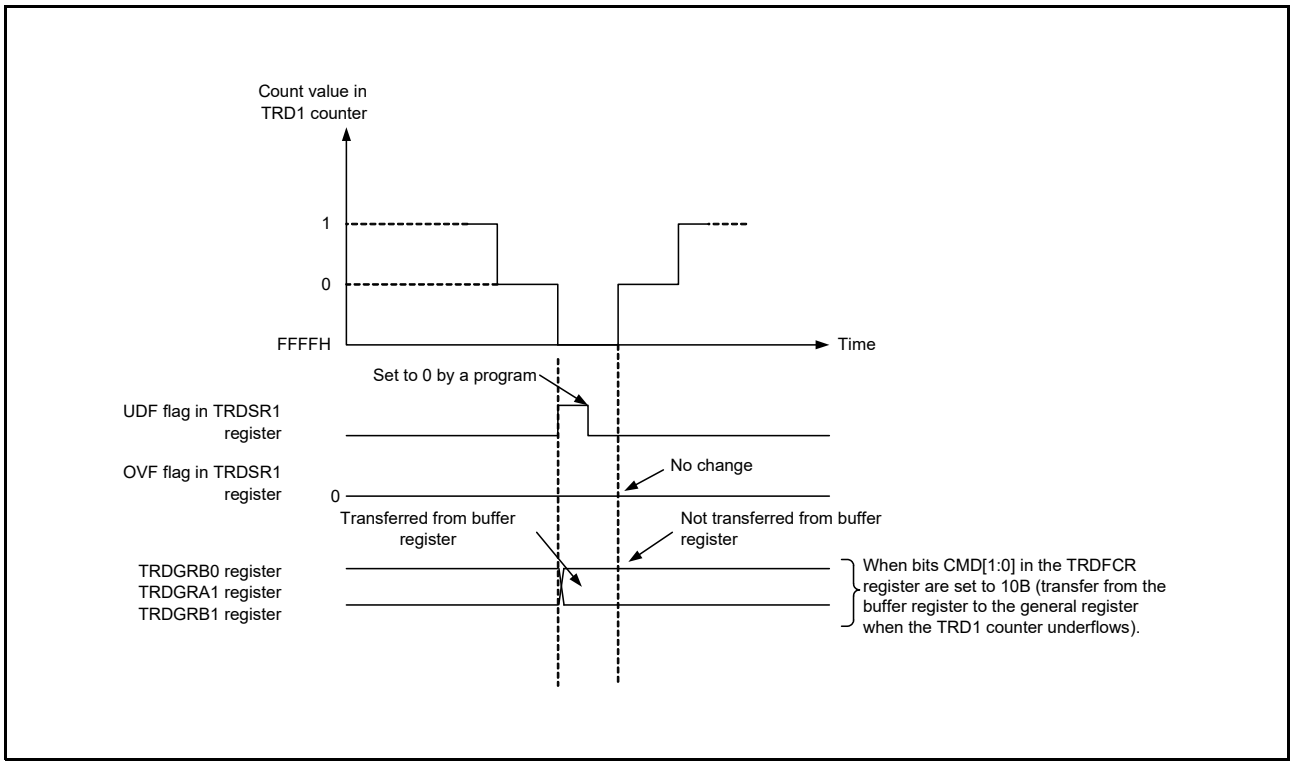
  [Changing procedure: When stopping complementary PWM mode]
  1. Set both the TSTART0 and TSTART1 bits in the TRDSTR register to 0 (count stops).
  2. Set bits CMD1 and CMD0 to 00B (timer mode, PWM mode, and PWM3 mode).
     - Do not write to the TRDGRA0, TRDGRB0, TRDGRA1, or TRDGRB1 register during operation.
       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 counter counts m – 1, m, m + 1, m, m – 1, in that order, when changing from increment to decrement operation.
       Counting from m to m + 1 causes the IMFA flag in the TRDSRi register to be set to 1. Also, when bits CMD1 and CMD0 in the TRDFCR register are set to 11B (complementary PWM mode, buffer data transferred at compare match between the TRD0 counter and TRDGRA0 register), the content of the buffer registers (TRDGRD0, TRDGRC1, and TRDGRD1) is transferred to the general registers (TRDGRB0, TRDGRA1, and TRDGRB1).
       During counting from m + 1 to m and then to m – 1, the IMFA flag remains unchanged and data is not transferred to registers such as the TRDGRA0 register.

Figure 12 - 111 Operation at Compare Match between TRD0 Counter and TRDGRA0 Register in Complementary PWM Mode



- The TRD1 counter 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 flag 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 counter), the content of the buffer registers (TRDGRD0, TRDGRC1, and TRDGRD1) is transferred to the general registers (TRDGRB0, TRDGRA1, and TRDGRB1).
  During counting from FFFFH to 0 and then to 1, data is not transferred to registers such as the TRDGRB0 register. Also, at this time, the OVF flag in the TRDSRi register remains unchanged.

Figure 12 - 112 Operation When TRD1 Counter Underflows in Complementary PWM Mode

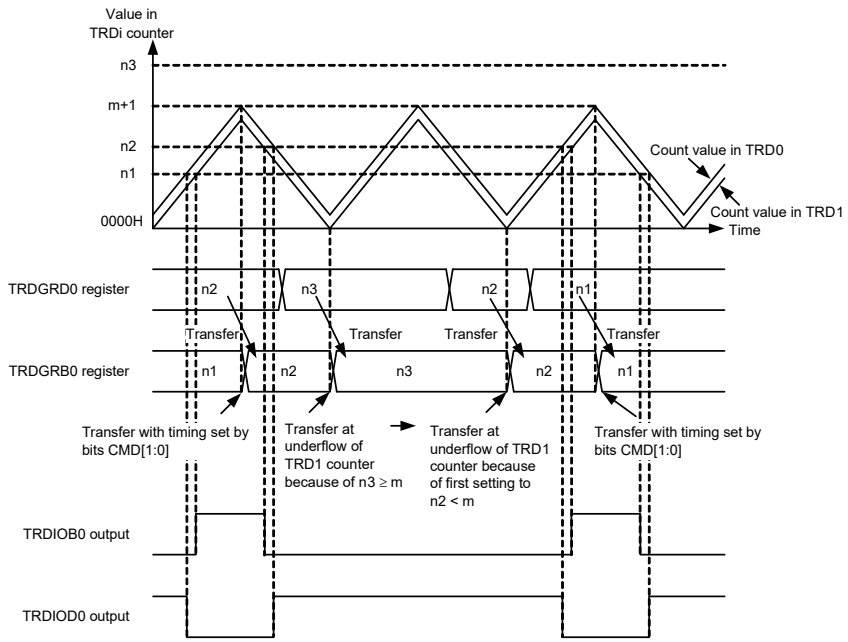


- The timing of data transfer from the buffer register to the general register should be selected using bits CMD1 and CMD0 in the TRDFCR register. However, regardless of the values of bits CMD1 and CMD0, transfer takes place with the following timing when duty cycle is 0% and duty cycle is 100%.

  Value in buffer register ≥ value in TRDGRA0 register (duty cycle is 0%): Transfer takes place at underflow of the TRD1 counter.
  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 counter 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 12 - 113 Operation When Value in Buffer Register ≥ Value in TRDGRA0 Register in Complementary PWM Mode



Remark
m: Value set in TRDGRA0 register

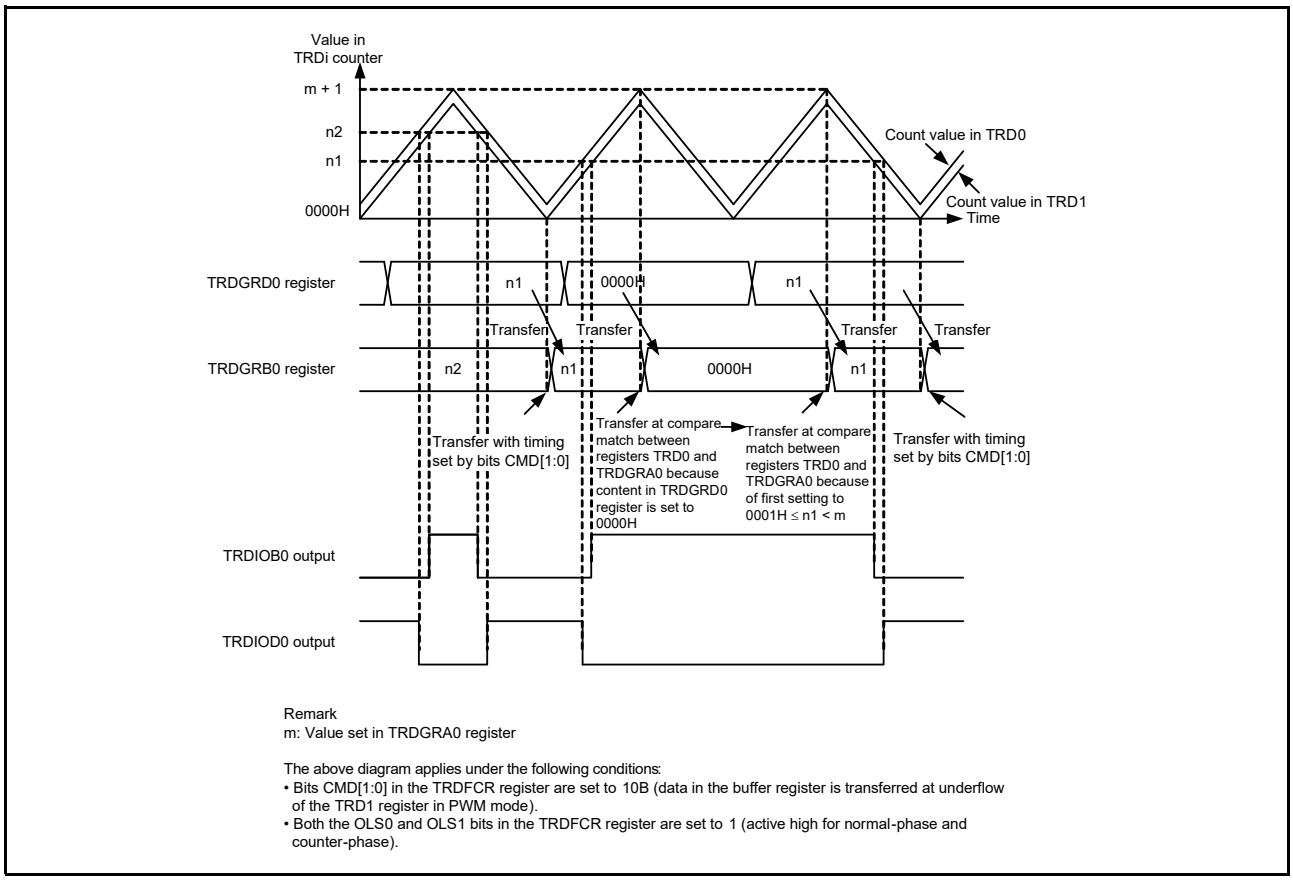
The above diagram applies under the following conditions:
- Bits CMD[1:0] in the TRDFCR register are set to 11B
  (data in the buffer register is transferred at compare match between the TRD0 counter and TRDGRA0 register in complementary PWM mode).
- Both the OSL0 and OLS1 bits in the TRDFCR register 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 counter and smaller than or equal to (TRDGRA0 register setting value minus TRD0 counter 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 the TRD0 counter and TRDGRA0 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 a compare match occurs between the TRD0 counter and TRDGRA0 register 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.

Figure 12 - 114 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 CMD[1:0] 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 the TRD0 counter and TRDGRA0 register, 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 counter and smaller than or equal to (TRDGRA0 register setting value minus TRD0 counter 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.

12.7.10 Extended PWM mode

- Use the following procedure to stop and restart counting.
  1. Set the TSTARTi bit in the TRDSTR register to 0 (count stops).
  2. Select the initial output level of the pins again by the TRDOCR register.
  3. Set the TRDi counter to 0000H.
  4. Set the TSTARTi bit in the TRDSTR register to 1 (count starts).

- Make the following settings to synchronize the counters.
  1. Set the SYNC bit in the TRDMR register to 1.
  2. Set the TRDCR0 and TRDCR1 registers to the same value.
     Set the TRDi counter to 0000H.
     Set the TRDGRA0 and TRDGRA1 registers to the same value.
  3. Set the TSTARTi bit in the TRDSTR register to 1 (count starts).

- To update the compare registers simultaneously, execute 16-bit access to the TRDRDT0 and TRDRDT1 registers to set the RDT bit in both registers at the same time.
- The initial value specified in a general register used to set a PWM output waveform must not be the same value as the PWM period. When the initial output level of a pin is set to the inactive level, a waveform with 100% duty cycle cannot be output in the first cycle after counting begins. To output a waveform with 100% duty cycle in the first cycle, select the active level as the initial output level. For details, see **12.7.7 PWM function in timer mode**.

12.7.11 Extended complementary PWM mode

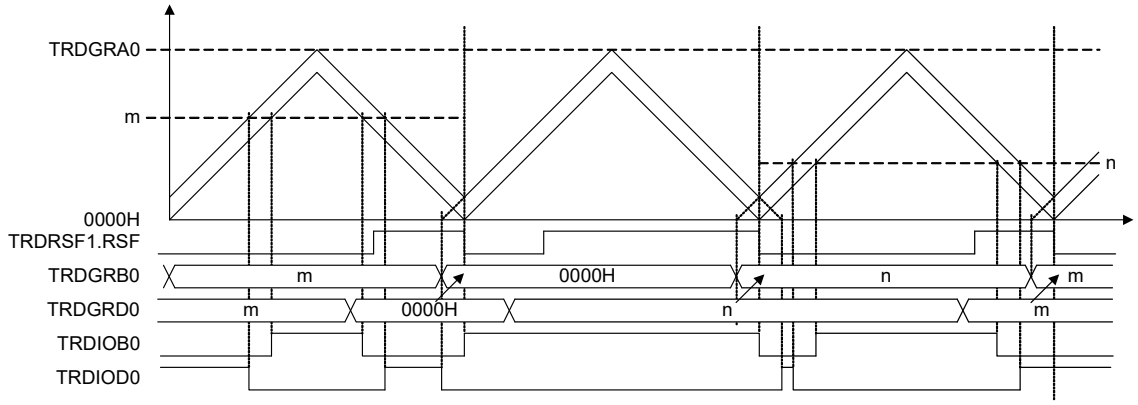
- When extended complementary PWM mode is used for motor control, make sure OLS0 = OLS1.
- Use the following procedure to stop and restart counting in extended complementary PWM mode.
  [Changing procedure]
  1. Set both the TSTART0 and TSTART1 bits in the TRDSTR register to 0 (count stops).
  2. Set the CMD1 and CMD0 bits in the TRDFCR register to 00B (timer mode and PWM3 mode).
  3. Set the CMD1 and CMD0 bits to 10B and EPWM bit to 1B (extended complementary PWM mode).
  4. Set the dead time and 0000H in the TRD0 and TRD1 counters, respectively.
  5. Set the TRDGRC0, TRDGRD0, TRDGRC1, and TRDGRD1 registers. Also set the TRDGRm and TRDCMPm (m = B0, D0, A1, B1, C1, D1) registers when the setting for the output of asymmetric waveforms has been made.
     - Set the same value in each of the following pairs of registers: TRDGRB0 and TRDGRD0, TRDGRA1 and TRDGRC1, TRDGRB1 and TRDGRD1, TRDCMPB0 and TRDCMPD0, TRDCMPA1 and TRDCMPC1, and TRDCMPB1and TRDCMPD1.
  6. To use the A/D conversion trigger output function, set the TRDADTB0 and TRDADTB1 registers. To change the initial levels on the pins, set the TRDOCR register.
     - Do not write to the TRDGRA0, TRDGRB0, TRDGRA1, TRDGRB1, TRDCMPB0, TRDCMPA1, and TRDCMPB1 registers during operation.
     - When changing the PWM waveform during operation, write the desired values to the TRDGRD0, TRDGRC1, TRDGRD1, TRDCMPD0, TRDCMPC1, and TRDCMPD1 registers and then set the TRDRDT1.RDT bit to 1. After that, use the buffer operation of the simultaneous update function to transfer the respective values written to these registers to the TRDGRB0, TRDGRA1, TRDGRB1, TRDCMPB0, TRDCMPA1, and TRDCMPB1 registers.
       The PWM period value cannot be changed.

[Settings of output with 100% duty cycle]

To output a waveform with 100% duty cycle, set the compare register to 0000H. When CPSS = 0, only the TRDGRji register needs to be set up. When CPSS = 1, the TRDGRji and TRDCMPm registers need to be set up. A waveform with 100% duty cycle can be output in the next and subsequent PWM cycles by setting the buffer registers and setting the TRDRDT1.RDT bit to 1 before the timing of the simultaneous update.

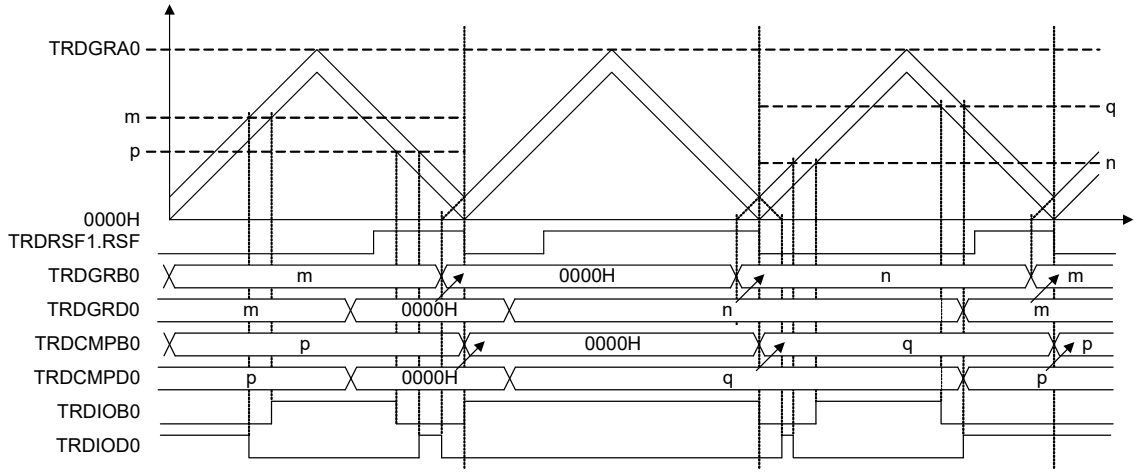
Figure 12 - 115 Examples of Setting the Output with 100% Duty Cycle

When TRDFCR.CPSS = 0 (symmetric waveform)



The above diagram applies under the following condition.
TRDFCR.CPSS = 0

When TRDFCR.CPSS = 1 (asymmetric waveform)

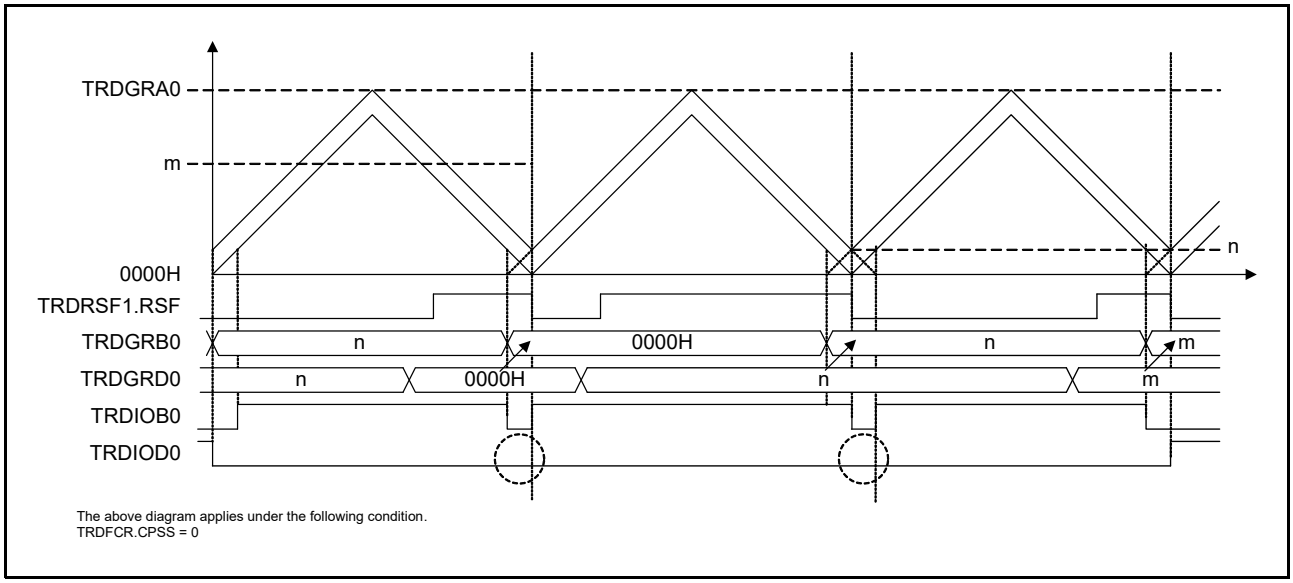


The above diagram applies under the following condition.
TRDFCR.CPSS = 1

[Restriction on output with 100% duty cycle]

The counter-phase output may be at the same inactive level as that when PWM output with 100% duty cycle is selected due to the settings of the PWM duty factor and dead time.

Figure 12 - 116 Restriction on Output with 100% Duty Cycle

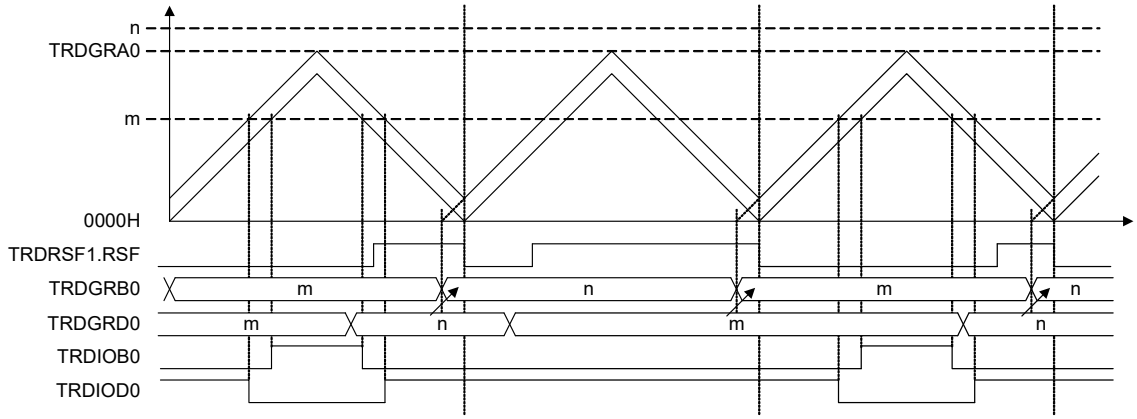


[Settings of output with 0% duty cycle]

To output a waveform with 0% duty cycle, set the compare register to a value equal to or greater than the value specified in the TRDGRA0 register. When CPSS = 0, only the TRDGRji register needs to be set up. When CPSS = 1, the TRDGRji and TRDCMPm registers need to be set up. A waveform with 0% duty cycle can be output in the next and subsequent PWM cycles by setting the buffer registers and setting the TRDRDT1.RDT bit to 1 before the timing of the simultaneous update.

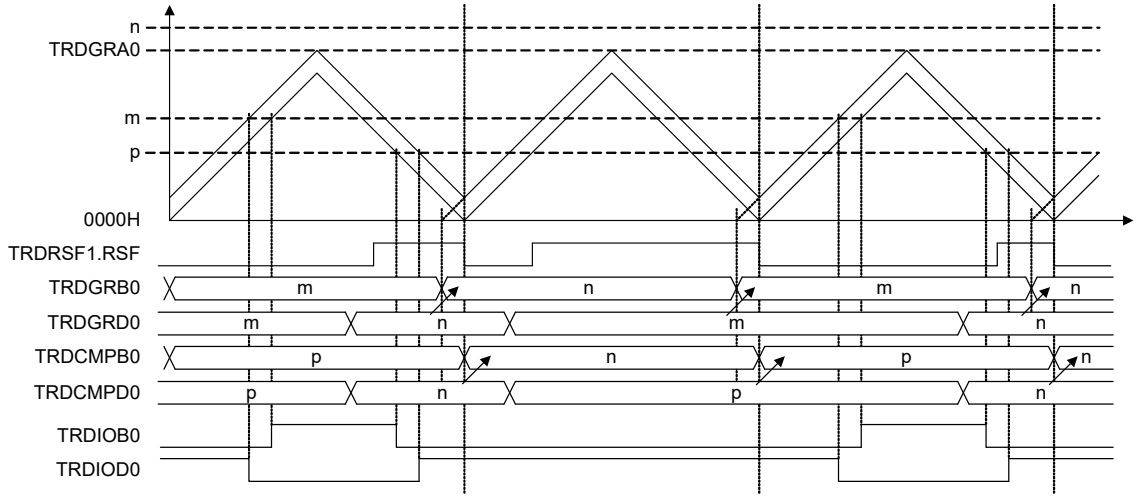
Figure 12 - 117 Examples of Setting the Output with 0% Duty Cycle

When TRDFCR.CPSS = 0 (symmetric waveform)



The above diagram applies under the following condition.
TRDFCR.CPSS = 0

When TRDFCR.CPSS = 1 (asymmetric waveform)

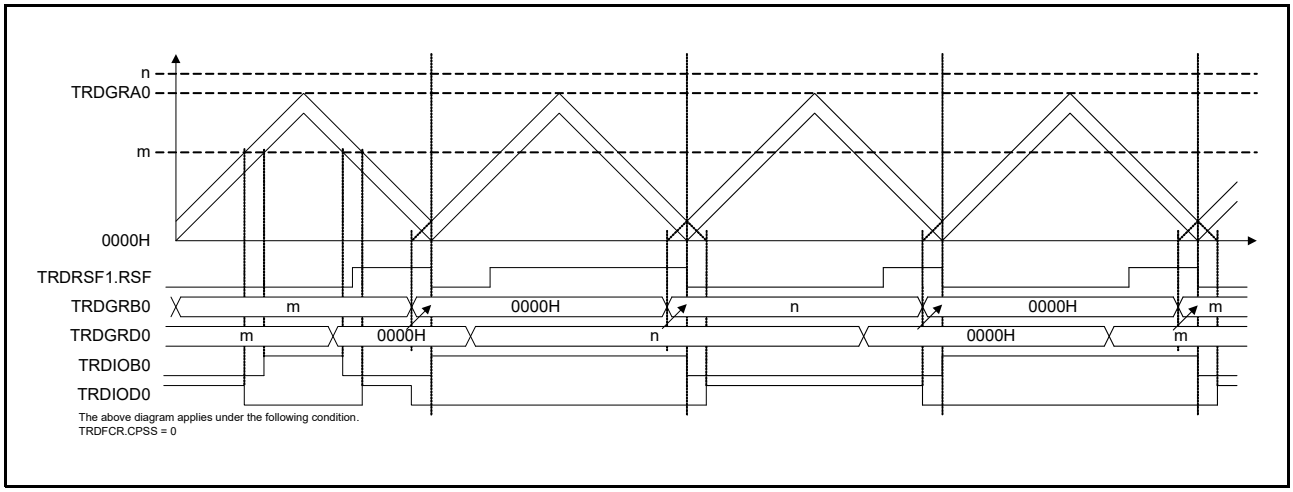


The above diagram applies under the following condition.
TRDFCR.CPSS = 1

[Operation example of waveform output with 100% duty cycle and 0% duty cycle]

In extended complementary PWM mode, output can be made to alternate between waveforms with 100% duty cycle and 0% duty cycle.

Figure 12 - 118 Operation Example of Waveform Output with 100% Duty Cycle and 0% Duty Cycle



12.8 PWM Option Unit A (PWMOPA)

The PWM option unit is used to cut off the signals output from timer RD2 and ports or release the signals from the cutoff state by using the comparator 3 output, external interrupt 0 (INTP0), or event link controller (ELC) as a trigger signal. The PWM option unit is a different function from the pulse forced cutoff incorporated in timer RD2.

Table 12 - 30 Functional Difference between Pulse Forced Cutoff and Output Forced Cutoff

	Pulse Forced Cutoff of Timer RD2	Output Forced Cutoff of PWM Option Unit
Mode supporting forced cutoff	• PWM function • Reset synchronous PWM mode • Complementary PWM mode • PWM3 mode • Extended PWM mode • Extended complementary PWM mode	• Supports all output modes for timer RD2 • Port output can also be cut off.
Source of cutoff	• ELC input • Low-level input of INTP0	• ELC input • INTP0 • Comparator 3 output
Release from cutoff state	Counting of timer RD2 is stopped and the output signals are released from the cutoff state via software	• Released via hardware • Released via software (stopping counting is unnecessary)
Pins that can be cut off	Pins set for timer RD2 output selected from among P17/TRDIOA0, P15/TRDIOB0, P16/TRDIOC0, P14/TRDIOD0, P13/TRDIOA1, P12/TRDIOB1, P11/TRDIOC1, and P10/TRDIOD1.	Selected from among P17/TRDIOA0, P15/TRDIOB0, P16/TRDIOC0, P14/TRDIOD0, P13/TRDIOA1, P12/TRDIOB1, P11/TRDIOC1, and P10/TRDIOD1. Port output can also be cut off.
Port state selection at cutoff	• High-impedance output • Low-level output • High-level output	• High-impedance output • Low-level output • High-level output However, when port output is cut off, only high-impedance output can be selected.

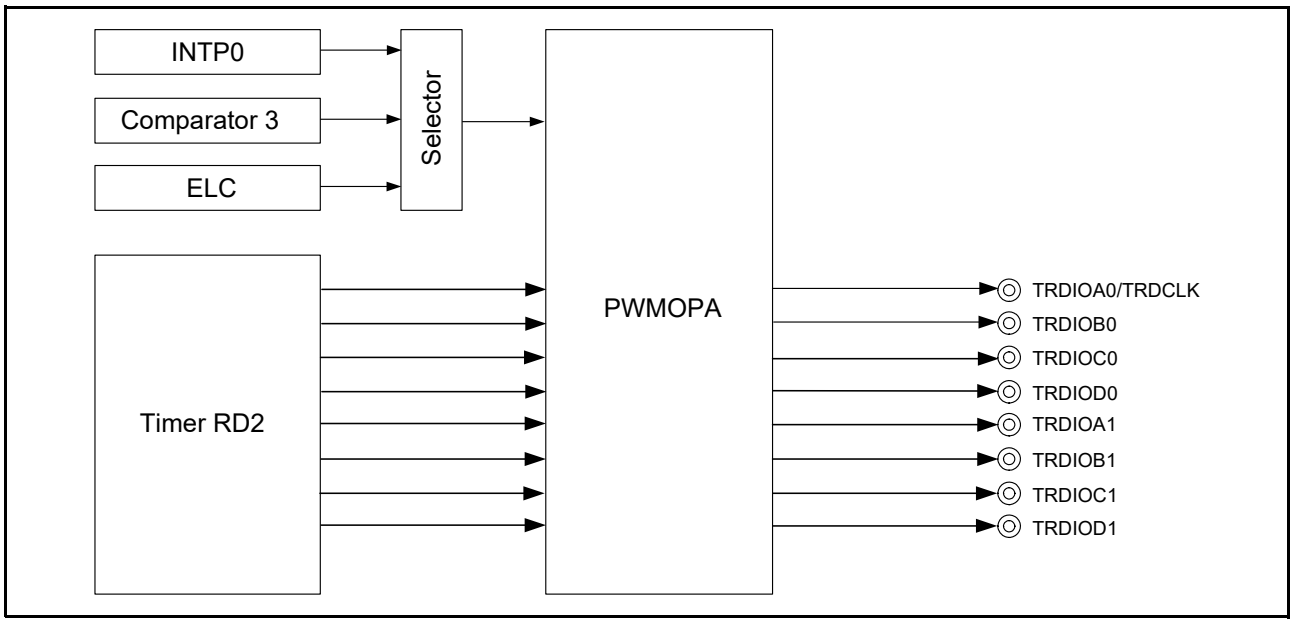
Caution When pulse forced cutoff and output forced cutoff are used simultaneously, the same cutoff source should not be selected.

12.8.1 Overview of PWM option unit

The PWM option unit has the following functions.

- Comparator 3, external interrupt 0, or the event link controller can be selected as an output cutoff source.
- When comparator 3 or external interrupt 0 is an output forced cutoff source, the edge to become a cutoff source can be selected.
- Software or hardware can be selected as the condition to release the output pins from the output forced cutoff state.
- High-level, low-level, or high-impedance can be selected for the output level at cutoff.

Figure 12 - 119 PWMOPA Control Logic



12.8.2 Registers controlling PWM option unit

The following registers are used to control the PWM option unit.

- **PWMOPA control register 0 (OPCTL0)**
- **PWMOPA cutoff control register 0 (OPDF0)**
- **PWMOPA cutoff control register 1 (OPDF1)**
- **PWMOPA edge selection register (OPEDGE)**)
- **PWMOPA status register (OPSR)**

12.8.2.1 PWMOPA control register 0 (OPCTL0)

The OPCTL0 register is used to control PWMOPA. This register can be set by an 8-bit memory manipulation instruction.
The value of this register following a reset is 00H.

Figure 12 - 120 Format of PWMOPA Control Register 0 (OPCTL0) (1/2)

Address: F0358H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
OPCTL0	0	HAZAD_SET	IN_EG	IN_SEL1	IN_SEL0	ACT	HZ_REL	HS_SEL

HAZAD_SET	Output cutoff hazard control selection[Note 1]
0	Hazard measure disabled
1	Hazard measure enabled

IN_EG	Output forced cutoff source edge/output forced cutoff release edge selection[Notes 2, 3]
0	Rising edge: Output forced cutoff Falling edge: Release from output forced cutoff
1	Rising edge: Release from output forced cutoff Falling edge: Output forced cutoff

IN_SEL1	IN_SEL0	Cutoff source selection[Notes 2, 4, 5]
0	0	No output cutoff source selection
0	1	Comparator 3 output
1	0	INTP0 pin input
1	1	Event input from ELC

Figure 12 - 120 Format of PWMOPA Control Register 0 (OPCTL0) (2/2)

ACT	When release via software is selected: Software release timing selection
0	When HZ_REL is set to 1 via software, pins are released from the output forced cutoff state and pulse output is resumed.
1	When HZ_REL is set to 1, pins are released from the forced cutoff state and pulse output is resumed at the following timing depending on the operating mode of timer RD2. • Timer RD2 complementary PWM mode and extended complementary PWM mode: Pins are released from the output forced cutoff state at the TRDIOC0 edge timing selected in the OPEDGE register and pulse output is resumed. • Timer RD2 reset synchronous PWM mode: Pins are released from the output forced cutoff state when the TRD0 count becomes 0000H. • Other than the above modes: TRDIOj0 (j = A to D) pins are released from the output forced cutoff state when the TRD0 count becomes 0000H. TRDIOj1 (j = A to D) pins are released from the output forced cutoff state when the TRD1 count becomes 0000H.**Note 6**

HZ_REL	When release via software is selected: Output cutoff release control
0	Output forced cutoff continues (after release from the forced cutoff state, the HZ_REL bit becomes 0).**Note 7**
1	Pins are released from the output forced cutoff state and pulse output is resumed.**Note 8**
The value that can be read from or written to the HZ_REL bit differs depending on the state. • Normal state: 0 or 1 can be written and only 0 can be read. • Output forced cutoff state: Only 1 can be written and only 1 can be read.	

HS_SEL	Output forced cutoff release mode selection
0	Released via hardware When releasing pins from the output forced cutoff state via hardware, the release timing varies depending on the operating mode of timer RD2. • Timer RD2 complementary PWM mode and extended complementary PWM mode: After a cutoff release source is detected, pins are released from the output forced cutoff state at the TRDIOC0 edge timing selected in OPEDGE. • Timer RD2 reset synchronous PWM mode: After a cutoff release source is detected, pins are released from the output forced cutoff state when the TRD0 count becomes 0000H.**Note 6** • Other than the above modes: After a cutoff release source is detected, TRDIOj0 (j = A to D) pins are released from the output forced cutoff state when the TRD0 count becomes 0000H. TRDIOj1 (j = A to D) pins are released from the output forced cutoff state when the TRD1 count becomes 0000H.**Note 9**
1	Released via software

Note 1. Do not change it while timer RD2 is operating.

Note 2. Set the IN_SEL1 and IN_SEL0 bits at least three clock cycles after the IN_EG bit has been set.

Note 3. It is enabled when comparator 3 output or INTP0 pin input is selected as an output cutoff source.

Note 4. To release pins from the output forced cutoff state with an ELC source, make sure to select release via software (set the HS_SEL bit to 1). There is no restriction on output cutoff release with external interrupt 0 (INTP0) and comparator 3.

Note 5. Set the input enabled level period of the comparator 3 output and INTP0 to one clock cycle or longer.

Note 6. When all bits 15 to 0 of the counter become 0 while timer RD20 and timer RD21 are operating, the timer RD20 and timer RD21 count values become 0000H.

Note 7. If timer RD2 operates in the output compare function, PWM function, PWM3 mode, or extended PWM mode, the operation at output cutoff release differs between cases where 2 channels are used and 1 channel is used.

- If timer RD2 is used with 2 channels
  If the HZ_REL bit is set to 1 via software, all cutoff state bits (HZOF0 and HZOF1) become 0 (cutoff release) and the HZ_REL bit becomes 0.
- If timer RD2 is used with 1 channel
  If the HZ_REL bit is set to 1 via software, the cutoff state bit (HZOF0 or HZOF1) corresponding to the used timer RD2 channel becomes 0 and the HZ_REL bit becomes 0.

Note 8. It cannot be set to 1 if forced cutoff has not occurred.

Note 9. When timer RD2 operates in the output compare function, PWM function, PWM3 mode, or extended PWM mode, the pins of the channel that is not operating cannot be released from the cutoff state (the cutoff state bits (HZOF0 and HZOF1) do not become 0).

12.8.2.2 PWMOPA cutoff control register 0 (OPDF0)

The OPDF0 register is used to control the pulse cutoff function for the PWM output TRDIOj0 (j = A to D) in PWMOPA. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 12 - 121 Format of PWMOPA Cutoff Control Register 0 (OPDF0)

Address: F0359H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
OPDF0	DFD01	DFD00	DFC01	DFC00	DFB01	DFB00	DFA01	DFA00

DFD01	DFD00	TRDIOD0 pin output forced cutoff control
0	0	Forced cutoff prohibited
0	1	High-impedance output
1	0	Low-level output
1	1	High-level output

DFC01	DFC00	TRDIOC0 pin output forced cutoff control
0	0	Forced cutoff prohibited
0	1	High-impedance output
1	0	Low-level output
1	1	High-level output

DFB01	DFB00	TRDIOB0 pin output forced cutoff control
0	0	Forced cutoff prohibited
0	1	High-impedance output
1	0	Low-level output
1	1	High-level output

DFA01	DFA00	TRDIOA0 pin output forced cutoff control
0	0	Forced cutoff prohibited
0	1	High-impedance output
1	0	Low-level output
1	1	High-level output

Caution 1. When the TRDIOj0 (j = A to D) pin is used for port output with forced cutoff enabled, select high-impedance output.

Caution 2. Do not change the register value in the forced cutoff state.

Caution 3. When one of the TRDIOji (j = A to D, i = 0 or 1) functions multiplexed in a pin is selected by the settings of the PIOR registers, only set the selected TRDIOji pin function for output.
Example: When TRDIOD0 is selected for P17 in PIOR2 and output from TRDIOD0 is enabled by the setting of the TRDOER1 register, only set the DFD0n (n = 0 or 1) bits for TRDIOD0 and set the DFA0n (n = 0 or 1) bits to prohibit forced cutoff for TRDIOA0.

12.8.2.3 PWMOPA cutoff control register 1 (OPDF1)

The OPDF1 register is used to control the pulse cutoff function for the PWM output TRDIOj1 (j = A to D) in PWMOPA. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 12 - 122 Format of PWMOPA Cutoff Control Register 1 (OPDF1)

Address: F035AH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
OPDF1	DFD11	DFD10	DFC11	DFC10	DFB11	DFB10	DFA11	DFA10

DFD11	DFD10	TRDIOD1 pin output forced cutoff control
0	0	Forced cutoff prohibited
0	1	High-impedance output
1	0	Low-level output
1	1	High-level output

DFC11	DFC10	TRDIOC1 pin output forced cutoff control
0	0	Forced cutoff prohibited
0	1	High-impedance output
1	0	Low-level output
1	1	High-level output

DFB11	DFB10	TRDIOB1 pin output forced cutoff control
0	0	Forced cutoff prohibited
0	1	High-impedance output
1	0	Low-level output
1	1	High-level output

DFA11	DFA10	TRDIOA1 pin output forced cutoff control
0	0	Forced cutoff prohibited
0	1	High-impedance output
1	0	Low-level output
1	1	High-level output

Caution 1. When the TRDIOj1 (j = A to D) pin is used for port output with forced cutoff enabled, select high-impedance output.

Caution 2. Do not change the register value in the forced cutoff state.

Caution 3. When one of the TRDIOji (j = A to D, i = 0 or 1) functions multiplexed in a pin is selected by the settings of the PIOR registers, only set the selected TRDIOji pin function for output.
Example: When TRDIOA1 is selected for P16 in PIOR2, and output from TRDIOA1 is enabled by the setting of the TRDOER1 register, only set the DFA1n (n = 0 or 1) bits for TRDIOA1 and set the DFC1n (n = 0 or 1) bits to prohibit forced cutoff for TRDIOC1.

12.8.2.4 PWMOPA edge selection register (OPEDGE)

The OPEDGE register selects the timing of release from the output forced cutoff state when timer RD2 is set to the complementary PWM mode or extended complementary PWM mode and the release is done by hardware. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 12 - 123 Format of PWMOPA Edge Selection Register (OPEDGE)

Address: F035BH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
OPEDGE	—	—	—	—	—	—	EG1	EG0

EG1	EG0	Output forced cutoff release edge selection
0	0	Released from the cutoff state at the rising edge of TRDIOC0
0	1	Released from the cutoff state at the falling edge of TRDIOC0
1	0	Released from the cutoff state at both edges of TRDIOC0
1	1	Input edge of TRDIOC0 disabled, cutoff retained

12.8.2.5 PWMOPA status register (OPSR)

The OPSR register indicates the output forced cutoff state and cutoff source state. This register can be read by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 12 - 124 Format of PWMOPA Status Register (OPSR)

Address: F035CH
After reset: 00H
R/W: R

Symbol	7	6	5	4	3	2	1	0
OPSR	0	0	0	0	0	HZOF1	HZOF0	HZIF0

HZOF1	Cutoff state[Note 1]
0	Normal timer output (TRDIOA1, TRDIOB1, TRDIOC1, TRDIOD1)
1	Cutoff state (TRDIOA1, TRDIOB1, TRDIOC1, TRDIOD1)

HZOF0	Cutoff state[Note 1]
0	Normal timer output (TRDIOA0, TRDIOB0, TRDIOC0, TRDIOD0)
1	Cutoff state (TRDIOA0, TRDIOB0, TRDIOC0, TRDIOD0)

HZIF0	Output cutoff source state[Notes 1, 2]
0	State in which output cutoff source has not exceeded threshold
1	State in which output cutoff source has exceeded threshold

Note 1. If the output cutoff source has exceeded the threshold before selecting the INTP0 or comparator 3 cutoff source with the IN_SEL1 and IN_SEL0 bits in the OPCTL0 register, the HZIF0 bit is set to 1 after the IN_SEL1 and IN_SEL0 bits are set, but the HZOF0 or HZOF1 bit is not set.

Note 2. Effective when the INTP0 or comparator 3 cutoff source is selected.

12.8.3 Operation

Output forced cutoff and release from the output forced cutoff state of the timer RD2 output pin TRDIOji (j = A to D, i = 0 or 1) can be controlled by using the INTP0 input, event input from ELC, or comparator 3 output as a trigger. When the INTP0 input or comparator 3 output is used as a cutoff source, the edge to trigger output forced cutoff or release from the output forced cutoff state can be selected.

12.8.3.1 Forced cutoff

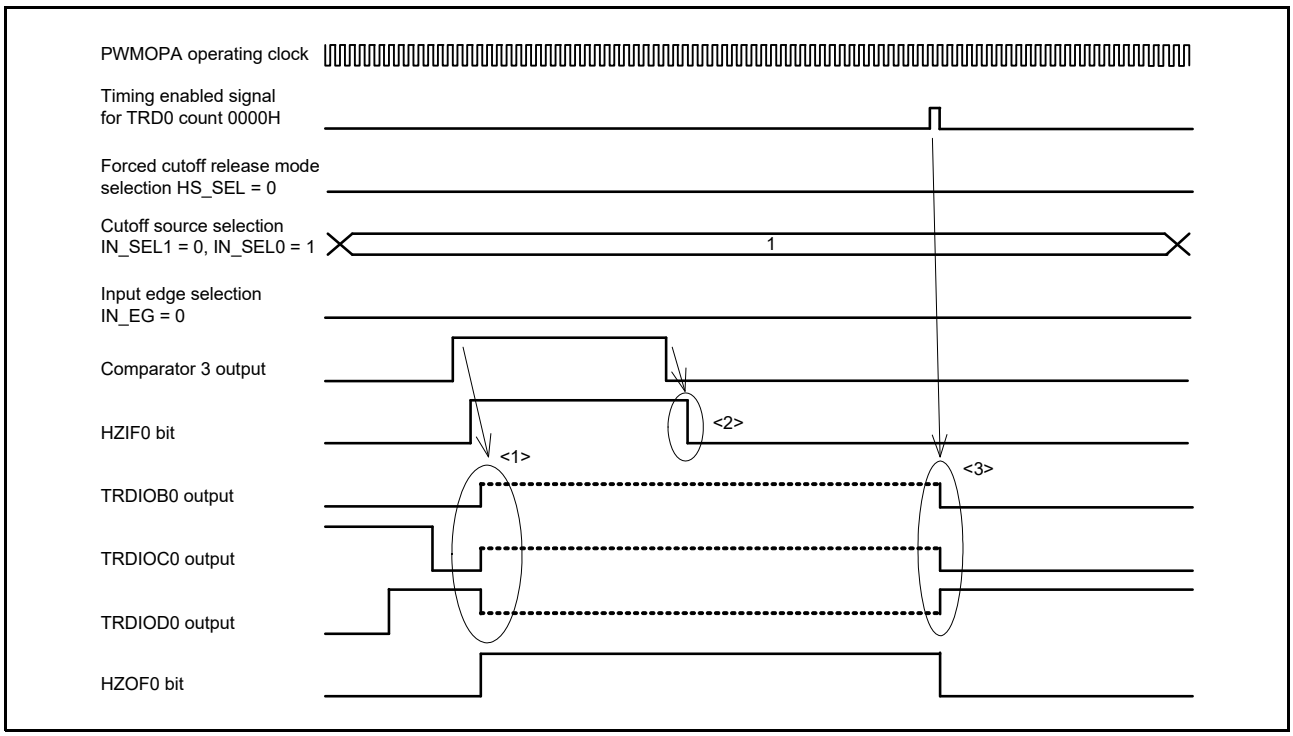
Pulses output from the timer RD2 output pin TRDIOji (j = A to D, i = 0 or 1) can be cut off by using the INTP0 input, event input from ELC, or comparator 3 output as a trigger.
When an output forced cutoff source is detected, the output of timer RD2 is forcibly cut off, and the output value specified in the OPDF0/OPDF1 register is output. For detailed operation, see **Figure 12 - 126 Detailed Timing Diagram of Forced Cutoff**.
Release from the forced cutoff state via hardware or software can be selected by using the setting value of the HS_SEL bit of the OPCTL0 register.

12.8.3.2 Release via hardware (HS_SEL = 0)

The timing of release from the output forced cutoff state varies depending on the function of timer RD2.

1. Output in modes other than timer RD2 complementary PWM and extended complementary PWM
   - Timer RD2 is in the output compare function, PWM function, PWM3 mode, or extended PWM mode:
     TRDIOA0, TRDIOB0, TRDIOC0, and TRDIOD0 are released from the output forced cutoff state when the TRD0 count value becomes 0000H after an output forced cutoff release source is detected. TRDIOA1, TRDIOB1, TRDIOC1, and TRDIOD1 are released from the output forced cutoff state when the TRD1 count value becomes 0000H.
   - Timer RD2 is in the reset synchronous PWM mode:
     All TRDIO pins are released from the output forced cutoff state when the TRD0 count value becomes 0000H after an output forced cutoff release source is detected.

Figure 12 - 125 Operation Example of Output Forced Cutoff and Release from Output Forced Cutoff State via Hardware (Example of Cutoff of TRDIOB0, TRDIOC0, and TRDIOD0 Pins)

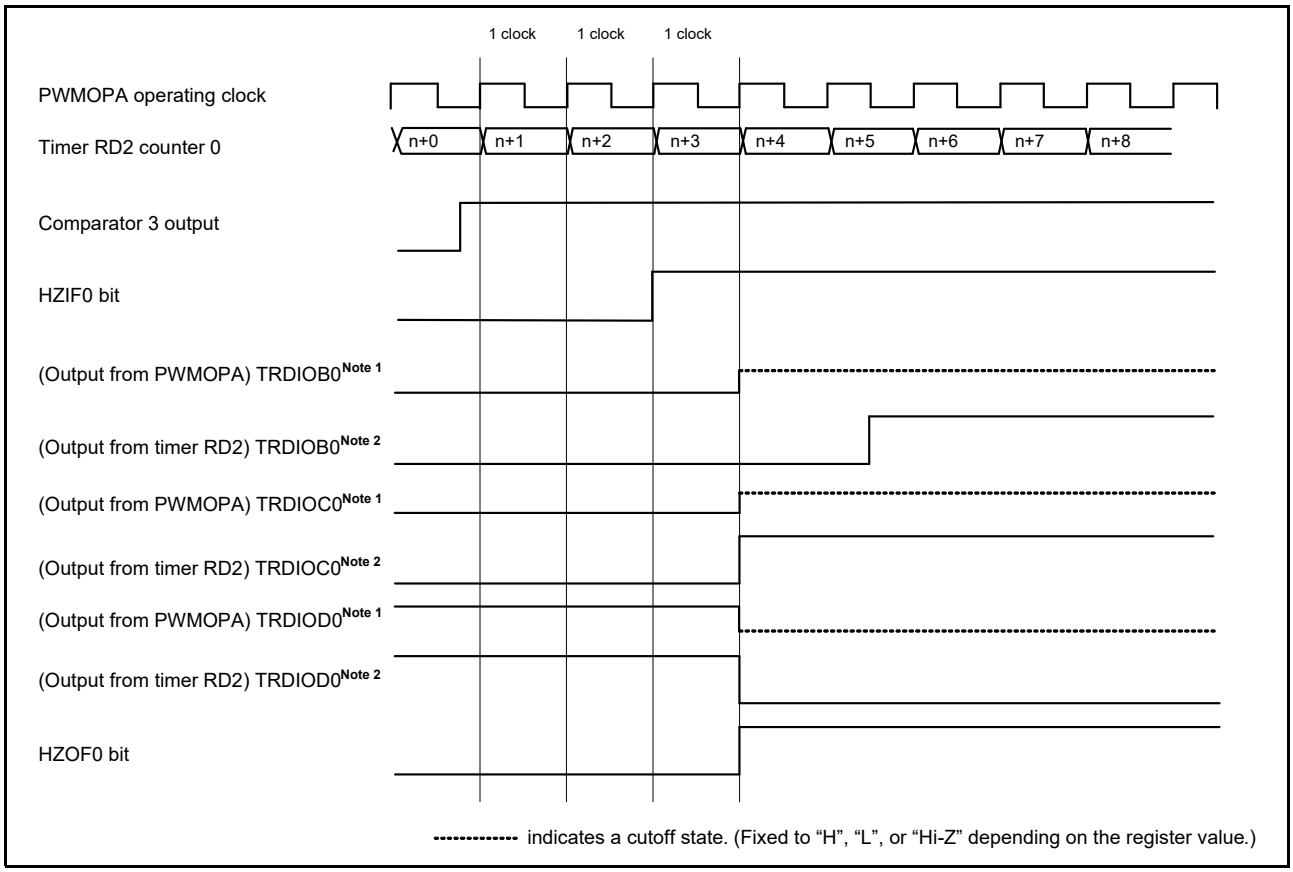


<1> The signals output from the TRDIOB0, TRDIOC0, and TRDIOD0 pins enter the output forced cutoff state when the rising edge of the comparator 3 output signal is detected.

<2> The HZIF0 bit is cleared after the falling edge of the comparator 3 output signal is detected.

<3> The pins are released from the forced cutoff state when the TRDi count value becomes 0000H.

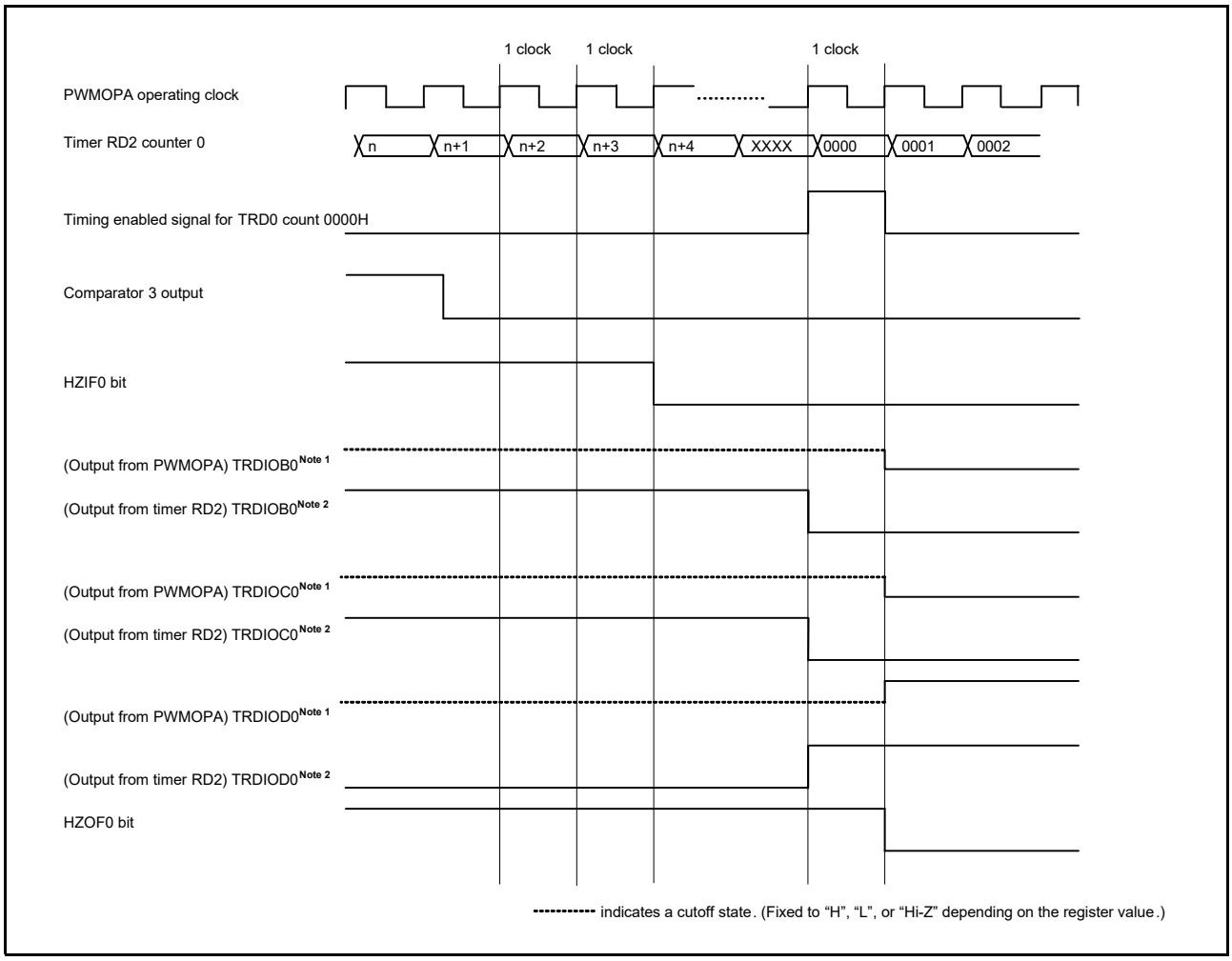
Figure 12 - 126 Detailed Timing Diagram of Forced Cutoff



Note 1. (Output from PWMOPA) TRDIO* (* = B to D) indicates the state of the multiplexed timer RD2 function pin.

Note 2. (Output from timer RD2) TRDIO* (* = B to D) indicates input to PWMOPA from timer RD2.

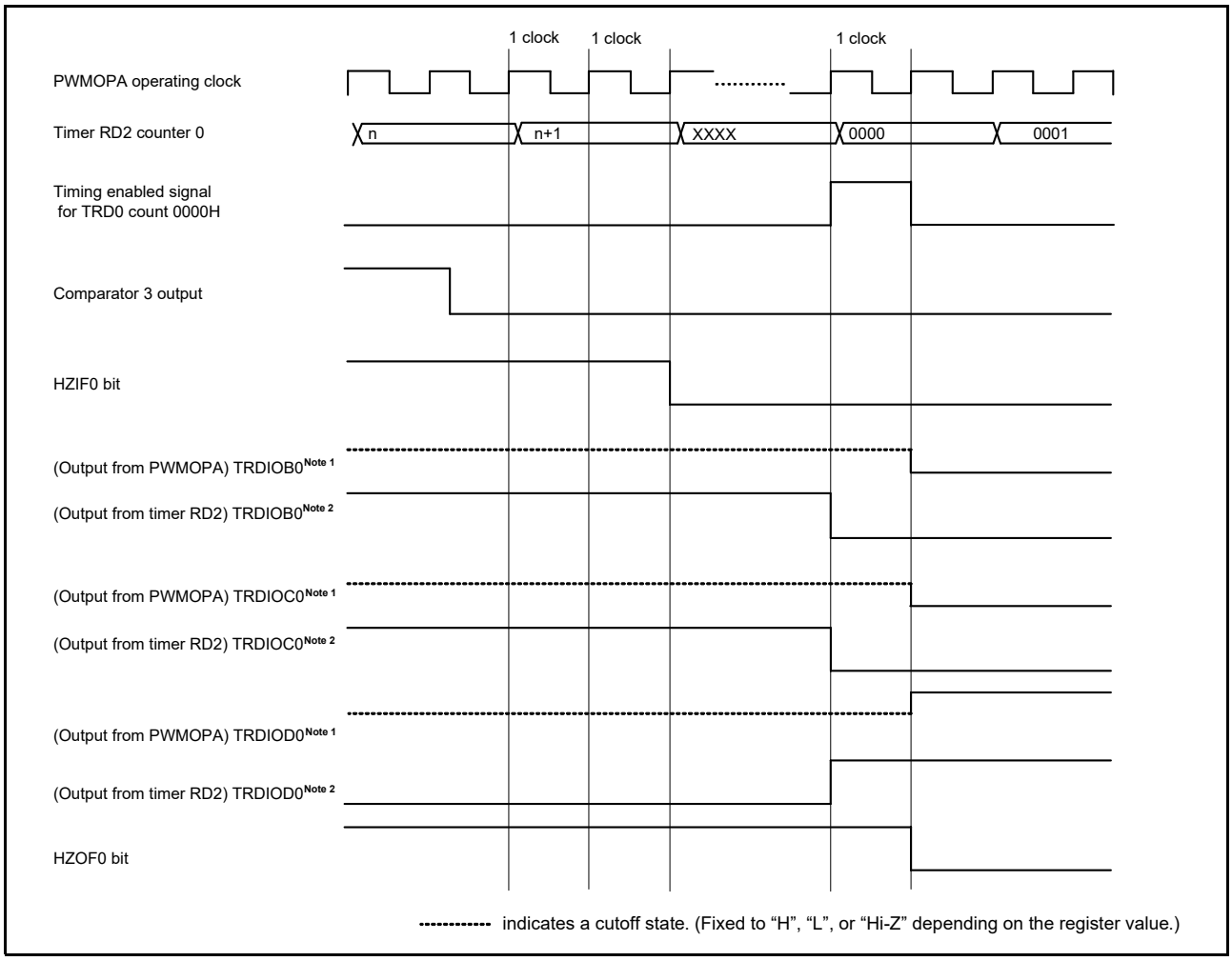
Figure 12 - 127 Detailed Timing Diagram of Release from Cutoff State (Timer RD2 Count Source = fCLK)



Note 1. (Output from PWMOPA) TRDIO* (* = B to D) indicates the state of the multiplexed timer RD2 function pin.

Note 2. (Output from timer RD2) TRDIO* (* = B to D) indicates input to PWMOPA from timer RD2.

Figure 12 - 128 Detailed Timing Diagram of Release from Cutoff State (Timer RD2 Count Source = $f_{CLK}/2$)

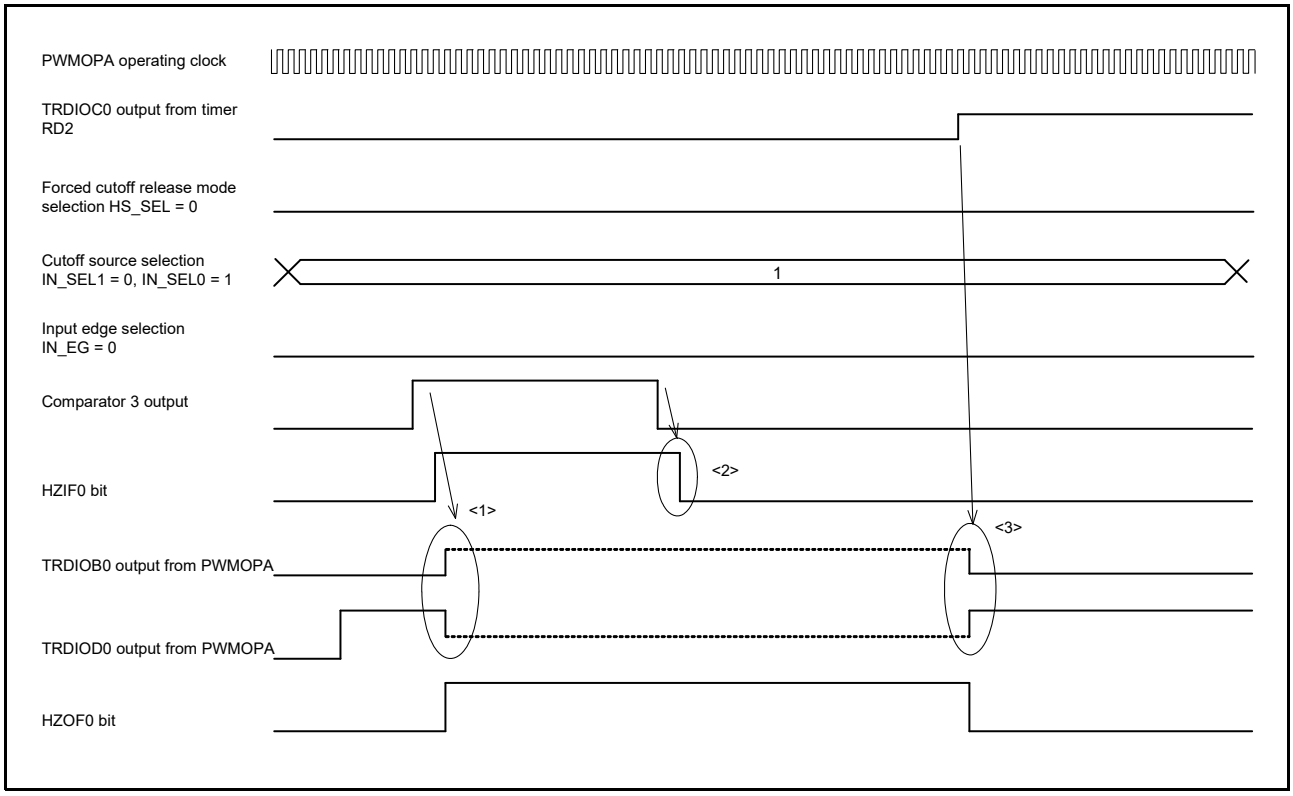


Note 1. (Output from PWMOPA) TRDIO* (* = B to D) indicates the state of the multiplexed timer RD2 function pin.

Note 2. (Output from timer RD2) TRDIO* (* = B to D) indicates input to PWMOPA from timer RD2.

2. Output in the timer RD2 complementary PWM mode and extended complementary PWM mode
   If the OPEDGE register is set after an output forced cutoff release source is detected, timer RD2 is released from the output forced cutoff state at the rising, falling, and both edges of the selected TRDIOC0.

Figure 12 - 129 Operation Example of Release from Cutoff State via Hardware (Example of TRDIOB0 and TRDIOD0)

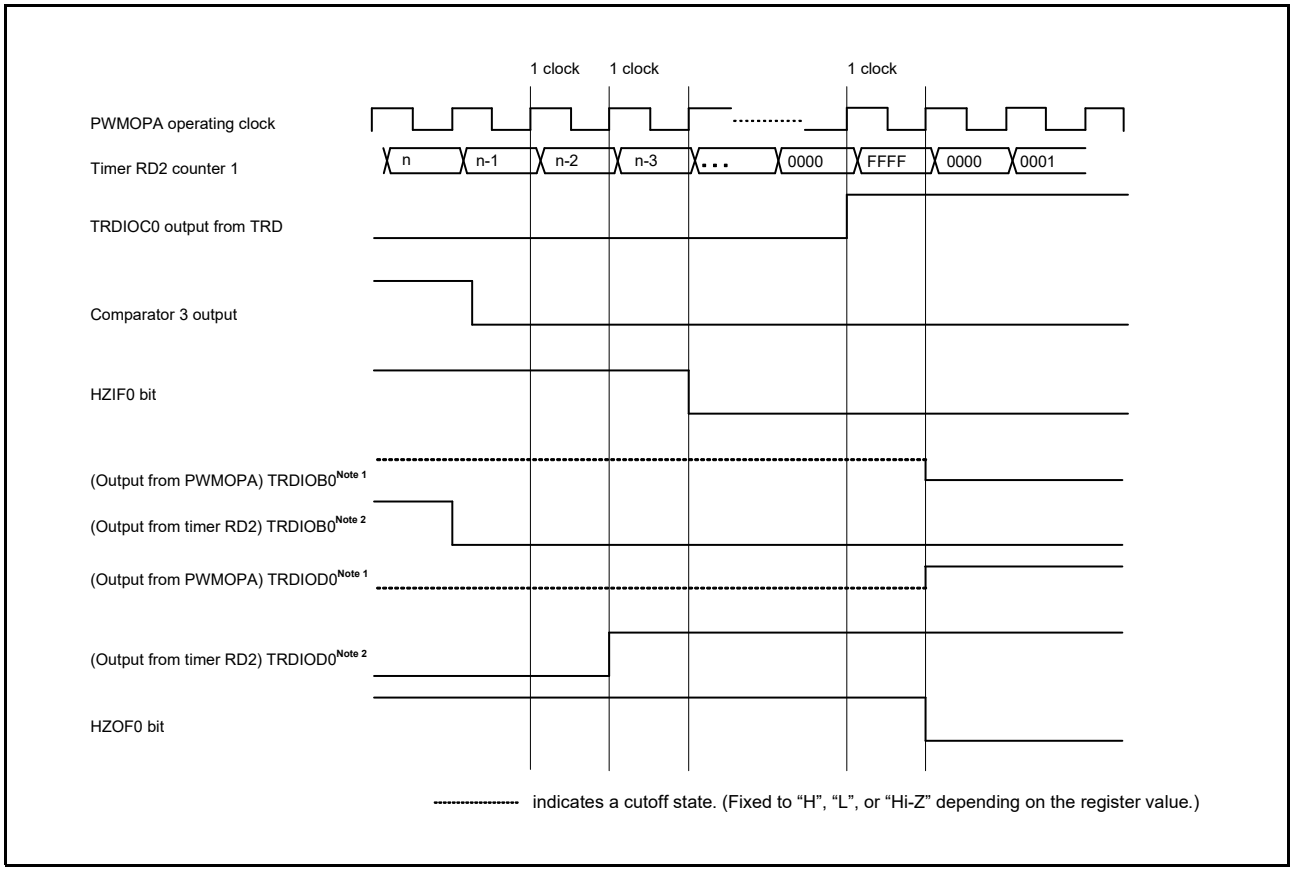


<1> The signals output from the TRDIOB0 and TRDIOD0 pins enter the forced cutoff state when the rising edge of the comparator 3 output signal is detected.

<2> The HZIF0 bit is cleared after the falling edge of the comparator 3 output signal is detected.

<3> The pins are released from the forced cutoff state at the rising edge of TRDIOC0.
For the detailed cutoff timing, see **Figure 12 - 126 Detailed Timing Diagram of Forced Cutoff**.

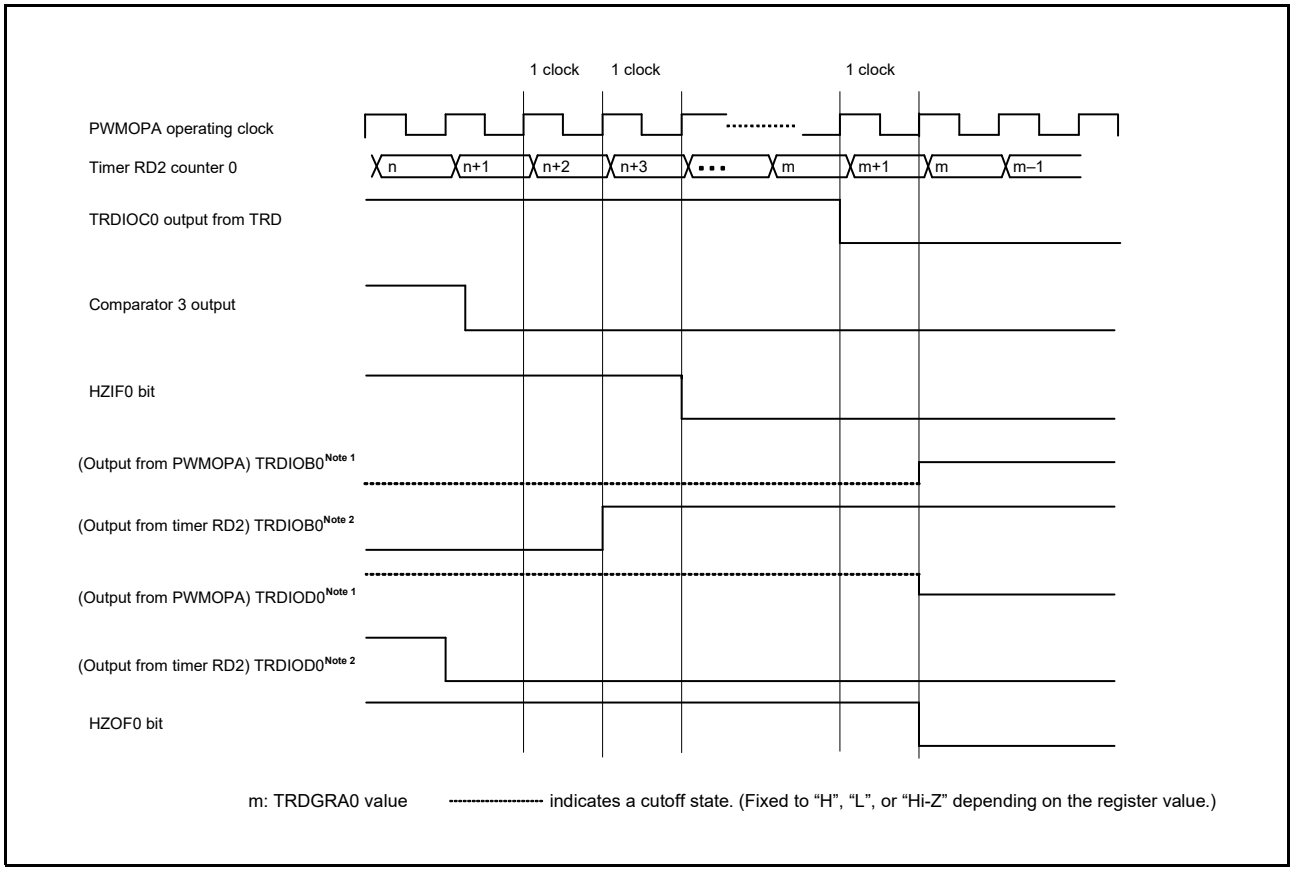
Figure 12 - 130 Detailed Timing Diagram of Release from Cutoff State (Timer RD2 Count Source = fCLK, Timer RD2 Decremented)



Note 1. (Output from PWMOPA) TRDIO* (* = B to D) indicates the state of the multiplexed timer RD2 function pin.

Note 2. (Output from timer RD2) TRDIO* (* = B to D) indicates input to PWMOPA from timer RD2.

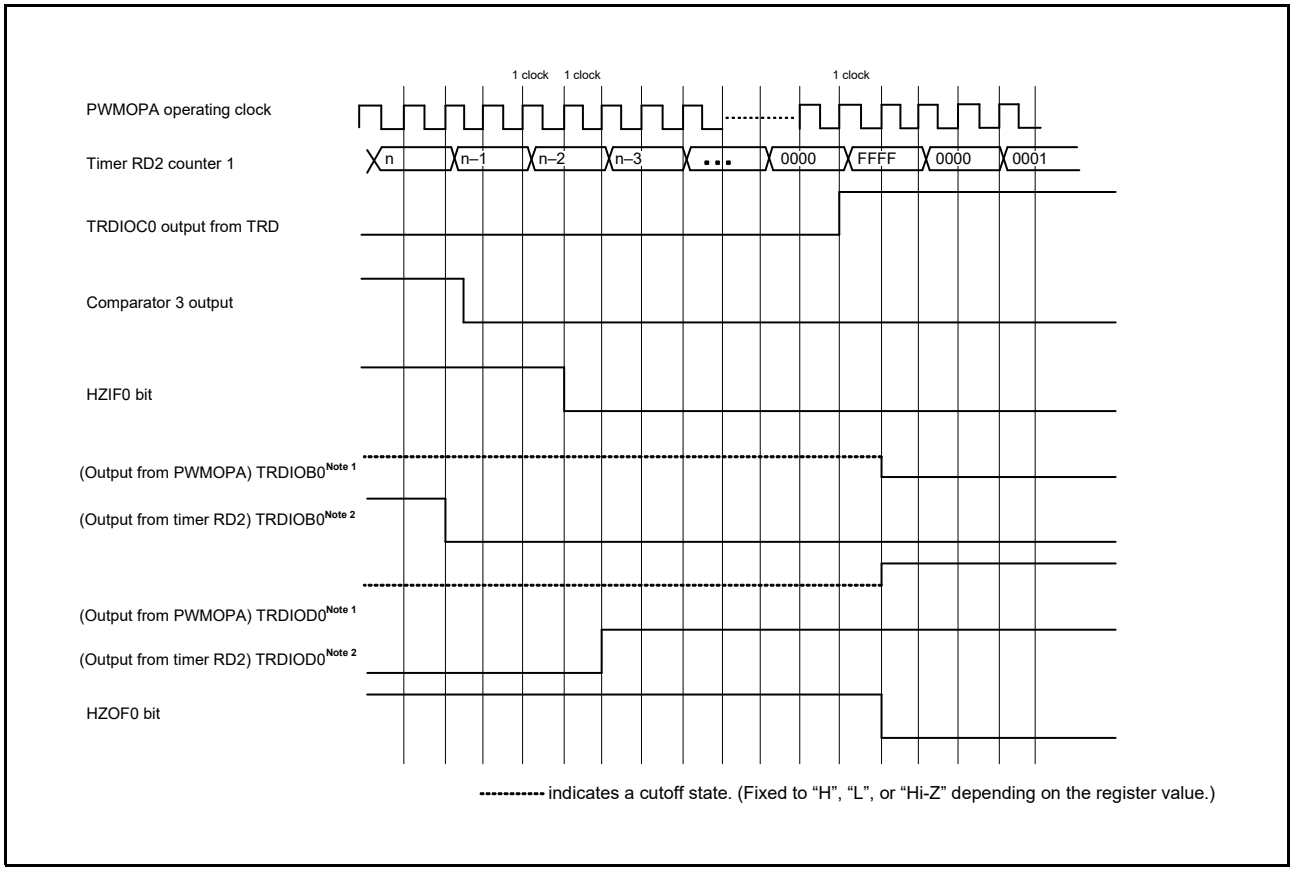
Figure 12 - 131 Detailed Timing Diagram of Release from Cutoff State (Timer RD2 Count Source = f_{CLK}, Timer RD2 Count = TRDGRA0)



Note 1. (Output from PWMOPA) TRDIO* (* = B to D) indicates the state of the multiplexed timer RD2 function pin.

Note 2. (Output from timer RD2) TRDIO* (* = B to D) indicates input to PWMOPA from timer RD2.

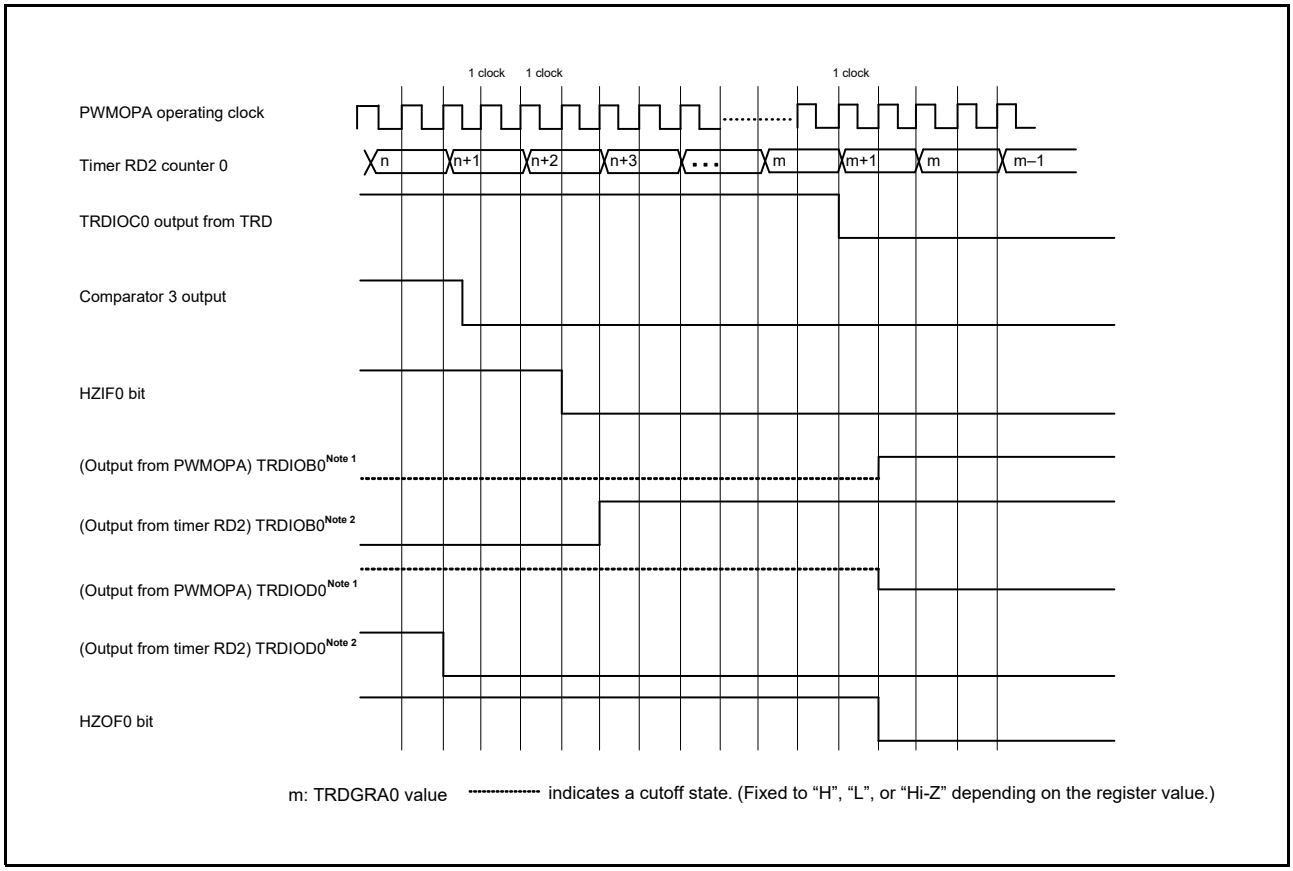
Figure 12 - 132 Detailed Timing Diagram of Release from Cutoff State (Timer RD2 Count Source = $f_{CLK}/2$, Timer RD2 Decremented)



Note 1. (Output from PWMOPA) TRDIO* (* = B to D) indicates the state of the multiplexed timer RD2 function pin.

Note 2. (Output from timer RD2) TRDIO* (* = B to D) indicates input to PWMOPA from timer RD2.

Figure 12 - 133 Detailed Timing Diagram of Release from Cutoff State (Timer RD2 Count Source = f_{CLK}/2, Timer RD2 Count = TRDGRA0)



Note 1. (Output from PWMOPA) TRDIO* (* = B to D) indicates the state of the multiplexed timer RD2 function pin.

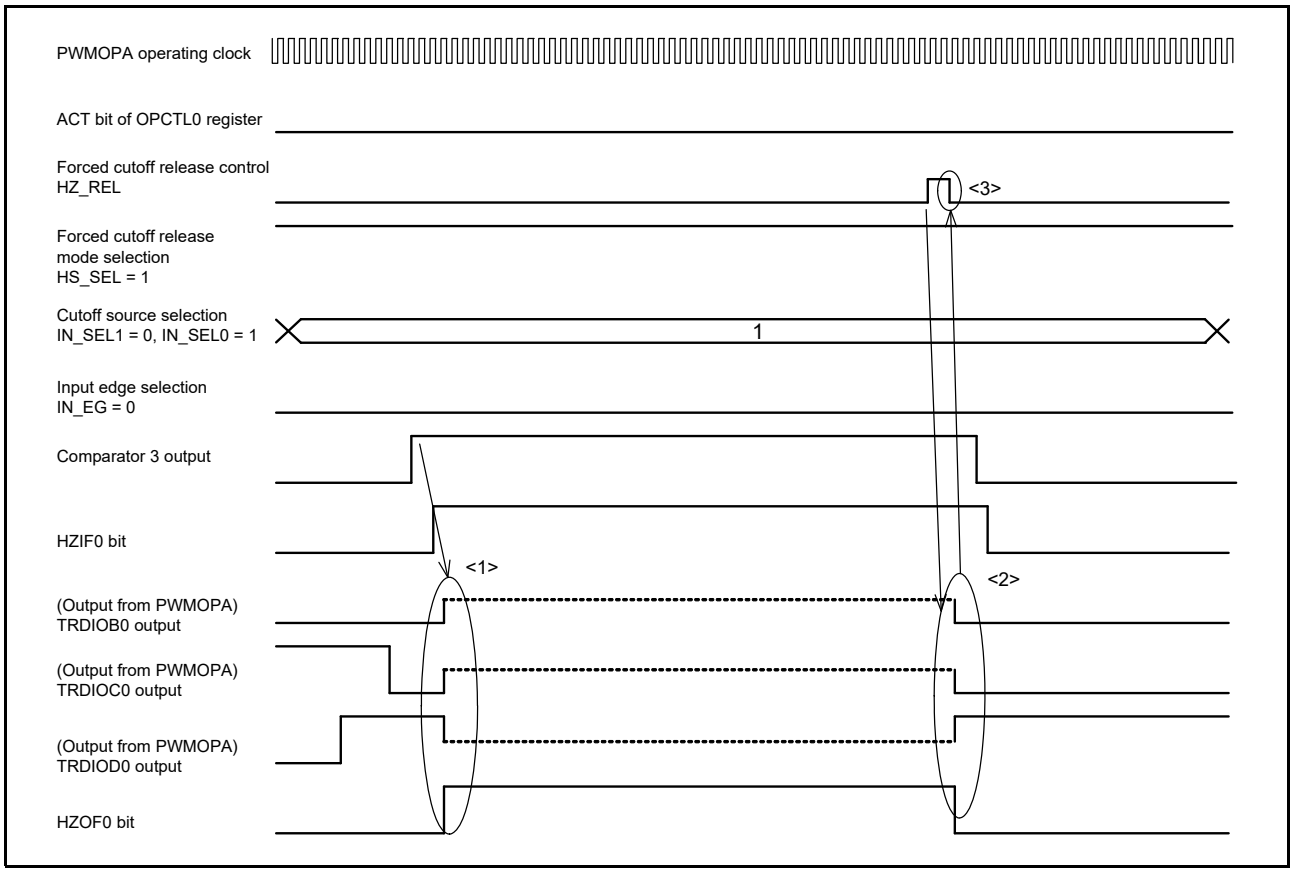
Note 2. (Output from timer RD2) TRDIO* (* = B to D) indicates input to PWMOPA from timer RD2.

12.8.3.3 Release via software (HS_SEL = 1)

The timing of release from the output forced cutoff state varies depending on the setting of the ACT bit of the OPCTL0 register.

1. Immediate release via software (ACT = 0)
   If ACT is set to 0, the pins are released from the forced cutoff state immediately when the HZ_REL bit of the OPCTL0 register is set to 1.
   After the release from the forced cutoff state, the HZ_REL bit automatically becomes 0.

Figure 12 - 134 Operation Example of Release from Cutoff State via Software (Example of TRDIOB0, TRDIOC0, and TRDIOD0)

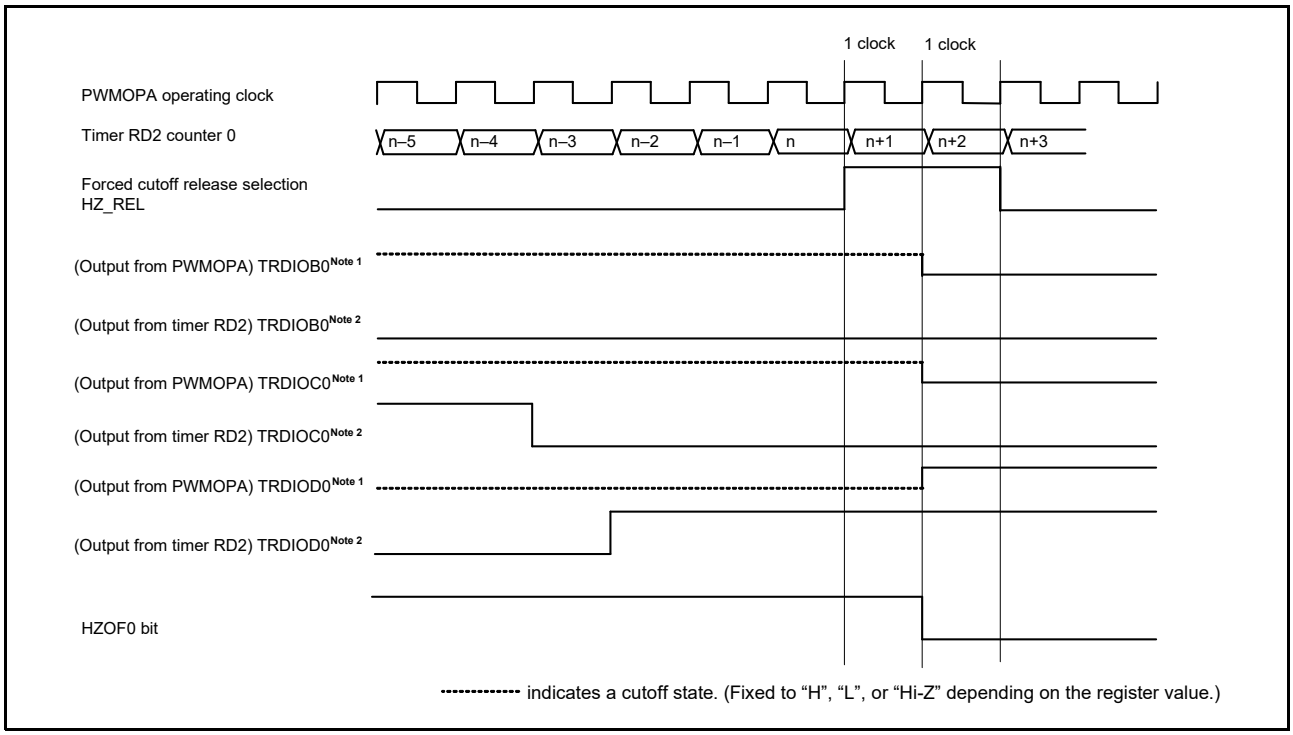


<1> The signals output from the TRDIOB0, TRDIOC0, and TRDIOD0 pins enter the cutoff state when the rising edge of the comparator 3 output signal is detected.

<2> The HZ_REL bit is set to 1 to release the pins from the forced cutoff state immediately.

<3> After the release from the forced cutoff state, the HZ_REL bit becomes 0.
For the detailed cutoff timing, see **Figure 12 - 126 Detailed Timing Diagram of Forced Cutoff**.

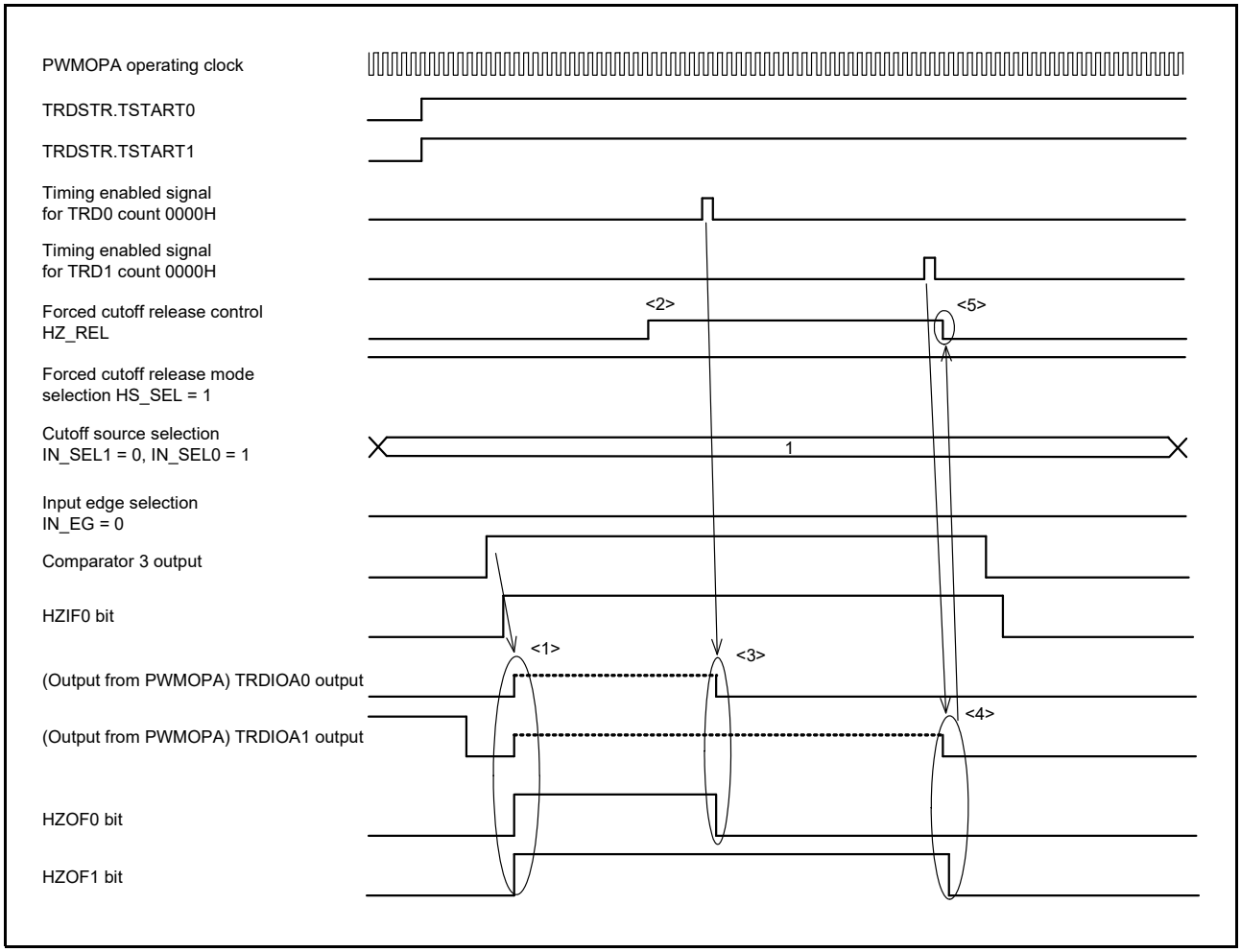
Figure 12 - 135 Detailed Timing Diagram of Release from Cutoff State



2. Conditional release via software (ACT = 1)
   If ACT is set to 1, the pins can be released from the forced cutoff state via a signal from timer RD2 after the HZ_REL bit of the OPCTL0 register is set to 1. After the release from the forced cutoff state, the HZ_REL bit automatically becomes 0.
   Release via hardware resumes output using the release signal from timer RD2 as a trigger after an output forced cutoff release source is detected. Release via software resumes output using the release signal from timer RD2 as a trigger after the HZ_REL bit is set to 1. The release timing is the same.

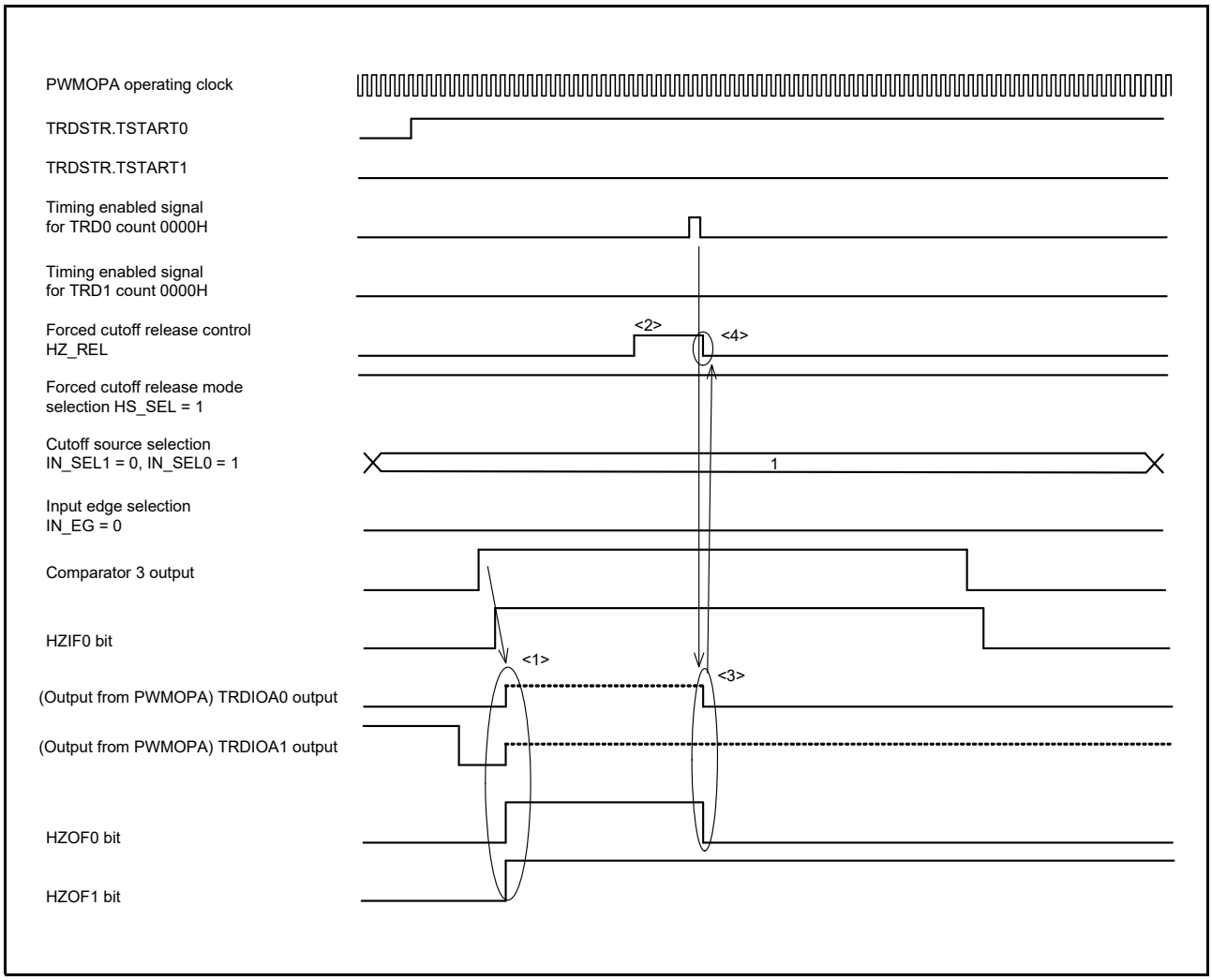
   a) Timer RD2 is in the output compare function, PWM function, PWM3 mode, or extended PWM mode:
      After the HZ_REL bit is set to 1, the TRDIOA0, TRDIOB0, TRDIOC0, and TRDIOD0 pins are released from the output forced cutoff state when the TRD0 count value becomes 0000H. The TRDIOA1, TRDIOB1, TRDIOC1, and TRDIOD1 pins are released from the output forced cutoff state when the TRD1 count value becomes 0000H.

Figure 12 - 136 Operation Example of Release from Cutoff State via Software (Timer RD2, 2-channel Count)



<1> The signals output from the TRDIOA0 and TRDIOA1 pins enter the cutoff state when the rising edge of the comparator 3 output signal is detected.
<2> After the HZ_REL bit is set to 1, PWMOPA waits until each counter value becomes 0000H.
<3> The TRDIOA0 pin is released from the forced cutoff state when the TRD0 count value becomes 0000H.
<4> The TRDIOA1 pin is released from the forced cutoff state when the TRD1 count value becomes 0000H.
<5> After the release of each channel from the forced cutoff state, the HZ_REL bit becomes 0 automatically.

Figure 12 - 137 Operation Example of Release from Cutoff State via Software (Timer RD2, 1-channel Count)



<1> The signals output from the TRDIOA0 and TRDIOA1 pins enter the cutoff state when the rising edge of the comparator 3 output signal is detected.

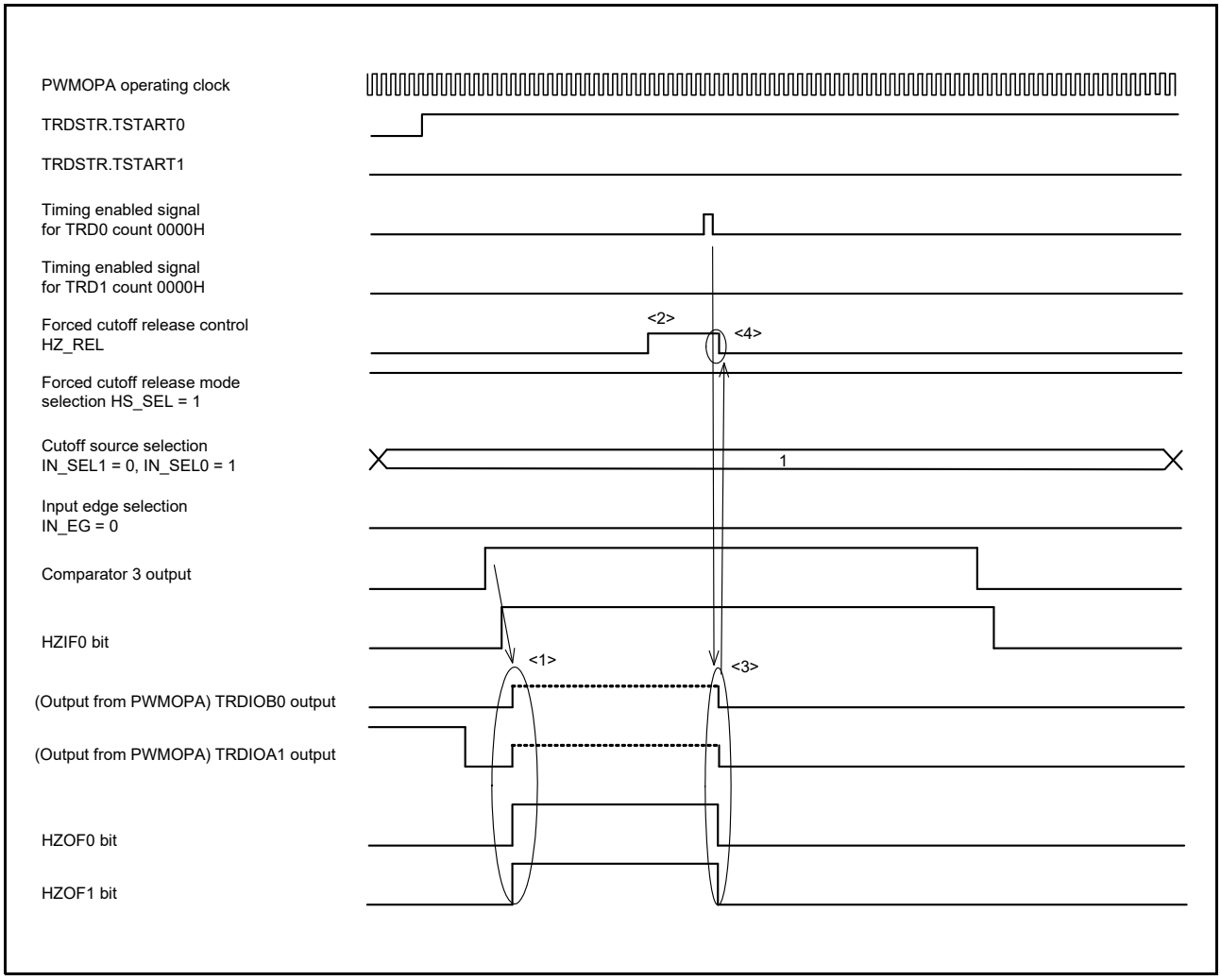
<2> After the HZ_REL bit is set to 1, PWMOPA waits until the counter value becomes 0000H.

<3> The TRDIOA0 pin is released from the forced cutoff state when the TRD0 count value becomes 0000H.

<4> After the release from the forced cutoff state, the HZ_REL bit becomes 0 automatically.
For the detailed cutoff timing, see **Figure 12 - 126 Detailed Timing Diagram of Forced Cutoff**.
For the detailed timing of the release from the cutoff state, see **Figure 12 - 127 Detailed Timing Diagram of Release from Cutoff State (Timer RD2 Count Source = f_{CLK})** and **Figure 12 - 128 Detailed Timing Diagram of Release from Cutoff State (Timer RD2 Count Source = $f_{CLK}/2$)**.
For the timing when HZ_REL bit becomes 0 automatically, see **Figure 12 - 135 Detailed Timing Diagram of Release from Cutoff State**.

b) Timer RD2 is in the reset synchronous PWM mode:
All TRDIO pins are released from the output forced cutoff state when the TRD0 count value becomes 0000H after the HZ_REL bit is set to 1.

Figure 12 - 138 Operation Example of Release from Cutoff State via Software



<1> The signals output from the TRDIOB0 and TRDIOA1 pins enter the cutoff state when the rising edge of the comparator 3 output signal is detected.

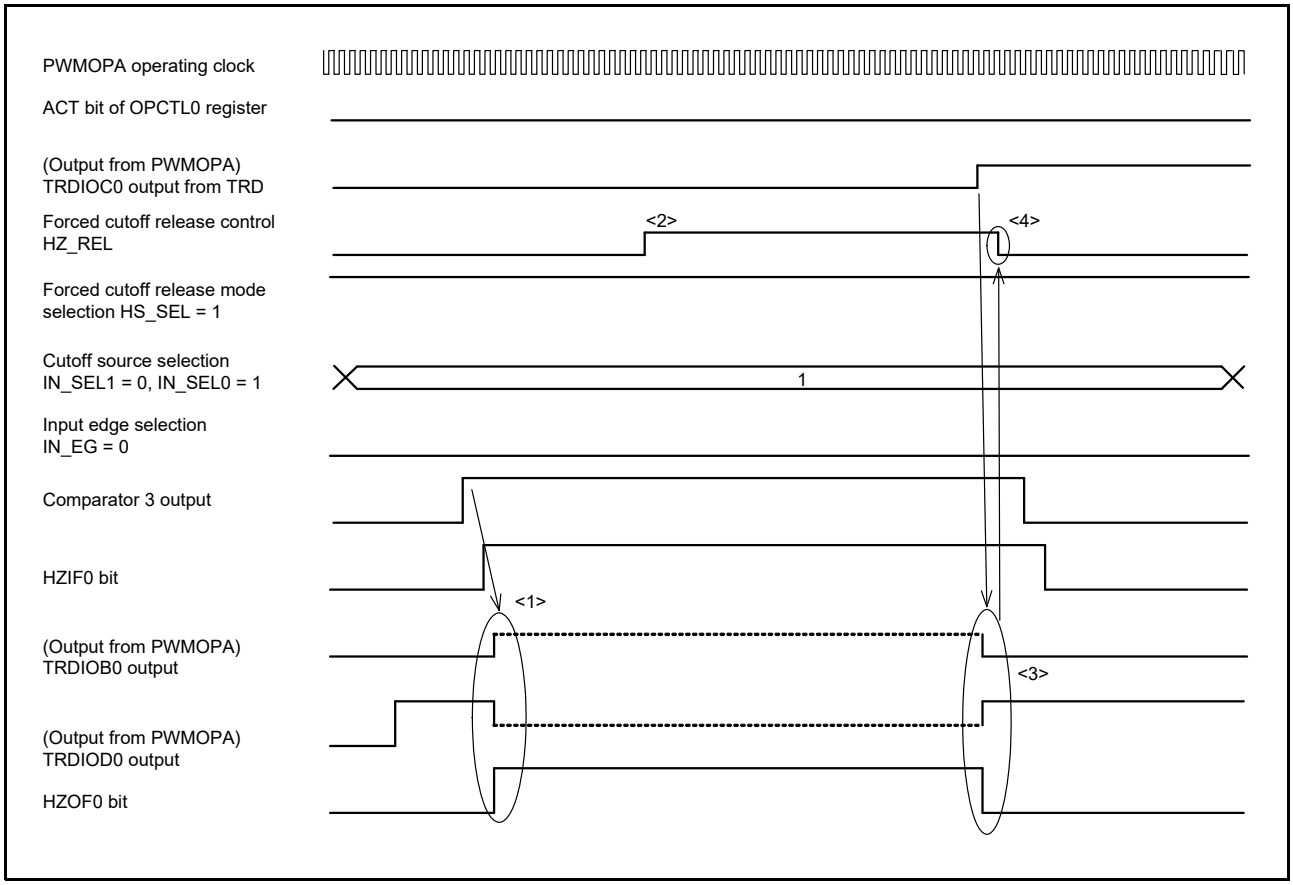
<2> After the HZ_REL bit is set to 1, PWMOPA waits until the counter value of channel 0 of timer RD2 becomes 0000H.

<3> The TRDIOB0 and TRDIOA1 pins are released from the forced cutoff state when the TRD0 count value becomes 0000H (timer RD2 channel 1 operation is not affected).

<4> After the release from the forced cutoff state, the HZ_REL bit becomes 0 automatically.
For the detailed cutoff timing, see **Figure 12 - 126 Detailed Timing Diagram of Forced Cutoff**.
For the detailed timing of the release from the cutoff state, see **Figure 12 - 127 Detailed Timing Diagram of Release from Cutoff State (Timer RD2 Count Source = f_{CLK})** and **Figure 12 - 128 Detailed Timing Diagram of Release from Cutoff State (Timer RD2 Count Source = $f_{CLK}/2$)**.
For the timing when HZ_REL bit becomes 0 automatically, see **Figure 12 - 135 Detailed Timing Diagram of Release from Cutoff State**.

c) Timer RD2 is in the complementary PWM mode or extended complementary PWM mode:
After the HZ_REL bit is set to 1, the output pins of timer RD2 are released from the forced cutoff state at both edges, rising edge, or falling edge of the TRDIOC0, which is selected by the setting in the OPEDGE register.

Figure 12 - 139 Operation Example of Release from Cutoff State via Software (Example of TRDIOB0 and TRDIOD0)



<1> The signals output from the TRDIOB0 and TRDIOD0 pins enter the cutoff state when the rising edge of the comparator 3 output signal is detected.

<2> After the HZ_REL bit is set to 1, PWMOPA waits for the TRDIOC0 rising signal.

<3> When the rising edge of TRDIOC0 is detected, the TRDIOB0 and TRDIOD0 pins are released from the forced cutoff state.

<4> After the release from the forced cutoff state, the HZ_REL bit becomes 0 automatically.

For the detailed cutoff timing, see **Figure 12 - 126 Detailed Timing Diagram of Forced Cutoff**.

For the detailed timing of the release from the cutoff state, see the following figures.

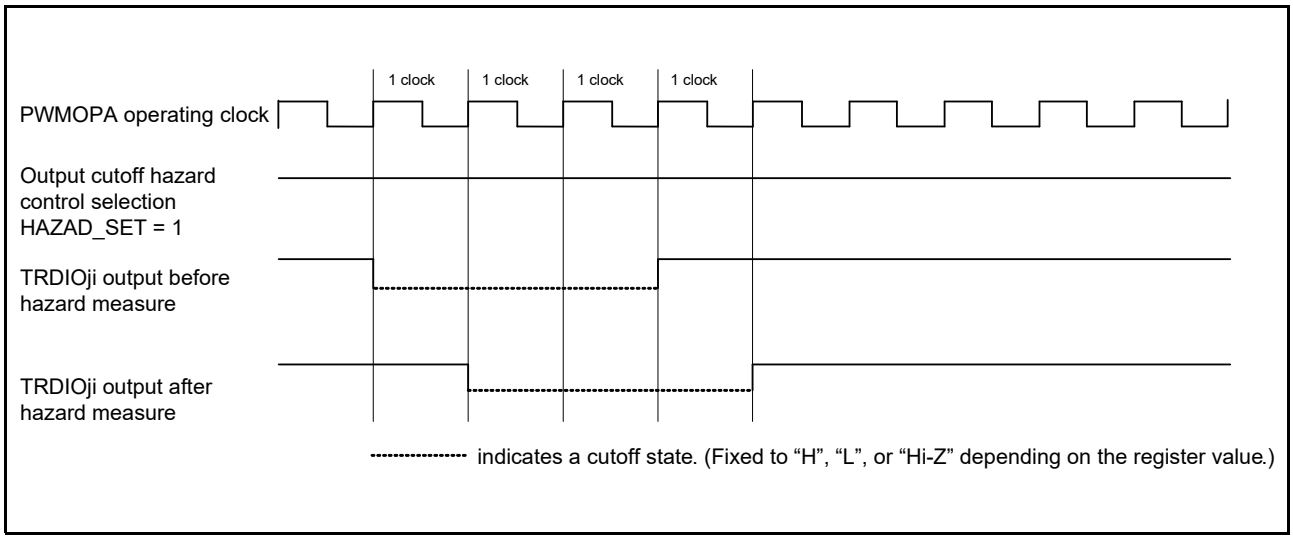
- **Figure 12 - 130 Detailed Timing Diagram of Release from Cutoff State (Timer RD2 Count Source = f_{CLK}, Timer RD2 Decremented)**
- **Figure 12 - 131 Detailed Timing Diagram of Release from Cutoff State (Timer RD2 Count Source = f_{CLK}, Timer RD2 Count = TRDGRA0)**
- **Figure 12 - 132 Detailed Timing Diagram of Release from Cutoff State (Timer RD2 Count Source = $f_{CLK}/2$, Timer RD2 Decremented)**
- **Figure 12 - 133 Detailed Timing Diagram of Release from Cutoff State (Timer RD2 Count Source = $f_{CLK}/2$, Timer RD2 Count = TRDGRA0)**

For the timing when HZ_REL bit becomes 0 automatically, see **Figure 12 - 135 Detailed Timing Diagram of Release from Cutoff State**.

12.8.3.4 Hazard measures

A hazard may occur when a pin enters the cutoff state, a pin is released from the cutoff state, or switching between the TRDIO pin function and port function is performed while timer RD2 is operating. Hazard risks can be handled by setting the hazard control selection bit (HAZAD_SET) to 1. However, the output from timer RD2 when hazard control is enabled is delayed by one clock cycle from the output when the control is disabled.

Figure 12 - 140 Timing Diagram of Hazard Control



Remark j = A to D, i = 0 or 1

12.8.3.5 Output cutoff source detected state and output cutoff source undetected state

Whether the output cutoff source has been detected (HZIF0 = 1) or has not been detected (HZIF0 = 0) is determined based on the level of the signal (INTP0 or comparator 3 output) selected with the cutoff source selection bits (OPCTL0.IN_SEL1 and OPCTL0.IN_SEL0).

If the output cutoff source edge/output cutoff release edge selection bit (OPCTL0.IN_EG) is set to 0, a high-level signal indicates that the output cutoff source has been detected, and a low-level signal indicates that the output cutoff source has not been detected.

If the output cutoff source edge/output cutoff release edge selection bit (OPCTL0.IN_EG) is set to 1, a low-level signal indicates that the output cutoff source has been detected, and a high-level signal indicates that the output cutoff source has not been detected.

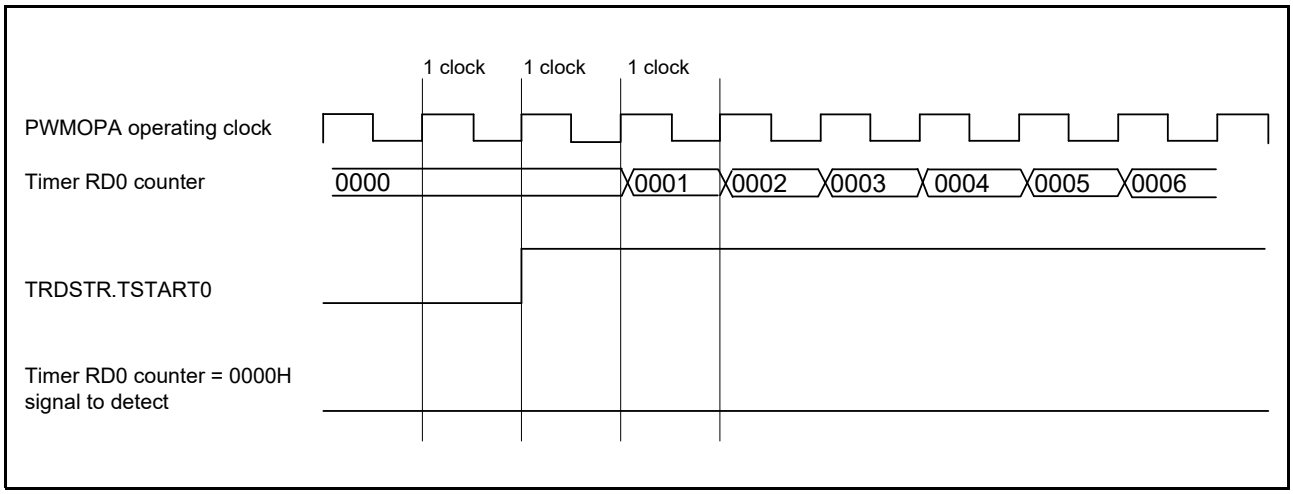
Remark If the output cutoff source has exceeded the threshold before selecting the INTP0 or comparator 3 cutoff source with the IN_SEL1 and IN_SEL0 bits in the OPCTL0 register, the HZIF0 bit is set to 1 after the IN_SEL1 and IN_SEL0 bits are set, but the HZOF0 and HZOF1 bits are not set.

12.8.3.6 Timing when timer RD2 counter value becomes 0000H

When using hardware to release the output pins from the forced cutoff state, the conditions for the release from the output cutoff state vary depending on the operating mode of timer RD2.

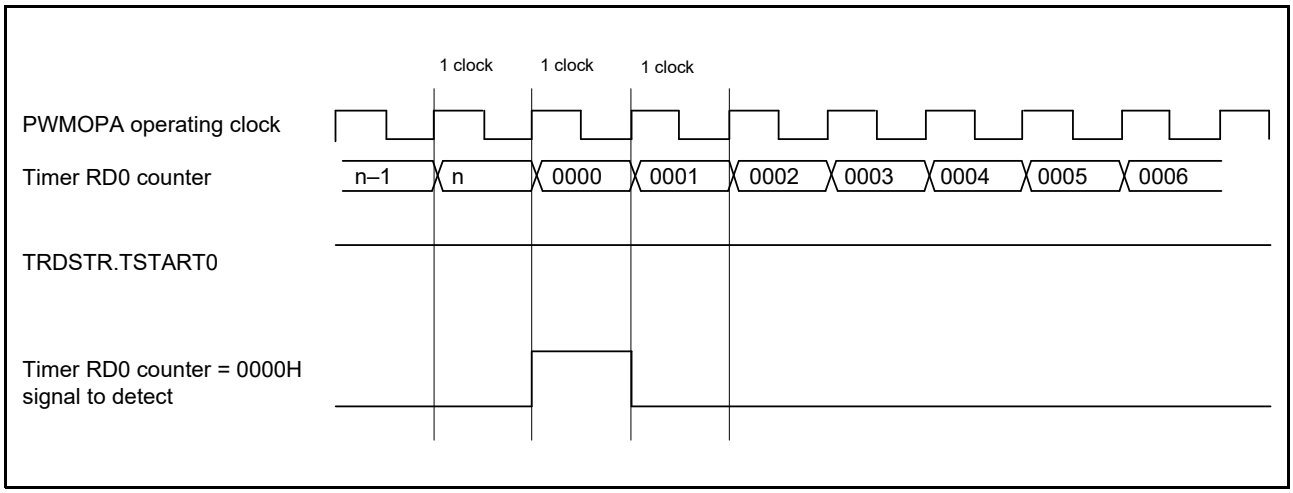
1. Timing when the count value becomes 0000H if timer RD2 is in the output compare function mode
   - Count value = 0000H and timer RD2 starts counting:
     Not released from the output cutoff state.
   - Timer RD2 is counting and 0000H is written to the counter with software:
     Released from the output cutoff state.
   - The counter overflows and becomes 0000H:
     Released from the output cutoff state.
   - The counter becomes 0000H at compare match with the TRDGRA0 register:
     Released from the output cutoff state.
2. Timing when the count value becomes 0000H if timer RD2 is in the PWM function mode
   - Count value = 0000H and timer RD2 starts counting:
     Not released from the output cutoff state.
   - Timer RD2 is counting and 0000H is written to the counter with software:
     Released from the output cutoff state.
   - The counter becomes 0000H at compare match with the TRDGRA0 register:
     Released from the output cutoff state.
3. Timing when the count value becomes 0000H if timer RD2 is in the reset synchronous PWM mode
   - Count value = 0000H and timer RD2 starts counting:
     Not released from the output cutoff state.
   - Timer RD2 is counting and 0000H is written to the counter with software:
     Released from the output cutoff state.
   - The counter becomes 0000H at compare match with the TRDGRA0 register:
     Released from the output cutoff state.
4. Timing when the count value becomes 0000H if timer RD2 is in the PWM3 mode
   - Count value = 0000H and timer RD2 starts counting:
     Not released from the output cutoff state.
   - Timer RD2 is counting and 0000H is written to the counter with software:
     Released from the output cutoff state.
   - The counter becomes 0000H at compare match with the TRDGRA0 register:
     Released from the output cutoff state.
5. Timing when the count value becomes 0000H if timer RD2 is in the extended PWM mode
   - Count value = 0000H and timer RD2 starts counting:
     Not released from the output cutoff state.
   - Timer RD2 is counting and 0000H is written to the counter with software:
     Released from the output cutoff state.
   - The counter becomes 0000H at compare match with the TRDGRA0 register:
     Released from the output cutoff state.

Figure 12 - 141 Judgment Timing for Count Value = 0000H (Timer RD2 Count Starts When Count Value = 0000H)



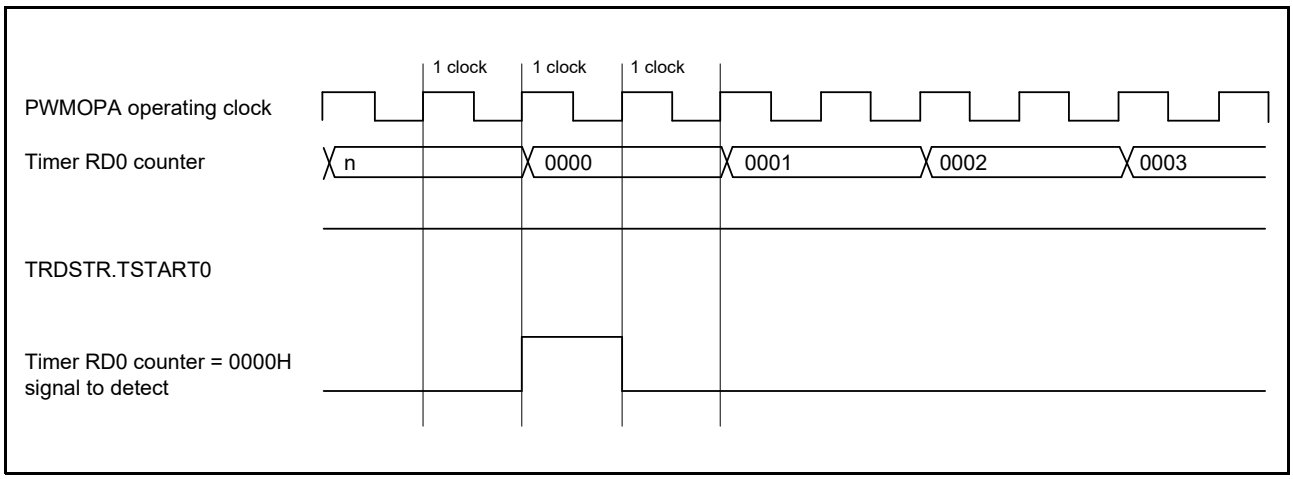
Remark Count value = 0000H is not detected.

Figure 12 - 142 Judgment Timing for Count Value = 0000H (Count Value Becomes 0000H While Counting with Count Source = Operating Clock)



Remark Count value = 0000H is detected.

Figure 12 - 143 Judgment Timing for Count Value = 0000H (Count Value Becomes 0000H While Counting with Count Source = Operating Clock/2)

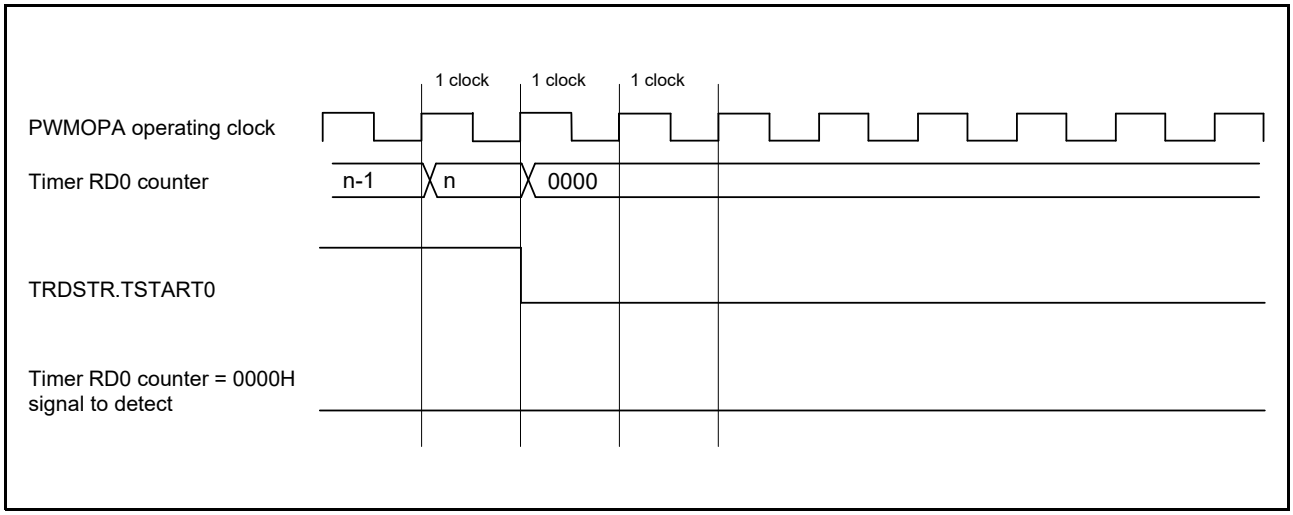


Remark Count value = 0000H is detected.

6. When timer RD2 count value = 0000H and timer RD2 is stopped

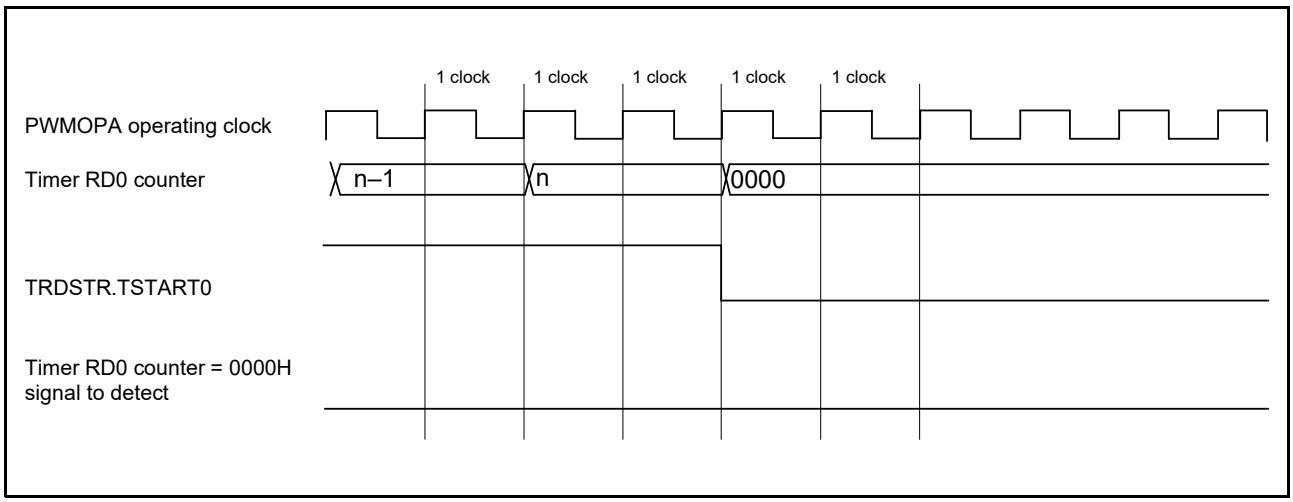
   If timer RD2 is stopped as soon as the timer RD2 count value becomes 0000H, the output pins are not released from the cutoff state.

Figure 12 - 144 Judgment Timing for Count Value = 0000H (Count Source = Operating Clock, Count Is Stopped as Soon as Timer RD2 Counter Value Becomes 0000H)



Remark Count value = 0000H is not detected.

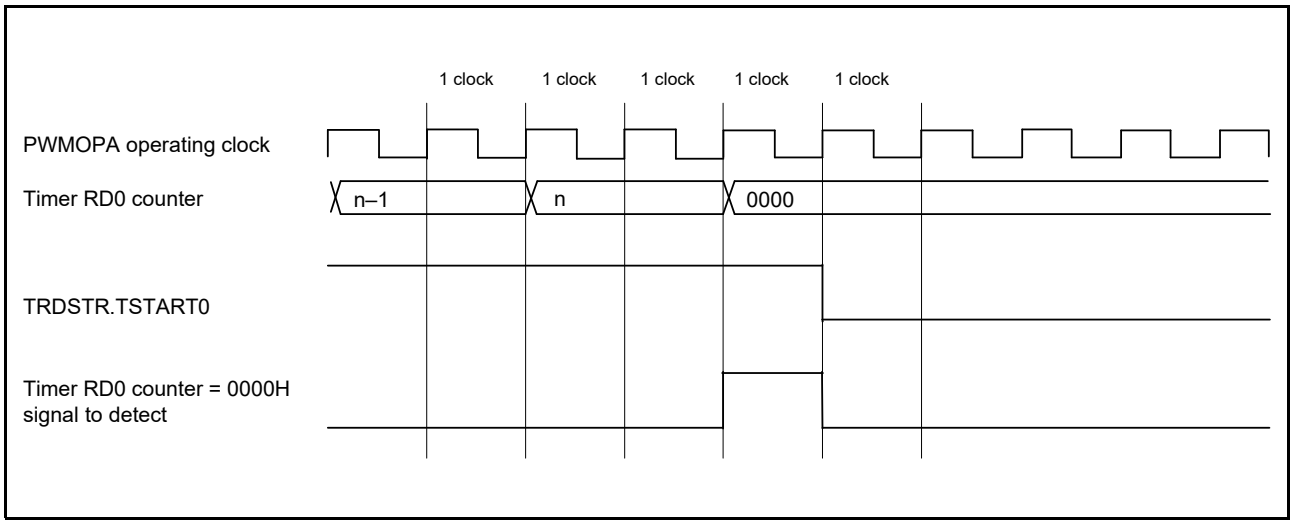
Figure 12 - 145 Judgment Timing for Count Value = 0000H (Count Source = fCLK/2, Count Is Stopped as Soon as Timer RD2 Counter Value Becomes 0000H)



Remark Count value = 0000H is not detected.

If timer RD2 is stopped one cycle after the timer RD2 count value becomes 0000H, the output pins are released from the forced cutoff state.

Figure 12 - 146 Judgment Timing for Count Value = 0000H (Count Source = Operating Clock/2, Count Is Stopped One Cycle after Timer RD2 Counter Value Becomes 0000H)



Remark Count value = 0000H is detected.

12.8.3.7 Setting procedure

PWMOPA can operate in coordination with timer RD2. The code for setting up the coordination should be added to the code for setting up timer RD2. The procedure is as follows:

After setting the clock and mode of timer RD2:

1. Set the PWMOPEN bit of the PER2 register to 1.
2. Set the OPCTL0 register.
3. Set the OPEDGE register.
4. Set the OPDF0 and OPDF1 registers.
   Start timer RD2 operation.
5. Wait until the HZOF1 and HZOF0 bits of the OPSR register indicate that the signals from the output pins are cut off.
6. Release the output pins from the cutoff state using the OPCTL0 register (release via software or hardware can be selected with the HS_SEL bit).

Remark 1. PWMOPA is a control module added to the output forced cutoff function of timer RD2 so that comparator 3 output, external interrupt 0 (INTP0), and event link controller (ELC) can be used as triggers. Accordingly, operation of PWMOPA must always be performed together with timer RD2.

Remark 2. To operate the timer RD2 function independently, do not set the control registers of PWMOPA.

12.8.4 Cautions

1. The following table lists the priorities when pulse output forced cutoff of timer RD2 operates simultaneously with output forced cutoff of PWMOPA.

Table 12 - 31 Forced Cutoff Priorities

		Pin State Control at Output Forced Cutoff of PWMOPA			
		Prohibited	Hi-Z	Low-level	High-level
Pin state control at output forced cutoff of timer RD2	Prohibited	Prohibited	Hi-Z	Low-level	High-level
	Hi-Z	Hi-Z	Hi-Z	Low-level	High-level
	Low-level	Low-level	Hi-Z	Low-level	High-level
	High-level	High-level	Hi-Z	Low-level	High-level

2. If timer RD2 enters the pulse output forced cutoff state when PWMOPA is in the output cutoff state in the complementary PWM mode or extended complementary PWM mode, an output cutoff release edge may be applied to PWMOPA depending on the state of TRDIOC0.
3. When output cutoff is triggered with an event link controller source, make sure to select the release from the cutoff state via software (set the HS_SEL bit to 1).
4. When output cutoff hazard control is selected, the timer RD2 output via PWMOPA is delayed for one cycle of PWMOPA operating clock.
5. If a timer RD2 output pin via PWMOPA is set to timer RD2 output when output cutoff hazard control is selected (HAZAD_SET bit is set to 1), switching between timer RD2 output and port output is possible while timer RD2 is counting.
6. If a timer RD2 output pin via PWMOPA is set to port operation, a hazard may occur upon output cutoff or release from the cutoff state.
7. Set the period of valid-level input of comparator 3 and INTP0 signals to two or more cycles of the PWMOPA operating clock.

Section 13 Timer RG2

13.1 Functions of Timer RG2

Timer RG2 supports the following four modes:

- Timer mode:
  - Input capture function: Counting on rising edges, falling edges, or both edges
  - Output compare function: Low-level output/high-level output/toggled output
- PWM mode: PWM output with any duty cycle is possible.
- PWM2 mode: The one-shot waveform or PWM waveform is output after the waiting time has elapsed following the arrival of a trigger. PWM output with any duty cycle is possible.
- Phase counting mode: Automatic measurement for a two-phase encoder by counting is possible. Phase change time measurement for a two-phase encoder by counting is possible.

13.2 Configuration of Timer RG2

Figure 13 - 1 shows the timer RG2 block diagram and **Table 13 - 1** lists the timer RG2 pin configuration.

Figure 13 - 1 Timer RG2 Block Diagram

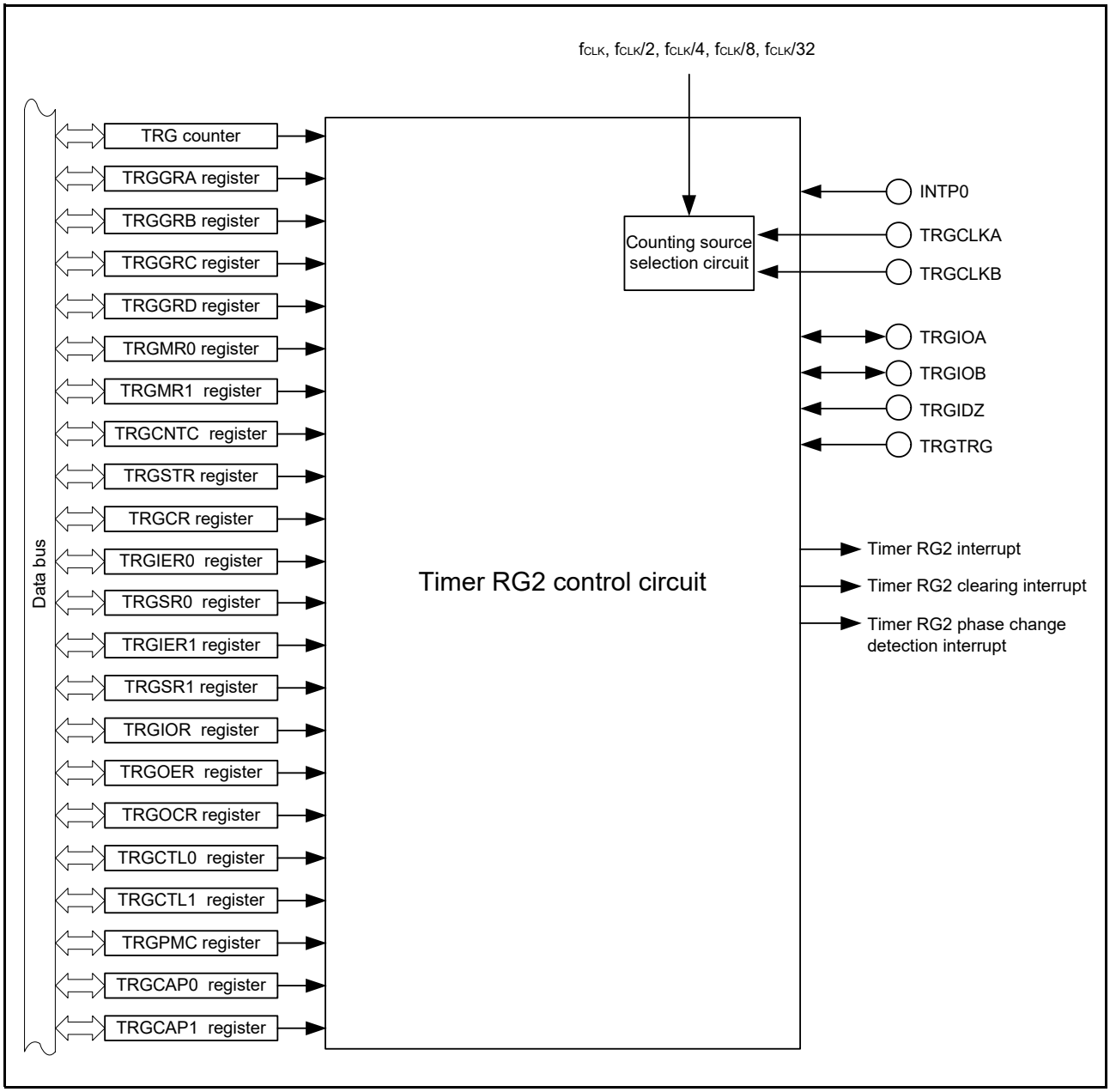


Table 13 - 1 Timer RG2 Pin Configuration

Pin Name	Alternate Port Name	I/O	Function
TRGCLKA	P00	Input	• In phase counting mode A-phase input • Other than in phase counting mode External clock A input
TRGCLKB	P01	Input	• In phase counting mode B-phase input • Other than in phase counting mode External clock B input
TRGIDZ	P120	Input	• In phase counting mode Z (INDEX) signal input • Other than in phase counting mode Not used
INTP0	P137	Input	• In timer mode (with the input capture function in use) Not used • Other than in timer mode (with the input capture function in use) Input of the signal for forcibly shutting off pulse output
TRGIOA	P50 (P17**Note**)	I/O	• In timer mode (with the output compare function in use) TRGGRA output-compare output • In timer mode (with the input capture function in use) TRGGRA input-capture input • In PWM mode or PWM2 mode PWM output
TRGIOB	P51 (P16**Note**)	I/O	• In timer mode (with the output compare function in use) TRGGRB output-compare output • In timer mode (with the input capture function in use) TRGGRB input-capture input • In PWM mode or PWM2 mode Not used
TRGTRG	P120	Input	• In PWM2 mode Input of the trigger for clearing the TRG counter • Other than in PWM2 mode Not used

Note The PIOR20 bit of the PIOR2 register can be used to select the port pins to which the TRGIOA and TRGIOB pin functions are assigned. For details, see **Section 7 Port Functions**.

13.3 Registers to Control Timer RG2

The following registers are used to control timer RG2.

- Peripheral enable register 2 (PER2)
- Peripheral reset control register 2 (PRR2)
- Timer RG mode register 0 (TRGMR0)
- Timer RG mode register 1 (TRGMR1)
- Timer RG count control register (TRGCNTC)
- Timer RG control register (TRGCR)
- Timer RG start register (TRGSTR)
- Timer RG interrupt enable register 0 (TRGIER0)
- Timer RG interrupt enable register 1 (TRGIER1)
- Timer RG status register 0 (TRGSR0)
- Timer RG status register 1 (TRGSR1)
- Timer RG I/O control register (TRGIOR)
- Timer RG output enable register (TRGOER)
- Timer RG output control register (TRGOCR)
- Timer RG phase counting control register 0 (TRGCTL0)
- Timer RG phase counting control register 1 (TRGCTL1)
- Timer RG counter (TRG)
- Timer RG general registers A, B, C, and D (TRGGRA, TRGGRB, TRGGRC, TRGGRD)
- Timer RG phase change time measurement counter (TRGPMC)
- Timer RG phase change time capture registers 0, 1 (TRGCAP0, TRGCAP1)
- Port mode registers xx (PMxx) (xx = 0, 5, 12)
- Port registers xx (Pxx) (xx = 0, 5, 12)

13.3.1 Peripheral enable register 2 (PER2)

The PER2 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. If timer RG2 is to be used, be sure to set bit 4 (TRGEN) of this register to 1. The PER2 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 13 - 2 Format of Peripheral Enable Register 2 (PER2)

Address: F00FCH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER2	FAAEN	MEMEN	TKBEN	TRGEN	TRD0EN	PWMOPEN	TRXEN	TRJ0EN

TRGEN	Control of supply of an input clock to timer RG2
0	Stops supply of an input clock. • The SFRs used by timer RG2 cannot be written.
1	Enables supply of an input clock. • The SFRs used by timer RG2 can be read and written.

Caution When setting timer RG2, be sure to set TRGEN to 1 first. If TRGEN = 0, writing to the registers which control timer RG2 is ignored, and the value read is 00H or 0000H (except for port mode registers 0, 5, 12 (PM0, PM5, PM12) and port registers 0, 5, 12 (P0, P5, P12)).

13.3.2 Peripheral reset control register 2 (PRR2)

The PRR2 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. To place timer RG2 in the reset state, set bit 4 (TRGRES) of this register to 1. The PRR2 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 13 - 3 Format of Peripheral Reset Control Register 2 (PRR2)

Address: F00FDH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PRR2	FAARES	MEMRES	TKBRES	TRGRES	TRD0RES	PWMOPRES	TRXRES	TRJ0RES

TRGRES	Control resetting of timer RG2
0	Timer RG2 is released from the reset state.
1	Timer RG2 is in the reset state. • The SFRs for use with timer RG2 are initialized.

13.3.3 Timer RG mode register 0 (TRGMR0)

Figure 13 - 4 Format of Timer RG Mode Register 0 (TRGMR0) (1/2)

Address: F03E0H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
TRGMR0	TRGSTART	TRGELCICE	TRGDFCK1	TRGDFCK0	TRGDFB	TRGDFA	TRGMDF	TRGPWM

TRGSTART	TRG counting start[Note 1]
0	Stops counting by the TRG counter and the PWM output signal on the TRGIOA pin is initialized (in PWM mode or PWM2 mode).
1	Starts counting by the TRG counter.

TRGELCICE	ELC input capture request select[Notes 2, 3]
0	External input signal B/digital filtering signal B is selected.
1	Event input (input capture) from the ELC is selected.

TRGDFCK1	TRGDFCK0	Digital filter function clock select[Note 2]
0	0	f_{CLK}/32
0	1	f_{CLK}/8
1	0	f_{CLK}
1	1	Clock selected by the TRGTCK[2:0] bits in the TRGCR register

TRGDFB	Digital filer function select for TRGIOB pin
0	Digital filter function not used
1	Digital filter function used
When the digital filter is used, edge detection is performed after up to five cycles of the sampling clock.	

TRGDFA	Digital filer function select for TRGIOA pin
0	Digital filter function not used
1	Digital filter function used
When the digital filter is used, edge detection is performed after up to five cycles of the sampling clock.	

TRGMDF	Phase counting mode select
0	Increment
1	Phase counting mode
When the TRGMDF bit is set to 0, the counter counts the counting source set by the TRGTCK[2:0] bits in the TRGCR register. When the TRGMDF bit is set to 1, the counter counts the phase of input signals from the TRGCLKj pin (j = A or B) as listed in **Table 13 - 18 Increment/Decrement Conditions for the TRG Counter**.	

Figure 13 - 4 Format of Timer RG Mode Register 0 (TRGMR0) (2/2)

TRGPWM	PWM mode select
0	Timer mode
1	PWM mode

Note 1. Write 0 to the TRGSTART bit while the TRGCSEL bit in the TRGSTR register is set to 1.

Note 2. Set this bit while the TRGSTART bit is 0 (to stop counting).

Note 3. To enable event input (input capture) from the ELC, set TRGIOB2 = 1 and TRGIOB[1:0] = 00B (rising edge) in the TRGIOR register.

13.3.4 Timer RG mode register 1 (TRGMR1)

Figure 13 - 5 Format of Timer RG Mode Register 1 (TRGMR1)

Address: F03F0H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	5	4	<3>	<2>	<1>	<0>
TRGMR1	TRGPENB	TRGPENA	0	0	TRGTCEG1	TRGTCEG0	TRGDFCLR	TRGPWM2

TRGPENB	Control over forcibly shutting off pulse output on the TRGIOB pin[Note 1]
0	Forcibly shutting off pulse output is disabled.
1	TRGIOB is in the high-Z state.

TRGPENA	Control over forcibly shutting off pulse output on the TRGIOA pin[Note 2]
0	Forcibly shutting off pulse output is disabled.
1	TRGIOA is in the high-Z state.

TRGTCEG1	TRGTCEG0	TRGTRG input edge select[Note 3]
0	0	TRGTRG input is disabled.
0	1	Rising edge
1	0	Falling edge
1	1	Rising and falling edges

TRGDFCLR	Digital filer function select for TRGTRG pin[Note 4]
0	Digital filter function not used
1	Digital filter function used

TRGPWM2	PWM2 mode select
0	Timer mode or PWM mode
1	PWM2 mode

Note 1. When the TRGIOB pin function is not to be used on an output port pin, set this bit to 0. The setting of the TRGPENB bit takes priority over that in the TRGOER register.

Note 2. When the TRGIOA pin function is not to be used on an output port pin, set this bit to 0. The setting of the TRGPENA bit takes priority over that in the TRGOER register.

Note 3. When the TRGPWM2 bit is set to 0, set the TRGTCEG[1:0] bits to 00B.

Note 4. When the TRGDFCLR bit is set to 1, the TRGDFCK[1:0] bits of the TRGMR0 register are used to select the clock for use in digital filtering.

Caution 1. Be sure to set bits 5 and 4 to 0.

Caution 2. When writing to the TRGMR1 register, ensure that the TRGSTART bit in the TRGMR0 register is set to 0 (to stop counting) at the time.

13.3.5 Timer RG count control register (TRGCNTC)

The TRGCNTC register is used to set the conditions for counting in phase counting mode. Set the TRGCNTC register while counting is stopped (TRGMR0.TRGSTART = 0).

Figure 13 - 6 Format of Timer RG Count Control Register (TRGCNTC) (1/2)

Address: F03E1H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TRGCNTC	CNTEN7	CNTEN6	CNTEN5	CNTEN4	CNTEN3	CNTEN2	CNTEN1	CNTEN0

CNTEN7	Counter enable 7
0	Disabled
1	Increment When TRGCLKA input is at the low level and on rising edges of TRGCLKB input

CNTEN6	Counter enable 6
0	Disabled
1	Increment When TRGCLKB input is at the high level and on rising edges of TRGCLKA input

CNTEN5	Counter enable 5
0	Disabled
1	Increment When TRGCLKA input is at the high level and on falling edges of TRGCLKB input

CNTEN4	Counter enable 4
0	Disabled
1	Increment When TRGCLKB input is at the low level and on falling edges of TRGCLKA input

CNTEN3	Counter enable 3
0	Disabled
1	Decrement When TRGCLKB input is at the high level and on falling edges of TRGCLKA input

CNTEN2	Counter enable 2
0	Disabled
1	Decrement When TRGCLKA input is at the low level and on falling edges of TRGCLKB input

CNTEN1	Counter enable 1
0	Disabled
1	Decrement When TRGCLKB input is at the low level and on rising edges of TRGCLKA input

Figure 13 - 6 Format of Timer RG Count Control Register (TRGCNTC) (2/2)

CNTEN0	Counter enable 0
0	Disabled
1	Decrement When TRGCLKA input is at the high level and on rising edges of TRGCLKB input

13.3.6 Timer RG control register (TRGCR)

Figure 13 - 7 Format of Timer RG Control Register (TRGCR)

Address: F03E2H
After reset: 00H
R/W: R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
TRGCR	0	TRGCCLR1	TRGCCLR0	TRGCKEG1	TRGCKEG0	TRGTCK2	TRGTCK1	TRGTCK0

TRGCCLR1	TRGCCLR0	TRG counter clearing source select
0	0	Clearing the TRG counter is disabled.
0	1	The TRG counter is cleared by input capture to or compare match with TRGGRA.
1	0	The TRG counter is cleared by input capture to or compare match with TRGGRB.
1	1	Setting prohibited

TRGCKEG1	TRGCKEG0	External clock active edge select Notes 1, 2, 3
0	0	Counting of rising edges
0	1	Counting of falling edges
1	0	Counting of rising and falling edges
1	1	Setting prohibited

TRGTCK2	TRGTCK1	TRGTCK0	Counting source select Note 1	
			In timer mode, PWM mode, or PWM2 mode	In phase counting mode
0	0	0	f_{CLK}	f_{CLK}
0	0	1	$f_{CLK}/2$	$f_{CLK}/2$
0	1	0	$f_{CLK}/4$	$f_{CLK}/4$
0	1	1	$f_{CLK}/8$	$f_{CLK}/8$
1	0	0	$f_{CLK}/32$	Setting prohibited
1	0	1	TRGCLKA input	Setting prohibited
1	1	0	Setting prohibited	Setting prohibited
1	1	1	TRGCLKB input	Setting prohibited

Note 1. In phase counting mode, the settings of the TRGTCK[2:0] and TRGCKEG[1:0] bits for the timer RG counter (TRG) are invalid and the operation in phase counting mode has priority. When the phase change time measurement function is to be used in phase counting mode, the setting of the TRGTCK[2:0] bits for the timer RG phase change time measurement counter (TRGPMC) is valid.

Note 2. The TRGCKEG[1:0] bits are valid when the TRGTCK[2:0] bits are set to an external clock (TRGCLKA or TRGCLKB). When not set to an external clock, they are invalid.

Note 3. In PWM2 mode, set the TRGCCLR[1:0] bits to 00B or 01B.

Caution When writing to the TRGCR register, make sure the TRGSTART bit in the TRGMR0 register is 0 (to stop counting).

13.3.7 Timer RG start register (TRGSTR)

Figure 13 - 8 Format of Timer RG Start Register (TRGSTR)

Address: F03F5H
After reset: 02H
R/W: R/W

Symbol	7	6	5	4	3	2	<1>	<0>
TRGSTR	0	0	0	0	0	0	TRGCSEL	TRGTSTART

TRGCSEL	TRG counting select[Notes 1, 2, 3]
0	The TRG counter stops counting in response to a compare match with the TRGGRA or TRGGRB register.
1	The TRG counter continues counting after a compare match with the TRGGRA or TRGGRB register.

TRGTSTART	TRG counting start flag[Note 4]
0	Counting has stopped.[Note 5]
1	Counting has started.

Note 1. The TRGCSEL bit is usable to set the response to a match in comparison during counting up or in PWM or PWM2 mode. When the phase counting mode or input capture function is to be selected, set this bit to 1.

Note 2. The counter operates under the conditions below:

- In timer mode (with the output compare function in use) or PWM mode
  - Compare match with the TRGGRA register when the setting of the TRGCCLR[1:0] bits in the TRGCR register is 01B
  - Compare match with the TRGGRB register when the setting of the TRGCCLR[1:0] bits in the TRGCR register is 10B
- In PWM2 mode
  - Compare match with the TRGGRA register

Note 3. To use one-shot pulse output in PWM2 mode, set this bit to 0.

Note 4. When a compare match with the TRGGRA or TRGGRB register occurs while the TRGCSEL bit is set to 0, this flag is set to 0 (indicating that counting has stopped).

Note 5. Write 0 to the TRGSTART bit while the TRGCSEL bit is set to 1. To stop counting without waiting for a compare match with the TRGGRA or TRGGRB register while the TRGCSEL bit is set to 0, set the TRGCSEL bit to 1 and then write 0 to the TRGSTART bit. This requires two rounds of register access to set the TRGCSEL and TRGSTART bits.

Caution 1. Be sure to set bits 7 to 2 to 0.

Caution 2. The TRGTSTART bit of this register and the TRGSTART bit in the TRGMR0 register mirror each other. When one is set or cleared, the other will have the same value.

13.3.8 Timer RG interrupt enable register 0 (TRGIER0)

Figure 13 - 9 Format of Timer RG Interrupt Enable Register 0 (TRGIER0)

Address: F03E3H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	5	4	<3>	<2>	<1>	<0>
TRGIER0	TRGIMIED	TRGIMIEC	0	0	TRGOVIE	TRGUDIE	TRGIMIEB	TRGIMIEA

TRGIMIED	Compare-match interrupt enable D
0	Interrupts in response to setting of the TRGIMFD flag are disabled.
1	Interrupts in response to setting of the TRGIMFD flag are enabled.

TRGIMIEC	Compare-match interrupt enable C
0	Interrupts in response to setting of the TRGIMFC flag are disabled.
1	Interrupts in response to setting of the TRGIMFC flag are enabled.

TRGOVIE	Overflow interrupt enable
0	Interrupts in response to setting of the TRGOVF flag are disabled.
1	Interrupts in response to setting of the TRGOVF flag are enabled.

TRGUDIE	Underflow interrupt enable
0	Interrupts in response to setting of the TRGUDF flag are disabled.
1	Interrupts in response to setting of the TRGUDF flag are enabled.

TRGIMIEB	Input-capture/compare-match interrupt enable B
0	Interrupts in response to setting of the TRGIMFB flag are disabled.
1	Interrupts in response to setting of the TRGIMFB flag are enabled.

TRGIMIEA	Input-capture/compare-match interrupt enable A
0	Interrupts in response to setting of the TRGIMFA flag are disabled.
1	Interrupts in response to setting of the TRGIMFA flag are enabled.

Remark TRGIMFA, TRGIMFB, TRGIMFC, TRGIMFD, TRGOVF, TRGUDF: Flags in the TRGSR0 register

13.3.9 Timer RG interrupt enable register 1 (TRGIER1)

Figure 13 - 10 Format of Timer RG Interrupt Enable Register 1 (TRGIER1)

Address: F03F3H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	<1>	<0>
TRGIER1	0	0	0	0	0	0	TRGPCIE	TRGZCIE

TRGPCIE	Counter clearing interrupt on phase period compare match enable
0	Interrupts in response to setting of the TRGPCLF flag are disabled.
1	Interrupts in response to setting of the TRGPCLF flag are enabled.

TRGZCIE	Counter clearing interrupt on Z-signal detection enable
0	Interrupts in response to setting of the TRGZCLF flag are disabled.
1	Interrupts in response to setting of the TRGZCLF flag are enabled.

Caution **Be sure to set bits 7 to 2 to 0.**

Remark TRGZCLF, TRGPCLF: Flags in the TRGSR1 register

13.3.10 Timer RG status register 0 (TRGSR0)

Figure 13 - 11 Format of Timer RG Status Register 0 (TRGSR0)

Address: F03E4H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	5	<4>	<3>	<2>	<1>	<0>
TRGSR0	TRGIMFD	TRGIMFC	0	TRGDIRF	TRGOVF	TRGUDF	TRGIMFB	TRGIMFA

TRGIMFD	Compare-match flag D
[Condition for setting to 0] Write 0 after reading**Notes 1, 2** [Condition for setting to 1] See **Table 13 - 2 Conditions for Setting Each Flag to 1**.	

TRGIMFC	Compare-match flag C
[Condition for setting to 0] Write 0 after reading**Notes 1, 2** [Condition for setting to 1] See **Table 13 - 2 Conditions for Setting Each Flag to 1**.	

TRGDIRF	Count direction flag
0	The TRG counter is decremented.
1	The TRG counter is incremented.

TRGOVF	Overflow flag**Note 3**
[Condition for setting to 0] Write 0 after reading**Note 1** [Condition for setting to 1] See **Table 13 - 2 Conditions for Setting Each Flag to 1**.	

TRGUDF	Underflow flag
[Condition for setting to 0] Write 0 after reading**Note 1** [Condition for setting to 1] See **Table 13 - 2 Conditions for Setting Each Flag to 1**.	

TRGIMFB	Input-capture/compare-match flag B
[Condition for setting to 0] Write 0 after reading**Notes 1, 2** [Condition for setting to 1] See **Table 13 - 2 Conditions for Setting Each Flag to 1**.	

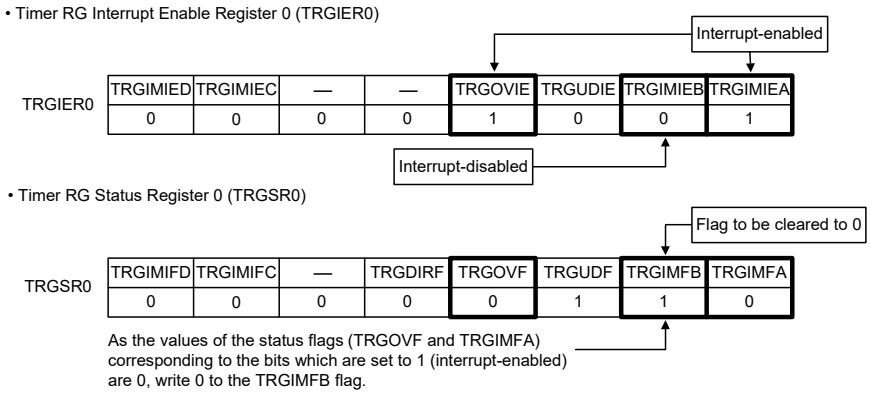
TRGIMFA	Input-capture/compare-match flag A
[Condition for setting to 0] Write 0 after reading**Notes 1, 2** [Condition for setting to 1] See **Table 13 - 2 Conditions for Setting Each Flag to 1**.	

Note 1. The writing results are as follows:

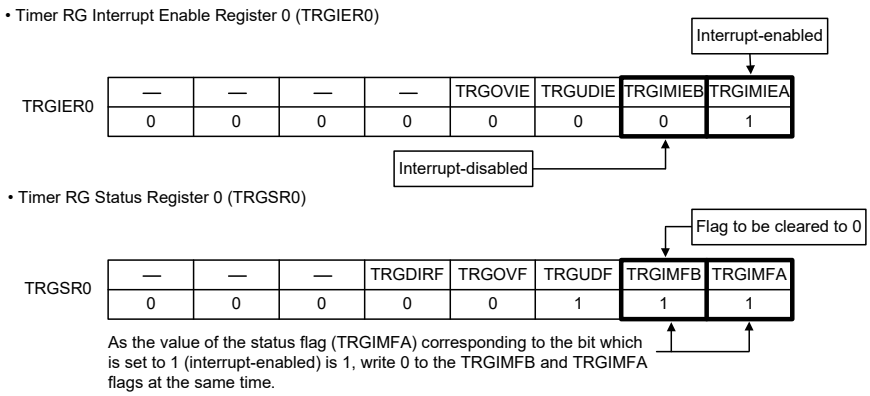
- Writing 1 has no effect.
- If the read value is 0, the bit remains unchanged even if 0 is written to it.
  (Even if the bit is changed from 0 to 1.)
- If the read value is 1, writing 0 to the bit sets it to 0.
  To clear the status flag of an interrupt source (applicable status flag) of timer RG2 to 0 while the corresponding interrupt is disabled in the timer RG interrupt enable register 0 (TRGIER0), use any of the following methods a) to c).

a) Set 00H (all interrupts disabled) in timer RG interrupt enable register 0 (TRGIER0) and write 0 to the applicable status flag.

b) When a bit is set to 1 (interrupt-enabled) in timer RG interrupt enable register 0 (TRGIER0) and the value of the status flag of the interrupt source corresponding to the bit is 0, write 0 to the applicable status flag.
Example: To clear the TRGIMFB flag to 0 when bits TRGIMIEA and TRGOVIE are set to 1 (interrupt-enabled) and the TRGIMIEB bit is set to 0 (interrupt-disabled).



c) When a bit is set to 1 (interrupt-enabled) in timer RG interrupt enable register 0 (TRGIER0) and the value of the status flag of the interrupt source corresponding to the bit is 1, write 0 to the status flag and applicable status flag at the same time.
Example: To clear the TRGIMFB flag to 0 when the TRGIMIEA bit is set to 1 (interrupt-enabled) and the TRGIMIEB bit is set to 0 (interrupt-disabled).



Note 2. When the DTC is used, the TRGIMFA, TRGIMFB, TRGIMFC, TRGIMFD flags are set to 1 after DTC transfer is completed.

Note 3. When the counter value of timer RG2 changes from FFFFH to 0000H, the TRGOVF flag is set to 1. Also, if the counter value of timer RG2 changes from FFFFH to 0000H due to an input capture/compare match during operation according to the setting of the TRGCCLR[1:0] bits in the TRGCR register, the TRGOVF flag is set to 1.

Table 13 - 2 Conditions for Setting Each Flag to 1

Flag	Timer Mode[Note 1]		PWM Mode/PWM2 Mode
	Input Capture Function	Output Compare Function	
TRGOVF	The TRG counter overflows.		
TRGUDF	The TRG counter underflows (only in phase counting mode).		
TRGIMFD	The values of the TRG counter and TRGGRD register match.[Note 2]		
TRGIMFC	The values of the TRG counter and TRGGRC register match.[Note 2]		
TRGIMFB	Input edge of TRGIOB pin[Note 3]	The values of the TRG counter and TRGGRB register match.	
TRGIMFA	Input edge of TRGIOA pin[Note 3]	The values of the TRG counter and TRGGRA register match.	

Note 1. Phase counting mode is the counting method of the timer RG count register. The above timer modes and PWM mode can be used by making the corresponding settings.

Note 2. Including when the TRGBUFA and TRGBUFB bits in the TRGIOR register are set to 1 (using as the buffer registers for TRGGRA and TRGGRB)

Note 3. Edge selected by the TRGIOj[1:0] bits (j = A or B) in the TRGIOR register.

13.3.11 Timer RG status register 1 (TRGSR1)

Figure 13 - 12 Format of Timer RG Status Register 1 (TRGSR1)

Address: F03F4H
After reset: 00H
R/W: R/W

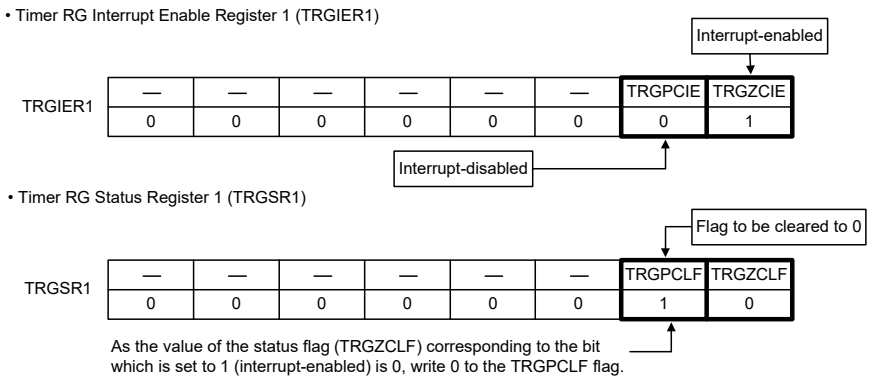
Symbol	7	6	5	4	3	2	<1>	<0>
TRGSR1	0	0	0	0	0	0	TRGPCLF	TRGZCLF

TRGPCLF	Flag to indicate clearing the counter on phase period compare match
0	Write 0 after reading.**Note**
1	See **Table 13 - 3 Conditions for Setting Each Flag to 1**.

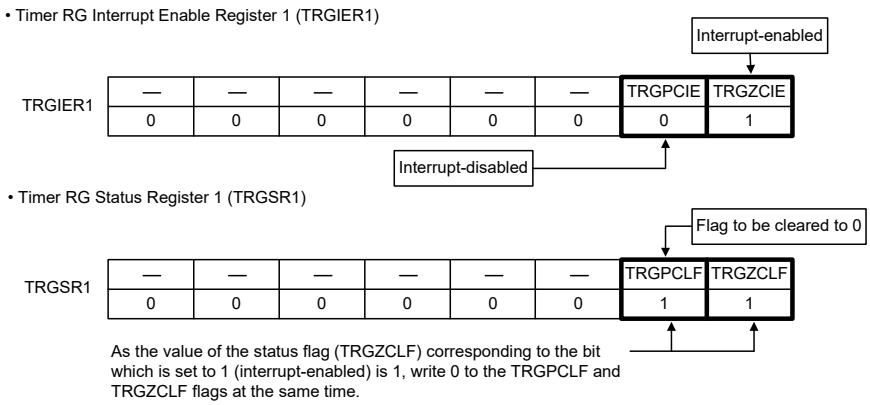
TRGZCLF	Flag to indicate clearing the counter on Z-signal detection
0	Write 0 after reading.**Note**
1	See **Table 13 - 3 Conditions for Setting Each Flag to 1**.

Note 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.)
- If the read value is 1, writing 0 to the bit sets it to 0.
  To clear the status flag of an interrupt source (applicable status flag) of timer RG2 to 0 while the corresponding interrupt is disabled in the timer RG interrupt enable register 1 (TRGIER1), use any of the following methods a) to c).
  a) Set 00H (all interrupts disabled) in timer RG interrupt enable register 1 (TRGIER1) and write 0 to the applicable status flag.
  b) When a bit is set to 1 (interrupt-enabled) in timer RG interrupt enable register 1 (TRGIER1) and the value of the status flag of the interrupt source corresponding to the bit is 0, write 0 to the applicable status flag.
  Example: To clear the TRGPCLF flag to 0 when the TRGZCIE bit is set to 1 (interrupt-enabled) and the TRGPCIE bit is set to 0 (interrupt-disabled).



c) When a bit is set to 1 (interrupt-enabled) in timer RG interrupt enable register 1 (TRGIER1) and the value of the status flag of the interrupt source corresponding to the bit is 1, write 0 to the status flag and applicable status flag at the same time.
Example: To clear the TRGPCLF flag to 0 when the TRGZCIE bit is set to 1 (interrupt-enabled) and the TRGPCIE bit is set to 0 (interrupt-disabled).



Caution Be sure to set bits 7 to 2 to 0.

Table 13 - 3 Conditions for Setting Each Flag to 1

Flag	Phase Counting Mode (TRGMR0.TRGMDF = 1)
TRGPCLF	• An active edge on the TRGIDZ pin is detected while the TRGSCE bit in the TRGCTL0 register is set to 0. • Any of the conditions for clearing the TRG counter selected by using the TRGACL, TRGBCL, and TRGZCL bits in the TRGCTL0 register is met while the TRGSCE bit in the TRGCTL0 register is set to 1.
TRGZCLF	• Counting up proceeds following detection of a match between the TRG counter and TRGGRC register while the TRGECM0 bit in the TRGCTL1 register is set to 1. • Counting down proceeds following detection of a match between the TRG counter and TRGGRD register while the TRGECM1 bit in the TRGCTL1 register is set to 1.

13.3.12 Timer RG I/O control register (TRGIOR)

The TRGIOR register controls I/O pins in timer mode. Set the TRGIOR register while counting is stopped (TRGMR0.TRGSTART = 0).

Figure 13 - 13 Format of Timer RG I/O Control Register (TRGIOR) (1/2)

Address: F03E5H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
TRGIOR	TRGBUFB	TRGIOB2	TRGIOB1	TRGIOB0	TRGBUFA	TRGIOA2	TRGIOA1	TRGIOA0

TRGBUFB	TRGGRD register function select
0	Not used as buffer register for TRGGRB register
1	Used as buffer register for TRGGRB register

TRGIOB2	TRGGRB mode select[Notes 1, 2, 3]
0	Output compare function
1	Input capture function

TRGIOB1	TRGIOB0	TRGGRB control[Note 3]
0	0	Pin output by compare match is disabled.
0	1	Low-level output
1	0	High-level output
1	1	Toggled output
Output on compare match between the TRG counter and TRGGRB register when the output compare function is selected		

TRGIOB1	TRGIOB0	TRGGRB control[Note 3]
0	0	Rising edge on TRGIOB
0	1	Falling edge on TRGIOB
1	0	Both edges on TRGIOB
1	1	Setting prohibited
Input capture of the value in the TRG counter to the TRGGRB register when the input capture function is selected		

TRGBUFA	TRGGRC register function select[Note 3]
0	Not used as buffer register for TRGGRA register
1	Used as buffer register for TRGGRA register

TRGIOA2	TRGGRA mode select[Notes 1, 2, 3]
0	Output compare function
1	Input capture function

Figure 13 - 13 Format of Timer RG I/O Control Register (TRGIOR) (2/2)

TRGIOA1	TRGIOA0	TRGGRA control[Note 3]
0	0	Pin output by compare match is disabled.
0	1	Low-level output
1	0	High-level output
1	1	Toggled output
Output on compare match between the TRG counter and TRGGRA register when the output compare function is selected		

TRGIOA1	TRGIOA0	TRGGRA control
0	0	Rising edge on TRGIOA
0	1	Falling edge on TRGIOA
1	0	Both edges on TRGIOA
1	1	Setting prohibited
Input capture of the value in the TRG counter to the TRGGRA register when the input capture function is selected		

Note 1. When the TRGIOj2 (j = A or B) bit is 1 (input capture function), the TRGGRj register functions as an input capture register.

Note 2. When the TRGIOj2 (j = A or B) bit is 0 (output compare function), the TRGGRj register functions as a compare match register. After a reset, the TRGIOj pin outputs as follows until the TRGIOj[1:0] bits are set and the first compare match occurs.

TRGIOj[1:0] = 01B: High-level output
10B: Low-level output
11B: Low-level output

Note 3. Set the TRGBUFA bit to 0 in PWM2 mode. In PWM mode or PWM2 mode, the settings of bits other than the TRGBUFA and TRGBUFB bits are invalid.

13.3.13 Timer RG output enable register (TRGOER)

Figure 13 - 14 Format of Timer RG Output Enable Register (TRGOER)

Address: F03F1H
After reset: 00H
R/W: R/W

Symbol	<7>	6	5	4	3	2	1	<0>
TRGOER	TRGPTO	0	0	0	0	0	0	TRGSHUTS

TRGPTO	INTP0 signal input for forcibly shutting off timer output enable
0	Input for forcibly shutting off timer output is disabled.
1	Input for forcibly shutting off timer output is enabled.

TRGSHUTS	State of forcible shut-off
0	Timer output is not forcibly shut off.
1	Timer output is forcibly shut off.
This flag is set to 1 on forcible shut-off of pulse output in response to INTP0 input with the TRGPTO bit set to 1. As clearing of the flag is not automatic, write 0 to it while counting is stopped (TRGSTART = 0) to release the pulse output from the forcible shut-off state. Writing 1 to the flag in a mode where this is effective also forcibly shuts off the pulse output.	

Caution Be sure to set bits 6 to 1 to 0.

13.3.14 Timer RG output control register (TRGOCR)

Figure 13 - 15 Format of Timer RG Output Control Register (TRGOCR)

Address: F03F2H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	<1>	<0>
TRGOCR	0	0	0	0	0	0	TRGTOB	TRGTOA

TRGTOB	Timer output level select B[Note 1]
0	Initial level: Low
1	Initial level: High

TRGTOA	Timer output level select A[Note 2]
0	Initial level: Low
1	Initial level: High

Note 1. Writing to the TRGTOB bit in the TRGOCR register causes reflection of the setting in the TRGIOB output. This setting is used in timer mode (with the output compare function in use).

Note 2. Writing to the TRGTOA bit in the TRGOCR register causes reflection of the setting in the TRGIOA output. This setting is used in timer mode (with the output compare function in use) and PWM2 mode.

Caution 1. Be sure to set bits 7 to 2 to 0.

Caution 2. When writing to the TRGOCR register, ensure that the TRGSTART bit in the TRGMR0 register is set to 0 (to stop counting) at the time.

13.3.15 Timer RG phase counting control register 0 (TRGCTL0)

Figure 13 - 16 Format of Timer RG Phase Counting Control Register 0 (TRGCTL0)

Address: F03F6H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	3	2	<1>	<0>
TRGCTL0	TRGSCE	TRGZCL	TRGBCL	TRGACL	0	0	TRGIDZ1	TRGIDZ0

TRGSCE	TRG counter clearing method select for phase counting mode
0	The TRG counter is cleared to 0000H in response to detection of an active edge on the TRGIDZ pin (specified in the TRGIDZ[1:0] bits).
1	The TRG counter is cleared to 0000H when the condition for clearing the TRG counter in response to the input level (specified in the TRGZCL, TRGBCL, or TRGACL bit) on the TRGCLKA, TRGCLKB, or TRGIDZ pin is met.

TRGZCL	TRGIDZ (Z signal) clearing condition setting**Note 1**
0	The low level of the TRGIDZ pin is set as the condition for clearing the TRG counter.
1	The high level of the TRGIDZ pin is set as the condition for clearing the TRG counter.

TRGBCL	TRGCLKB (B phase) clearing condition setting**Note 1**
0	The low level of the TRGCLKB pin is set as the condition for clearing the TRG counter.
1	The high level of the TRGCLKB pin is set as the condition for clearing the TRG counter.

TRGACL	TRGCLKA (A phase) clearing condition setting**Note 1**
0	The low level of the TRGCLKA pin is set as the condition for clearing the TRG counter.
1	The high level of the TRGCLKA pin is set as the condition for clearing the TRG counter.

TRGIDZ1	TRGIDZ0	TRGIDZ input edge select**Note 2**
0	0	Trigger input is disabled.
0	1	Rising edge
1	0	Falling edge
1	1	Rising and falling edges

Note 1. The setting of this bit is valid while TRGSCE = 1. The setting of this bit is invalid while TRGSCE = 0.
Note 2. The setting of these bits is valid while TRGSCE = 0. The setting of these bits is invalid while TRGSCE = 1.

Caution 1. Be sure to set bits 4 and 3 to 0.
Caution 2. When writing to the TRGCTL0 register, ensure that the TRGSTART bit in the TRGMR0 register is set to 0 (to stop counting) at the time.

13.3.16 Timer RG phase counting control register 1 (TRGCTL1)

Figure 13 - 17 Format of Timer RG Phase Counting Control Register 1 (TRGCTL1)

Address: F03F7H
After reset: 00H
R/W: R/W

Symbol	<7>	6	5	4	<3>	2	<1>	<0>
TRGCTL1	TRGPMCE	0	0	0	TRGLDE	0	TRGECM1	TRGECM0

TRGPMCE	TRGPMC counting enable
0	Counting by TRGPMC is stopped.
1	Counting by TRGPMC is enabled.

TRGLDE	Control over loading of the TRGGRC register value to the counter on an underflow
0	Loading the setting of the TRGGRC register to the counter when the counter underflows is disabled.
1	Loading the setting of the TRGGRC register to the counter when the counter underflows is enabled.

TRGECM1	TRGGRD match TRG counter clearing select
0	The counter is not cleared to 0000H.
1	If counting down is to proceed following detection of a match between the TRG counter and TRGGRD register, the counter is cleared to 0000H.

TRGECM0	TRGGRC match TRG counter clearing select
0	The counter is not cleared to 0000H.
1	If counting up is to proceed following detection of a match between the TRG counter and TRGGRC register, the counter is cleared to 0000H.

Caution 1. Be sure to set bits 6 to 4 and 2 to 0.

Caution 2. When writing to the TRGCTL1 register, ensure that the TRGSTART bit in the TRGMR0 register is set to 0 (to stop counting) at the time.

13.3.17 Timer RG counter (TRG)

The TRG counter is connected to the CPU via the internal 16-bit bus and should be always accessed in 16-bit units.
This counter operates incrementing and can also operate free-running, period counting, or external event counting (incrementing or decrementing). It can be cleared to 0000H by compare match with the corresponding TRGGRA or TRGGRB register, or input capture to registers TRGGRA and TRGGRB (counter clearing function).
In PWM2 mode, the counter can be cleared to 0000H by input on the TRGTRG pin. In phase counting mode, the counter can be cleared to 0000H by TRGIDZ input or compare match between the TRG counter and TRGGRC register or TRGGRD register. Moreover, if the TRGLDE bit of the TRGCTL1 register is set to 1, the setting of the TRGGRC register will be loaded to the counter when the counter underflows.
When the TRG counter overflows (FFFFH → 0000H), the TRGOVF flag in the TRGSR0 register is set to 1.
When the TRG counter underflows (0000H → FFFFH), the TRGUDF flag in the TRGSR0 register is set to 1.

Figure 13 - 18 Format of Timer RG Counter (TRG)

Address: F03E6H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRG																

—	Function
Bits 15 to 0	In phase counting mode, counting up and down proceeds. In other modes, counting up proceeds.

13.3.18 Timer RG general registers A, B, C, and D (TRGGRA, TRGGRB, TRGGRC, TRGGRD)

The TRGGRA, TRGGRB, TRGGRC, and TRGGRD registers function as the output compare or input capture registers. These functions can be switched by setting the TRGIOR register.

When the TRGGRA, TRGGRB, TRGGRC, and TRGGRD registers function as the output compare registers, the values of these registers and the value of the TRG counter are always compared. When their values match (compare match), the TRGIMFA, TRGIMFB, TRGIMFC, and TRGIMFD flags in the TRGSR0 register are set to 1. Compare match output can be set by the TRGIOR register.

When the TRGGRA, TRGGRB, TRGGRC, and TRGGRD registers function as the input capture registers, the value of the TRG counter is stored upon detecting externally input capture signals. At this time, the TRGIMFA and TRGIMFB flags are set to 1. The detection edge of input capture signals is selected by setting the TRGIOR register.

In PWM or PWM2 mode, the settings of bits of the TRGIOR register other than the TRGBUFA and TRGBUFB bits are ignored. When the TRG counter is to operate in phase counting mode, the TRGGRC and TRGGRD registers can be specified as selecting the upper and lower period limits by settings of the TRGECM0 and TRGECM1 bits of the TRGCTL1 register, respectively.

The TRGGRC register can also be used as the buffer register for the TRGGRA register and the TRGGRD register can be used as the buffer register for the TRGGRB register, respectively. These functions can be selected by setting bits TRGBUFA and TRGBUFB in the TRGIOR register.

For example, when the TRGGRA register is set as the output compare register and the TRGGRC register is set as the buffer register for the TRGGRA register, the value of the TRGGRC register is transferred to the TRGGRA register each time a compare match A occurs.

When the TRGGRA register is set as the input capture register and the TRGGRC register is set as the buffer register for the TRGGRA register, the value of the TRG counter is transferred to the TRGGRA register and the value of the TRGGRA register is transferred to the TRGGRC register each time an input capture occurs.

Registers TRGGRA, TRGGRB, TRGGRC, and TRGGRD can be read or written in 16-bit units.

Figure 13 - 19 Format of Timer RG General Registers A, B, C, and D (TRGGRA, TRGGRB, TRGGRC, TRGGRD)

Address: F03E8H (TRGGRA), F03EAH (TRGGRB), FFF60H (TRGGRC), FFF62H (TRGGRD)
After reset: FFFFH
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRGGRj																

—	Function
Bits 15 to 0	The function varies depending on the mode or the function. **Table 13 - 4** lists the TRGGRA, TRGGRB, TRGGRC, and TRGGRD register functions.

Remark j = A, B, C, D

Table 13 - 4 TRGGRA, TRGGRB, TRGGRC, and TRGGRD Register Functions (1/2)

Mode	Register	Setting	Function
Input capture	TRGGRA	TRGIOR (TRGIOA2 = 1) TRGMR0 (TRGPWM = 0)	Input capture register Holds the value of the TRG counter.
	TRGGRB	TRGIOR (TRGIOB2 = 1) TRGMR0 (TRGPWM = 0)	Input capture register Holds the value of the TRG counter.
	TRGGRC	TRGIOR (TRGIOA2 = 1) TRGMR0 (TRGPWM = 0) TRGIOR (TRGBUFA = 0) TRGCTL1 (TRGECM0 = 0)	Output compare register Holds the value for comparison with the TRG counter.
		TRGIOR (TRGIOA2 = 1) TRGMR0 (TRGPWM = 0) TRGIOR (TRGBUFA = 1) TRGCTL1 (TRGECM0 = 0)	Buffer register for TRGGRA (for use in transfer from TRGGRA) Receives the previous input capture value from TRGGRA in response to the input capture signal.
		TRGIOR (TRGIOA2 = 1) TRGMR0 (TRGPWM = 0) TRGCTL1 (TRGECM0 = 1)	Upper phase counting period limit setting register Holds the value for comparison with the TRG counter. If counting up is to proceed following a compare match, the TRG counter is cleared to 0000H.
	TRGGRD	TRGIOR (TRGIOB2 = 1) TRGMR0 (TRGPWM = 0) TRGIOR (TRGBUFB = 0) TRGCTL1 (TRGECM1 = 0)	Output compare register Holds the value for comparison with the TRG counter.
		TRGIOR (TRGIOB2 = 1) TRGMR0 (TRGPWM = 0) TRGIOR (TRGBUFB = 1) TRGCTL1 (TRGECM1 = 0)	Buffer register for TRGGRB (for use in transfer from TRGGRB) Receives the previous input capture value from TRGGRB in response to the input capture signal.
		TRGIOR (TRGIOB2 = 1) TRGMR0 (TRGPWM = 0) TRGCTL1 (TRGECM1 = 1)	Lower phase counting period limit setting register Holds the value for comparison with the TRG counter. If counting down is to proceed following a compare match, the TRG counter is cleared to 0000H.
Output compare	TRGGRA	TRGIOR (TRGIOA2 = 0) TRGMR0 (TRGPWM = 0)	Output compare register Holds the value for comparison with the TRG counter. The specified level is output to the TRGIOA pin on compare match.
	TRGGRB	TRGIOR (TRGIOB2 = 0) TRGMR0 (TRGPWM = 0)	Output compare register Holds the value for comparison with the TRG counter. The specified level is output to the TRGIOB pin on compare match.
	TRGGRC	TRGIOR (TRGIOA2 = 0) TRGMR0 (TRGPWM = 0) TRGIOR (TRGBUFA = 0) TRGCTL1 (TRGECM0 = 0)	Output compare register Holds the value for comparison with the TRG counter.
		TRGIOR (TRGIOA2 = 0) TRGMR0 (TRGPWM = 0) TRGIOR (TRGBUFA = 1) TRGCTL1 (TRGECM0 = 0)	Buffer register for TRGGRA (for use in transfer to TRGGRA) The value expected to be next for comparison is transferred to TRGGRA in response to detection of a compare match between the TRG counter and TRGGRA register.
		TRGIOR (TRGIOA2 = 1) TRGMR0 (TRGPWM = 0) TRGCTL1 (TRGECM0 = 1)	Upper phase counting period limit setting register Holds the value for comparison with the TRG counter. If counting up is to proceed following a compare match, the TRG counter is cleared to 0000H.

Table 13 - 4 TRGGRA, TRGGRB, TRGGRC, and TRGGRD Register Functions (2/2)

Mode	Register	Setting	Function
Output compare	TRGGRD	TRGIOR (TRGIOB2 = 0) TRGMR0 (TRGPWM = 0) TRGIOR (TRGBUFB = 0) TRGCTL1 (TRGECM1 = 0)	Output compare register Holds the value for comparison with the TRG counter.
		TRGIOR (TRGIOB2 = 0) TRGMR0 (TRGPWM = 0) TRGIOR (TRGBUFB = 1) TRGCTL1 (TRGECM1 = 0)	Buffer register for TRGGRB (for use in transfer to TRGGRB) The value expected to be next for comparison is transferred to TRGGRB in response to detection of a compare match between the TRG counter and TRGGRB register.
		TRGIOR (TRGIOB2 = 1) TRGMR0 (TRGPWM = 0) TRGCTL1 (TRGECM1 = 1)	Upper phase counting period limit setting register Holds the value for comparison with the TRG counter. If counting up is to proceed following a compare match, the TRG counter is cleared to 0000H.
PWM	TRGGRA	TRGMR0 (TRGPWM = 1) TRGMR1 (TRGPWM2 = 0)	Output compare register The high level is output on the TRGIOA pin following a compare match.
	TRGGRB		Output compare register The low level is output on the TRGIOA pin following a compare match.
	TRGGRC	TRGMR0 (TRGPWM = 1) TRGMR1 (TRGPWM2 = 0) TRGIOR (TRGBUFA = 1) TRGCTL1 (TRGECM0 = 0)	Buffer register for TRGGRA (for use in transfer to TRGGRA) The value expected to be next for comparison is transferred to TRGGRA in response to detection of a compare match between the TRG counter and TRGGRA register.
		TRGMR0 (TRGPWM = 1) TRGMR1 (TRGPWM2 = 0) TRGCTL1 (TRGECM0 = 1)	Upper phase counting period limit setting register Holds the value for comparison with the TRG counter. If counting up is to proceed following a compare match, the TRG counter is cleared to 0000H.
	TRGGRD	TRGMR0 (TRGPWM = 1) TRGMR1 (TRGPWM2 = 0) TRGIOR (TRGBUFB = 1) TRGCTL1 (TRGECM1 = 0)	Buffer register for TRGGRB (for use in transfer to TRGGRB) The value expected to be next for comparison is transferred to TRGGRB in response to detection of a compare match between the TRG counter and TRGGRB register.
		TRGMR0 (TRGPWM = 1) TRGMR1 (TRGPWM2 = 0) TRGCTL1 (TRGECM1 = 1)	Lower phase counting period limit setting register Holds the value for comparison with the TRG counter. If counting down is to proceed following a compare match, the TRG counter is cleared to 0000H.
PWM2	TRGGRA	TRGMR1 (TRGPWM2 = 1)	PWM period setting register
	TRGGRB	TRGMR1 (TRGPWM2 = 1)	PWM output transition point setting register
	TRGGRC	TRGMR1 (TRGPWM2 = 1)	Register for setting the waiting time following the start of counting or a trigger.
	TRGGRD	TRGMR1 (TRGPWM2 = 1) TRGIOR (TRGBUFB = 1)	Buffer register for TRGGRB (for use in transfer to TRGGRB) The value expected to be next for comparison is transferred to TRGGRB in response to detection of a compare match between the TRG counter and TRGGRA register or an active edge on the TRGTRG pin.

Caution When the setting of the TRGTCK[2:0] bits in the TRGCR register is 000B (f_{CLK}) and the value for comparison is set to 0000H, a request signal to the DTC and the ELC is generated only once immediately after counting starts. When the value for comparison is non-zero, a request signal is generated each time a compare match occurs.

13.3.19 Timer RG phase change time measurement counter (TRGPMC)

The TRGPMC register only operates in phase counting mode (when the TRGMDF bit in the TRGMR0 register is 1 and the TRGPMCE bit in the TRGCTL1 register is 1). The TRGPMC register is re-set to 0001H in response to counting up or down by the TRG counter according to the conditions for A- and B-phase detection specified in the TRGCNTC register. The TRGPMC register is incremented by cycles of the source clock specified in the TRGTCK[2:0] bits of the TRGCR register. Be sure to access the TRGPMC register in 16-bit units.

Figure 13 - 20 Format of Timer RG Phase Change Time Measurement Counter (TRGPMC)

Address: F03F8H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRGPMC																

—	Function
Bits 15 to 0	The A- and B-phase change times are counted in phase counting mode.

13.3.20 Timer RG phase change time capture registers 0, 1 (TRGCAP0, TRGCAP1)

Figure 13 - 21 Format of Timer RG Phase Change Time Capture Registers 0, 1 (TRGCAP0, TRGCAP1)

Address: F03FAH (TRGCAP0), F03FCH (TRGCAP1)
After reset: FFFFH
R/W: R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRGCAPi																

TRGCAP0 **Note**	Function
Bits 15 to 0	Input capture register Holds the value of the TRGPMC register in response to counting up or down by the TRG counter according to the conditions for A- and B-phase detection specified in the TRGCNTC register.

TRGCAP1 **Note**	Function
Bits 15 to 0	Buffer register for TRGCAP0 (for use in transfer from TRGCAP0) Receives the previous input capture value from the TRGCAP0 register in response to the input capture signal.

Note Capturing does not proceed at the time the TRG counter starts counting in response to detecting the A- and B-phase states specified in the TRGCNTC register for the first time after the TRGSTART bit in the TRGMR0 register has been set to 1. Also, a timer RG phase change detection interrupt is not generated at this time.

Caution **This register is only enabled when TRGMR0.TRGMDF = 1 and TRGCTL1.TRGPMCE = 1.**

Remark i = 0, 1

13.3.21 Registers for controlling the port functions multiplexed with the inputs and outputs of timer RG

Set the following registers to control the port functions multiplexed with the inputs and outputs of timer RG.

- Port mode registers xx (PMxx)
- Port registers xx (Pxx)

For details, see **7.3.1 Port mode registers xx (PMxx)** and **7.3.2 Port registers xx (Pxx)**.

When the pins multiplexed with TRGIOA and TRGIOB are to be used for outputs of the timer, set the port mode register xx (PMxx) bit and port register xx (Pxx) bit corresponding to each port to 0.

Example: When using P50/TRGIOA for timer output
Set the PM50 bit of port mode register 5 to 0.
Set the P50 bit of port register 5 to 0.

When the pins multiplexed with TRGIOA and TRGIOB are to be used for inputs of the timer, set the port mode register xx (PMxx) bit corresponding to each port to 1. At this time, the port register xx (Pxx) bit may be 0 or 1.

Example: When using P50/TRGIOA for timer input
Set the PM50 bit of port mode register 5 to 1.
Set the P50 bit of port register 5 to 0 or 1.

Remark xx = 0, 5, 12

13.4 Timer RG2 Operation

13.4.1 Items common to multiple modes and functions

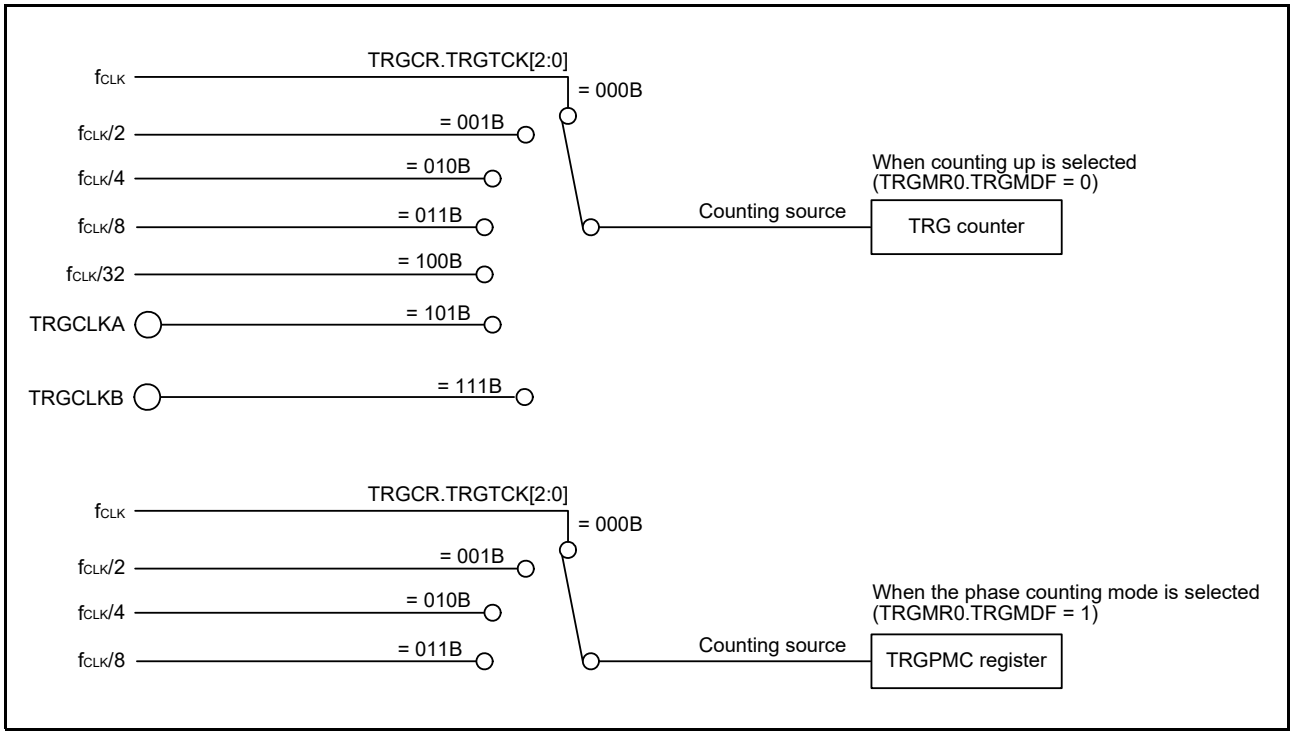
1. Sources for counting

   Table 13 - 5 lists the counting source selection and **Figure 13 - 22** shows the counting source block diagram. When the phase counting mode is selected, the TRGTCK[2:0] bits in the TRGCR register are used to select the clock source for counting by the TRGOMC counter. The setting of the TRGCKEG[1:0] bits in the TRGCR register is invalid.

Table 13 - 5 Counting Source Selection

Counting Source	Selection Method
f_{CLK}, $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, $f_{CLK}/32$	The counting source is selected by the TRGTCK[2:0] bits in the TRGCR register.
External signal input to the TRGCLKA or TRGCLKB pin	The TRGTCK[2:0] bits in the TRGCR register are set to 101B (TRGCLKA input) or 111B (TRGCLKB input). The active edge is selected by the TRGCKEG[1:0] bits in the TRGCR register. The corresponding bit of the port mode register xx is set to 1 (input mode).

Figure 13 - 22 Counting Source Block Diagram



The pulse width of an external clock input to the TRGCLKj pin (j = A or B) should be set to three cycles or more of the timer RG2 operating clock (f_{CLK}).

2. Buffer operation

The TRGBUFA or TRGBUFB bit in the TRGIOR register can be used to select the TRGGRC or TRGGRD register as the buffer register for the TRGGRA or TRGGRB register.

- Buffer register for TRGGRA register: TRGGRC register
- Buffer register for TRGGRB register: TRGGRD register

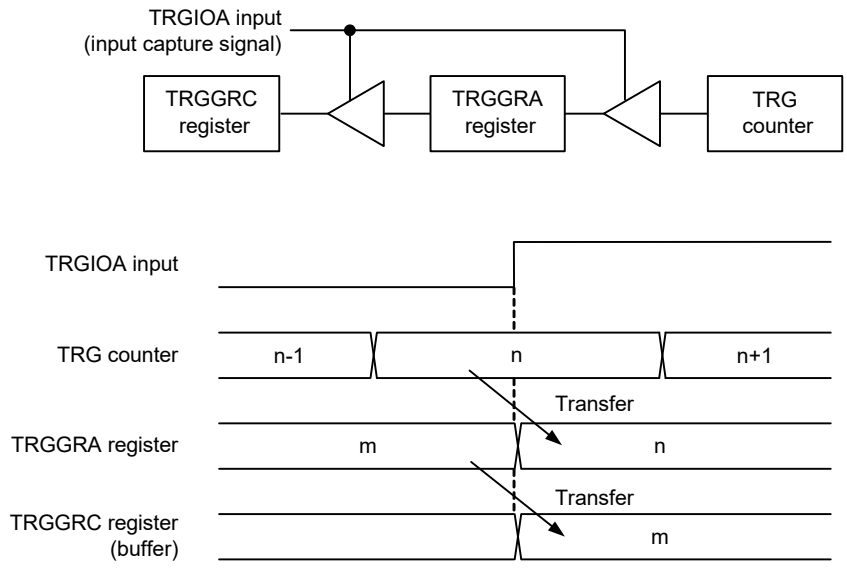
Buffer operation differs depending on the mode. **Table 13 - 6** lists buffer operation in each mode, **Figure 13 - 23** shows buffer operation for the input capture function, **Figure 13 - 24** shows buffer operation for the output compare function, and **Figure 13 - 25** shows buffer operation in PWM2 mode.

Table 13 - 6 Buffer Operation in Each Mode

Function, Mode	Transfer Timing	Transfer Destination Register
Input capture function	Input capture signal input	The content of the TRGGRA (TRGGRB) register is transferred to the buffer register.
Output compare function	Compare match between the TRG counter and the TRGGRA (TRGGRB) register	The content of the buffer register is transferred to the TRGGRA (TRGGRB) register.
PWM mode		
PWM2 mode**Note**	Compare match between the TRG counter and TRGGRA register or detection of an active edge on the TRGTRG pin	The content of the buffer register (TRGGRD) is transferred to the TRGGRB register.

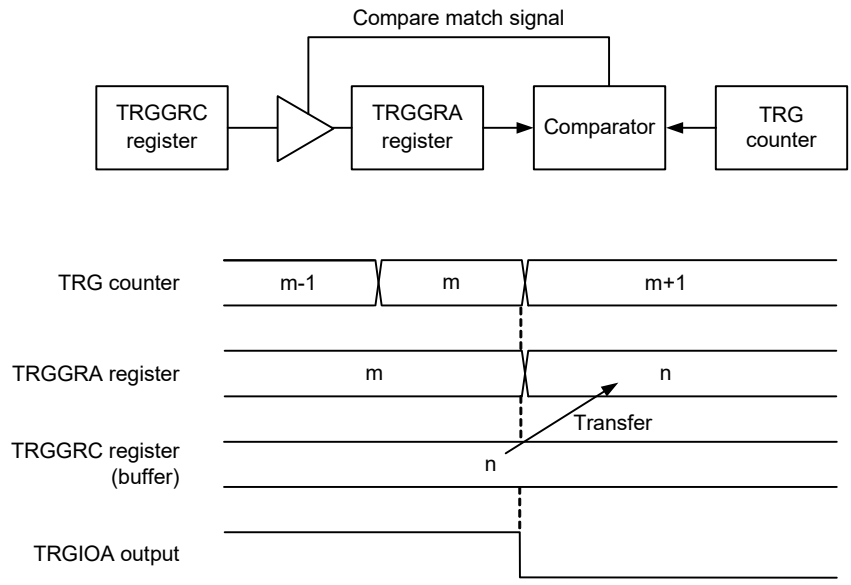
Note When the PWM2 mode is selected, buffer operation of the TRGGRC register is disabled.

Figure 13 - 23 Buffer Operation for the Input Capture Function



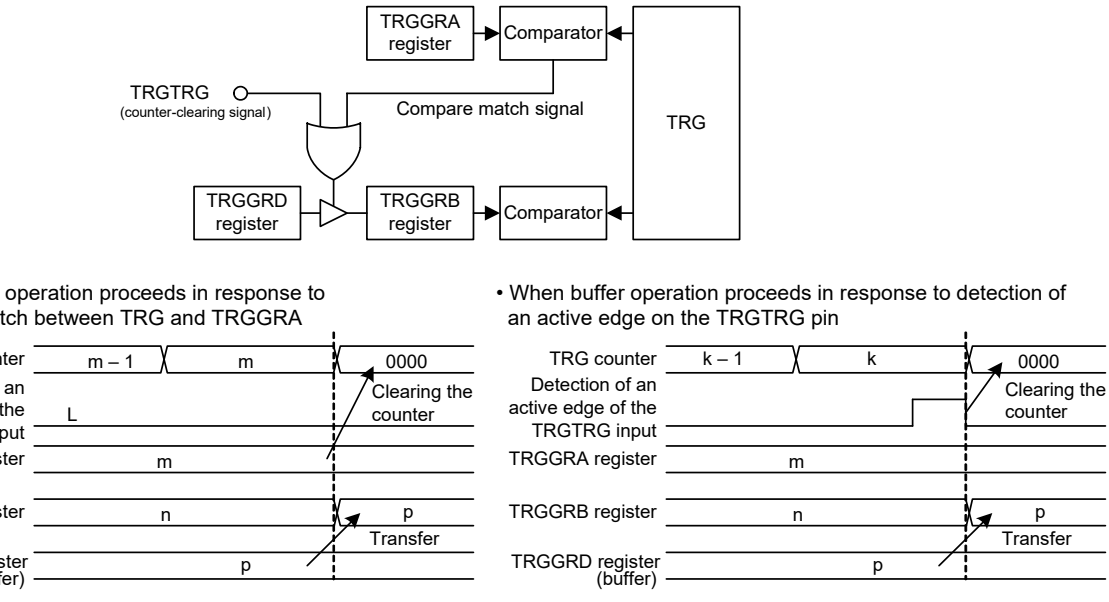
The above diagram applies under the following conditions:
- The TRGBUFA bit in the TRGIOR register is set to 1 (the TRGGRC register is used as a buffer register for the TRGGRA register).
- The TRGIOA[2:0] bits in the TRGIOR register are set to 100B (input capture on the rising edge).

Figure 13 - 24 Buffer Operation for the Output Compare Function



The above diagram applies under the following conditions:
- The TRGBUFA bit in the TRGIOR register is set to 1 (the TRGGRC register is used as a buffer register for the TRGGRA register).
- The TRGIOA[2:0] bits in the TRGIOR register are set to 001B (low-level output by compare match).

Figure 13 - 25 Buffer Operation in PWM2 Mode



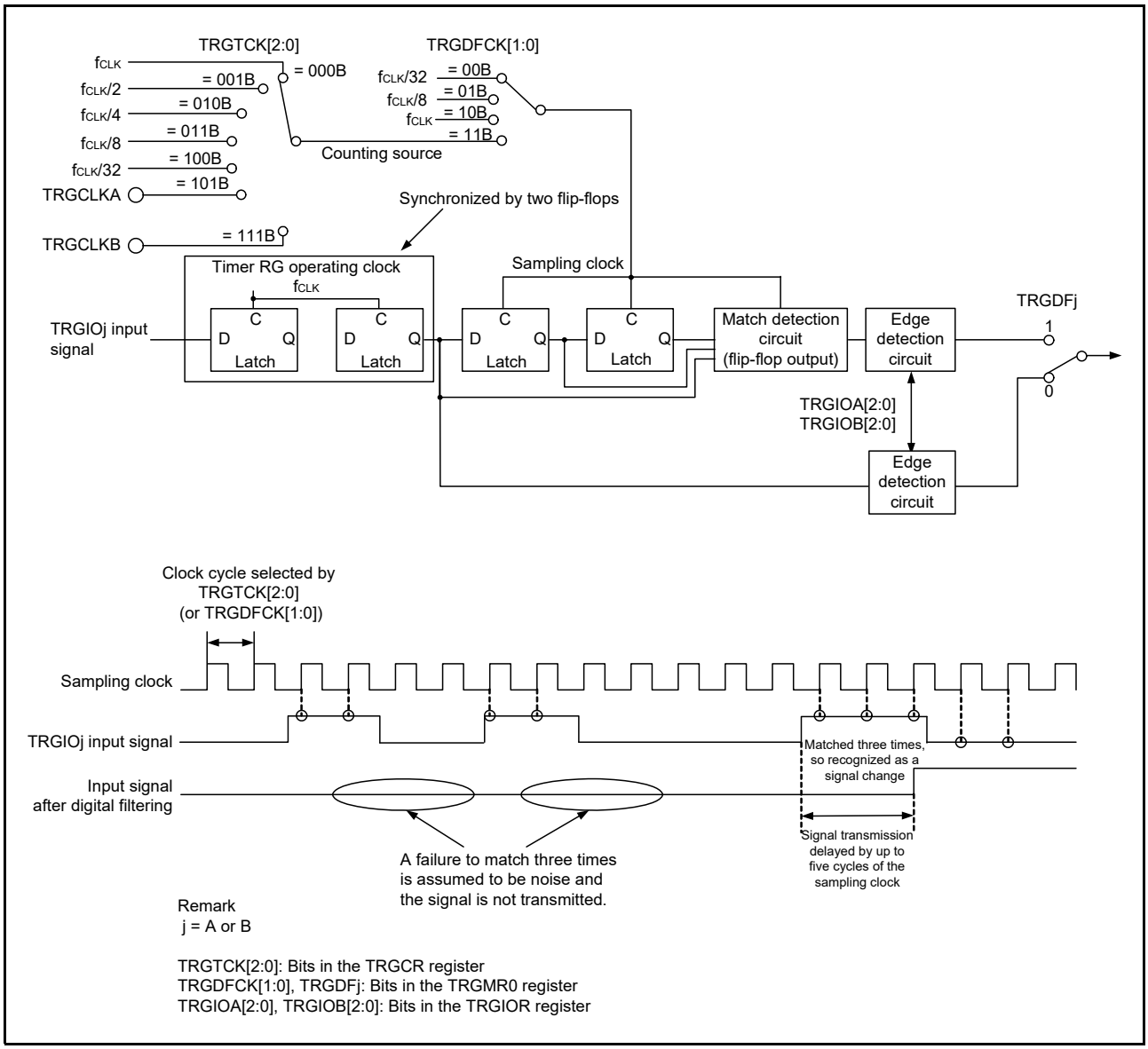
The above diagram applies under the following conditions:
- The TRGBUFA bit in the TRGIOR register is set to 1 (the TRGGRC register is used as a buffer register for the TRGGRA register).
- The TRGTCEG[1:0] bits in the TRGMR1 register are set to 01B (rising edge on TRGTRG).

3. Digital filter

The TRGIOj input (j = A or B) 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 TRGMR0 register.

Figure 13 - 26 shows a block diagram of the digital filter.

Figure 13 - 26 Block Diagram of the Digital Filter



Remark
j = A or B

TRGTCK[2:0]: Bits in the TRGCR register
TRGDFCK[1:0], TRGDFj: Bits in the TRGMR0 register
TRGIOA[2:0], TRGIOB[2:0]: Bits in the TRGIOR register

4. Event input from event link controller (ELC)
   Timer RG2 performs input capture operation B by event input from the ELC. The TRGIMFB flag in the TRGSR0 register is set to 1 at this time.
   To use this function, select the input capture function of the timer mode/phase counting mode, and set the TRGELCICE bit in the TRGMR0 register to 1. This function is disabled in other modes (the output compare function of the timer mode/phase counting mode, PWM mode, and PWM2 mode).
   Setting procedure
   <1> Set timer RG2 as the ELC event link destination.
   <2> Set the TRGELCICE bit in the TRGMR0 register to 1.

5. Event output to event link controller (ELC)
   Table 13 - 7 lists ELC event output according to the TRGIMFA flag. **Table 13 - 8** lists ELC event output according to the TRGIMFB flag. **Table 13 - 9** lists ELC event output according to the TRGIMFC bit. **Table 13 - 10** lists ELC event output according to the TRGIMFD bit.

Table 13 - 7 ELC Event Output According to the TRGIMFA Flag

Function, Mode	ELC Source
Input capture function (TRGPWM = 0, TRGIOA2 = 1)	Detection of TRGIOA edge set by the TRGIOA[1:0] bits
Output compare function (TRGPWM = 0, TRGIOA2 = 0)	Compare match between the TRG counter and TRGGRA register
PWM mode (TRGPWM = 1)	Compare match between the TRG counter and TRGGRA register
PWM2 mode (TRGPWM2 = 1)	Compare match between the TRG counter and TRGGRA register

Remark TRGPWM: Bit in the TRGMR0 register
TRGPWM2: Bit in the TRGMR1 register
TRGIOA2, TRGIOA[1:0]: Bits in the TRGIOR register

Table 13 - 8 ELC Event Output According to the TRGIMFB Flag

Function, Mode	ELC Source
Input capture function (TRGPWM = 0, TRGIOB2 = 1)	Detection of TRGIOB edge set by the TRGIOB[1:0] bits
Output compare function (TRGPWM = 0, TRGIOB2 = 0)	Compare match between the TRG counter and TRGGRB register
PWM mode (TRGPWM = 1)	Compare match between the TRG counter and TRGGRB register
PWM2 mode (TRGPWM2 = 1)	Compare match between the TRG counter and TRGGRB register

Remark TRGPWM: Bit in the TRGMR0 register
TRGPWM2: Bit in the TRGMR1 register
TRGIOB2, TRGIOB[1:0]: Bits in the TRGIOR register

Table 13 - 9 ELC Event Output According to the TRGIMFC Bit

Function, Mode	ELC Source
Output compare function (TRGPWM = 0, TRGIOA2 = 0)	Compare match between the TRG counter and TRGGRC register
PWM mode (TRGPWM = 1)	Compare match between the TRG counter and TRGGRC register
PWM2 mode (TRGPWM2 = 1)	Compare match between the TRG counter and TRGGRC register

Remark TRGPWM: Bit in the TRGMR0 register
TRGPWM2: Bit in the TRGMR1 register
TRGIOA2: Bit in the TRGIOR register

Table 13 - 10 ELC Event Output According to the TRGIMFD Bit

Function, Mode	ELC Source
Output compare function (TRGPWM = 0, TRGIOB2 = 0)	Compare match between the TRG counter and TRGGRD register
PWM mode (TRGPWM = 1)	Compare match between the TRG counter and TRGGRD register
PWM2 mode (TRGPWM2 = 1)	Compare match between the TRG counter and TRGGRD register

Remark TRGPWM: Bit in the TRGMR0 register
TRGPWM2: Bit in the TRGMR1 register
TRGIOB2: Bit in the TRGIOR register

6. Forcibly shutting off pulse output

In timer mode (with the output compare function in use), PWM mode, or PWM2 mode, input on the INTP0 pin can be used to forcibly make the TRGIOA and TRGIOB output pins become programmable I/O port pins, thus shutting off pulse output. When the TRGIOA and TRGIOB pins are to be used as outputs in timer mode (with the output compare function in use), the TRGIOA and TRGIOB pins are made to operate as output port pins by setting the TRGIOA2 and TRGIOB2 bits in the TRGIOR register to 1, respectively. In PWM or PWM2 mode, the TRGIOA pin is made to operate as an output port pin and the TRGIOB pin is fixed to input. Accordingly, forcibly shutting off pulse output is disabled. When a low level is input to the INTP0 pin while the TRGPTO bit in the TRGOER register is set to 1 (enabling INTP0 signal input for forcibly shutting off pulse output), an output pin used as an output port pin outputs the output level corresponding to the setting in the TRGPENA or TRGPENB bit in the TRGMR1 register. To use this function, make the following settings.

- Use the TRGMR1 register to set the pin state (high-impedance state) for use when pulse output is forcibly shut off.
- Execution of forcible shut-off of pulse output sets the TRGSHUTS bit in the TRGOER register to 1. To halt the forcible shut-off of pulse output, set the TRGSHUTS bit to 0 while counting is stopped (TRGSTART = 0).
- Set the corresponding bit in the port direction register to 0 (input mode) for the I/O port pin function multiplexed with the INTP0 pin function.
- Set the TRGPTO bit of the TRGOER register to 1 (enabling INTP0 signal input for forcibly shutting off pulse output).

13.4.2 Timer mode (with the input capture function in use)

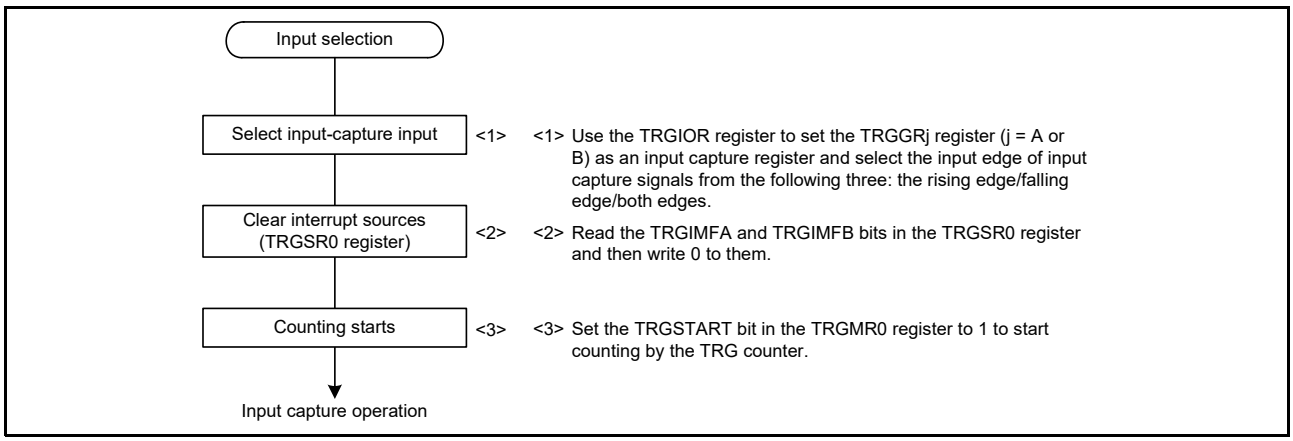
The value of the TRG counter can be transferred to registers TRGGRA and TRGGRB upon detecting the input edge of the input capture/output compare pins (TRGIOA and TRGIOB). The detection edge can be selected from the rising edge/falling edge/both edges.
The input capture function can be used for measuring pulse widths and periods. **Table 13 - 11** lists the input capture function specifications.

Table 13 - 11 Input Capture Function Specifications

Item	Specification
Sources for counting	f_{CLK}, $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, $f_{CLK}/32$ External signal input to the TRGCLKA or TRGCLKB pin (active edge selectable by the program)
Direction of counting	Increment
Period of counting	When the TRGCCLR[1:0] bits in the TRGCR register are set to 00B (free-running operation) $1/f_K \times 65{,}536$ f_K: Frequency of the source for counting
Condition to start counting	1 (to start counting) is written to the TRGSTART bit in the TRGMR0 register.
Condition to stop counting	1 is written to the TRGCSEL bit in the TRGSTR register and 0 (to stop counting) is written to the TRGSTART bit in the TRGMR0 register.
Interrupt request generation timing	• Input capture (active edge of TRGIOA and TRGIOB pin input) • The TRG counter overflows.
TRGIOA, TRGIOB pin function	I/O port or input-capture input (selectable for each pin)
TRGCLKA, TRGCLKB pin function	I/O port or external clock input
Read from timer	The counter value can be read by reading the TRG counter.
Write to timer	The TRG counter can be written to.
Selectable functions	• Input-capture input pin selection Either one or both of pins TRGIOA and TRGIOB • Active edge selection for input-capture input Rising edge, falling edge, or both rising and falling edges • Timing for setting the TRG counter to 0000H Overflow or input capture • Buffer operation (see **13.4.1 Items common to multiple modes and functions**, **2. Buffer operation**) • Digital filter (see **13.4.1 Items common to multiple modes and functions**, **3. Digital filter**) • Input capture operation by the event input signal (input capture) from the ELC

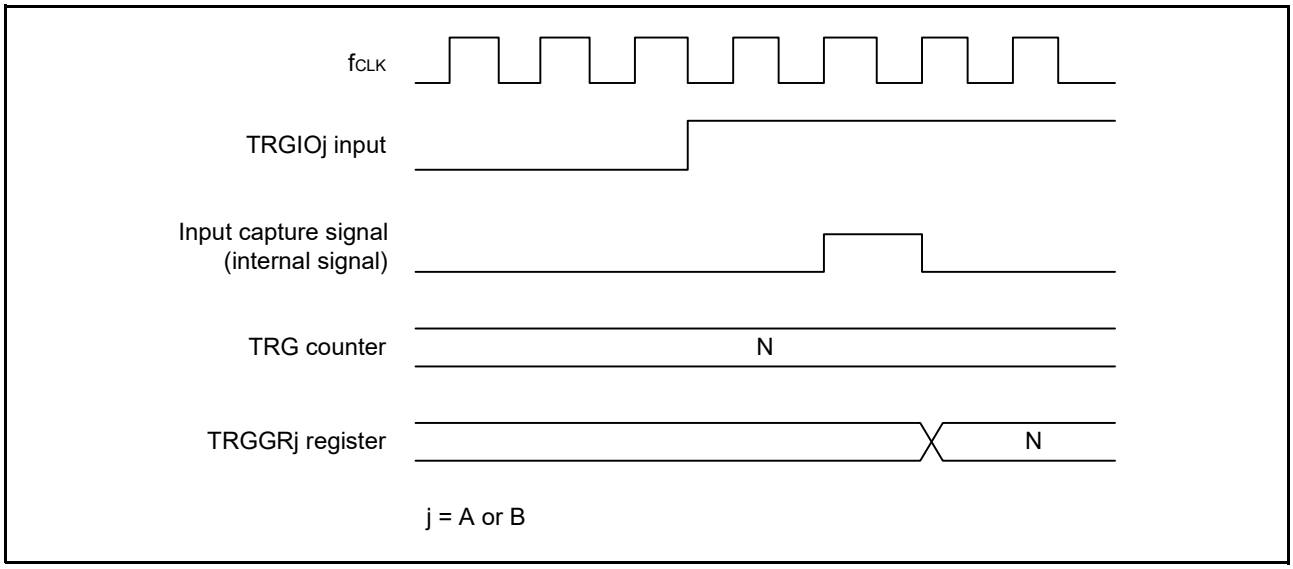
1. Procedure example for setting input capture operation
   Figure 13 - 27 shows a procedure example for setting input capture operation.

Figure 13 - 27 Procedure Example for Setting Input Capture Operation



2. Input-capture input timing
   For input-capture input, the rising edge/falling edge/both edges can be selected by setting the TRGIOR register.
   Figure 13 - 28 shows the input-capture input timing.
   The pulse width of input-capture input signals should be 1.5 f_{CLK} cycles or more for a single edge and 2.5 f_{CLK} cycles or more for both edges.

Figure 13 - 28 Input-capture Input Timing



3. Operation example

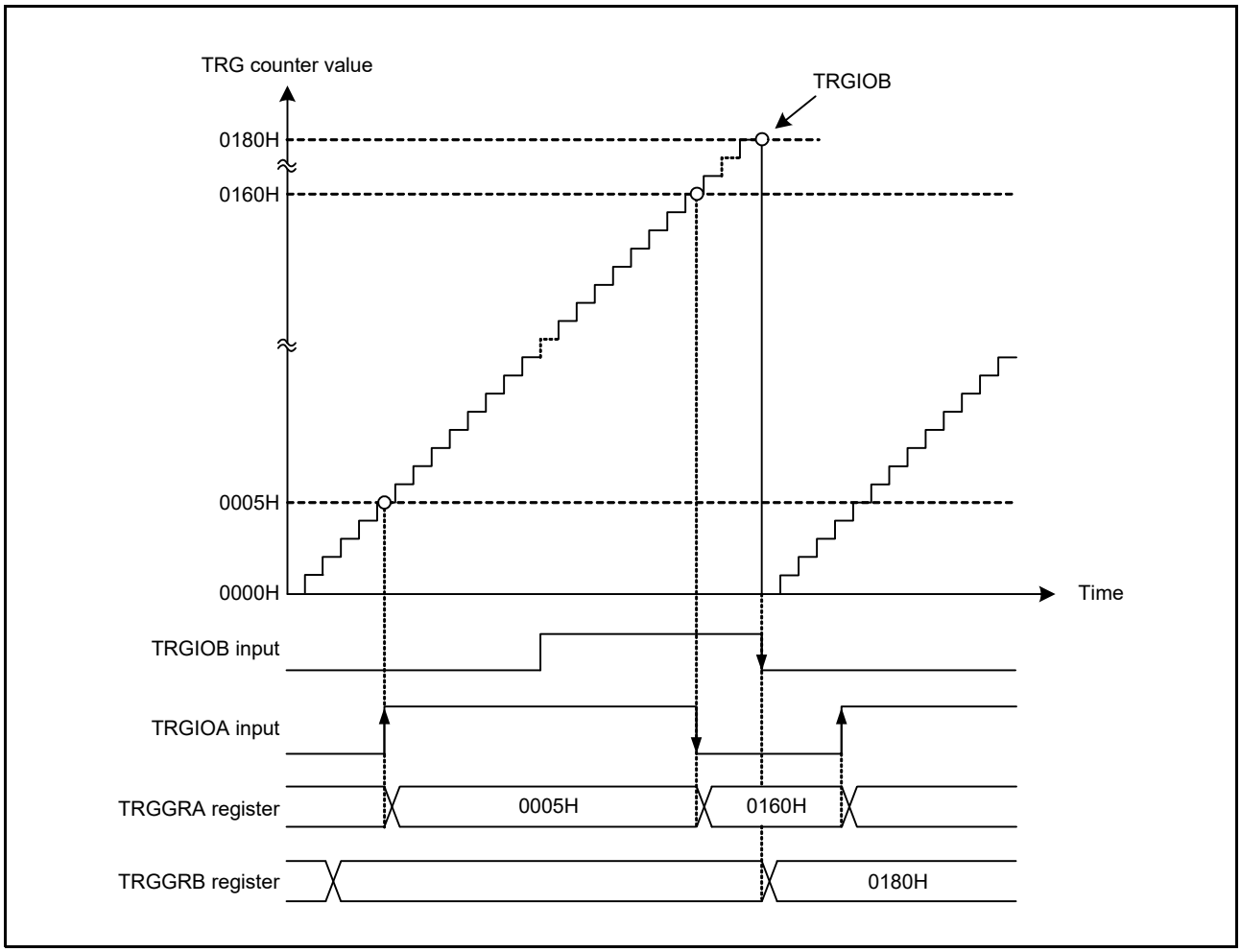
Figure 13 - 29 shows an operation example of input capture.

This example applies when both the rising/falling edges are selected as the input-capture input edge of the TRGIOA pin and the falling edge is selected as the input-capture input edge of the TRGIOB pin, and the TRG counter is cleared by input capture to the TRGGRB register.

<1> Use the TRGIOR register to set registers TRGGRA and TRGGRB as input capture registers and select the input edge of input capture signals from the following three: the rising edge/falling edge/both edges.

<2> Set the TRGSTART bit in the TRGMR0 register to 1 and start counting by the TRG counter.

Figure 13 - 29 Operation Example of Input Capture



By setting the TRGCCLR[1:0] bits in the TRGCR register, the counter can be cleared by input capture A or B.

Figure 13 - 29 shows an operation example with the TRGCCLR[1:0] bits set to 10B. If the input capture operation has been set to clear the counter during operation and is performed when the timer counter value is FFFFH, depending on the timing between the source for counting and input capture operation interrupt flags TRGIMFA, TRGIMFB, and TRGOVF may be set to 1 simultaneously.

13.4.3 Timer mode (with the output compare function in use)

This mode (with the output compare function in use) is for detecting compare matches between the contents of the TRG counter and TRGGRA, TRGGRB, TRGGRC, or TRGGRD register. When the value of the TRG counter matches that of the TRGGRA register, a given level is output on the TRGIOA pin. When the value of the TRG counter matches that of the TRGGRB register, a given level is output on the TRGIOB pin. A compare match between the TRG counter and TRGGRC or TRGGRD register does not affect the output levels of the TRGIOA and TRGIOB pins.

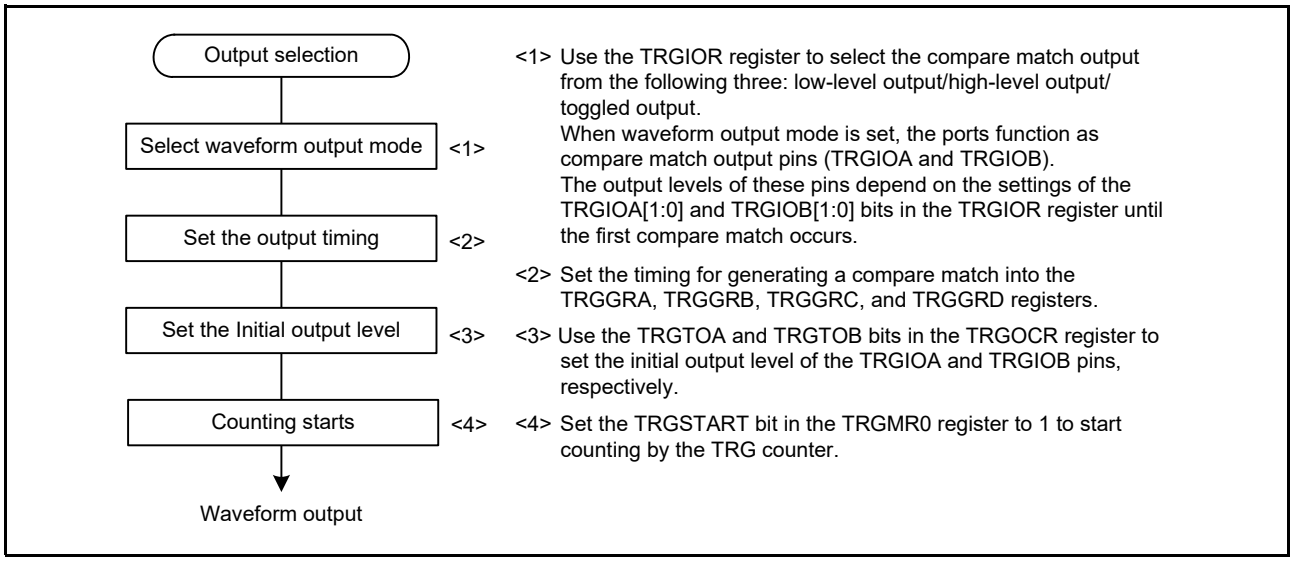
Table 13 - 12 lists the output compare function specifications.

Table 13 - 12 Output Compare Function Specifications

Item	Specification
Sources for counting	f_{CLK}, $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, $f_{CLK}/32$ External signal input to the TRGCLKA or TRGCLKB pin (active edge selectable by the program)
Direction of counting	Increment
Period of counting	• When the TRGCCLR[1:0] bits in the TRGCR register are set to 00B (free-running operation) 1/fK × 65,536 fK: Frequency of the source for counting • When the TRGCCLR[1:0] bits in the TRGCR register are set to 01B (the TRG counter is set to 0000H by compare match with the TRGGRA register) 1/fK × (n + 1) n: Value set in the TRGGRA register • When the TRGCCLR[1:0] bits in the TRGCR register are set to 10B (the TRG counter is set to 0000H by compare match with the TRGGRB register) 1/fK × (n + 1) n: Value set in the TRGGRB register
Waveform output timing	Compare match (contents of the TRG counter and TRGGRj register match)
Condition to start counting	1 (to start counting) is written to the TRGSTART bit in the TRGMR0 register.
Condition to stop counting	• The TRG counter stops counting by writing 0 (to stop counting) to the TRGSTART bit in the TRGMR0 register while the TRGCSEL bit in the TRGSTR register is 1. The output-compare output pin stays at the same level as before the counter was stopped. • The TRG counter stops counting on a compare match with the TRGGRA register while the TRGCSEL bit in the TRGSTR register is 0 and the TRGCCLR[1:0] bits in the TRGCR register are 01B. The output-compare output pin stays at the same level after the level has been changed in response to a compare match. • The TRG counter stops counting on a compare match with the TRGGRB register while the TRGCSEL bit in the TRGSTR register is 0 and the TRGCCLR[1:0] bits in the TRGCR register are 10B. The output-compare output pin stays at the same level after the level has been changed in response to a compare match.
Interrupt request generation timing	• Compare match (contents of the TRG counter and TRGGRA, TRGGRB, TRGGRC, or TRGGRD register match) • The TRG counter overflows.
TRGIOA, TRGIOB pin function	I/O port or output-compare output (selectable for each pin)
TRGCLKA, TRGCLKB pin function	I/O port or external clock input
Read from timer	The counter value can be read by reading the TRG counter.
Write to timer	The TRG counter can be written to.
Selectable functions	• Output-compare output pin selection Either one or both of pins TRGIOA and TRGIOB • Output level selection on compare match Low-level output, high-level output, or toggled output • Initial output level selection Setting the level during the period from the start of counting until a compare match • Timing for setting the TRG counter to 0000H Overflow or compare match with the TRGGRA or TRGGRB register • Buffer operation (see **13.4.1 Items common to multiple modes and functions**, **2. Buffer operation**) • Forcibly shutting off pulse output (see **13.4.1 Items common to multiple modes and functions**, **6. Forcibly shutting off pulse output**)

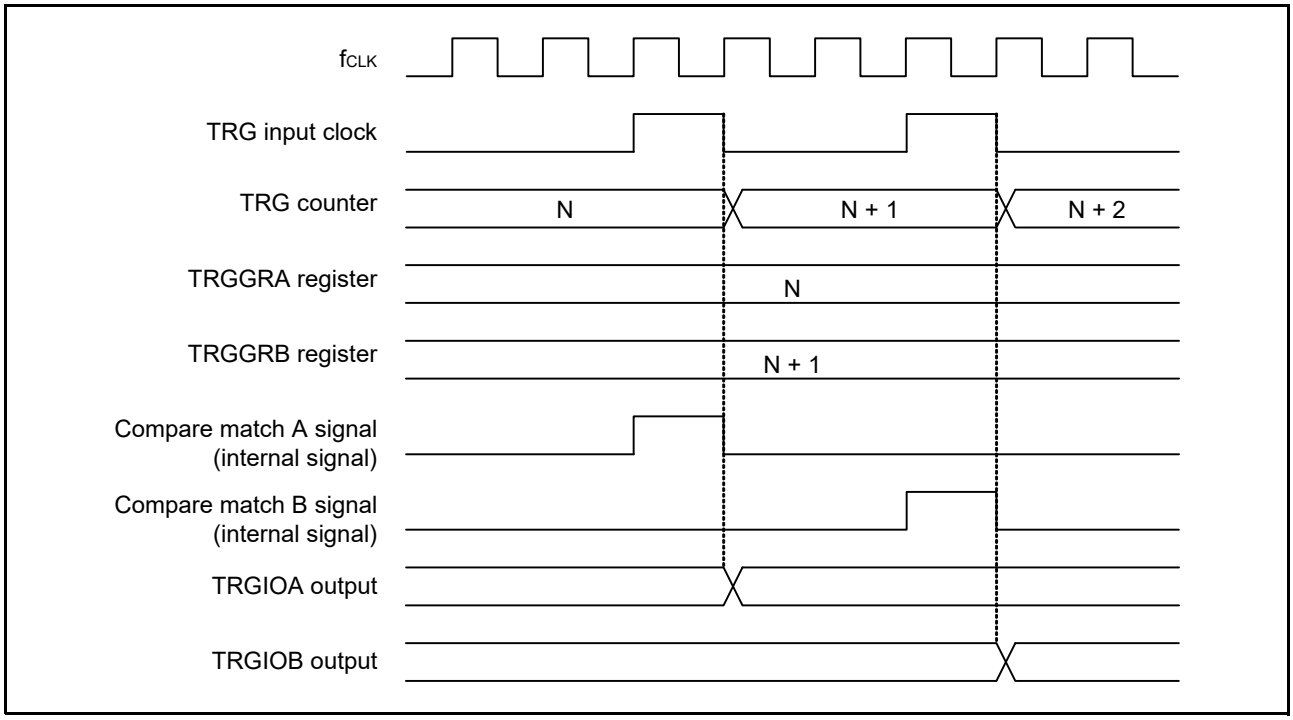
1. Procedure example for setting waveform output by compare match
   Figure 13 - 30 shows a procedure example for setting waveform output by compare match.

Figure 13 - 30 Procedure Example for Setting Waveform Output by Compare Match



2. Output-compare output timing
   A compare match signal is generated at the last state when the TRG counter and the TRGGRA, TRGGRB, TRGGRC, or TRGGRD register match (at the timing for updating the counter value that the TRG counter matches). When the compare match signal is generated in response to a match with the TRGGRA or TRGGRB register, the output value set by the TRGIOR register is output to the output-compare output pin (TRGIOA or TRGIOB). After the TRG counter and the TRGGRA, TRGGRB, TRGGRC, or TRGGRD register match, no compare match signal is generated until the TRG counter input clock is generated.
   Figure 13 - 31 shows the output-compare output timing.

Figure 13 - 31 Output-compare Output Timing



3. Operation example

Figure 13 - 32 shows an operation example of low-level output and high-level output.

This example applies when the TRG counter is set for free-running operation, and low-level output is set on compare match A, and high-level output is set on compare match B. When the set level and the pin level match, the pin level does not change.

After release from the reset state, the initial output level is determined by using the TRGIOR register to set it. If you intend to start operation from an output level other than that determined by the setting of the TRGIOR register, the TRGOCR register can be used to change the initial output level.

Figure 13 - 32 Operation Example of Low-level Output and High-level Output

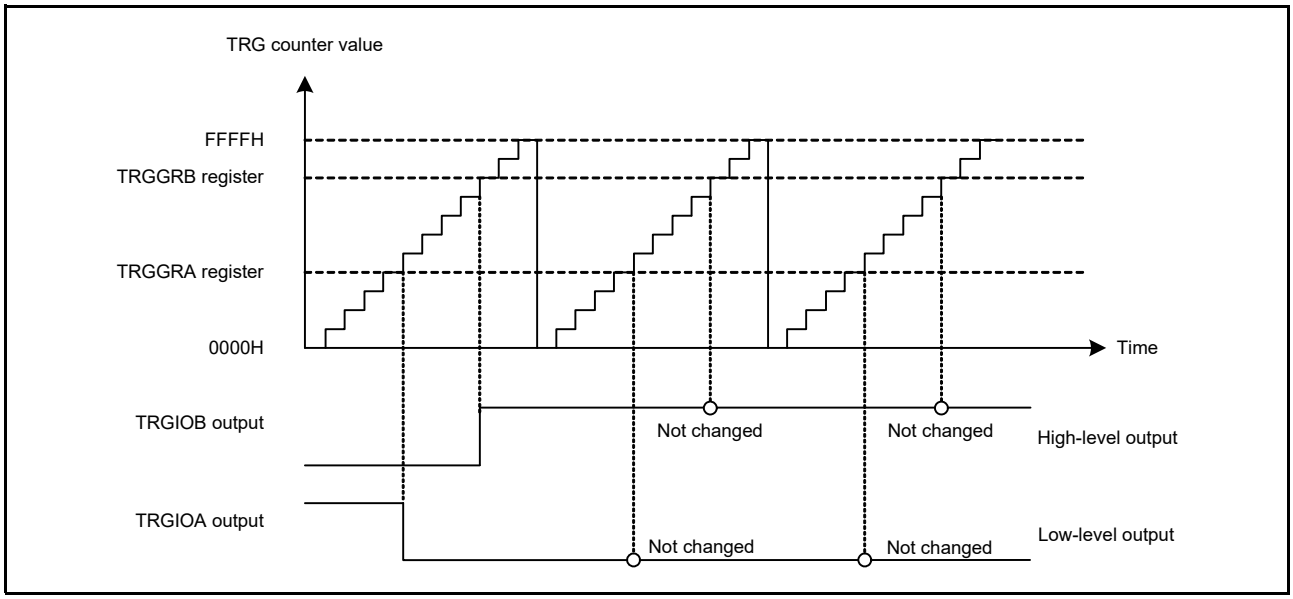


Figure 13 - 33 shows an operation example of toggled output. This example applies when the TRG counter is set for period counting operation (clearing the counter on compare match B), and toggled output is set for both compare matches A and B.

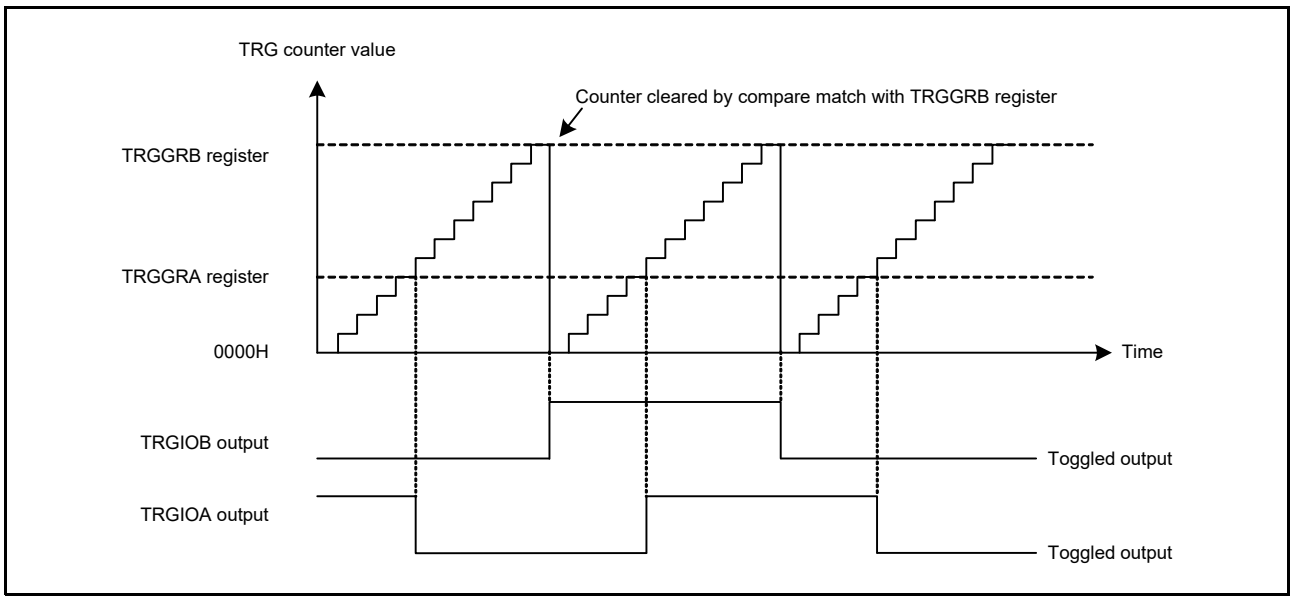
<1> Use the TRGIOR register to select the compare match output from the following three: low-level output/ high-level output/toggled output. When waveform output mode is set, the ports function as compare match output pins (TRGIOA and TRGIOB).

<2> Set the timing for generating a compare match into registers TRGGRA and TRGGRB.

<3> Set the TRGSTART bit in the TRGMR register to 1 to start counting by the TRG counter.

The compare match output pins (TRGIOA and TRGIOB) are not initialized by setting the TRGSTART bit to 0 during operation. To return to their initial values, write to the TRGIOR register to initialize the output. The output is only initialized when the TRGIOA[1:0] and TRGIOB[1:0] bits in the TRGIOR register are set to low-level output or high-level output. By setting the TRGCCLR[1:0] bits in the TRGCR register, the timer RG2 counter value is reset by an input capture/compare match (match with the TRGGRA or TRGGRB register). If the expected value for comparison is FFFFH at this time, FFFFH changes to 0000H, in the same way as the overflow operation, and the TRGOVF flag is set to 1. This operation is the same for modes where the output compare function is used on the timer RG2 counter value and expected value for comparison.

Figure 13 - 33 Operation Example of Toggled Output



13.4.4 PWM mode

In PWM mode, registers TRGGRA and TRGGRB are used as a pair and a PWM waveform is output from the TRGIOA output pin. The output setting by the TRGIOR register is invalid for the pins set to PWM mode. Set the timing of high-level output for the PWM waveform in the TRGGRA register and the timing of low-level output for the PWM waveform in the TRGGRB register.

By setting the compare match with either the TRGGRA or TRGGRB register as the source to derive clearing of the TRG counter, a PWM waveform with a duty cycle from 0% to 100% can be output from the TRGIOA pin.

Table 13 - 13 lists the PWM mode specifications and **Table 13 - 14** lists the combination of PWM output pins and registers. When the setting values in registers TRGGRA and TRGGRB are the same, the output value does not change even if a compare match occurs.

Table 13 - 13 PWM Mode Specifications

Item	Specification
Sources for counting	f_{CLK}, $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, $f_{CLK}/32$ External signal input to the TRGCLKA or TRGCLKB pin (active edge selectable by the program)
Direction of counting	Increment
Period of counting	• The high-level output timing of a PWM waveform is set into the TRGGRA register. • The low-level output timing of a PWM waveform is set into the TRGGRB register.
Condition to start counting	1 (to start counting) is written to the TRGSTART bit in the TRGMR0 register.
Condition to stop counting	• The TRG counter stops counting by writing 0 (to stop counting) to the TRGSTART bit in the TRGMR0 register while the TRGCSEL bit in the TRGSTR register is 1. The TRGIOA output pin retains its initial level (the inverse of the TRGCCLR1 bit in the TRGCR register). • The TRG counter stops counting on a compare match with the TRGGRA register while the TRGCSEL bit in the TRGSTR register is 0 and the TRGCCLR[1:0] bits in the TRGCR register are 01B. The TRGIOA output pin retains its initial level (the inverse of the TRGCCLR1 bit in the TRGCR register). • The TRG counter stops counting on a compare match with the TRGGRB register while the TRGCSEL bit in the TRGSTR register is 0 and the TRGCCLR[1:0] bits in the TRGCR register are 10B. The TRGIOA output pin retains its initial level (the inverse of the TRGCCLR1 bit in the TRGCR register).
Interrupt request generation timing	• Compare match (contents of the TRG counter and TRGGRA, TRGGRB, TRGGRC, or TRGGRD register match) • The TRG counter overflows.
TRGIOA pin function	PWM output
TRGIOB pin function	I/O port
TRGCLKA, TRGCLKB pin function	I/O port or external clock input
Read from timer	The counter value can be read by reading the TRG counter.
Write to timer	The TRG counter can be written to.
Selectable functions	• Timing for setting the TRG counter to 0000H Overflow or compare match with the TRGGRA or TRGGRB register • Buffer operation (see **13.4.1 Items common to multiple modes and functions**, **2. Buffer operation**) • Forcibly shutting off pulse output

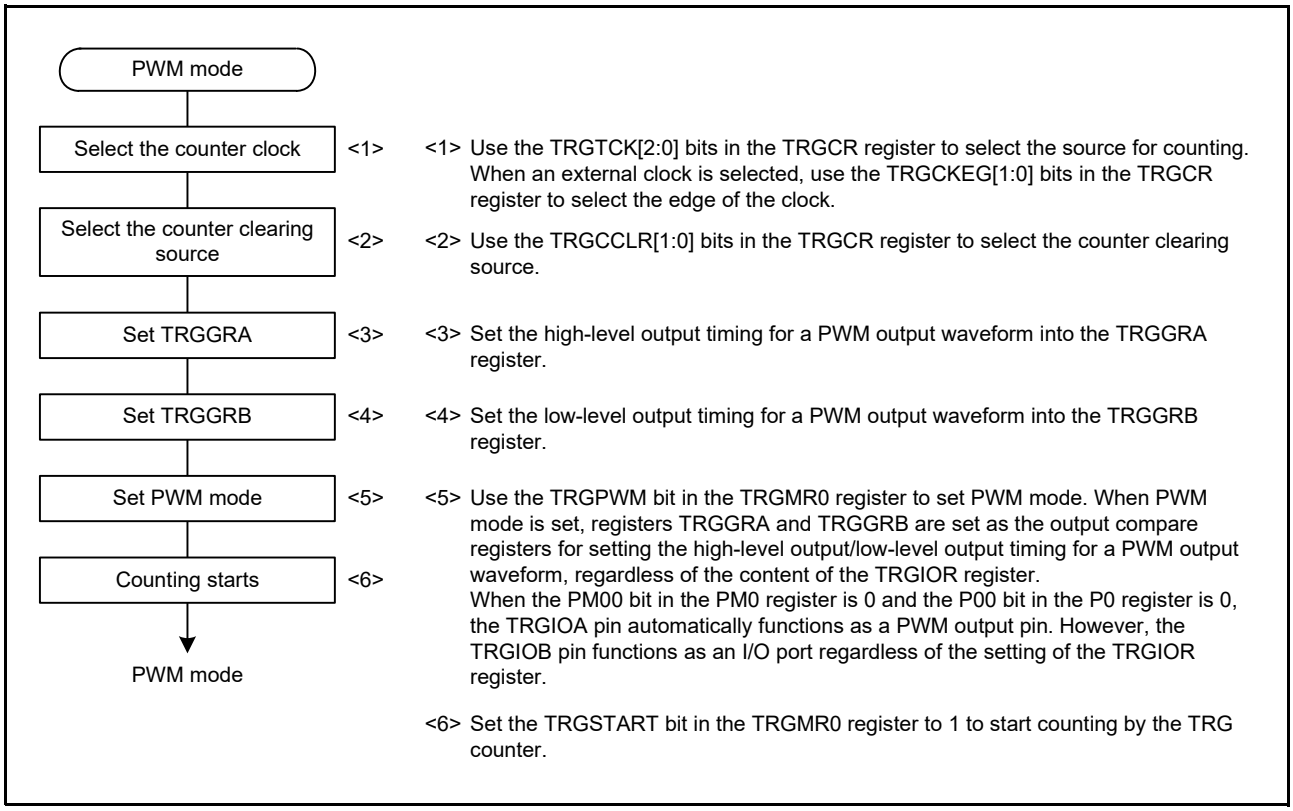
Table 13 - 14 Combination of PWM Output Pins and Registers

Output Pin	High-level Output	Low-level Output
TRGIOA	TRGGRA	TRGGRB
TRGIOB	I/O port function	

1. Procedure example for setting the PWM mode

   Figure 13 - 34 shows a procedure example for setting the PWM mode.

Figure 13 - 34 Procedure Example for Setting the PWM Mode



2. Operation example

Figure 13 - 35 shows an operation example in PWM mode.

When the PWM mode is selected while the PM00 bit in the PM0 register is 0 and the P00 bit in the P0 register is 0, the TRGIOA pin automatically functions as an output pin, and high-level output is set on the compare match with the TRGGRA register and low-level output is set on the compare match with the TRGGRB register. However, regardless of the setting of the TRGIOR register, the TRGIOB pin functions as an I/O port.

This example applies when the compare match with the TRGGRA or TRGGRB register is set as the counter clearing source for the TRG counter. The initial state of the TRGIOA pin depends only on the counter clearing source. This correspondence is shown in **Table 13 - 15**.

This initialization is performed when the TRGSTART bit in the TRGMR0 register is 0 (to stop counting).

Table 13 - 15 Correspondence between the Initial State of the TRGIOA Pin and Counter Clearing Sources

Counter Clearing Source	Initial State of the TRGIOA Pin
Compare match with the TRGGRA register	High level
Compare match with the TRGGRB register	Low level

When the TRGCCLR[1:0] bits in the TRGCR register are set to 00B (clearing disabled), the initial level of the TRGIOA pin becomes high.

Figure 13 - 35 Operation Example (1) in PWM Mode

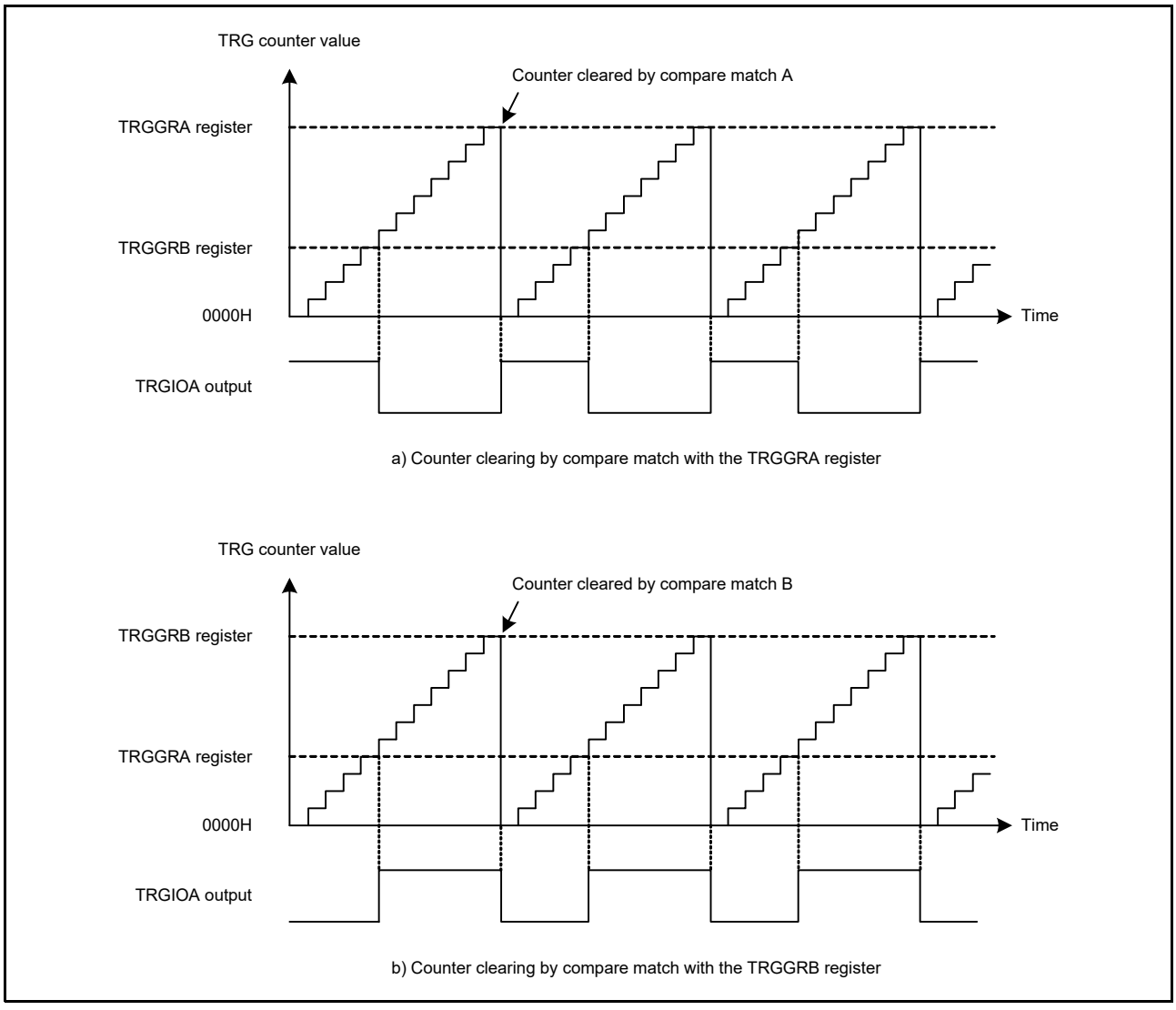


Figure 13 - 36 shows an example for outputting a PWM waveform with duty cycle 0% and duty cycle 100% in PWM mode.

A PWM waveform is set to duty cycle 0% when the compare match with the TRGGRB register is set as the counter clearing source with the following settings:

Value set in TRGGRA register > Value set in TRGGRB register

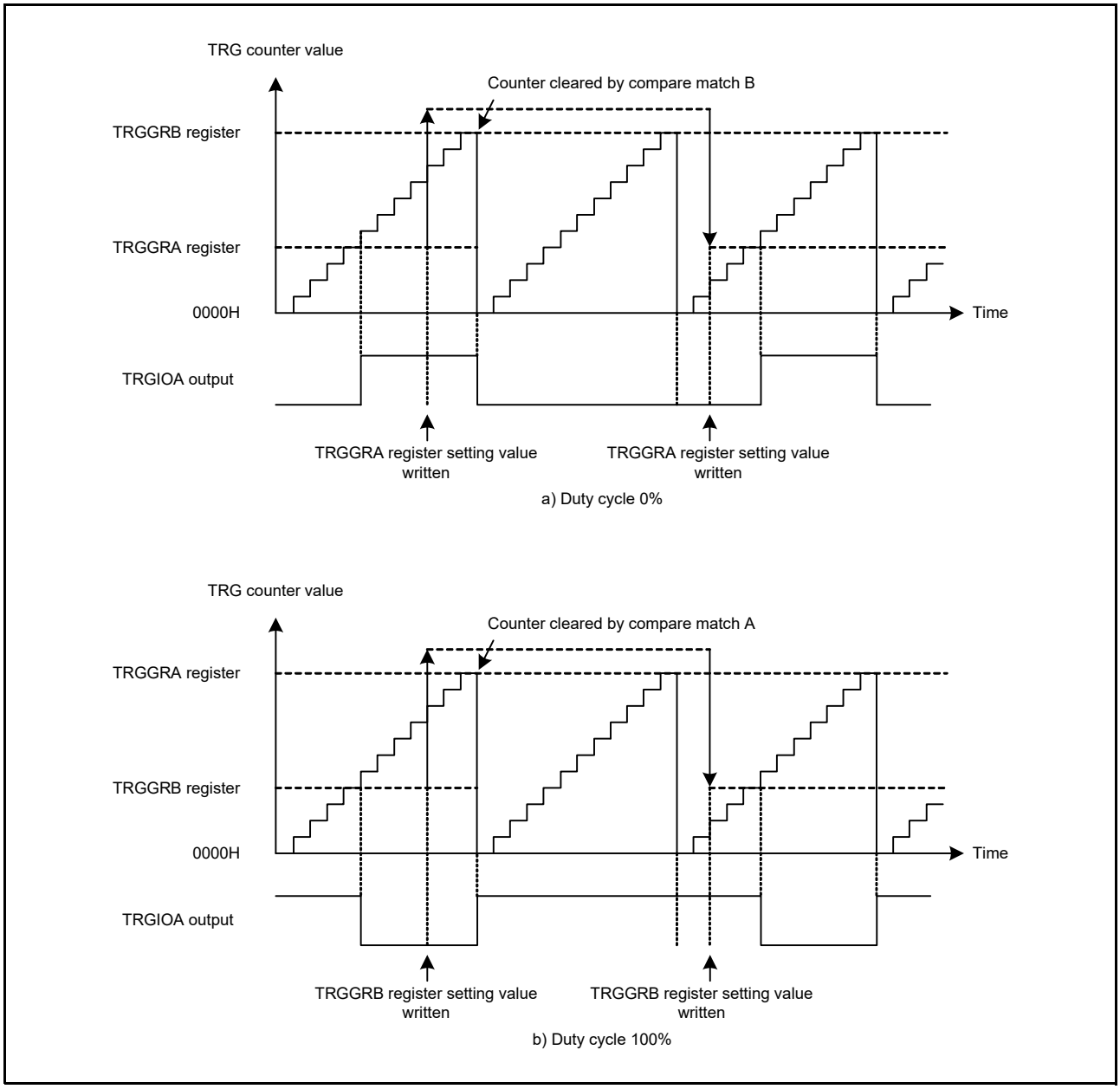
A PWM waveform is set to duty cycle 100% when the compare match with the TRGGRA register is set as the counter clearing source with the following settings:

Value set in TRGGRB register > Value set in TRGGRA register

The output value is unchanged even if a compare match is generated with the following settings:

Value set in TRGGRA register = Value set in TRGGRB register

Figure 13 - 36 Operation Example (2) in PWM Mode



13.4.5 PWM2 mode

In PWM2 mode, a PWM waveform is output through the TRGIOA pin with the timing determined by compare matches between the TRG counter and TRGGRB and TRGGRC registers. The pin output goes to the active level after a certain waiting time has elapsed following a trigger and returns to the inactive level after the certain time has elapsed.
The TRGTOA bit in the TRGOCR register is used to set the output level of the TRGIOA pin. When the TRGTOA bit is set to 0, the TRGIOA pin outputs a low-level signal on compare match between the TRG counter and TRGGRB register and a high-level signal on compare match between the TRG counter and TRGGRC register. When the TRGTOA bit is set to 1, the TRGIOA pin outputs a high-level signal on compare match between the TRG counter and TRGGRB register and a low-level signal on compare match between the TRG counter and TRGGRC register. The TRGGRD register can be used as the buffer register for the TRGGRB register by setting the TRGBUFB bit in the TRGIOR register to 1.
The counter is cleared to 0000H in response to detection of a compare match between the TRG counter and TRGGRA register or an active edge of the TRGTRG input.
The TRGTCEG[1:0] bits in the TRGMR1 register are used to set the active edge of the TRGTRG input.
Active edges of TRGTRG input will not be detected under the following conditions:
the output of the TRGIOA pin is at the high level and the TRGTOA bit is set to 0, or
the output of the TRGIOA pin is at the low level and the TRGTOA bit is set to 1.
One-shot pulse output can be used by setting the TRGCSEL bit in the TRGSTR register to 0. When the TRGTCEG[1:0] bits in the TRGMR1 register are set to 00B (to disable TRGTRG input), the TRG counter starts counting by setting the TRGSTART bit to 1 and stops counting by a match between the TRG counter ant TRGGRA register.
When the TRGTCEG[1:0] bits in the TRGMR1 register are not set to 00B (to enable TRGTRG input), the TRG counter does not start counting by setting the TRGSTART bit to 1. The counter starts counting by detection of an active edge of the TRGTRG input and stops counting by a match between the TRG counter ant TRGGRA register. A one-shot pulse is output by using the settings of the TRGGRB and TRGGRC registers within the period specified in the TRGGRA register.

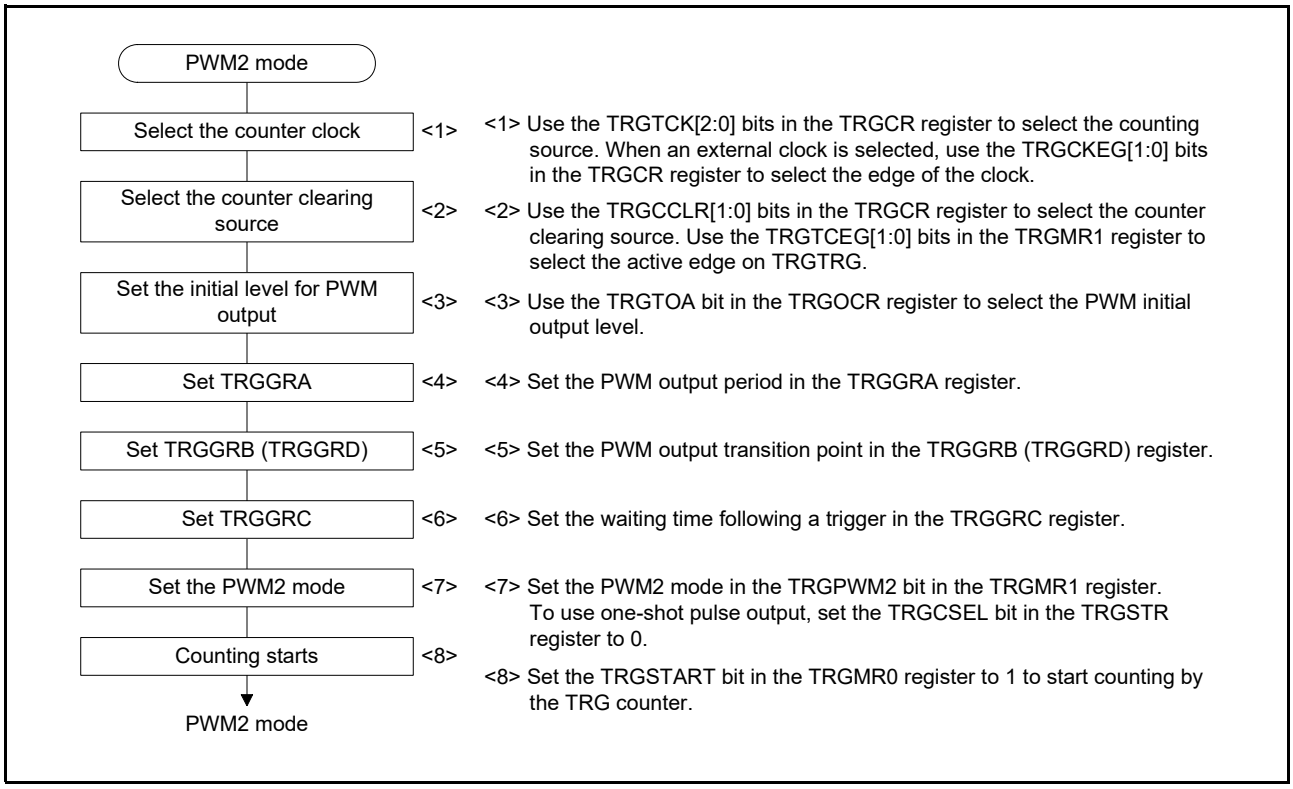
Table 13 - 16 lists the PWM2 mode specifications.

Table 13 - 16 PWM2 Mode Specifications

Item	Specification
Sources for counting	f_{CLK}, $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, $f_{CLK}/32$ External signal input to the TRGCLKA or TRGCLKB pin (active edge selectable by the program)
Direction of counting	Increment
PWM waveform	PWM period: $1/f_K \times (m + 1)$ (if there is no input on TRGTRG) Active level width: $1/f_K \times (n - p)$ Waiting time from start of counting or a trigger: $1/f_K \times (p + 1)$ f_K: Frequency of the source for counting m: Setting of the TRGGRA register n: Setting of the TRGGRB register p: Setting of the TRGGRC register
Condition to start counting	• 1 (to start counting) is written to the TRGSTART bit in the TRGMR0 register while the TRGTCEG[1:0] bits in the TRGMR1 register are set to 00B (to disable input of the TRGTRG trigger signal) or the TRGCSEL bit in the TRGSTR register is set to 1 (to continue counting). • An active edge on the TRGTRG pin is detected while the TRGTCEG[1:0] bits in the TRGMR1 register are set to 01B, 10B, or 11B (to enable input of the TRGTRG trigger signal) and the TRGTSTART bit in the TRGSTR register is set to 1 (to start counting).
Condition to stop counting	• The TRG counter stops counting by writing 0 (to stop counting) to the TRGSTART bit in the TRGMR0 register while the TRGCSEL bit in the TRGSTR register is 1. The TRGIOA output pin retains its initial level (the setting of the TRGTOA bit in the TRGOCR register). • The TRG counter stops counting by a compare match with the TRGGRA register while the TRGCSEL bit in the TRGSTR register is 0 and the TRGCCLR[1:0] bits in the TRGCR register are 01B. The TRGIOA output pin retains its initial level (the setting of the TRGTOA bit in the TRGOCR register).
Interrupt request generation timing	• Compare match (contents of the TRG counter and TRGGRA, TRGGRB, or TRGGRC register match) • The TRG counter overflows.
TRGIOA pin function	PWM output
TRGIOB pin function	Programmable I/O port
TRGCLKA, TRGCLKB pin function	Programmable I/O port or external clock input
Read from timer	The counter value can be read by reading the TRG counter.
Write to timer	The TRG counter can be written to.
Selectable functions	• Timing for setting the TRG counter to 0000H Overflow, compare match with the TRGGRA register, or detection of an active edge of the TRGTRG input under the following conditions: the output of the TRGIOA pin is at the low level while the TRGTOA bit is set to 0, or the output of the TRGIOA pin is at the high level while the TRGTOA bit is set to 1. • Selection of the active edge of the TRGTRG external trigger signal Disabling detection of edges, rising edge, falling edge, or both edges • One-shot pulse output • Selection of the initial level of the TRGIOA output pin • Buffer operation (see **13.4.1 Items common to multiple modes and functions**, **2. Buffer operation**) • Digital Filter (see **13.4.1 Items common to multiple modes and functions**, **3. Digital filter**) • Signal input for forcibly shutting off pulse output (see **13.4.1 Items common to multiple modes and functions**, **6. Forcibly shutting off pulse output**)

1. Procedure example for setting the PWM2 mode
   Figure 13 - 37 shows a procedure example for setting the PWM2 mode.

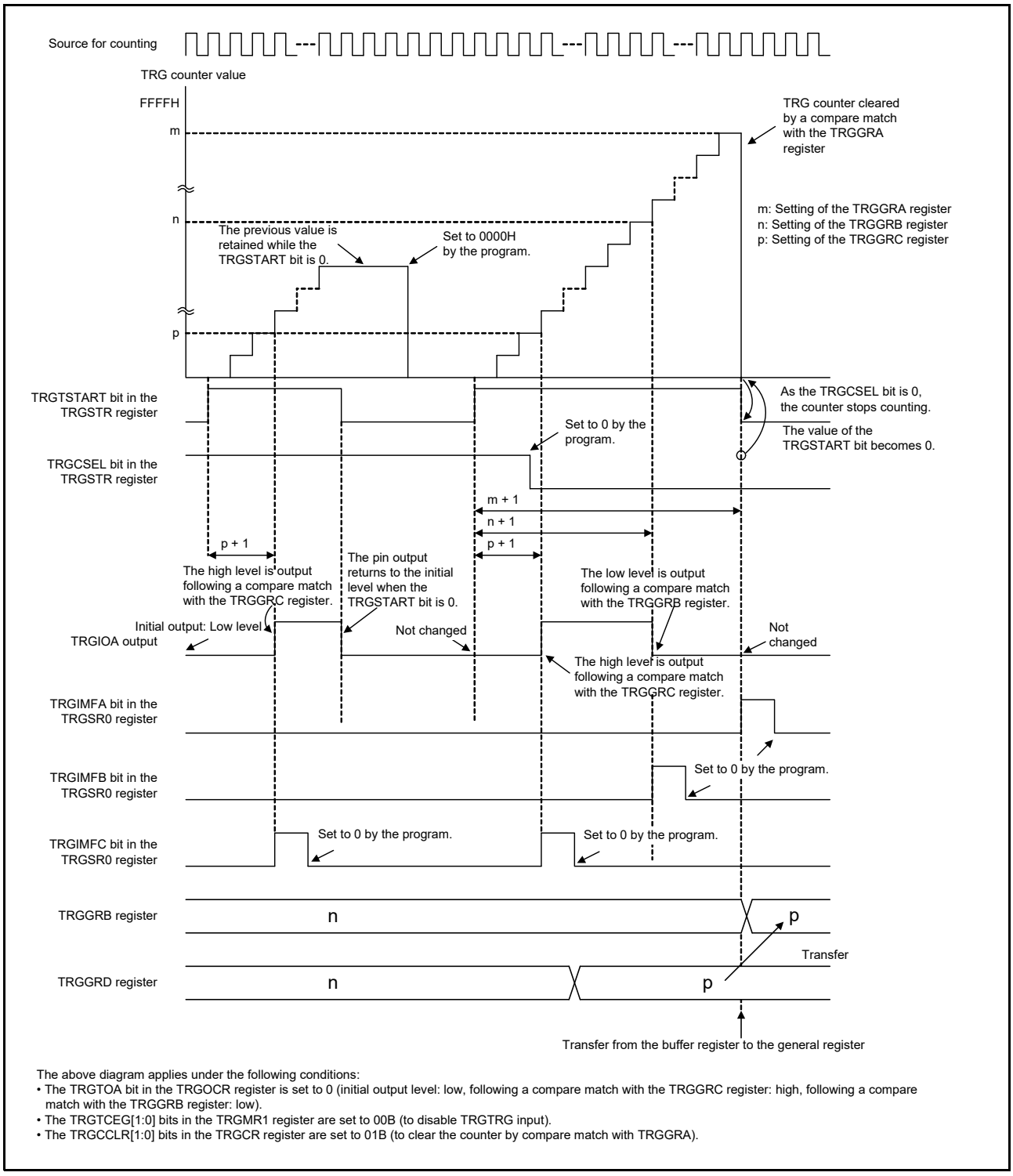
Figure 13 - 37 Procedure Example for Setting the PWM2 Mode



2. Operation example

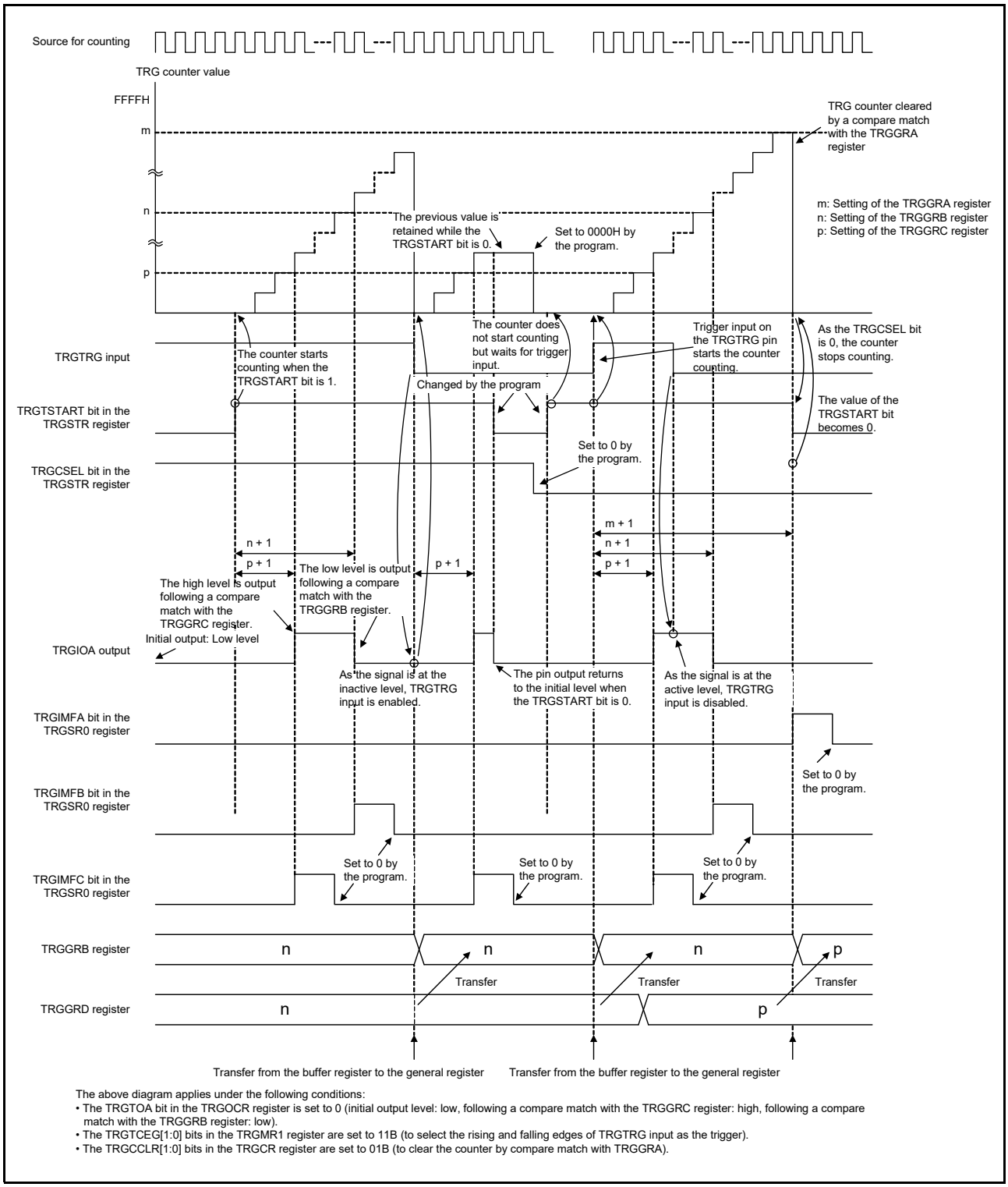
Figure 13 - 38 shows an example of operation in PWM2 mode when input of the TRGTRG trigger signal is disabled and **Figure 13 - 39** shows an example of operation in PWM2 mode when input of the TRGTRG trigger signal is enabled.

Figure 13 - 38 Operation Example (1) in PWM2 Mode



The above diagram applies under the following conditions:
- The TRGTOA bit in the TRGOCR register is set to 0 (initial output level: low, following a compare match with the TRGGRC register: high, following a compare match with the TRGGRB register: low).
- The TRGTCEG[1:0] bits in the TRGMR1 register are set to 00B (to disable TRGTRG input).
- The TRGCCLR[1:0] bits in the TRGCR register are set to 01B (to clear the counter by compare match with TRGGRA).

Figure 13 - 39 Operation Example (2) in PWM2 Mode

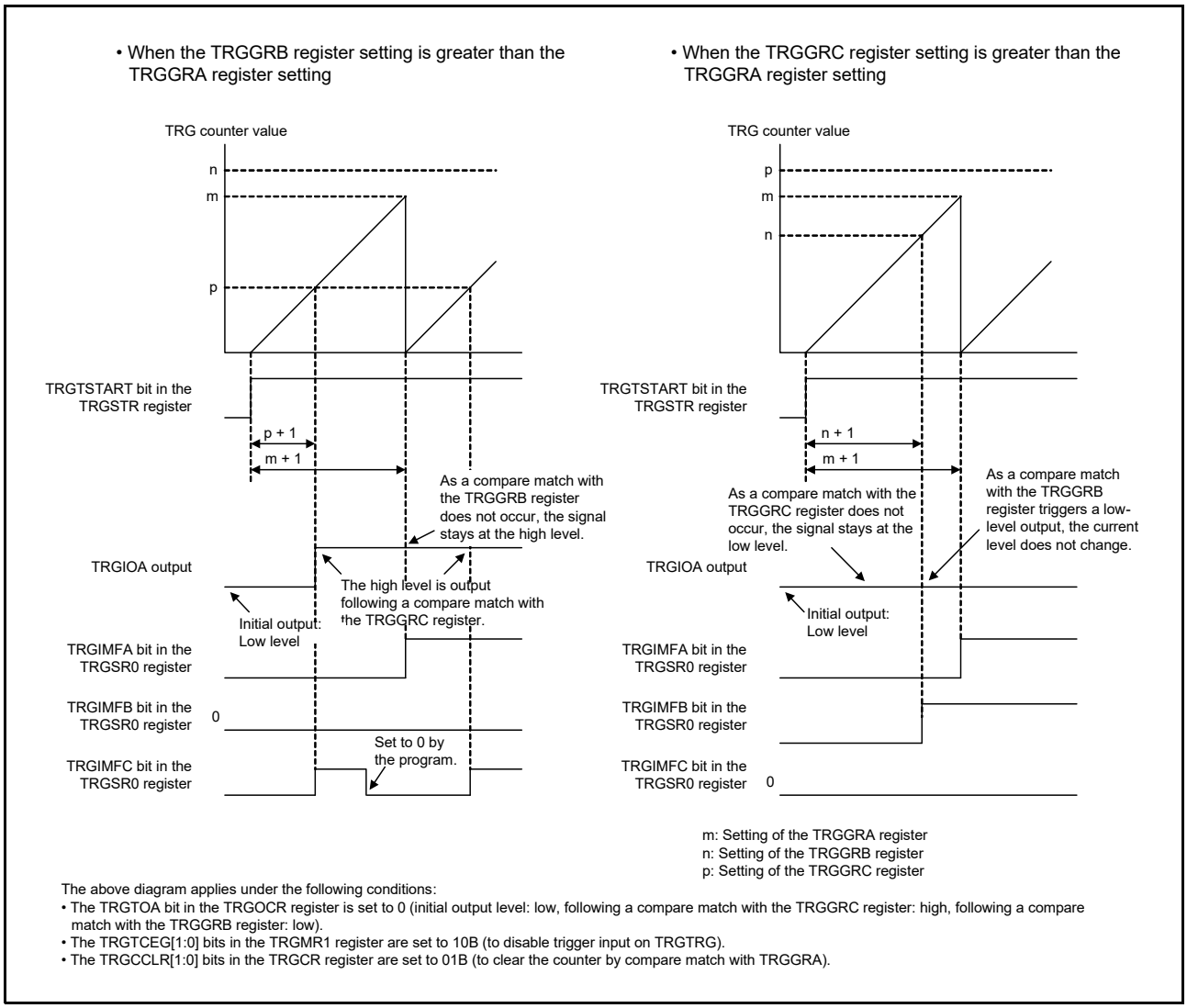


The above diagram applies under the following conditions:

- The TRGTOA bit in the TRGOCR register is set to 0 (initial output level: low, following a compare match with the TRGGRC register: high, following a compare match with the TRGGRB register: low).
- The TRGTCEG[1:0] bits in the TRGMR1 register are set to 11B (to select the rising and falling edges of TRGTRG input as the trigger).
- The TRGCCLR[1:0] bits in the TRGCR register are set to 01B (to clear the counter by compare match with TRGGRA).

Figure 13 - 40 shows operation examples in PWM2 mode with duty cycle 0% and duty cycle 100%.

Figure 13 - 40 Operation Example (3) in PWM2 Mode

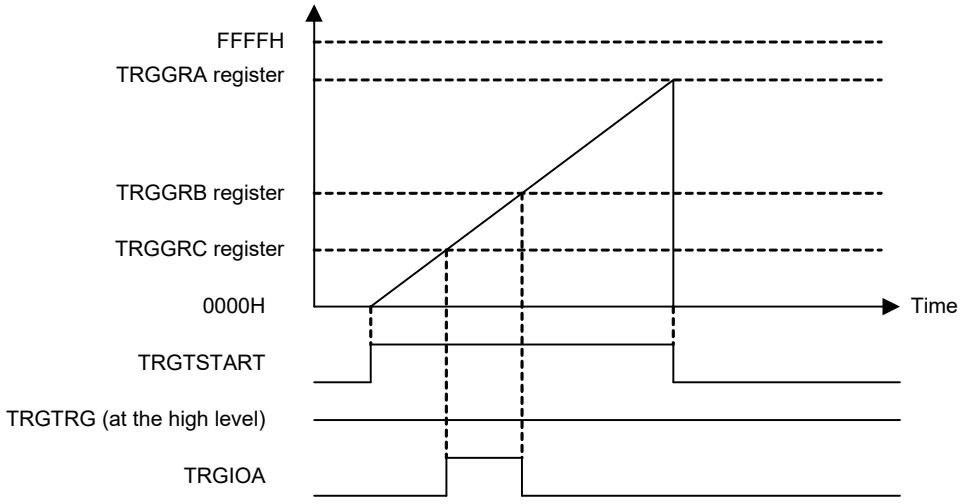


The above diagram applies under the following conditions:

- The TRGTOA bit in the TRGOCR register is set to 0 (initial output level: low, following a compare match with the TRGGRC register: high, following a compare match with the TRGGRB register: low).
- The TRGTCEG[1:0] bits in the TRGMR1 register are set to 10B (to disable trigger input on TRGTRG).
- The TRGCCLR[1:0] bits in the TRGCR register are set to 01B (to clear the counter by compare match with TRGGRA).

Figure 13 - 41 shows an operation example in PWM2 mode when a one-shot pulse waveform is used and **Figure 13 - 42** shows an operation example in PWM2 mode when a one-shot pulse waveform is used and counting starts in response to TRGTRG input.

Figure 13 - 41 Operation Example (4) in PWM2 Mode

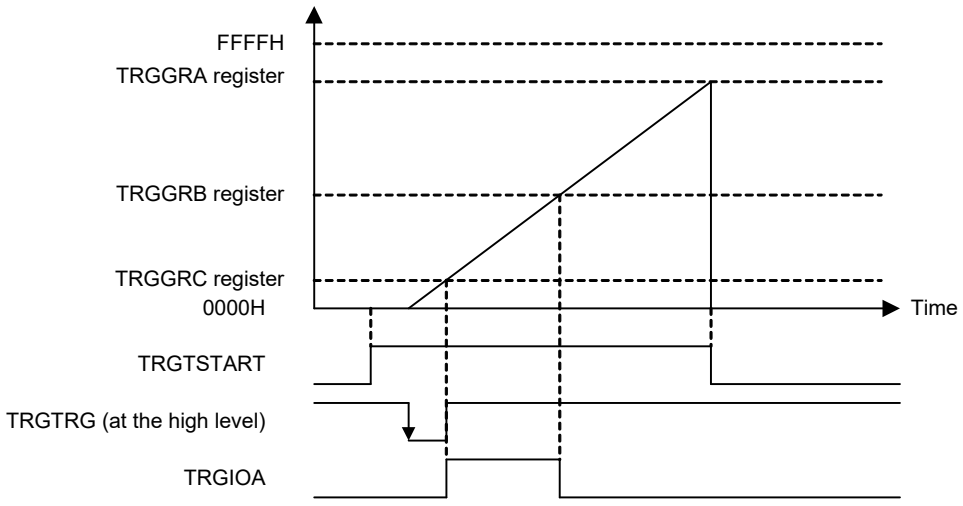


TRGTSTART: Bit in the TRGSTR register

The above diagram applies under the following conditions:

- The TRGTOA bit in the TRGOCR register is set to 0 (initial output level: low, following a compare match with the TRGGRC register: high, following a compare match with the TRGGRB register: low).
- The TRGTCEG[1:0] bits in the TRGMR1 register are set to 00B (to disable TRGTRG input).
- The TRGCCLR[1:0] bits in the TRGCR register are set to 01B (to clear the counter by compare match with TRGGRA).

Figure 13 - 42 Operation Example (5) in PWM2 Mode



TRGTSTART: Bit in the TRGSTR register

The above diagram applies under the following conditions:

- The TRGTOA bit in the TRGOCR register is set to 0 (initial output level: low, following a compare match with the TRGGRC register: high, following a compare match with the TRGGRB register: low).
- The TRGTCEG[1:0] bits in the TRGMR1 register are set to 10B (to select the falling edge of TRGTRG input as the trigger).
- The TRGCCLR[1:0] bits in the TRGCR register are set to 01B (to clear the counter by compare match with TRGGRA).

13.4.6 Phase counting mode

In phase counting mode, a phase difference between external input signals from two pins TRGCLKA and TRGCLKB is detected and the TRG counter is incremented or decremented.

When the phase counting mode is set when bits PM00 and PM01 in the PM0 register are 1, regardless of the settings of the TRGTCK[2:0] and TRGCKEG[1:0] bits in the TRGCR register, pins TRGCLKA and TRGCLKB automatically function as external clock input pins and the TRG counter is incremented or decremented by the CNTEN7 to CNTEN0 bits in the TRGCNTC register. However, the TRGCCLR[1:0] bits in the TRGCR register and registers TRGIOR, TRGIER0, TRGSR0, TRGIER1, TRGSR1, TRGGRA, and TRGGRB are enabled. This allows the input capture and output compare functions, PWM output function, and interrupt sources to be used.

Counting by the TRG counter proceeds on both the rising and falling edges of pins TRGCLKA and TRGCLKB by the CNTEN7 to CNTEN0 bits. The condition for clearing the TRG counter is selectable from among the input level on the TRGIDZ, TRGCLKA, or TRGCLKB pin or compare match between the TRG counter and TRGGRC or TRGGRD register.

Table 13 - 17 lists the phase counting mode specifications and **Table 13 - 18** lists the increment/decrement conditions for the TRG counter.

Table 13 - 17 Phase Counting Mode Specifications (1/2)

Item	Specification
Sources for counting	External signal input to the TRGCLKA or TRGCLKB pin
Direction of counting	Increment/decrement
Condition to start counting	1 (to start counting) is written to the TRGSTART bit in the TRGMR0 register.
Condition to stop counting	0 (to stop counting) is written to the TRGSTART bit in the TRGMR0 register while the TRGCSEL bit in the TRGSTR register is set to 1.
Conditions for clearing the counter	• Detection of an active edge on the TRGIDZ pin • The level on the TRGCLKA, TRGCLKB, or TRGIDZ pin meets the condition for clearing the counter. • Counting up proceeds after a match between the TRG counter and TRGGRC register is detected. • Counting down proceeds after a match between the TRG counter and TRGGRD register is detected.
Interrupt request generation timing (timer RG2 interrupt)	• Input capture (active edge of TRGIOA or TRGIOB input) • Compare match (contents of the TRG counter and TRGGRA, TRGGRB, TRGGRC, or TRGGRD register match) • The TRG counter overflows. • The TRG counter underflows.
Interrupt request generation timing (timer RG2 clearing interrupt)	Any of the conditions for clearing the TRG counter is detected.
Interrupt request generation timing (timer RG2 phase change detection interrupt)**Note**	Counting up or down by the TRG counter according to the conditions for A- and B-phase detection specified in the TRGCNTC register.
TRGIOA pin function	I/O port, input-capture input, output-compare output, or PWM output
TRGIOB pin function	I/O port, input-capture input, or output-compare output
TRGCLKA, TRGCLKB pin function	External clock input
TRGIDZ pin function	Clearing the TRG counter
Read from timer	• The counter value can be read by reading the TRG counter. • The counter value can be read by reading the TRGPMC register.
Write to timer	• The TRG counter can be written to. • The TRGPMC register can be written to.

Table 13 - 17 Phase Counting Mode Specifications (2/2)

Item	Specification
Selectable functions	• Selection of counter increment/decrement conditions Selectable by the CNTEN7 to CNTEN0 bits in the TRGCNTC register. • Input capture and output compare functions and PWM function can be used. • Clearing the counter • Phase change time measurement function

Note Capturing does not proceed at the time the TRG counter starts counting in response to detecting the A- and B-phase states specified in the TRGCNTC register for the first time after the TRGSTART bit in the TRGMR0 register has been set to 1. Also, a timer RG2 phase change detection interrupt is not generated at this time.

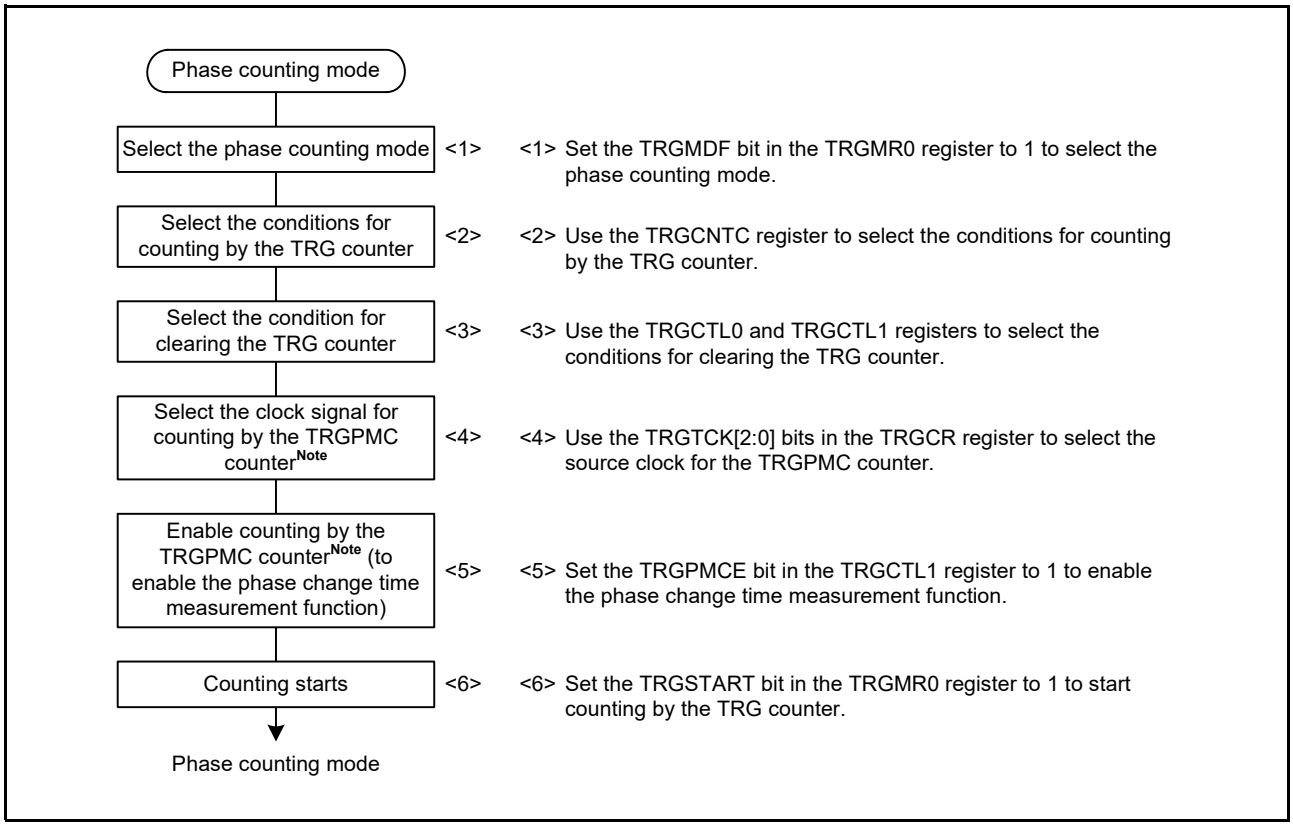
Table 13 - 18 Increment/Decrement Conditions for the TRG Counter

TRGCLKB pin	↑	High-level input	↓	Low-level input	High-level input	↓	Low-level input	↑
TRGCLKA pin	Low-level input	↑	High-level input	↓	↓	Low-level input	↑	High-level input
CNTEN7 to CNTEN0 bits in TRGCNTC register	CNTEN7	CNTEN6	CNTEN5	CNTEN4	CNTEN3	CNTEN2	CNTEN1	CNTEN0
Direction of counting**Note**	+1	+1	+1	+1	–1	–1	–1	–1

Note The direction of counting when each bit in the TRGCNTC register is 1 (decrement or increment) is shown. When the TRGCNTC[7:0] bits are set to 00H, counting does not proceed.

1. Procedure example for setting the phase counting mode
   Figure 13 - 43 shows a procedure example for setting the phase counting mode.

Figure 13 - 43 Procedure Example for Setting the Phase Counting Mode



Note Measuring the phase change time in inputs on the TRGCLKA (A phase) and TRGCLKB (B phase) pins with the use of the TRGPMC counter requires this setting. For details, see **13.4.6 Phase counting mode**, **<4> Phase change time measurement function**.

2. Operation example

Figures 13 - 44 to **13 - 47** show operation examples in phase counting mode.

In phase counting mode, the TRG counter is incremented or decremented on both the rising(↑) and falling(↓) edges of pins TRGCLKA and TRGCLKB by the CNTEN7 to CNTEN0 bits in the TRGCNTC register.

Figure 13 - 44 Operation Example (1) in Phase Counting Mode

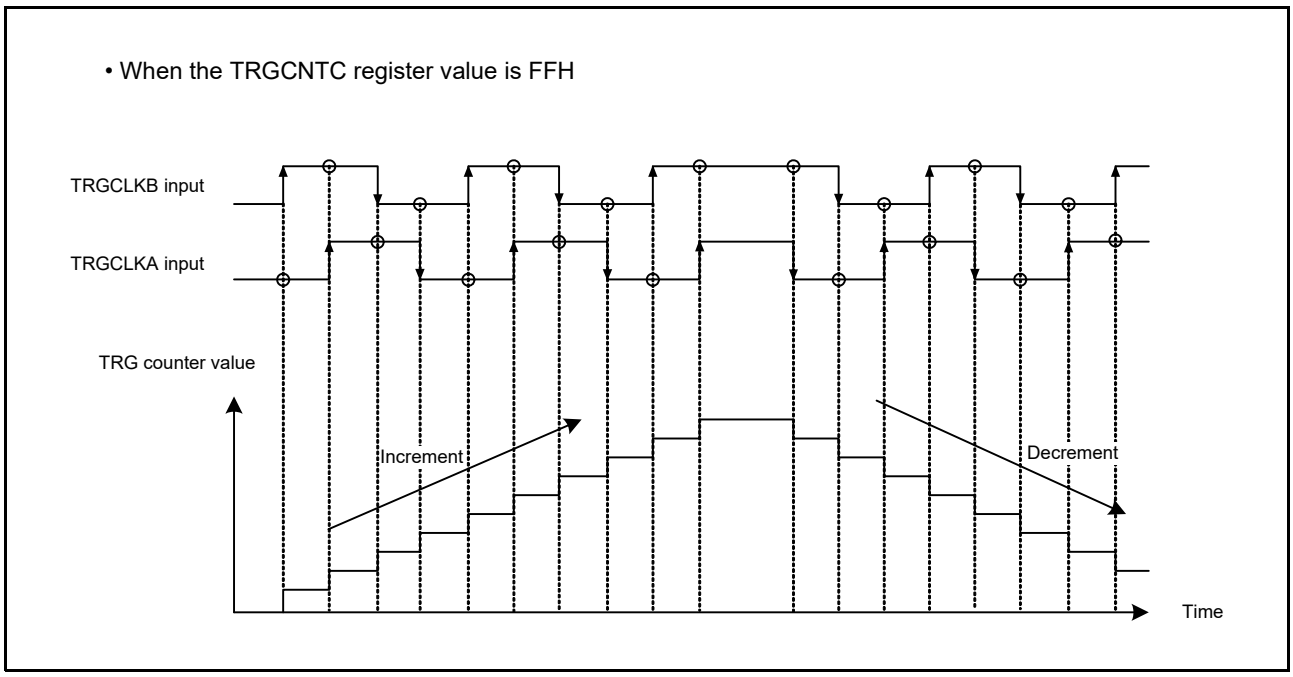


Figure 13 - 45 Operation Example (2) in Phase Counting Mode

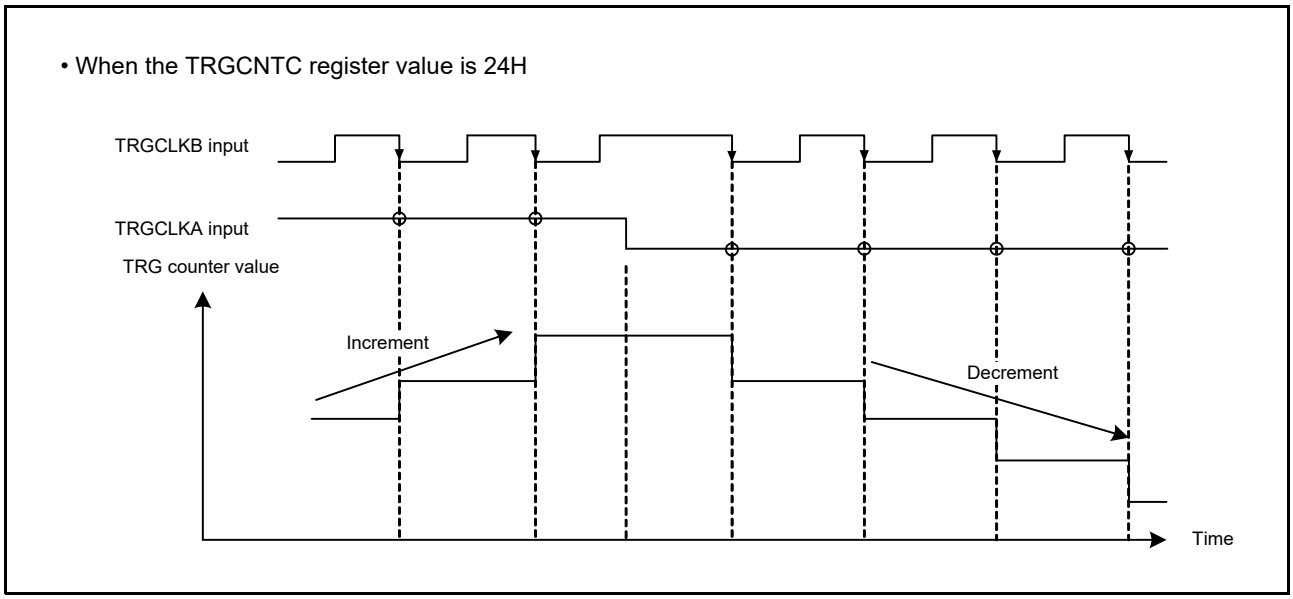


Figure 13 - 46 Operation Example (3) in Phase Counting Mode

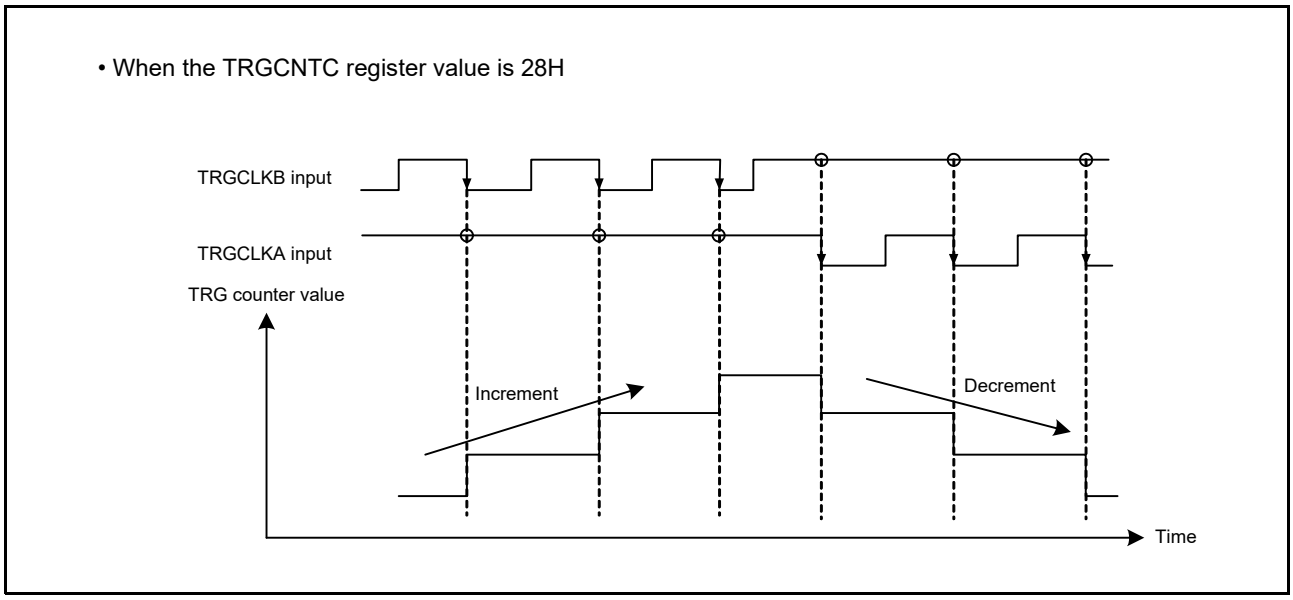
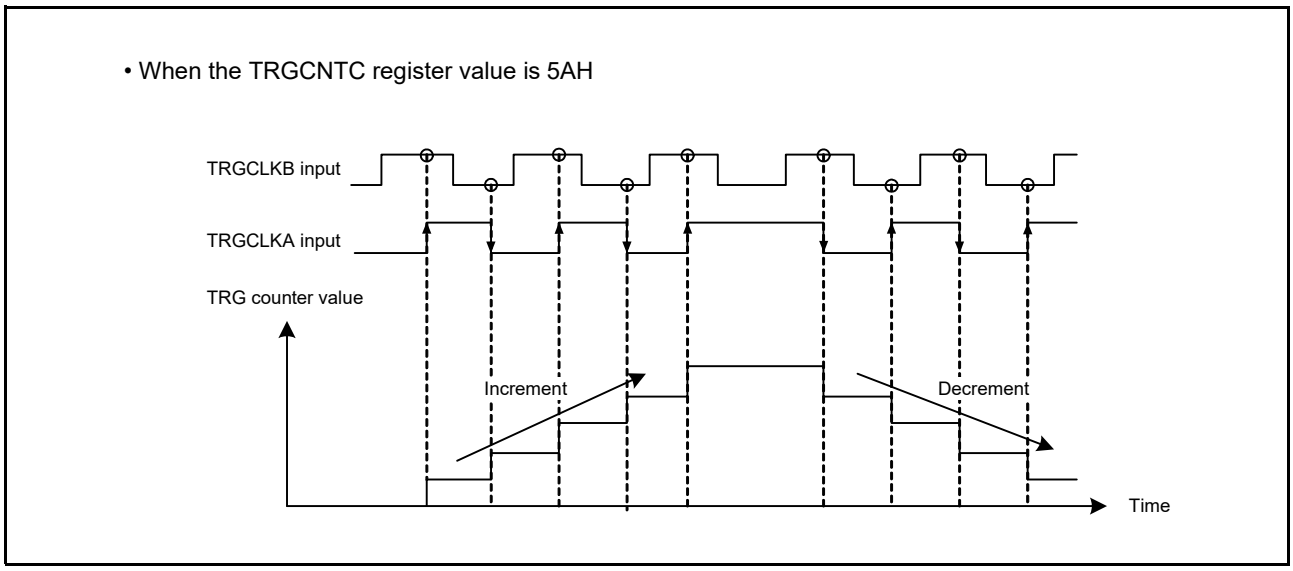


Figure 13 - 47 Operation Example (4) in Phase Counting Mode



3. Counter clearing function

The phase counting mode supports three methods of clearing the TRG counter.

<1> Clearing by detection of an active edge (specified in TRGIDZ[1:0]) on the TRGIDZ pin

The TRG counter is cleared to 0000H by detection of an active edge of the TRGIDZ input specified in the TRGIDZ[1:0] bits while the TRGSCE bit in the TRGCTL0 register is set to 0.

<2> Clearing by a level of any of the TRGCLKA, TRGCLKB, and TRGIDZ inputs meeting the condition for clearing (specified in TRGZCL, TRGBCL, or TRGACL)

While the TRGSCE bit in the TRGCTL0 register is set to 1, the TRG counter is cleared to 0000H when a level of any of the TRGIDZ, TRGCLKB, and TRGCLKA inputs meets the condition for clearing specified in the TRGZCL, TRGBCL, or TRGACL bit.

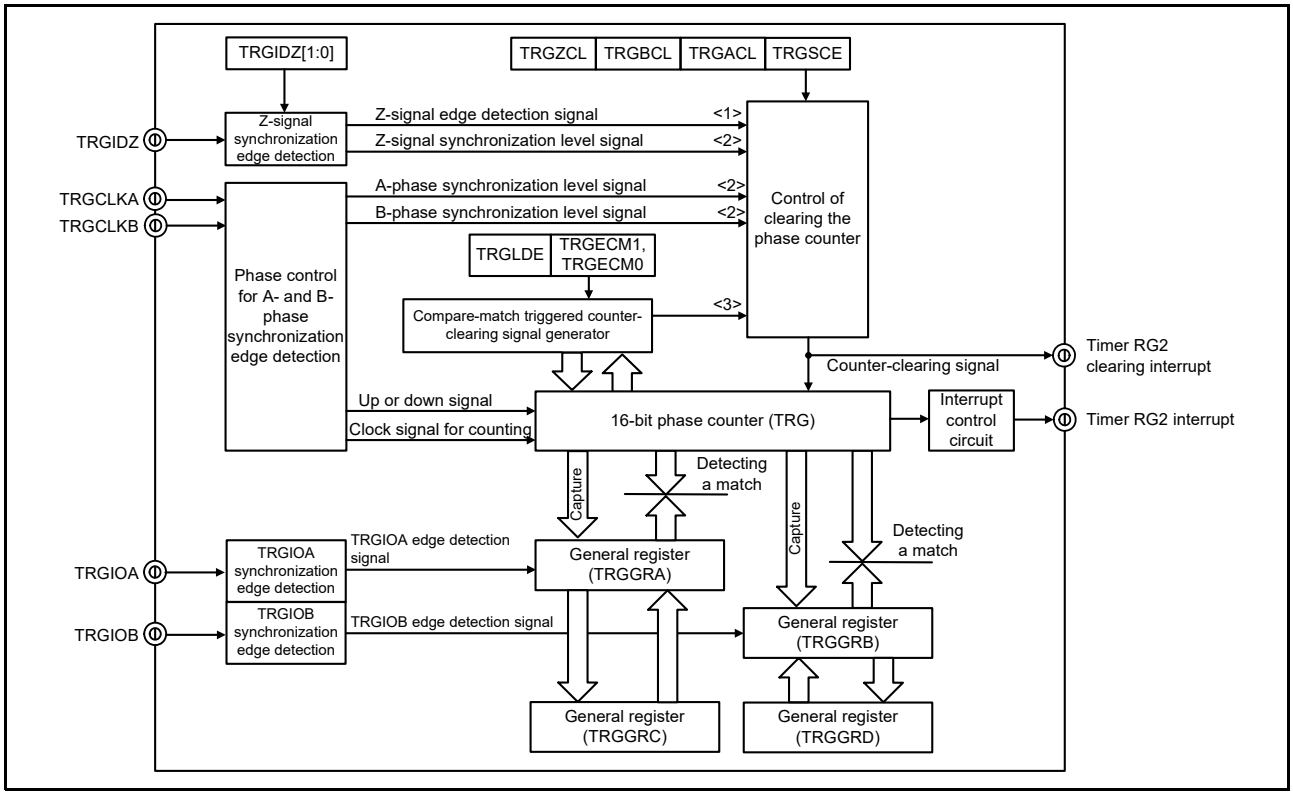
<3> Clearing by detection of a compare match between the TRG counter and TRGGRC register or TRGGRD register

While the TRGECM0 bit in the TRGCTL1 register is set to 1, the TRGGRC register operates as the general register and the setting of the TRGBUFA bit in the TRGIOR register is invalid. If counting up is to proceed following detection of a match between the TRG counter and TRGGRC register, the TRG counter is cleared to 0000H. If counting down is to proceed following an underflow of the TRG counter while the TRGLDE bit in the TRGCTL1 register is set to 1, the setting of the TRGGRC register is loaded to the TRG counter.

While the TRGECM1 bit in the TRGCTL1 register is set to 1, the TRGGRD register operates as the general register and the setting of the TRGBUFB bit in the TRGIOR register is invalid. If counting down is to proceed following detection of a match between the TRG counter and TRGGRD register, the TRG counter is cleared to 0000H.

When clearing proceeds in response to detecting an active edge on the TRGIDZ pin or a level of any of the three inputs meeting the condition for clearing, the TRGZCLF flag in the TRGSR1 register is set. When clearing proceeds in response to detecting a match with the TRGGRC or TRGGRD register, the TRGPCLF flag in the TRGSR1 register is set. A timer RG2 clearing interrupt is generated in both cases. **Figure 13 - 48** is a block diagram for clearing the counter in phase counting mode.

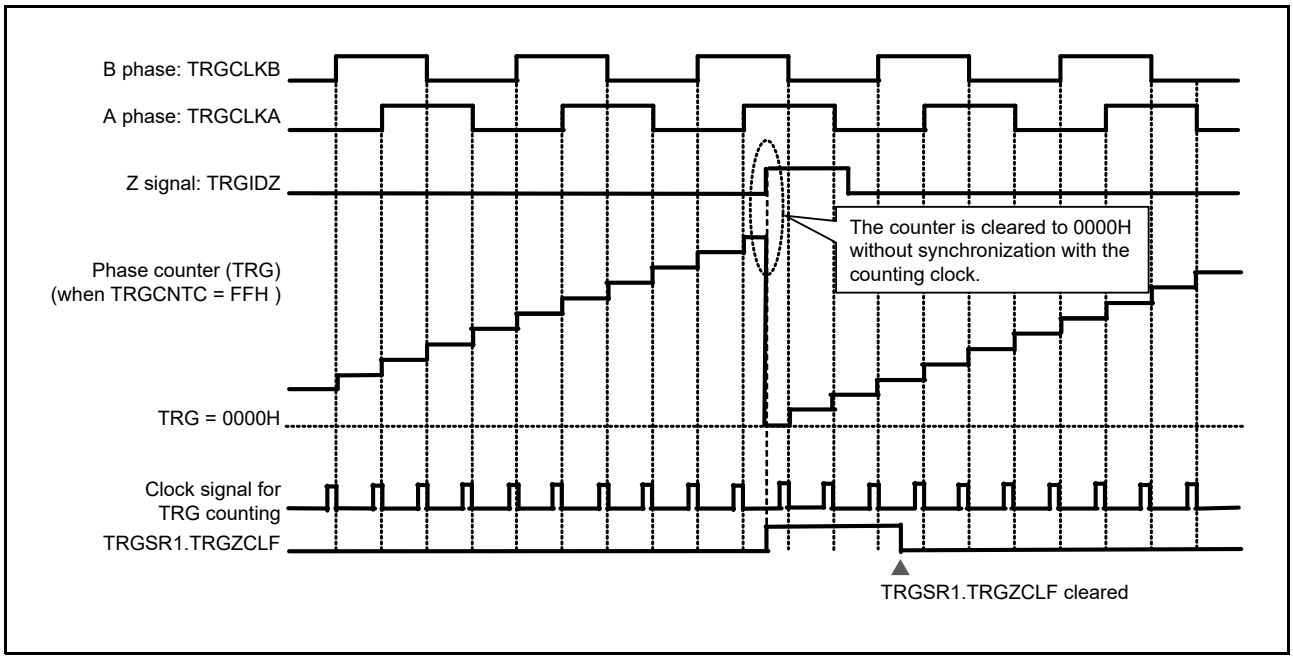
Figure 13 - 48 Block Diagram for Clearing the Counter in Phase Counting Mode



The following shows details of the function for clearing the TRG counter in phase counting mode.

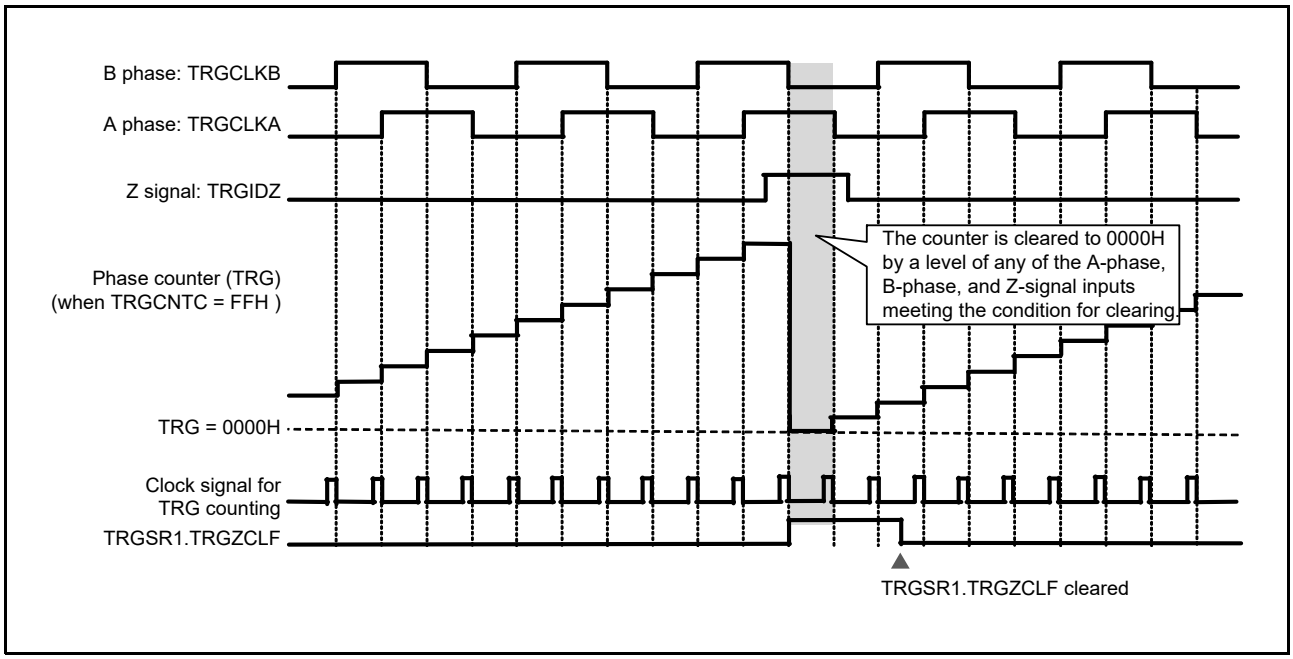
<1> Clearing by detection of an active edge (specified in TRGIDZ[1:0]) on the TRGIDZ pin

Figure 13 - 49 Clearing the Counter by Detection of an Active Edge on the TRGIDZ Pin



<2> Clearing by a level of any of the TRGCLKA, TRGCLKB, and TRGIDZ inputs meeting the condition for clearing (specified in TRGZCL, TRGBCL, or TRGACL)

Figure 13 - 50 Clearing the Counter by an Input Level Meeting the Condition for Clearing



<3> Clearing by detection of a compare match between the TRG counter and TRGGRC or TRGGRD register

Figures 13 - 51 to **13 - 54** show configuration diagrams and examples of operation with the use of the two different settings of the TRGCTL1 register.

Operation example 1: Operation when TRGECM0 = 1, TRGECM1 = 0, and TRGLDE = 1 in the TRGCTL1 register

If counting up is to proceed following detection of a compare match between the TRG counter and TRGGRC register, the counter is cleared to 0000H. If counting down is to proceed following an underflow of the TRG counter while the TRGLDE bit in the TRGCTL1 register is set to 1, the setting of the TRGGRC register is loaded to the TRG counter.

When clearing proceeds in response to detecting a compare match with the TRGGRC register, the TRGPCLF flag is set.

Figure 13 - 51 Configuration Diagram When TRGECM0 = 1, TRGECM1 = 0, and TRGLDE = 1

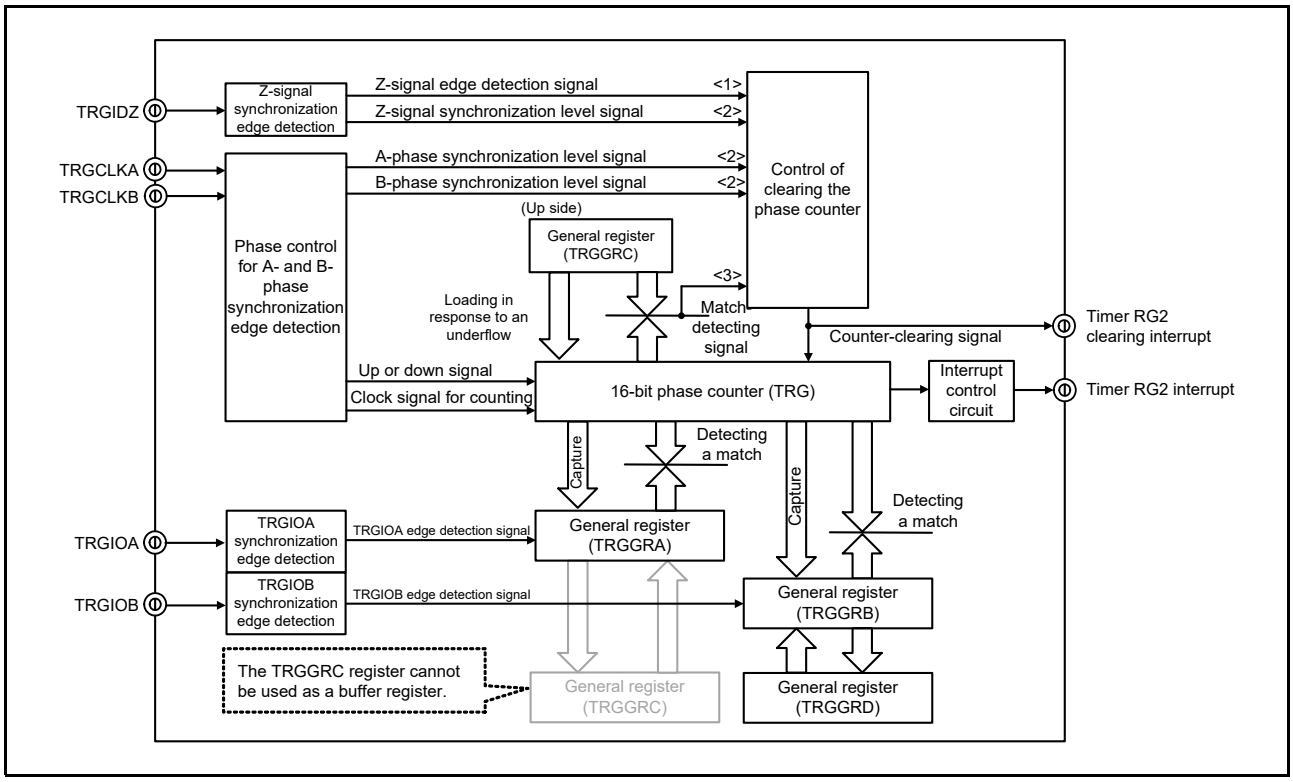
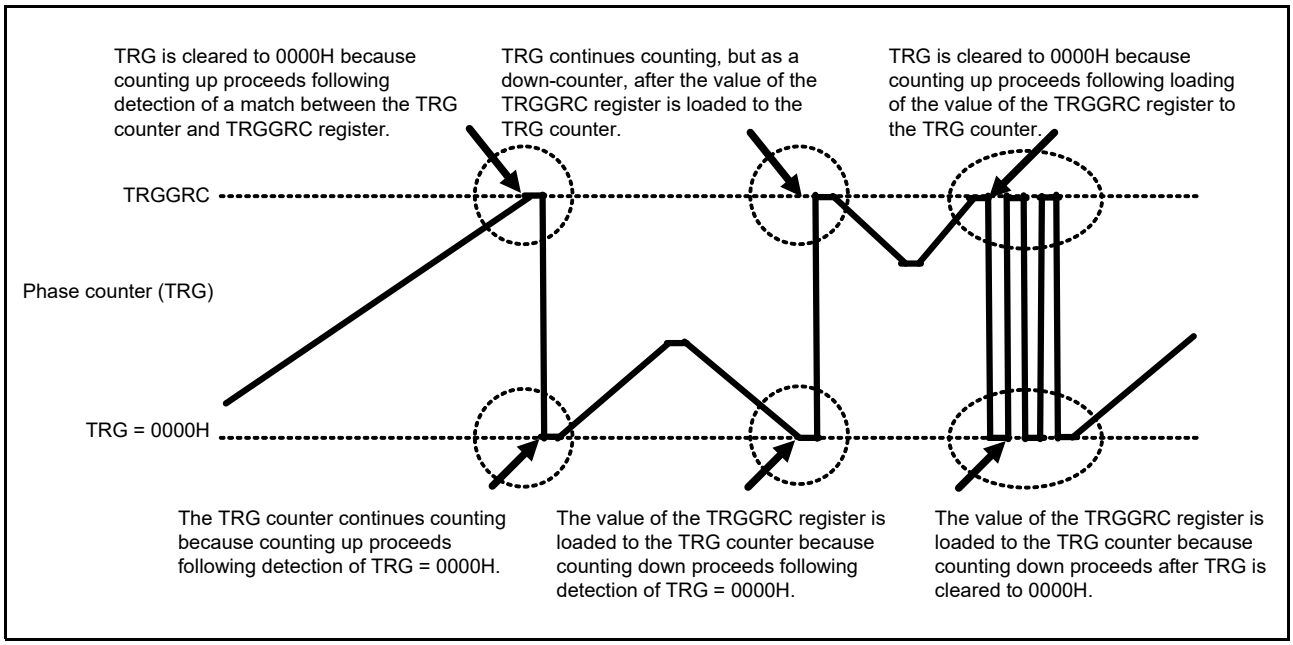


Figure 13 - 52 Operation Example When TRGECM0 = 1, TRGECM1 = 0, and TRGLDE = 1



Operation example 2: Operation when TRGECM0 = 1, TRGLDE = 0, and TRGECM1 = 1 in the TRGCTL1 register

When TRGECM0 = 1, TRGECM1 = 1, and TRGLDE = 0, the counter is not cleared to 0000H with the TRGGRC register setting used as the upper threshold, but starts counting from FFFFH with the TRGGRD register setting used as the lower threshold. The values of the TRGGRC and TRGGRD registers should be set under the condition TRGGRC ≤ TRGGRD.

If counting up is to proceed following detection of a match between the TRG counter and TRGGRC register, the TRG counter is cleared to 0000H.

If counting down is to proceed following detection of a match between the TRG counter and TRGGRD register, the TRG counter is cleared to 0000H.

When clearing proceeds in response to detecting a match with the TRGGRC or TRGGRD register, the TRGPCLF flag is set.

Figure 13 - 53 Configuration Diagram When TRGECM0 = 1, TRGLDE = 0, and TRGECM1 = 1

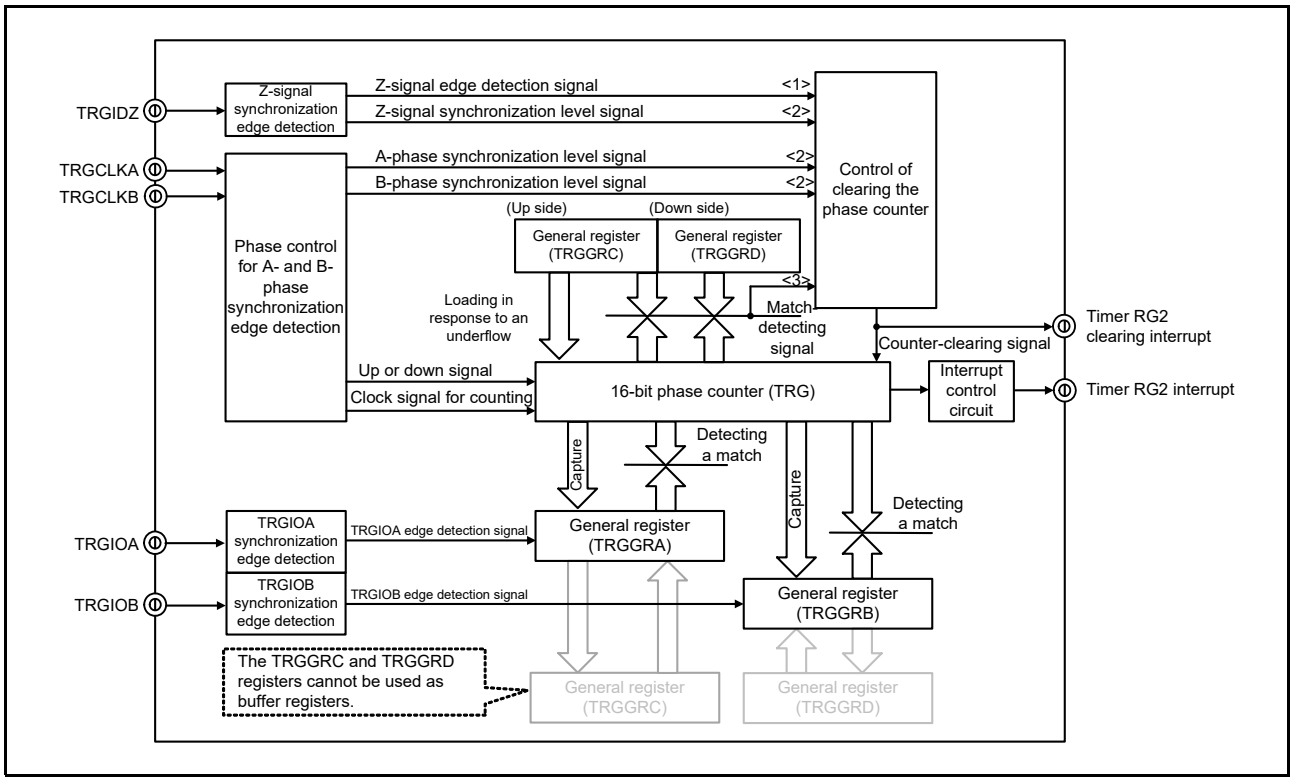
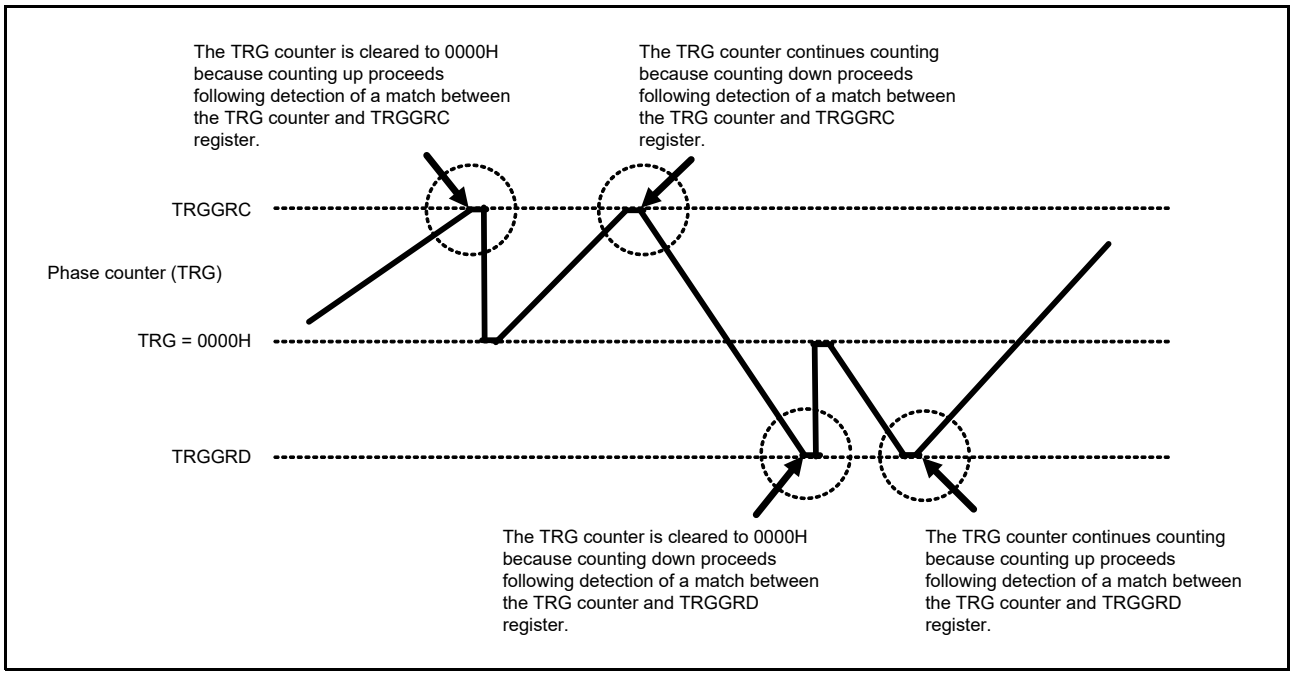


Figure 13 - 54 Operation Example When TRGECM0 = 1, TRGLDE = 0, and TRGECM1 = 1



<4> Phase change time measurement function

The phase counting mode supports the phase change time measurement function. This function measures the phase change times of the TRGCLKA and TRGCLKB inputs with the use of the phase change time measurement counter (TRGPMC) and two phase change time measurement capture registers 0 and 1 (TRGCAP0 and TRGCAP1) for use in capturing the TRGPMC counter values.

The TRGPMC register is cleared to 0001H in response to counting up or down by the TRG counter according to the conditions for A- and B-phase detection specified in the TRGCNTC register. The TRGPMC register is incremented by cycles of the source clock specified in the TRGTCK[2:0] bits of the TRGCR register. The value of the TRGPMC register is loaded to the TRGCAP0 register and the value of the TRGCAP0 register (the previous captured value) is loaded to the TRGCAP1 register at the times the TRG counter counts up and down according to the conditions for A- and B-phase detection specified in the TRGCNTC register. The phase change times of the inputs on the TRGCLKA and TRGCLKB pins can be calculated by using the values stored in the TRGCAP0 and TRGCAP1 registers.

A- or B-phase change time = TRGCAP0 setting × TRGPMC counting source (f_{CLK}, $f_{CLK}/2$, $f_{CLK}4$, $f_{CLK}/8$)

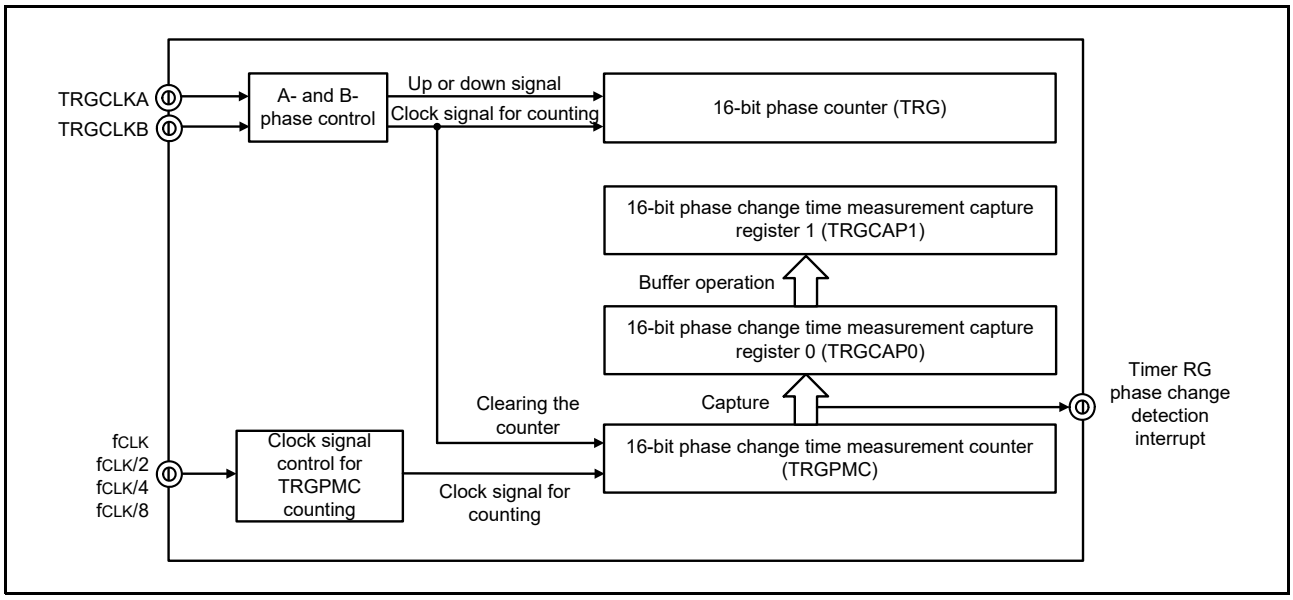
Caution When $f_{CLK}/2$, $f_{CLK}/4$, or $f_{CLK}/8$ is selected as the TRGPMC count source, an error at a maximum of one cycle of the TRGPMC count source ($f_{CLK}/2$, $f_{CLK}/4$, or $f_{CLK}/8$) is generated because the TRGCLKA and TRGCLKB signals are detected in synchronization with f_{CLK}.

Table 13 - 19 lists the specifications of the phase change time measurement function.

Table 13 - 19 Phase Counting Mode Specifications

Item	Specification
Sources for TRGPMC counting	f_{CLK}, $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$
Direction of TRGPMC counting	Increment
Condition to start TRGPMC counting	The TRG counter starts counting up and down in response to A- and B-phase detection after writing of 1 (to start counting) to the TRGSTART bit in the TRGMR0 register.
Condition to stop TRGPMC counting	1 is written to the TRGCSEL bit in the TRGSTR register and 0 (to stop counting) is written to the TRGSTART (TRGTSTART) bit in the TRGMR0 (TRGSTR) register.
Condition for clearing TRGPMC counting	Detection of counting up and down by the TRG counter in response to A- and B-phase detection
Interrupt request generation timing timer RG2 phase change detection interrupt	Counting up and down by the TRG counter in response to A- and B-phase detection specified in the TRGCNTC register
Read from timer	The counter value can be read by reading the TRGPMC counter.
Write to timer	The TRGPMC counter can be written to.
Selectable functions	Buffer operation (see **13.4.1 Items common to multiple modes and functions**, **2. Buffer operation**)

Figure 13 - 55 Phase Counting Mode and Phase Change Time Measurement Function



Figures 13 - 56 to **13 - 59** show operation examples in phase counting mode with the phase change time measurement function in use.

Figure 13 - 56 Phase Change Time Measurement Function in Phase Counting Mode (1)

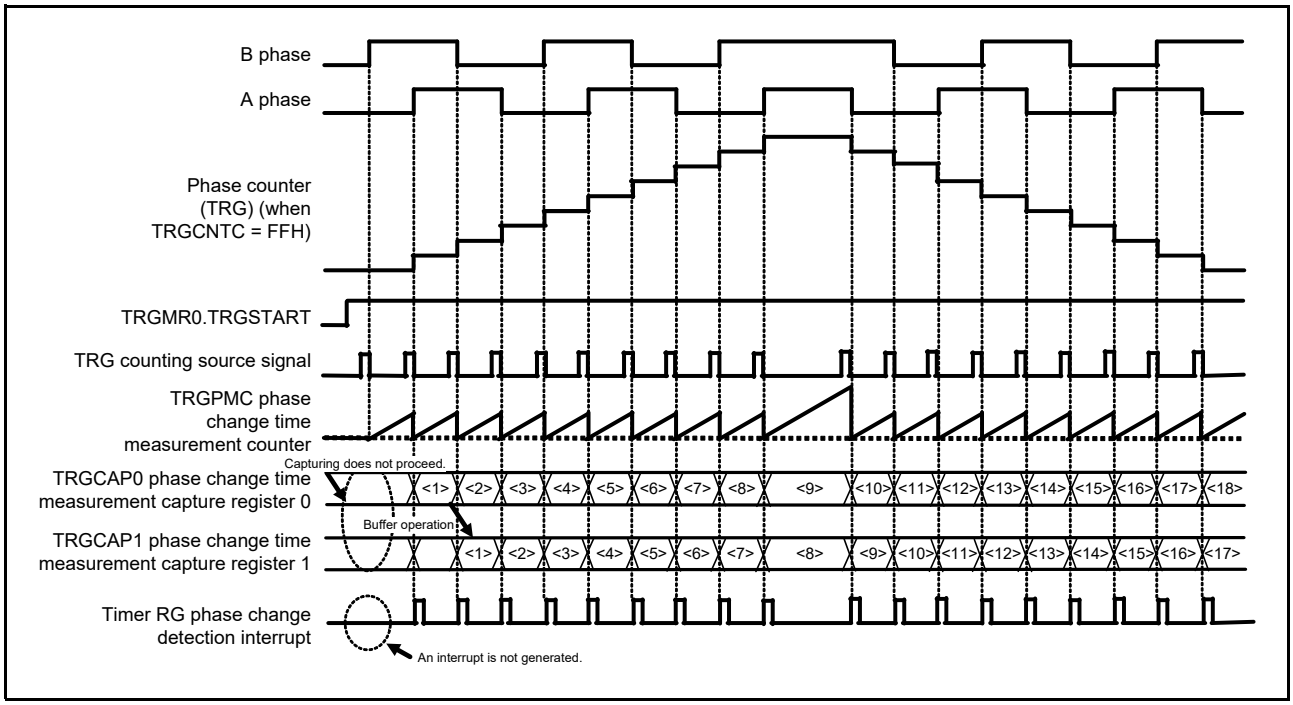


Figure 13 - 57 Phase Change Time Measurement Function in Phase Counting Mode (2)

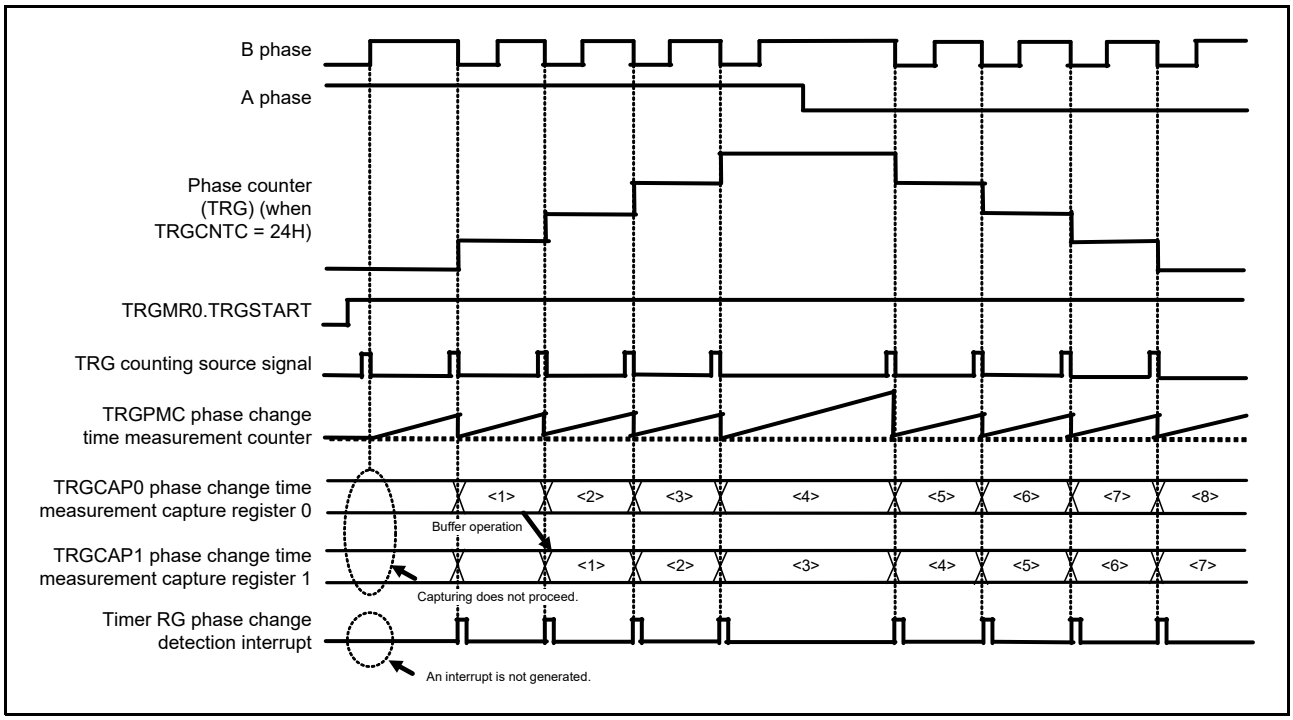


Figure 13 - 58 Phase Change Time Measurement Function in Phase Counting Mode (3)

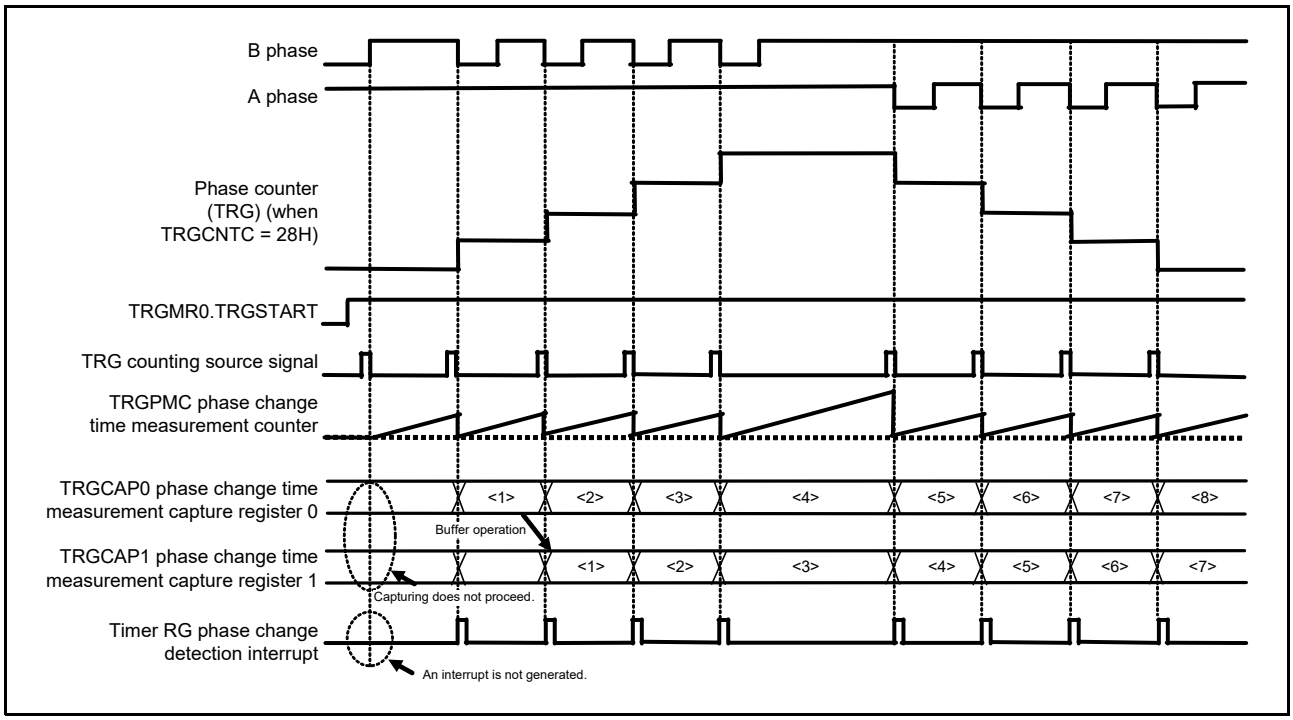
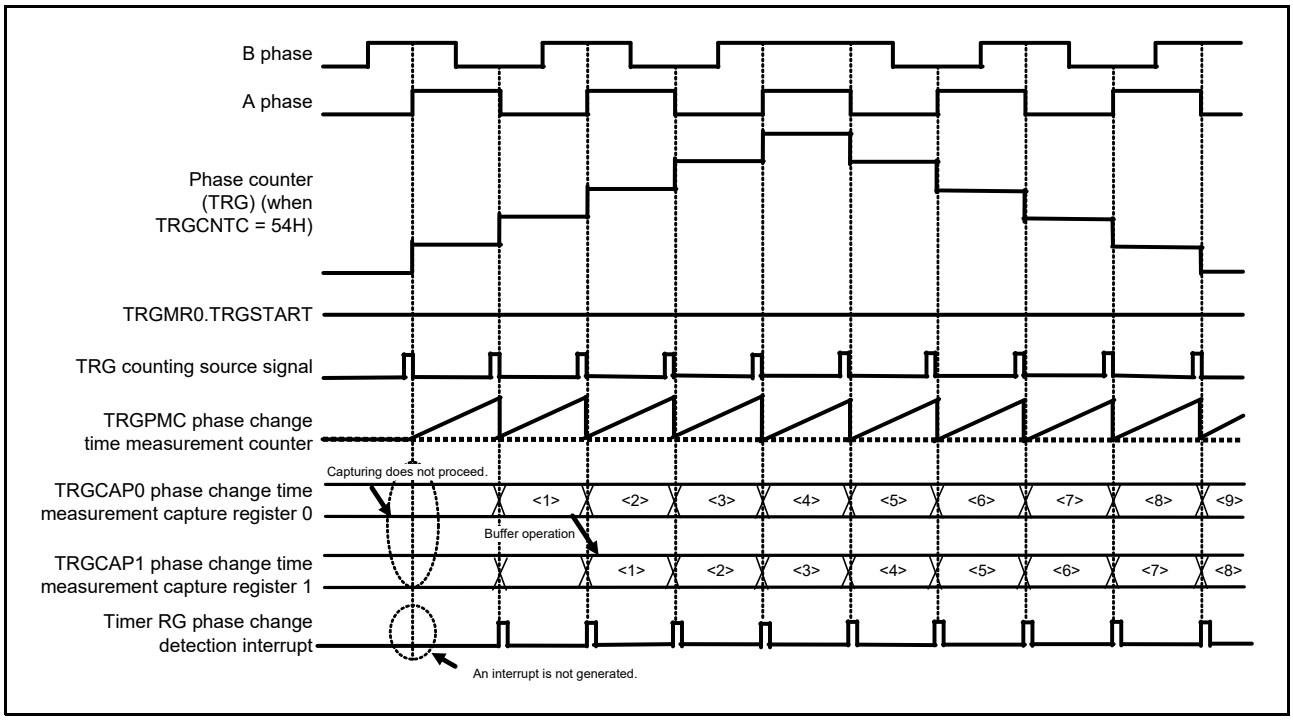


Figure 13 - 59 Phase Change Time Measurement Function in Operation Example (4) in Phase Counting Mode (4)



13.5 Interrupt Sources

Timer RG2 generates the following three types of interrupt: timer RG2, timer RG2 clearing, and timer RG2 phase change detection interrupts.

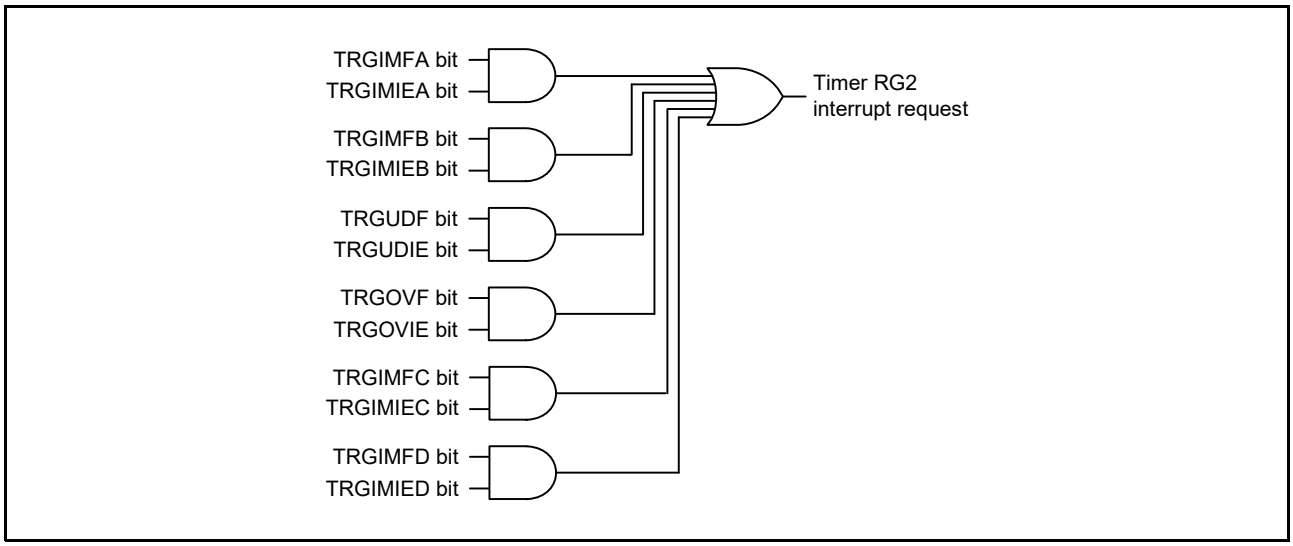
13.5.1 Timer RG2 interrupt

Timer RG2 generates the timer RG2 interrupt request from six sources. **Table 13 - 20** lists the registers associated with the timer RG2 interrupt and **Figure 13 - 60** shows the timer RG2 interrupt block diagram.

Table 13 - 20 Registers Associated with the Timer RG2 Interrupt

	Timer RG Status Register	Timer RG Interrupt Enable Register	Interrupt Request Flag (Register)	Interrupt Mask Flag (Register)	Priority Specification Flag (Register)
Timer RG2	TRGSR0	TRGIER0	TRGIF (IF1H)	TRGMK (MK1H)	TRGPR0 (PR01H) TRGPR1 (PR11H)

Figure 13 - 60 Timer RG2 Interrupt Block Diagram



TRGIMFA, TRGIMFB, TRGUDF, TRGOVF, TRGIMFC, TRGIMFD: Bits in the TRGSR0 register
TRGIMIEA, TRGIMIEB, TRGUDIE, TRGOVIE, TRGIMIEC, TRGIMIED: Bits in the TRGIER0 register

Since timer RG2 generates an interrupt request (timer RG2 interrupt) from multiple interrupt request sources, the following differences from other maskable interrupts except the timer RD2 interrupt apply:

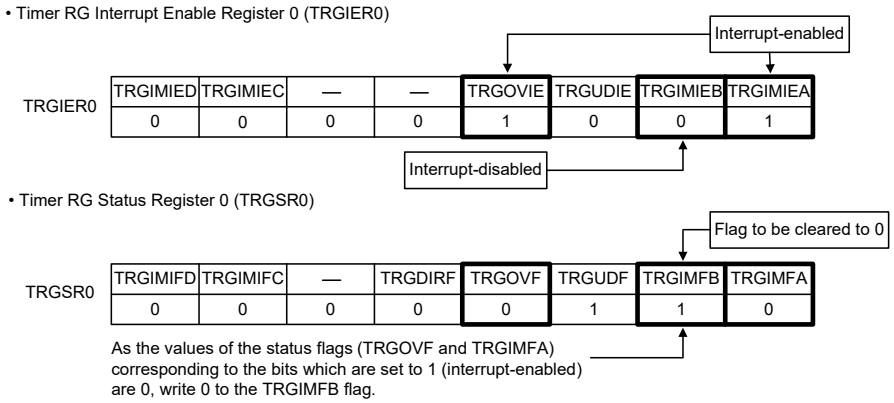
- When a bit in the TRGSR0 register is 1 and the corresponding bit in the TRGIER register is 1 (interrupt enabled), the TRGIF bit in the IF1H register is set to 1 (interrupt requested).
- If multiple bits in the TRGIER0 register are set to 1, use the TRGSR0 register to determine the source of the interrupt request.
- Since the bits in the TRGSR0 register are not automatically set to 0 even if the interrupt is acknowledged, set the corresponding bit to 0 in the interrupt routine.

- To clear the status flag of an interrupt source (applicable status flag) of timer RG2 to 0 while the corresponding interrupt is disabled in the timer RG interrupt enable register i (TRGIER0), use any of the following methods a) to c).

a) Set 00H (all interrupts disabled) in timer RG interrupt enable register 0 (TRGIER0) and write 0 to the applicable status flag.

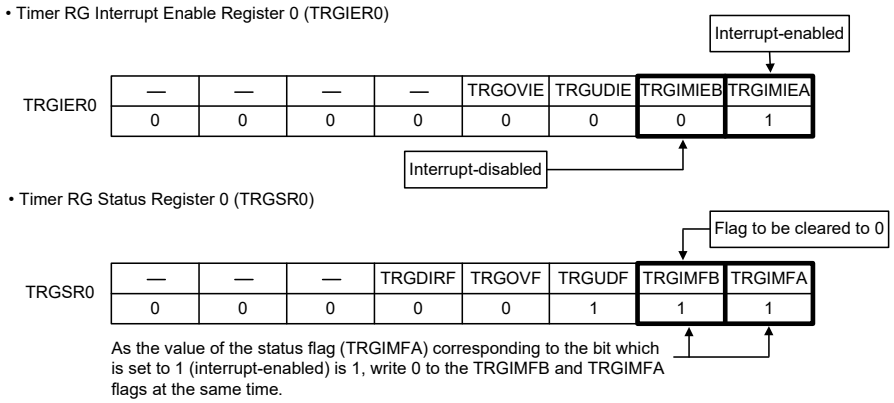
b) When a bit is set to 1 (interrupt-enabled) in timer RG interrupt enable register 0 (TRGIER0) and the value of the status flag of the interrupt source corresponding to the bit is 0, write 0 to the applicable status flag.

Example: To clear the TRGIMFB flag to 0 when bits TRGIMIEA and TRGOVIE are set to 1 (interrupt-enabled) and the TRGIMIEB bit is set to 0 (interrupt-disabled).



c) When a bit is set to 1 (interrupt-enabled) in timer RG interrupt enable register 0 (TRGIER0) and the value of the status flag of the interrupt source corresponding to the bit is 1, write 0 to the status flag and the applicable status flag at the same time.

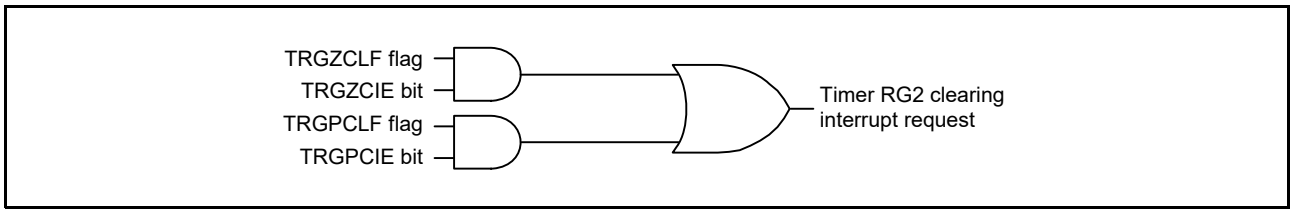
Example: To clear the TRGIMFB flag to 0 when the TRGIMIEA bit is set to 1 (interrupt-enabled) and the TRGIMIEB bit is set to 0 (interrupt-disabled).



13.5.2 Timer RG2 clearing interrupt

Timer RG2 generates the timer RG2 clearing interrupt request from two sources. **Figure 13 - 61** shows the timer RG2 clearing interrupt block diagram.

Figure 13 - 61 Timer RG2 Clearing Interrupt Block Diagram



TRGZCLF, TRGPCLF: Bits in the TRGSR1 register
TRGZCIE, TRGPCIE: Bits in the TRGIER1 register

13.5.3 Timer RG2 phase change detection interrupt

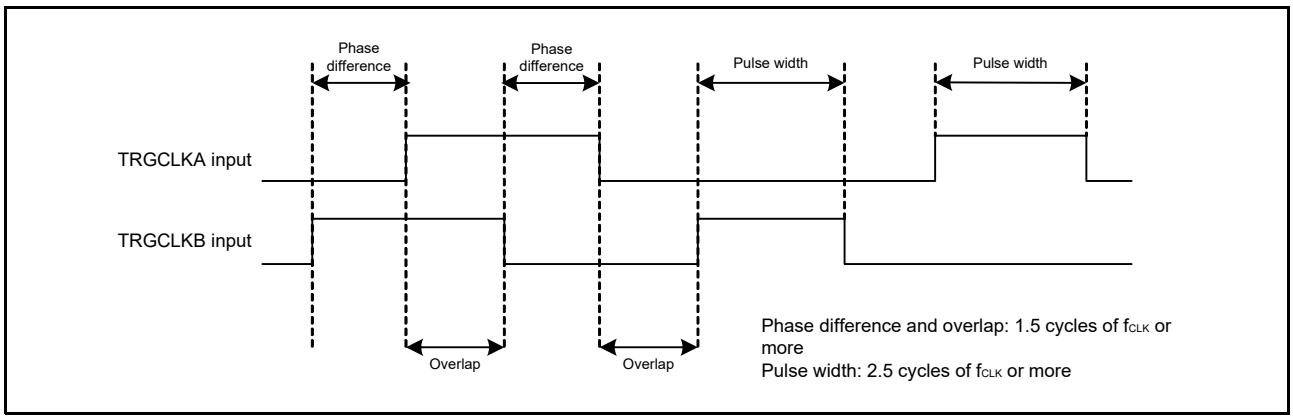
When the phase change time measurement function is in use (selected by setting the TRGPMCE bit of the TRGCTL1 register to 1) while the phase counting mode is selected (by setting the TRGMDF bit of the TRGMR0 register to 1), timer RG2 generates a timer RG2 phase change detection interrupt when the detected A- and B-phases meet any of the counting conditions specified in the TRGCNTC register. This interrupt has a different configuration from the timer RG2 interrupt and timer RG2 clearing interrupt and a flag for use in detecting phase changes is not in place.

13.6 Cautions for Timer RG2

13.6.1 Phase difference, overlap, and pulse width in phase counting mode

The phase difference and overlap between external input signals from pins TRGCLKA and TRGCLKB should be 1.5 cycles of f_{CLK} or more, respectively. The pulse width should be 2.5 cycles of f_{CLK} or more.
Figure 13 - 62 shows the phase difference, overlap, and pulse width in phase counting mode.

Figure 13 - 62 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode



13.6.2 Mode switching

- When switching modes during operation, set the TRGSTART bit in the TRGMR0 register to 0 (to stop counting) before switching**Note**.
- After switching modes, set the TRGIF bit to 0 before starting operation.
  Refer to **Section 29 Interrupt Functions** for details.

Note The registers and bits that cannot be rewritten during counting are as follows:
- All bits except TRGSTART in the TRGMR0 register
- TRGMR1 register
- TRGCNTC register
- TRGCR register
- TRGIOR register
- TRGCTL0 and TRGCTL1 registers
- TRGOCR register

13.6.3 Switching the source for counting

- Stop counting before switching the source for counting.
  [Changing procedure]
  <1> Set the TRGSTART bit in the TRGMR0 register to 0 (to stop counting).
  <2> Change the TRGTCK[2:0] bits in the TRGCR register.

13.6.4 Procedure for setting pins TRGIOA, TRGIOB, and TRGTRG

After release from the reset state, the I/O port pin functions multiplexed with the TRGIOA, TRGIOB, and TRGTRG pin functions operate as input port pins.

- To use the TRGIOA and TRGIOB pins as outputs, use the following setting procedure:
  [Changing procedure]
  <1> Set the mode and the initial value/output enabled (the settings for the initial value and to enable output are made with a single SFR.).
  <2> Set the bits in the port register xx corresponding to pins TRGIOA and TRGIOB to 0.
  <3> Set the bits in the port mode register xx corresponding to pins TRGIOA and TRGIOB to output mode (output is started from pins TRGIOA and TRGIOB).
  <4> Start counting (TRGSTART in the TRGMR0 register = 1).

- To change the bits in the port mode register xx corresponding to pins TRGIOA, TRGIOB, and TRGTRG from output mode to input mode, use the following setting procedure:
  [Changing procedure]
  <1> Set the bits in the port mode register xx corresponding to pins TRGIOA, TRGIOB, and TRGTRG to input mode (input is started from pins TRGIOA, TRGIOB, and TRGTRG).
  <2> Set the TRGIOA and TRGIOB pins to the input capture function and the TRGTRG pin to the PWM2 mode.
  <3> Start counting (TRGSTART in the TRGMR0 register = 1).

- When switching pins TRGIOA, TRGIOB, and TRGTRG from output mode to input mode, input capturing may proceed depending on the pin state. When the digital filter is not used, edge detection is performed after two or more cycles of the CPU clock have elapsed. When the digital filter is enabled, edge detection is performed after up to five cycles of the digital filter sampling clock. The synchronization and digital filter circuits for the TRGTRG input pin are also used with the TRGIOB input pin, so input capturing or TRG counter clearing may proceed depending on the pin state.

- Restrictions on switching the direction of a port from output to input
  When a TRGIOi pin is switched to an external input pin, an edge detection signal may be incorrectly generated before a change of the input signal level (a valid edge) is detected. To prevent this, the following restrictions apply to the port setting procedure.
  – When the TRGDFi bit in the TRGMR0 register is 0 (the digital filter function is not used), change the port settings, wait for two cycles of f_{CLK}, and then set the TRGIOi2 bit in the TRGIOR register to 1.
  – When the TRGDFi bit in the TRGMR0 register is 1 (the digital filter function is used), change the port settings, wait for five cycles of the filter clock, and then set the TRGIOi2 bit in the TRGIOR register to 1.
  – When the TRGDFCLR bit in the TRGMR1 register is 0 (the digital filter function is not used), change the port settings, wait for two cycles of f_{CLK}, and then set the TRGPWM2 bit in the TRGMR1 register to 1.
  – When the TRGDFCLR bit in the TRGMR1 register is 1 (the digital filter function is used), change the port settings, wait for five cycles of the filter clock, and then set the TRGPWM2 bit in the TRGMR1 register to 1.

  Remark i = A or B

13.6.5 External clocks TRGCLKA, TRGCLKB

The pulse width of external clocks input to the TRGCLKA and TRGCLKB pins should be set to three cycles or more of the timer RG2 operating clock (f_{CLK}).

13.6.6 SFR read/write access

When setting timer RG2, set the TRGEN bit in the PER2 register to 1 first. If the TRGEN bit is 0, writes to the timer RG2 control registers are ignored and all the read values are 00H or 0000H (except for the port register xx and the port mode register xx).

1. TRGMR0 register
   Use the following setting procedure when switching the digital filter clock.
   <1> With the TRGSTART bit set to 0 (to stop counting), set bits TRGDFA and TRGDFB (digital filter function select bits of pins TRGIOA and TRGIOB) in the TRGMR0 register, and the TRGDFCK[1:0] bits (clock select bits used by the digital filter function) in the TRGMR0 register.
   <2> Set the TRGSTART bit to 1.

   However, when the digital filter is not set and TRGDFCK[1:0] = 00B remains unchanged after a reset, the setting can be performed in a single step.
   Besides external input pins (TRGIOA and TRGIOB), event input from the ELC can also be selected as an operating source for input capture. To use this function, set the TRGELCICE bit in the TRGMR0 register to 1, and set the input capture function (the rising edge as the active edge for input capture (TRGIOB[2:0] = 100B)).
   This function is disabled in PWM mode and timer mode (with the output compare function in use) (TRGPWM = 1 and TRGIOB2 = 0, respectively).

2. TRG counter
   Writing to the TRGMR0 register has priority over resetting of the counter generated by timer RG2 operating conditions.

3. TRGSTR register
   The TRGSTR register can be set by an 8-bit memory manipulation instruction.
   The TRGTSTART bit of this register and the TRGSTART bit in the TRGMR0 register mirror each other.
   When the TRGCSEL bit is set to 0 (so that a compare match between the TRG counter and TRGGRA register stops counting), writing 0 to the TRGTSTART bit neither stops counting nor affects the value of the TRGTSTART bit. The TRGTSTART bit is only set to 0 (to stop counting) by compare match with the TRGGRA or TRGGRB register.
   In rewriting the TRGSTR register, when changing the TRGCSEL bit to 1 without affecting counting while its value is 0, write 0 to the TRGSTART bit. Writing 1 to the TRGTSTART bit while the counter is stopped may lead to the counter starting to count. When counting is to be stopped under program control, write 0 to the TRGTSTART bit after writing 1 to the TRGCSEL bit. Writing 1 and 0 to the TRGCSEL and TRGTSTART bits at the same time (with a single instruction) does not stop counting.
   In PWM2 mode, do not write to the TRGSTR register at the times a compare match between the TRG counter and TRGGRA register occurs while the setting of the TRGCSEL bit in the TRGSTR register is 0.

   Table 13 - 21 describes the output levels of the TRGIOA and TRGIOB pins while counting is stopped when they are being used as timer RG2 outputs.

Table 13 - 21 Output Levels of the TRGIOA and TRGIOB Pins While Counting Is Stopped

Method of Stopping Counting / Operating Mode	Counting is stopped by writing 0 to the TRGTSTART bit while the TRGCSEL bit is 1.	Counting is stopped in response to a compare match between the TRG counter and TRGGRA or TRGGRB register while the TRGCSEL bit is 0.
Timer mode (with the output compare function in use)	The output level before counting stops is retained.	The output level that was changed in response to a compare match is retained.
PWM mode	The initial level (the inverse of the TRGCCLR1 bit in the TRGCR register) is output.	The output level that was changed in response to a compare match is retained (the initial level specified in the TRGCCLR[1:0] bits in the TRGCR register is output).
PWM2 mode	The output level before counting stops is retained.	The output level that was changed in response to a compare match is retained (the initial output level specified in the TRGTOA bit in the TRGCR register is output).

13.6.7 Input capture while counting is stopped

In input capture mode, even if the TRGSTART bit in the TRGMR0 register is set to 0 (to stop counting), input of the edge specified in the TRGCKEG[1:0] bits in the TRGCR register on the TRGIOA or TRGIOB pin sets the TRGIMFA or TRGIMFB flag in the TRGSR0 register, respectively. If the corresponding TRGIMIEA or TRGIMIEB bit in the TRGIER0 register is set to 1, a timer RG2 (capture) interrupt request is also generated at this time. When the TRGCCLR[1:0] bits in the TRGCR register are set to 01B or 10B, the TRG counter is cleared.

13.6.8 Clearing the counter in response to TRGTRG input while counting is stopped

In PWM2 mode, even if the TRGSTART bit in the TRGMR0 register is set to 0 (to stop counting), the TRG counter is cleared in response to input of the edge specified in the TRGTCEG[1:0] bits in the TRGMR1 register on the TRGTRG pin while the output of the TRGIOA pin is at the inactive level.

13.6.9 Clearing the counter in phase counting mode in response to Z-signal input while counting is stopped

In phase counting mode, even if the setting of the TRGSTART bit in the TRGMR0 register is 0 (to stop counting), the TRGZCLF flag in the TRGSR1 register is set and the TRG counter is cleared when any of the conditions for clearing the TRG counter selected by using the TRGSEC, TRGZCL, TRGBCL, TRGACL, TRGIDZ[1:0] bits in the TRGCTL0 register is met. When the TRGZCIE bit in the TRGIER1 register is set to 1, a timer RG2 clearing interrupt request is generated.

Section 14 Timer RX

14.1 Functions of Timer RX

Timer RX is an input capture timer, the operation of which is triggered by software, a comparator, or timer RD2.

Timer RX operates as follows:

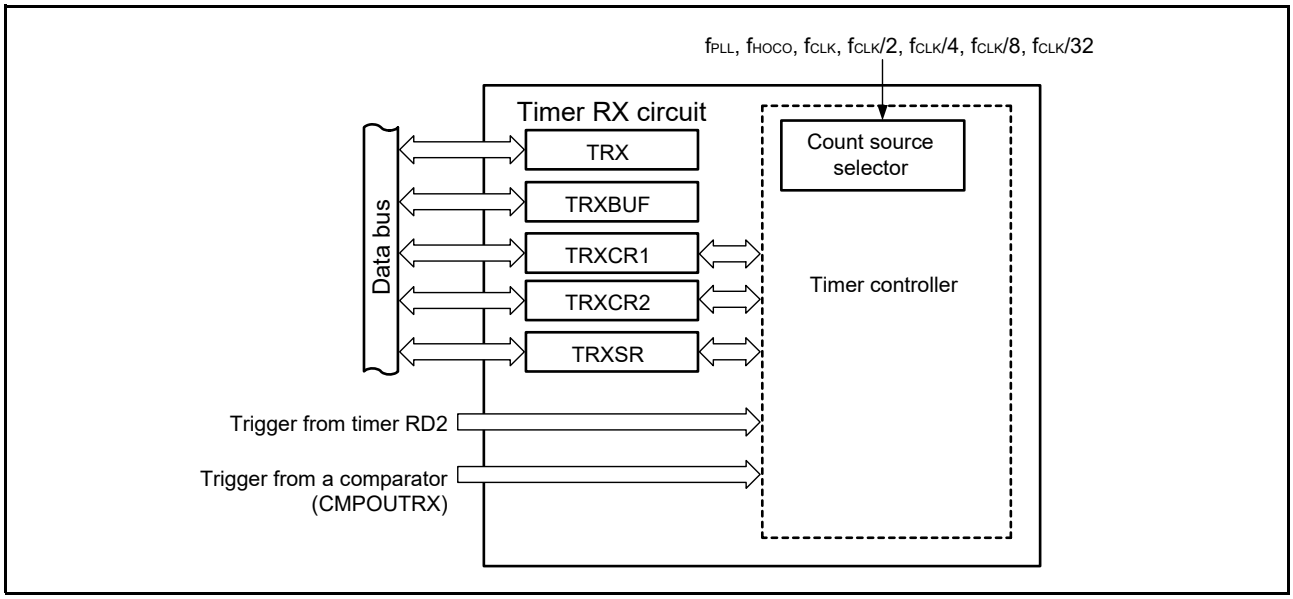
- Starting timer RX counting: Counting by timer RX can be started by a trigger from timer RD2 or software.
- Stopping timer RX counting: Stoppage of counting is triggered by a signal from comparator 0, 1, 2, or 3, or software.
- Input capture operation: Input capture enables transfer of the counted value to a buffer when an interrupt is generated from comparator 0, 1, 2, or 3.
- Resetting the timer RX counter: Resetting of the counter is triggered by a signal from timer RD2, or comparator 0, 1, 2, or 3.

Timer RX operates with the operating clock f_{CLK}, f_{HOCO}, or f_{PLL}.

14.2 Configuration of Timer RX

Figure 14 - 1 shows the block diagram of timer RX.

Figure 14 - 1 Block Diagram of Timer RX



14.3 Registers to Control Timer RX

The following registers are used to control timer RX.

- Peripheral enable register 2 (PER2)
- Peripheral reset control register 2 (PRR2)
- Timer RX counter (TRX)
- Timer RX count buffer counter (TRXBUF)
- Timer RX function control register 1 (TRXCR1)
- Timer RX function control register 2 (TRXCR2)
- Timer RX status register (TRXSR)

14.3.1 Peripheral enable register 2 (PER2)

The PER2 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. If timer RX is to be used, be sure to set bit 1 (TRXEN) of this register to 1. The PER2 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 14 - 2 Format of Peripheral Enable Register 2 (PER2)

Address: F00FCH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER2	FAAEN	MEMEN	TKBEN	TRGEN	TRD0EN	PWMOPEN	TRXEN	TRJ0EN

TRXEN	Control of supply of an input clock to timer RX
0	Stops supply of an input clock. • The SFRs used by timer RX cannot be written.
1	Enables supply of an input clock. • The SFRs used by timer RX can be read and written.

Caution 1. Be sure to set TRXEN to 1 before setting timer RX. If TRXEN = 0, writing to the control registers of timer RX is ignored and the value to be read is 00H or 0000H.

Caution 2. To select f_{HOCO} as the count source for timer RX, set f_{CLK} to f_{IH} before setting the TRXEN bit of the PER2 register. To change f_{CLK} to a clock other than f_{IH}, clear the TRXEN bit of the PER2 register before making the change.

14.3.2 Peripheral reset control register 2 (PRR2)

The PRR2 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. To place timer RX in the reset state, set bit 1 (TRXRES) of this register to 1. The PRR2 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 14 - 3 Format of Peripheral Reset Control Register 2 (PRR2)

Address: F00FDH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PRR2	FAARES	MEMRES	TKBRES	TRGRES	TRD0RES	PWMOPRES	TRXRES	TRJ0RES

TRXRES	Control resetting of timer RX
0	Timer RX is released from the reset state.
1	Timer RX is in the reset state.

14.3.3 Timer RX counter (TRX)

Figure 14 - 4 Format of Timer RX Counter (TRX)

Address: F0350H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRX																

—	Function
Bits 15 to 0	These bits count up cycles of the source signal for counting. If counting overflows, the TRXOVF bit of the TRXSR register is set to 1.

Caution If FRQSEL4 of the option byte (000C2H/010C2H) is 1 and TRXEN of the PER2 register is 0, the value of this register following a reset is undefined. If it is necessary to read the default value, set f_{CLK} to f_{IH} and set TRXEN to 1 before reading the value.

14.3.4 Timer RX count buffer counter (TRXBUF)

Figure 14 - 5 Format of Timer RX Count Buffer Counter (TRXBUF)

Address: F0352H
After reset: 0000H
R/W: R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRXBUF																

—	Function
Bits 15 to 0	Buffer register of the timer RX counter The value of the timer RX counter is transferred to the buffer register when an interrupt trigger is generated from a comparator selected from among 0 to 3.

Caution If FRQSEL4 of the option byte (000C2H/010C2H) is 1 and TRXEN of the PER2 register is 0, the value of this register following a reset is undefined. If it is necessary to read the default value, set f_{CLK} to f_{IH} and set TRXEN to 1 before reading the value.

14.3.5 Timer RX function control register 1 (TRXCR1)

Figure 14 - 6 Format of Timer RX Function Control Register 1 (TRXCR1)

Address: F0354H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TRXCR1	TCK2	TCK1	TCK0	START_MD	TRIG_MD_SW	TRIG_MD_HW	TRD_TRIG	OVIE

TCK2	TCK1	TCK0	Selection of the source for counting**Note 1**
0	0	0	f_{CLK}, f_{PLL}, f_{HOCO} **Notes 2, 4**
0	0	1	$f_{CLK}/2$, $f_{PLL}/2$ **Notes 3, 4**
0	1	0	$f_{CLK}/4$, $f_{PLL}/4$ **Notes 3, 4**
0	1	1	$f_{CLK}/8$, $f_{PLL}/8$ **Notes 3, 4**
1	0	0	$f_{CLK}/32$, $f_{PLL}/32$ **Notes 3, 4**
Other than above			Setting prohibited

START_MD	Selects count start source
0	Counting is triggered by software.
1	Counting is triggered by a signal from timer RD2.
Counting is started by setting the TSTART bit in the TRXCR2 register to 1 when the setting of this bit is 1.	

TRIG_MD_SW	Signal for enabling timer RX reset by software**Note 5**
0	Disables resetting of the timer RX counter by software.
1	Enables resetting of the timer RX counter by software.

TRIG_MD_HW	Selects operation when counting is triggered by a signal from timer RD2**Note 6**
0	Starts counting after resetting the timer RX counter.
1	Starts counting by the timer RX counter.
Selects the timer RX operation when counting is triggered by a signal from timer RD2.	

TRD_TRIG	Selects a hardware start trigger from timer RD2**Note 6**
0	Counting is triggered by count start of counter 0 in timer RD2 (TRDSTR.TSTART0 bit being set to 1).
1	Counting is triggered by count start of counter 1 in timer RD2 (TRDSTR.TSTART1 bit being set to 1).

OVIE	Enables overflow interrupt
0	Disables the interrupt generated when the counter of timer RX overflows.
1	Enables the interrupt generated when the counter of timer RX overflows.

(Notes and Caution are listed on the next page.)

Note 1. When timer RX operates in coordination with timer RD2, select the count source of timer RX to have the same frequency as the count source of timer RD2.

Note 2. f_{CLK} is selected when FRQSEL4 = 0 and f_{HOCO} is selected when FRQSEL4 = 1 in the user option byte (000C2H/010C2H). When selecting f_{HOCO} as the count source for timer RD2, set f_{CLK} to f_{IH} before setting bit 3 (TRD0EN) in peripheral enable register 2 (PER2). When changing f_{CLK} to a clock other than f_{IH}, clear bit 3 (TRD0EN) in peripheral enable register 2 (PER2) before changing.

Note 3. Do not set this value when FRQSEL4 = 1 in the user option byte (000C2H/010C2H).

Note 4. f_{PLL} is supplied to timer RX when the setting of DSCON in the PLL control register (DSCCTL) is 1.

Note 5. The setting of this bit has no effect when the setting of the START_MD bit is 1.

Note 6. The setting of this bit has no effect when the setting of the START_MD bit is 0.

Caution If FRQSEL4 of the option byte (000C2H/010C2H) is 1 and TRXEN of the PER2 register is 0, the value of this register following a reset is undefined. If it is necessary to read the default value, set f_{CLK} to f_{IH} and set TRXEN to 1 before reading the value.

14.3.6 Timer RX function control register 2 (TRXCR2)

Figure 14 - 7 Format of Timer RX Function Control Register 2 (TRXCR2)

Address: F0355H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TRXCR2	0	0	0	0	0	CMP1_TCR1	CMP1_TCR0	TSTART

CMP1_TCR1	CMP1_TCR0	Selects operation when a trigger is generated from a comparator
0	0	Counting by the timer RX counter is stopped.
0	1	The value of the timer RX counter is transferred to the timer RX count buffer counter. Counting by the timer RX counter is continued.
1	0	The value of the timer RX counter is reset to 0000H and counting is continued.
1	1	The value of the timer RX counter is transferred to the timer RX count buffer counter. The value of the timer RX counter is reset to 0000H and counting is continued.

TSTART	Controls start of operation of timer RX**Note**
0	Stops counting by the timer RX counter.
1	Starts counting by the timer RX counter.

Note When a stop signal from a comparator conflicts with manipulation of the TSTART bit, the stop signal from the comparator takes precedence.

Caution **If FRQSEL4 of the option byte (000C2H/010C2H) is 1 and TRXEN of the PER2 register is 0, the value of this register following a reset is undefined. If it is necessary to read the default value, set f_{CLK} to f_{IH} and set TRXEN to 1 before reading the value.**

14.3.7 Timer RX status register (TRXSR)

Figure 14 - 8 Format of Timer RX Status Register (TRXSR)

Address: F0356H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	<1>	<0>
TRXSR	0	0	0	0	0	0	TRXSB	TRXOVF

TRXSB	Timer RX counter status flag[Note 1]
0	Counting is stopped.
1	Counting is in progress.

TRXOVF	Overflow state of timer RX counter[Notes 2, 3]
0	Overflow has not occurred.
1	Overflow has occurred.

Note 1. Only reading is enabled. Writing is disabled.

Note 2. If 0 is written to the TRXOVF bit, the TRXOVF bit becomes 0. However, even though 1 is written to the TRXOVF bit, the bit value does not change.

Note 3. If the timer RX counter overflows and 0 is written to TRXOVF at the same time, the overflow takes precedence.

Caution If FRQSEL4 of the option byte (000C2H/010C2H) is 1 and TRXEN of the PER2 register is 0, the value of this register following a reset is undefined. If it is necessary to read the default value, set f_{CLK} to f_{IH} and set TRXEN to 1 before reading the value.

14.4 Operation of Timer RX

Counting by timer RX can be triggered by a signal from timer RD2 and can be stopped by a signal from a comparator selected from among 0 to 3.

14.4.1 Count source

The count source of timer RX is selected by the option byte and frequency divider of timer RX.

- If FRQSEL4 of the option byte (000C2H/010C2H) is set to 1 and the high-speed on-chip oscillator clock (f_{HOCO}) is selected for the CPU/peripheral hardware clock frequency (f_{CLK}), the count source of timer RX is the high-speed on-chip oscillator clock (f_{HOCO}).
- If FRQSEL4 of the option byte (000C2H/010C2H) is cleared to 0 or the high-speed system clock (f_{MX}) is selected for the CPU/peripheral hardware clock frequency (f_{CLK}), the count source of timer RX is the CPU/peripheral hardware clock frequency (f_{CLK}).
- If the PLL clock is selected for the main system clock (f_{MAIN}) in the main clock control register (MCKC), the count source of timer RX is the CPU/peripheral hardware clock frequency (f_{CLK}).
- A frequency selected with the TRXCR1 register is used.
  When counting by the timer RX counter is started by a signal from timer RD2, select the count source of timer RX to have the same frequency as the count source of timer RD2.

14.4.2 Starting timer RX counting

Counting by timer RX can be started by a trigger from timer RD2 or software.

14.4.2.1 Setting and operation when trigger from timer RD2 is selected

1. Setting procedure for resetting the timer RX count and starting counting (TRXCR1.TRIG_MD_HW = 0)
    1. Set TRXCR1.START_MD to 1 to select a trigger from timer RD2 as a count start source.
    2. Set TRXCR1.TRIG_MD_HW to 0 to start counting after resetting the timer RX counter.
    3. Set TRXCR1.TRD_TRIG to 1 or 0 to select a hardware start trigger from timer RD2.
    4. Set TRXCR2.TSTART to 1 to start counting by timer RX.

Figure 14 - 9 Operation Example of Resetting the Timer RX Count and Starting Counting (TRXCR1.TRIG_MD_HW = 0) When Timer RD2 Is Used in Complementary PWM Mode

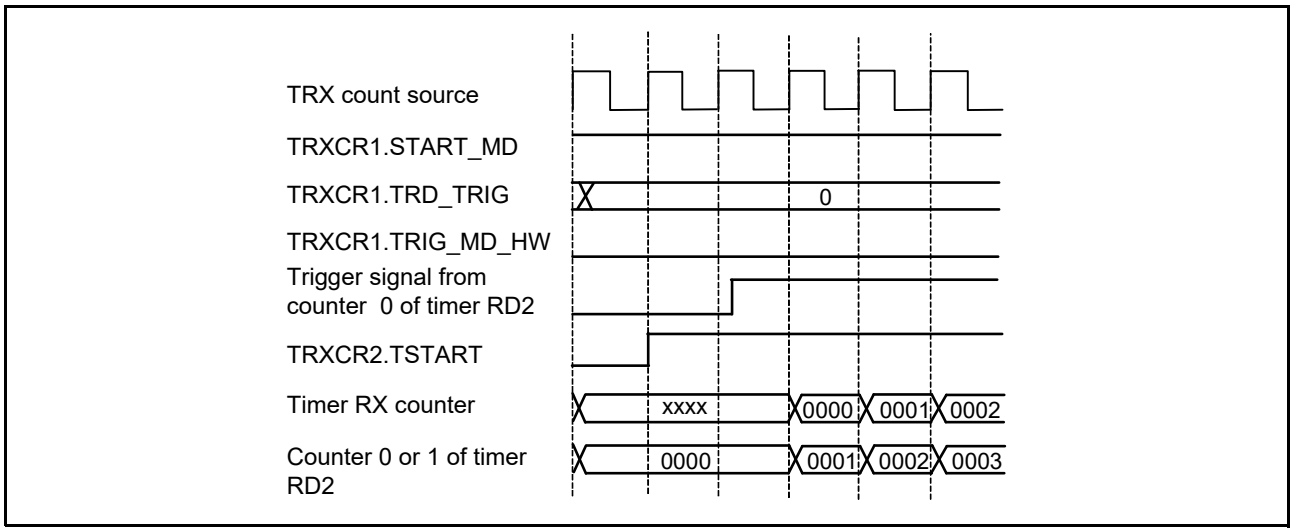
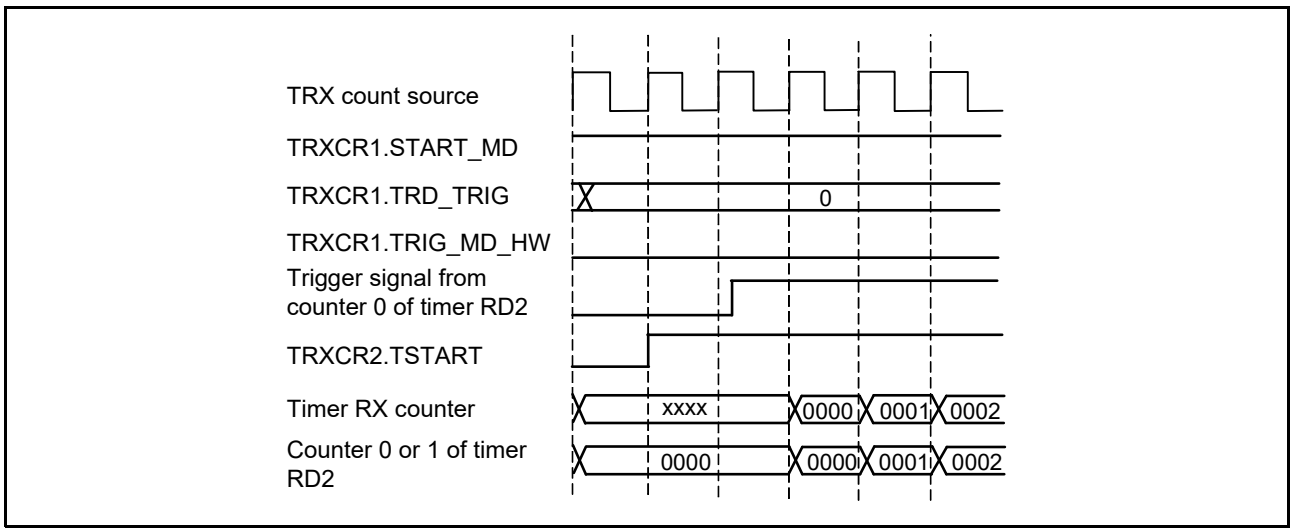
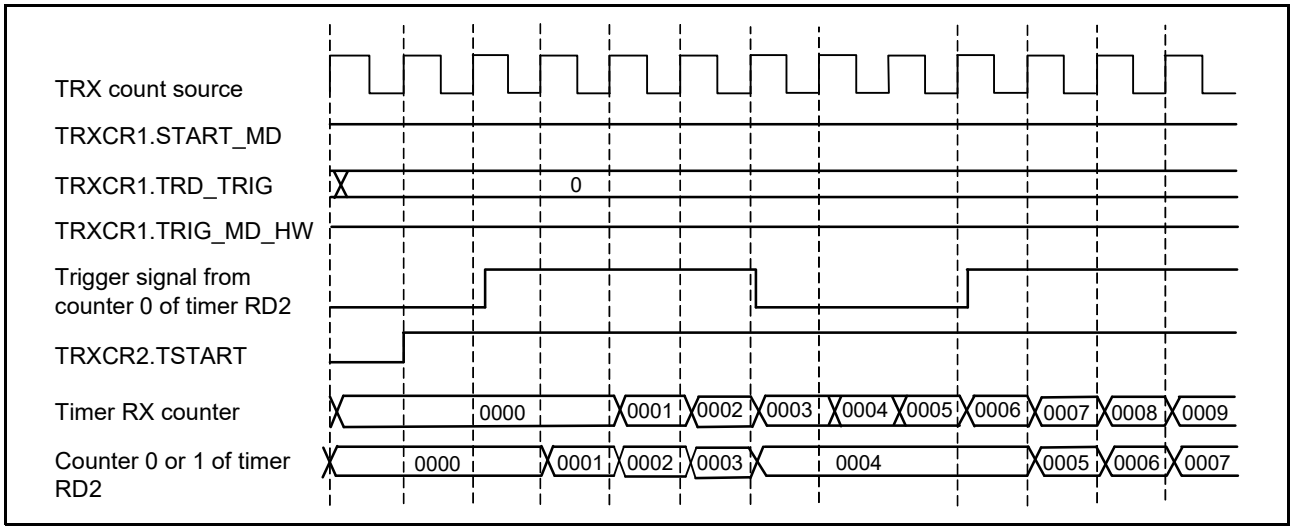


Figure 14 - 10 Operation Example of Resetting the Timer RX Count and Starting Counting (TRXCR1.TRIG_MD_HW = 0) When Timer RD2 Is Used in Extended Complementary PWM Mode or in Timer-KB PWM Output Gating Mode



2. Setting procedure for starting timer RX counting (TRXCR1.TRIG_MD_HW = 1)
    1. Set TRXCR1.START_MD to 1 to select a trigger from timer RD2 as a count start source.
    2. Set TRXCR1.TRIG_MD_HW to 1 to start counting by the timer RX counter when counting is triggered by a signal from timer RD2.
    3. Set TRXCR1.TRD_TRIG to 1 or 0 to select a hardware start trigger from timer RD2.
    4. Set TRXCR2.TSTART to 1 to start counting by timer RX.

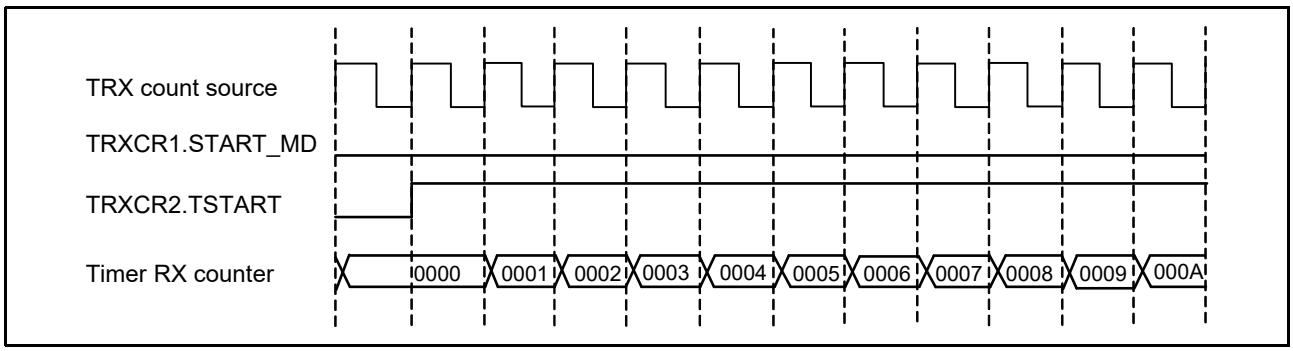
Figure 14 - 11 Operation Example of Starting Timer RX Counting (TRXCR1.TRIG_MD_HW = 1)



14.4.2.2 Setting and operation when software trigger is selected

1. Set TRXCR1.START_MD to 0 to select software as a count start source.
2. Set TRXCR2.TSTART to 1 to start counting by timer RX.

Figure 14 - 12 Operation Example of Starting Timer RX Counting by Software



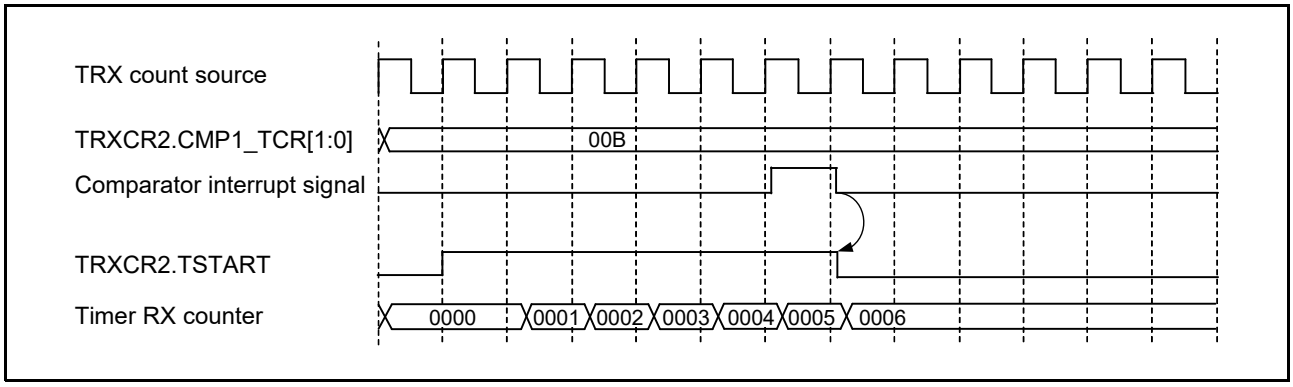
14.4.3 Stopping timer RX counting

The operation of timer RX, which has been started by hardware or software, can be stopped by a trigger selected from among comparators 0 to 3 or software.

14.4.3.1 Setting and operation when trigger from a comparator is selected

1. Set TRXCR2.CMP1_TCR[1:0] to 00B to stop the timer RX counter when a trigger is generated from a comparator.
2. Set TRXCR2.TSTART to 1 to start the timer RX counter.

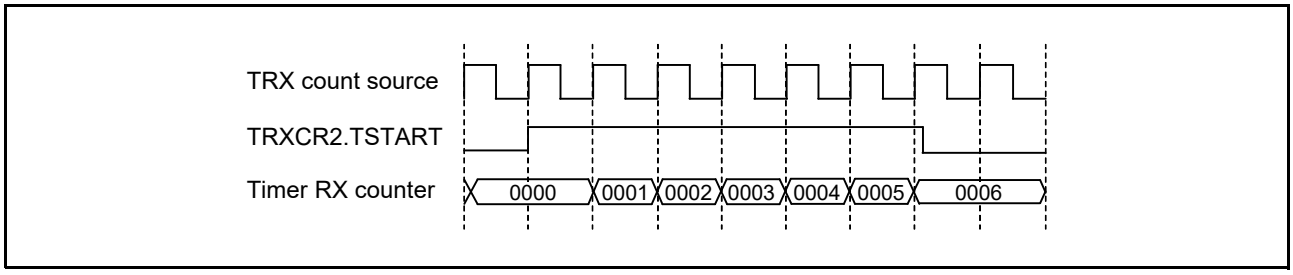
Figure 14 - 13 Operation Example of Stopping Timer RX Counting by a Trigger from a Comparator



14.4.3.2 Setting and operation when software trigger is selected

1. Set the TRXCR2.TSTART bit to 1 to start timer RX counting.
2. Write 0 to the TRXCR2.TSTART bit by software to stop timer RX counting.

Figure 14 - 14 Operation Example When the Software Trigger Is Selected

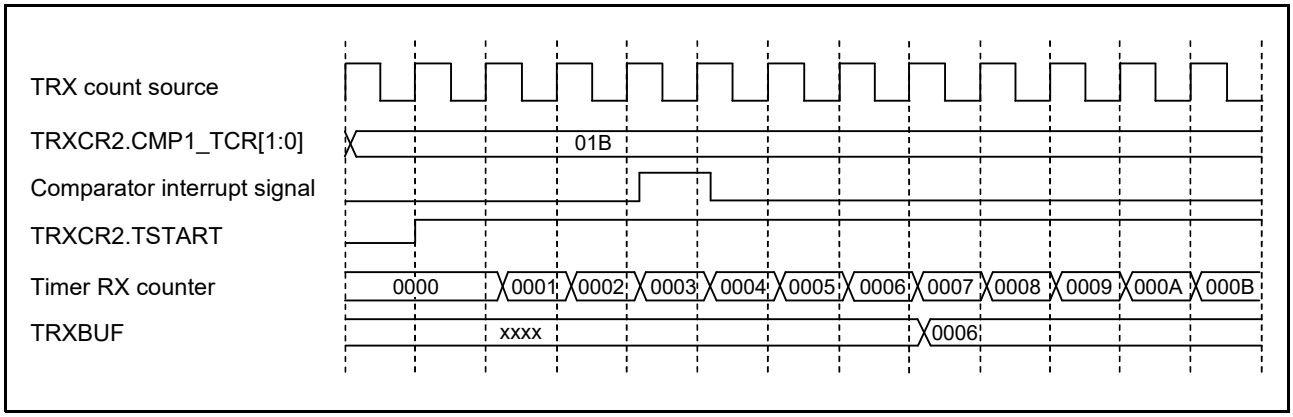


14.4.4 Input capture operation

An interrupt signal from a comparator during operation of timer RX causes the timer to behave differently.

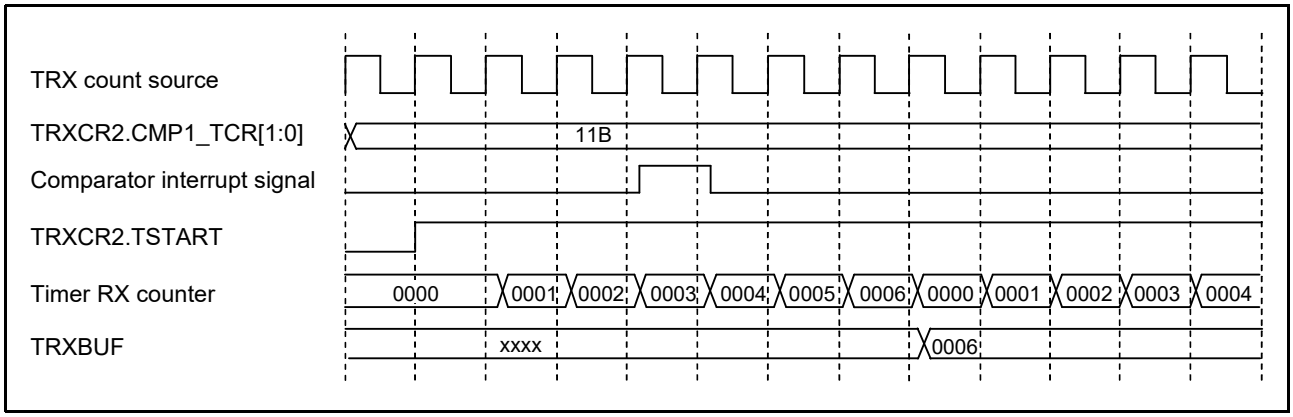
1. Case 1:
   The value of the timer RX counter is transferred to the count buffer by the setting of TRXCR2.CMP1_TCR[1:0] = 01B.
   1. Set TRXCR2.CMP1_TCR[1:0] to 01B to transfer the counter value to the buffer.
   2. Set TRXCR2.TSTART to 1 to start the timer RX counter.

Figure 14 - 15 Operation Example of Input Capture



2. Case 2:
   The value of the timer RX counter is transferred to the count buffer and the value of the timer RX counter is reset by the setting of TRXCR2.CMP1_TCR[1:0] = 11B.
   1. Set TRXCR2.CMP1_TCR[1:0] to 11B to select the input capture function and to reset the value of the timer RX counter.
   2. Set TRXCR2.TSTART to 1 to start the timer RX counter.

Figure 14 - 16 Operation Example of Input Capture (to Reset Counter Value at the Same Time)

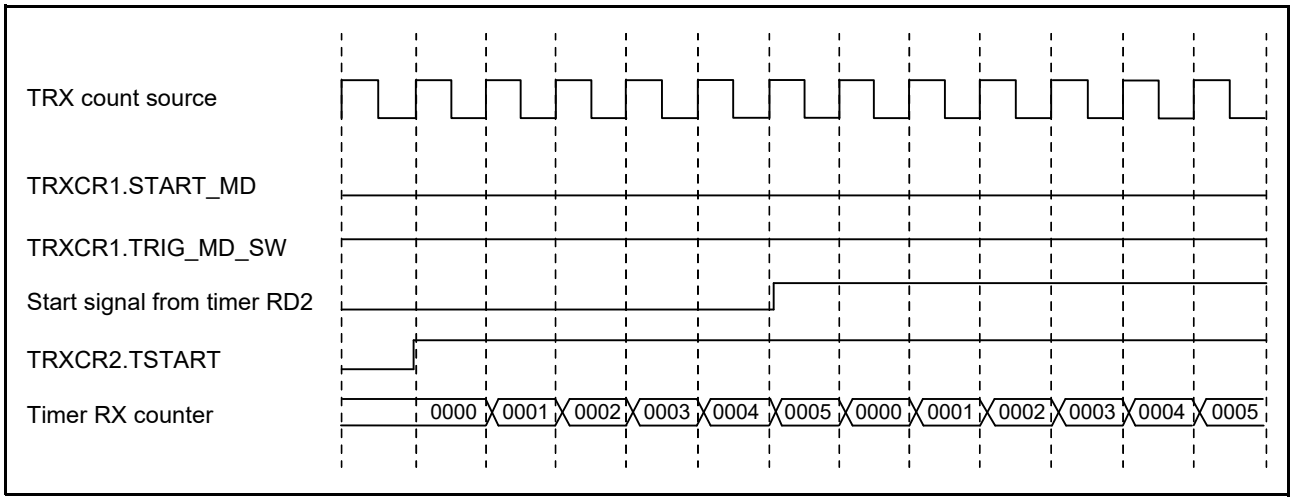


14.4.5 Resetting timer RX count

Resetting the count register can be controlled by a trigger from timer RD2 or a comparator if starting of timer RX counting has been selected by software.

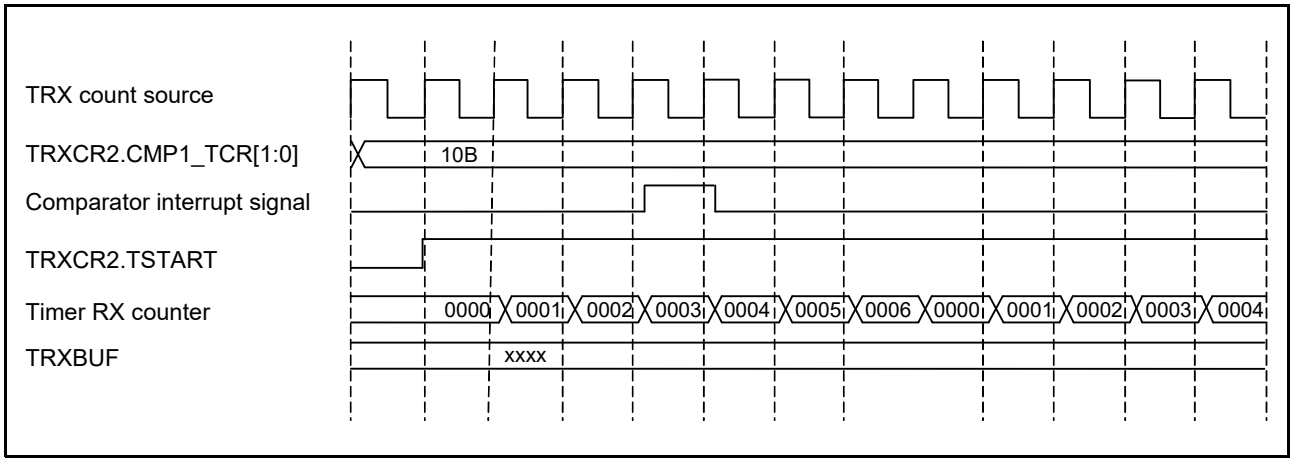
1. Case 1:
   The value of the timer RX counter is reset by a trigger from timer RD2 when TRXCR1.TRIG_MD_SW = 1 and starting of timer RX counting has been selected by software.
   1. Set TRXCR1.START_MD to 0 to select software as a count start source.
   2. Set TRXCR1.TRIG_MD_SW to 1 to enable software to reset counting.
   3. Set TRXCR2.TSTART to 1 to start the timer RX counter.

Figure 14 - 17 Operation Example of Resetting the Counter Value by a Trigger from Timer RD2



2. Case 2:
   The value of the timer RX counter is reset by a trigger from a comparator by the setting of TRXCR2.CMP1_TCR[1:0] = 10B.
   1. Set TRXCR2.CMP1_TCR[1:0] to 10B to reset the counter value and to continue counting.
   2. Set TRXCR2.TSTART to 1 to start the timer RX counting.

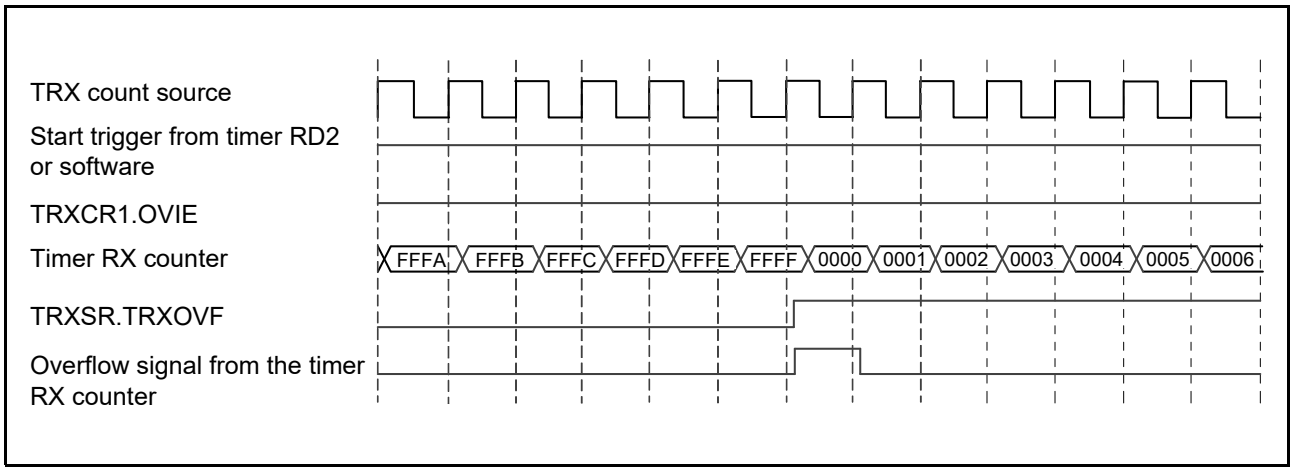
Figure 14 - 18 Operation Example of Resetting the Counter Value by a Trigger from a Comparator



14.4.6 Timer RX interrupt

An overflow interrupt signal can be generated when the counter of timer RX overflows while TRXCR1.OVIE = 1.

Figure 14 - 19 Operation Example of Generating an Overflow Interrupt



14.5 Usage Notes

14.5.1 SFR read/write access

To set timer RX, be sure to set the TRXEN bit of the PER2 register to 1 first. If this bit is 0, writing to the control registers of timer RX is ignored and only the initial values are read.
While clock supply is stopped, the timer RX counter and TRXBUF counter cannot be written to. The other registers related to timer RX can be read from or written to.

Caution Writing to the following registers is prohibited during the counting operation.

- **TRXCR1**
- **TRXCR2**

14.5.2 Overflow interrupt

When the value of the timer RX counter is FFFFH, if the counter value is reset by an external trigger signal before the counter value could be set to 0000H by an overflow, an overflow interrupt is not generated.

14.5.3 Input capture and timer RX count reset operations

The input capture and resetting of timer RX, that are triggered by a signal from timer RD2 or a comparator, still proceed even when the TRXSB bit in the TRXSR register is 0 (stopping the counting).

Caution When the value of the timer RX counter becomes FFFFH while timer RX is counting, an overflow interrupt of timer RX is not generated if the counter value is reset by an external trigger signal.

14.5.4 Procedure for coordinating the operations of timer RX, timer RD2, and a comparator

To use timer RX in coordination with timer RD2 or a comparator, follow the procedure below.

1. Set PER1.PGACMPEN to 1 to start clock supply to the comparators.
2. Select the comparator trigger to be used for timer RX. For details, see **Table 22 - 2 Procedure for Setting the Registers to Control the Comparators**.
3. Enable the comparator interrupt and output. For details, see **Table 22 - 2 Procedure for Setting the Registers to Control the Comparators**.
4. Set PER2.TRXEN to 1 to start clock supply to timer RX.
5. Set the TRXCR1 register.
6. Set the TRXCR2 register.
7. Set TRXCR2.TSTART to 1 to start counting by timer RX.
8. Set TRDSTR.TSTART0 or TRDSTR.TSTART1 to 1 to start counting by timer RD2.

Remark 1. When using timer RX together with timer RD2 and a comparator, select the count source of timer RX to have the same frequency as the count source of timer RD2.

Remark 2. First set the TRXCR1 register and then TRXCR2.TSTART, in that order.

Section 15 16-bit Timers KB30, KB31, and KB32

16-bit timers KB30, KB31, and KB32 are timers that can generate PWM output which is suitable for power sources and lighting control.
The number of channels of the 16-bit timers differs, depending on the product.

Item	20-pin	24- to 64-pin
16-bit timer KB30	✓	✓
16-bit timer KB31	✓	✓
16-bit timer KB32	—	✓

Caution 1. Most of the following descriptions in this section use the 64-pin products as an example.
Caution 2. 16-bit timer KB32 is generated as an external pin for the 20 pins products.

15.1 Functions of 16-bit Timers KB30, KB31, and KB32

16-bit timers KB30, KB31, and KB32 are dedicated PWM output timers and have two outputs each, enabling the generation of up to six PWM outputs. Complementary PWM output can also be generated to control a half-bridge circuit (2 outputs), full-bridge circuit (4 outputs), or 3-phase inverter circuit (6 outputs). Also, by linking with a comparator, INTP20, INTP21, or INTP0, PWM output can be stopped forcibly and timers can be restarted. With these functions, the power factor correction (PFC) circuit and the DC/DC converter can perform sufficiently.
16-bit timers KB30, KB31, and KB32 are provided with the following functions.

1. PWM output
   Each 16-bit timer KB3n is capable of producing two PWM outputs. The duty and the cycle of PWM output can be changed during timer operation. The default level while the timer is stopped and the active level while the timer is operating can be set to the high level or low level. Using these functions enables control over the PFC in the continuous current mode (CCM-PFC) and a half-bridge circuit.

2. A/D conversion start timing signal output
   The A/D conversion start timing signal can be output by using the 16-bit timer KB trigger compare register n (TKBTGCRn). The 16-bit timer KB3n and A/D conversion start timing can be synchronized with this function.

3. Simultaneous start/stop mode
   By setting 16-bit timer KB30 as the master and 16-bit timers KB31 and KB32 as slaves, slave 16-bit timers KB31 and KB32 can be started/stopped at the same time synchronized with the count start/stop timing of 16-bit timer KB30.

4. Simultaneous start/clear mode
   By setting 16-bit timer KB30 as the master and 16-bit timers KB31 and KB32 as slaves, the counting cycles of the master and slave timers can be synchronized. This mode allows the generation of up to six complementary PWM outputs in three phases. This enables control over bridge and 3-phase inverter circuits.

5. PWM output gating function (by interlocked operation of timer RD2)
   Up to 6 timer KB3n outputs can be gate-controlled by using the output of timer RD2 in the timer-KB PWM output gating mode (the TRDIOB1, TRDIOC1, TRDIOD1, TRDIOA1, TRDIOB0, and TRDIOD0 outputs).

6. Timer restart function (by interlocked operation of a comparator, INTPn, and timer RD2)
   Timer output can be restarted directly (not via the CPU) when a trigger source occurs (outputs of comparators 0 to 3, INTP20, INTP21, INTP0, and timer RD2 outputs). By using this function, PFC control in the critical conduction mode can be implemented, for example.

7. Multi-phase function (by interlocked operation of timer RD2)
   The timer output of a 16-bit timer KB3n can be started or restarted by active edges of each output of timer RD2 in the timer-KB PWM output gating mode. By using this function, the multi-phase DC/DC converter can be controlled.

8. Forced output stop function 1 (by interlocked operation of a comparator and INTPn)
   Timer output can be fixed to Hi-Z, high, or low level directly (not via the CPU) and asynchronously with the operating clock f_{KBKC} of the 16-bit timer KB3n and timer RD2 when a trigger source occurs (outputs of comparators 0 to 3, INTP20, INTP21, and INTP0). The forced output stop status can be canceled synchronously with the operating clock f_{KBKC} of the 16-bit timer KB3n and timer RD2 by setting the stop trigger of forced output stop function 1. With these functions, PWM output can be stopped forcibly for over current protection (OCP) and over voltage protection (OVP).

9. Forced output stop function 2 (by interlocked operation of a comparator and INTPn)
   Timer output can be fixed to high or low level directly (not via the CPU) and asynchronously with the operating clock f_{KBKC} of the 16-bit timer KB3n and timer RD2 when a trigger source occurs (outputs of comparators 0 to 3, INTP20, INTP21, and INTP0). The forced output stop status is canceled at the beginning of the next counter cycle after the trigger source occurs or after the trigger source signal changes to the inactive level.
   Furthermore, the use of the fixed off function enables placement of the period over which the active level of the trigger source signal is detected in the forced output stop state. With these functions, multiple operations of the DC/DC converter, including controlling peak current, fixed off output, and protection, can be implemented.

10. PWM output dithering function
    The “set duty + 1” waveform in each 16-period cycle can be output in the range of periods 0 to 15. By using this function, PWM waveform that is 16 times the count clock can be output as the average resolution of 16 cycles of the 16-bit timer KB3n.

11. PWM output smooth start function
    It is possible to make a smooth start that automatically increases duty after PWM output starts until it reaches to the configured duty value.
    It is possible to configure initial duty and duty plus one incremental period.

12. Maximum frequency limit function
    When using the timer restart function, if a trigger occurs earlier than the set maximum frequency, restart can be suspended until the set maximum frequency. With this function, on PFC control in the critical conduction mode, switching in excessively high frequency can be prevented.

13. Interleaved PFC output mode
    With the timer restart function, it is possible to use external factors to automatically alternate restart output between two outputs. It is possible to make interleaved PFC control in the critical conduction mode.

14. Pulse characteristics measurement function
    The characteristics of the pulses in the PWM output signal actually output to a TKBOnp pin can be measured.

Remark The critical conduction mode is a PFC control method that activates a switching FET by detecting zero level of inductor current.

15.2 Configuration of 16-bit Timers KB30, KB31, and KB32

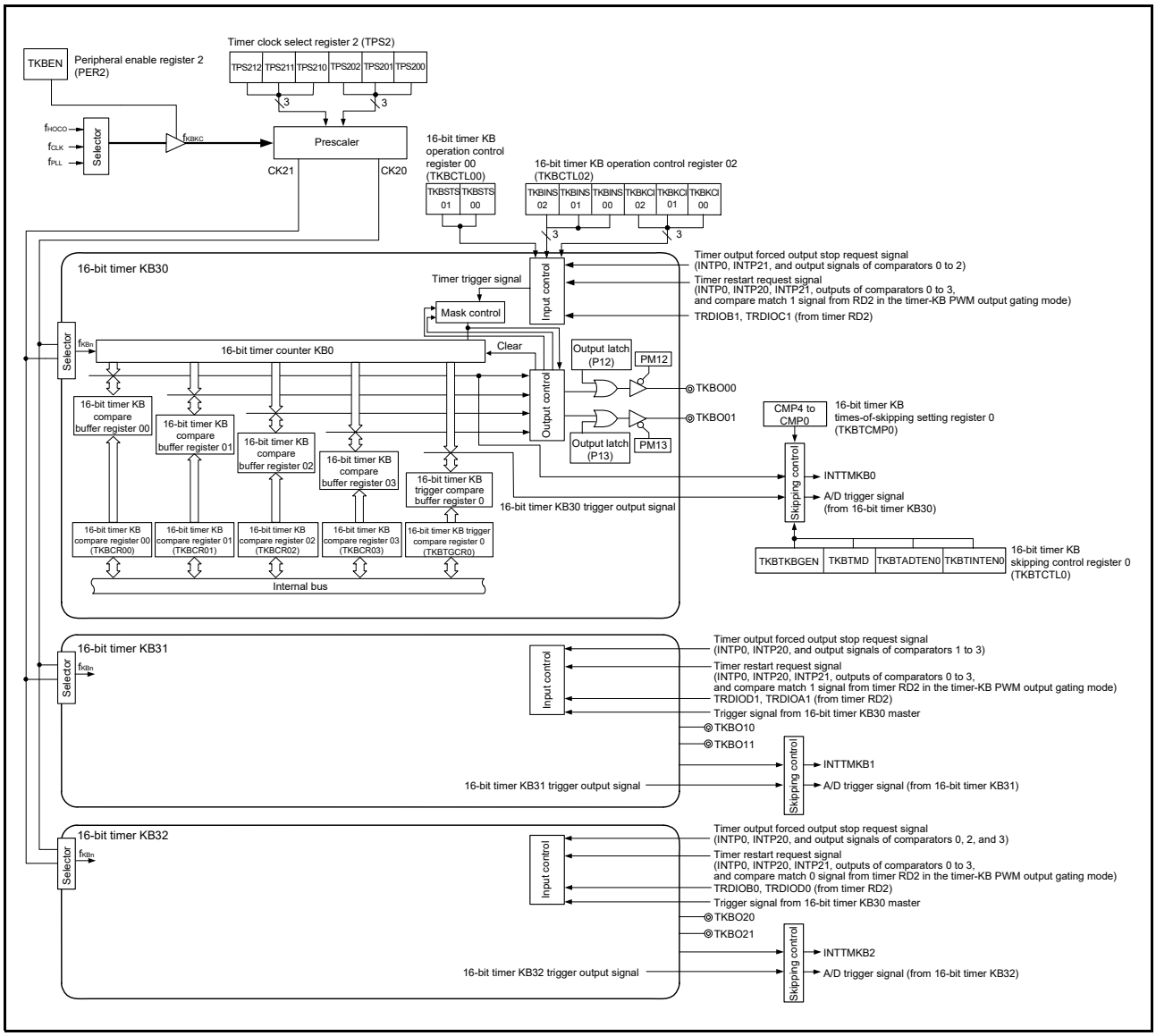
16-bit timers KB30, KB31, and KB32 include the following hardware.

Table 15 - 1 Configuration of 16-bit Timers KB30, KB31, and KB32

Item	Configuration
Timer/counter	16-bit timer counter KBn (TKBCNTn)
Timer output	TKBOn0, TKBOn1
Control registers	• Peripheral enable register 2 (PER2) • Peripheral reset control register 2 (PRR2) • Timer clock select register 2 (TPS2) • 16-bit timer KB compare register nm (TKBCRnm) (n = 0 to 2, m = 0 to 3) • 16-bit timer KB trigger compare register n (TKBTGCRn) (n = 0 to 2) • 16-bit timer KB operation control register n0 (TKBCTLn0) (n = 0 to 2) • 16-bit timer KB operation control register n1 (TKBCTLn1) (n = 0 to 2) • 16-bit timer KB operation control register n2 (TKBCTLn2) (n = 0 to 2) • 16-bit timer KB output control register n0 (TKBIOCn0) (n = 0 to 2) • 16-bit timer KB output control register n1 (TKBIOCn1) (n = 0 to 2) • 16-bit timer KB flag register n (TKBFLGn) (n = 0 to 2) • 16-bit timer KB trigger register n (TKBTRGn) (n = 0 to 2) • 16-bit timer KB flag clear trigger register n (TKBCLRn) (n = 0 to 2) • 16-bit timer KB dithering count register np (TKBDNRnp) (n = 0 to 2; p = 0, 1) • 16-bit timer KB compare 1L & dithering count register n0 (TKBCRLDn0) (n = 0 to 2) • 16-bit timer KB compare 3L & dithering count register n1 (TKBCRLDn1) (n = 0 to 2) • 16-bit timer KB smooth start initial duty register np (TKBSIRnp) (n = 0 to 2; p = 0, 1) • 16-bit timer KB smooth start step width register np (TKBSSRnp) (n = 0 to 2; p = 0, 1) • 16-bit timer KB maximum frequency limit setting register n (TKBMFRn) (n = 0 to 2) • 16-bit timer KB skipping control register n (TKBTCTLn) (n = 0 to 2) • 16-bit timer KB times-of-skipping setting register n (TKBTCMPn) (n = 0 to 2) • External interrupt control register n (INTPCTLn) (n = 0 to 2) • Port mode register xx (PMxx) (xx = 1) • Port register xx (Pxx) (xx = 1) • Port mode control A register xx (PMCAxx) (xx = 1)

Figure 15 - 1 shows a block diagram.

Figure 15 - 1 Block Diagram of 16-bit Timer KB3n



Remark 1. f_{KBKC}: Operating clock of whole 16-bit timer KB3n and timer RD2
f_{KBn}: Count clock of 16-bit timer KB3n

Remark 2. n = 0 to 2

15.3 Registers Controlling 16-bit Timers KB30, KB31, and KB32

16-bit timers KB30, KB31, and KB32 are controlled by the following registers.

- Peripheral enable register 2 (PER2)
- Peripheral reset control register 2 (PRR2)
- Timer clock select register 2 (TPS2)
- 16-bit timer KB compare register nm (TKBCRnm) (n = 0 to 2, m = 0 to 3)
- 16-bit timer KB trigger compare register n (TKBTGCRn) (n = 0 to 2)
- 16-bit timer KB operation control register n0 (TKBCTLn0) (n = 0 to 2)
- 16-bit timer KB operation control register n1 (TKBCTLn1) (n = 0 to 2)
- 16-bit timer KB operation control register n2 (TKBCTLn2) (n = 0 to 2)
- 16-bit timer KB output control register n0 (TKBIOCn0) (n = 0 to 2)
- 16-bit timer KB output control register n1 (TKBIOCn1) (n = 0 to 2)
- 16-bit timer KB flag register n (TKBFLGn) (n = 0 to 2)
- 16-bit timer KB trigger register n (TKBTRGn) (n = 0 to 2)
- 16-bit timer KB flag clear trigger register n (TKBCLRn) (n = 0 to 2)
- 16-bit timer KB dithering count register np (TKBDNRnp) (n = 0 to 2; p = 0, 1)
- 16-bit timer KB compare 1L & dithering count register n0 (TKBCRLDn0) (n = 0 to 2)
- 16-bit timer KB compare 3L & dithering count register n1 (TKBCRLDn1) (n = 0 to 2)
- 16-bit timer KB smooth start initial duty register np (TKBSIRnp) (n = 0 to 2; p = 0, 1)
- 16-bit timer KB smooth start step width register np (TKBSSRnp) (n = 0 to 2; p = 0, 1)
- 16-bit timer KB maximum frequency limit setting register n (TKBMFRn) (n = 0 to 2)
- 16-bit timer KB skipping control register n (TKBTCTLn) (n = 0 to 2)
- 16-bit timer KB times-of-skipping setting register n (TKBTCMPn) (n = 0 to 2)
- External interrupt control register n (INTPCTLn) (n = 0 to 2)
- Port mode registers xx (PMxx) (xx = 1)
- Port registers xx (Pxx) (xx = 1)
- Port mode control A registers xx (PMCAxx) (xx = 1)

15.3.1 Peripheral enable register 2 (PER2)

The PER2 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. To use the 16-bit timer KB3n, set bit 5 (TKBEN) of this register to 1. The PER2 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 15 - 2 Format of Peripheral Enable Register 2 (PER2)

Address: F00FCH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER2	FAAEN	MEMEN	TKBEN	TRGEN	TRD0EN	PWMOPEN	TRXEN	TRJ0EN

TKBEN	Control of supply of an input clock to the 16-bit timer KB3n
0	Stops supply of an input clock. • The SFRs used by the 16-bit timer KB3n cannot be written.
1	Enables supply of an input clock. • The SFRs used by the 16-bit timer KB3n can be read and written.

Caution 1. When setting a 16-bit timer KB3n, make sure that the setting of the TKBEN bit is 1 beforehand. If TKBEN = 0, writing to any of the registers which control the 16-bit timer KB3n is ignored and the value read from any of those registers will be 00H or 0000H (except for timer clock select register 2 (TPS2), external interrupt control registers 0 to 2 (INTPCTL0 to INTPCTL2), port mode register 1 (PM1), and port register 1 (P1)).

Caution 2. When selecting f_{HOCO} as the operating clock of the 16-bit timer KB3n (CK20, CK21), select f_{IH} as f_{CLK} before setting the PER2.TKBEN bit. When changing the clock selected for f_{CLK} from f_{IH} to other clock, clear the PER2.TKBEN bit beforehand.

15.3.2 Peripheral reset control register 2 (PRR2)

The PRR2 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. To place the 16-bit timer KB3n in the reset state, set bit 5 (TKBRES) of this register to 1. The PRR2 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 15 - 3 Format of Peripheral Reset Control Register 2 (PRR2)

Address: F00FDH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PRR2	FAARES	MEMRES	TKBRES	TRGRES	TRD0RES	PWMOPRES	TRXRES	TRJ0RES

TKBRES	Control resetting of the 16-bit timer KB3n
0	The 16-bit timer KB3n is released from the reset state.
1	The 16-bit timer KB3n is in the reset state. • The SFRs for use with the 16-bit timer KB3n are initialized.

15.3.3 Timer clock select register 2 (TPS2)

The TPS2 register is a 8-bit register that is used to select two types of operating clocks (CK20, CK21) that are commonly supplied to 16-bit timers KB30, KB31, KB32, and timer RD2 from external prescaler. CK21 is selected by using bits 6 to 4 of the TPS2 register, and CK20 is selected by using bits 2 to 0. Rewriting of the TPS2 register during timer operation is possible only in the following cases.

- If the TPS20[2:0] bits can be rewritten (n = 0 to 2):
  All timers for which CK20 is selected as the operating clock (TKBCTLn1.TKBCKSn = 0, TRDBCR.GCKS = 0) are stopped (TKBCTLn1.TKBCEn = 0, TRDBCR.GCE = 0).
- If the TPS21[2:0] bits can be rewritten (n = 0 to 2):
  All timers for which CK21 is selected as the operating clock (TKBCTLn1.TKBCKSn = 1, TRDBCR.GCKS = 1) are stopped (TKBCTLn1.TKBCEn = 0, TRDBCR.GCE = 0).

The TPS2 register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 15 - 4 Format of Timer Clock Select Register 2 (TPS2)

Address: F0373H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TPS2	0	TPS212	TPS211	TPS210	0	TPS202	TPS201	TPS200

TPS 2k2	TPS 2k1	TPS 2k0	Selection of operating clock (CK2k)Notes 1, 2 (k = 0, 1)					
				f_{CLK} = 20 MHz	f_{CLK} = 32 MHz	f_{CLK} = 48 MHz	f_{PLL} = 64 MHz	f_{PLL} = 96 MHz	f_{HOCO} = 64 MHz
0	0	0	f_{CLK}, f_{PLL}, f_{HOCO}	20 MHz	32 MHz	48 MHz	64 MHz	96 MHz	64 MHz
0	0	1	$f_{CLK}/2$, $f_{PLL}/2$	10 MHz	16 MHz	24 MHz	32 MHz	48 MHz	Setting prohibited
0	1	0	$f_{CLK}/2^2$, $f_{PLL}/2^2$	5 MHz	8 MHz	12 MHz	16 MHz	24 MHz	
0	1	1	$f_{CLK}/2^3$, $f_{PLL}/2^3$	2.5 MHz	4 MHz	6 MHz	8 MHz	12 MHz	
1	0	0	$f_{CLK}/2^4$, $f_{PLL}/2^4$	1.25 MHz	2 MHz	3 MHz	4 MHz	6 MHz	
1	0	1	$f_{CLK}/2^5$, $f_{PLL}/2^5$	625 kHz	1 MHz	1.5 MHz	2 MHz	3 MHz	
Other than above			Setting prohibited						

Note 1. When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop 16-bit timers KB30, KB31, KB32, and timer RD2 (TKBCTLn1.TKBCEn = 0, TRDBCR.GCE = 0).

Note 2. When the CKSELR bit is 1 in the main clock control register (MCKC), f_{PLL} is supplied. When FRQSEL4 in user option byte (000C2H/010C2H) = 0 and 1, f_{CLK} and f_{HOCO} is selected, respectively. When selecting f_{HOCO} as the operating clock of the 16-bit timer KB3n (CK20, CK21), select f_{IH} as f_{CLK} before setting bit 5 (TKBEN) of the peripheral enable register 2 (PER2). When changing the clock selected for f_{CLK} from f_{IH} to other clock, clear bit 5 (TKBEN) of the peripheral enable register 2 (PER2) beforehand.

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

Remark f_{CLK}: CPU/peripheral hardware clock frequency
f_{PLL}: PLL clock frequency
f_{HOCO}: High-speed on-chip oscillator clock frequency (64 MHz max.)

15.3.4 16-bit timer KB compare register nm (TKBCRnm) (n = 0 to 2, m = 0 to 3)

The TKBCRnm register can be refreshed (writing the same value) and its value can be rewritten while the timer is counting (TKBCTLn1.TKBCEn = 1). When the value of the TKBCRnm register is rewritten while the timer is operating, that value is latched, transferred to the TKBCRnm register at the following timing, and the value of the TKBCRnm register is changed.

- When starting count operation of counter (changing TKBCTLn1.TKBCEn from 0 to 1)
- When a batch overwrite trigger occurs (TKBFLGn.TKBRSFn = 1)

This register can be read or written in 16-bit units. The value of this register following a reset is 0000H.

Figure 15 - 5 Format of 16-bit Timer KB Compare Register nm (TKBCRnm)

Address: F0740H (TKBCR00), F0742H (TKBCR01), F0744H (TKBCR02), F0746H (TKBCR03), F0780H (TKBCR10), F0782H (TKBCR11), F0784H (TKBCR12), F0786H (TKBCR13), F0400H (TKBCR20), F0402H (TKBCR21), F0404H (TKBCR22), F0406H (TKBCR23)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKBCRnm																

Remark n = 0 to 2, m = 0 to 3

15.3.5 16-bit timer KB trigger compare register n (TKBTGCRn) (n = 0 to 2)

The TKBTGCRn register can be refreshed (writing the same value) and its value can be rewritten while the timer is counting (TKBCTLn1.TKBCEn = 1). When the value of the TKBTGCRn register is rewritten while the timer is operating, that value is latched, transferred to the TKBTGCRn register at the following timing, and the value of the TKBTGCRn register is changed.

- When starting count operation of counter (changing TKBCTLn1.TKBCEn from 0 to 1)
- When a batch overwrite trigger occurs (TKBFLGn.TKBRSFn = 1)

Periodic signals from this register can be used as a hardware trigger for A/D conversion. This register can be read or written in 16-bit units. The value of this register following a reset is 0000H.

Figure 15 - 6 Format of 16-bit Timer KB Trigger Compare Register n (TKBTGCRn)

Address: F0748H (TKBTGCR0), F0788H (TKBTGCR1), F0408H (TKBTGCR2)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKBTGCRn																

Remark n = 0 to 2

15.3.6 16-bit timer KB operation control register n0 (TKBCTLn0) (n = 0 to 2)

The TKBCTLn0 register sets the following.

- Output gating function
- Smooth start function
- Dithering function
- Maximum frequency limit function
- Interleaved PFC output mode
- Compare register batch overwrite function set by external trigger
- Timer restart trigger

The TKBCTLn0 register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 15 - 7 Format of 16-bit Timer KB Operation Control Register n0 (TKBCTLn0) (1/2)

Address: F0762H (TKBCTL00), F07A2H (TKBCTL10), F0422H (TKBCTL20)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TKBCTLn0	0	TKBGTEn1	TKBSSEn1	TKBDIEn1	0	TKBGTEn0	TKBSSEn0	TKBDIEn0

7	6	5	4	3	2	1	0
TKBMFEn	0	TKBIRSn1	TKBIRSn0	0	TKBTSEn	TKBSTSn1	TKBSTSn0

TKBGTEnp	Control of TKBOnp PWM output gating function by timer RD2 output
0	Does not use the PWM output gating function.
1	Uses the PWM output gating function.

TKBSSEnp	Control of TKBOnp PWM output smooth start function
0	Does not use the PWM output smooth start function.
1	Uses the PWM output smooth start function.

TKBDIEnp	Control of TKBOnp PWM output dithering function
0	Does not use the PWM output dithering function.
1	Uses the PWM output dithering function.

TKBMFEn	Control of TKBOn0 and TKBOn1 maximum frequency limit function
0	Does not use the maximum frequency limit function.
1	Use the maximum frequency limit function.

TKBIRSn1	TKBIRSn0	Configuration of acceptable range of INTP21 input that immediately outputs TKBOn1 in interleaved PFC output mode
0	0	T/2 to T/2 + T/64
0	1	T/2 to T/2 + T/32
1	0	T/2 to T/2 + T/16
1	1	T/2 to T/2 + T/8

Figure 15 - 7 Format of 16-bit Timer KB Operation Control Register n0 (TKBCTLn0) (2/2)

TKBTSEn	Control of compare register batch overwrite function set by external trigger
0	Does not use the compare register batch overwrite function set by external trigger.
1	Uses the compare register batch overwrite function set by external trigger.

TKBSTSn1	TKBSTSn0	Selection of the restart trigger for the 16-bit timer KB3n
0	0	Does not use trigger input.
0	1	External interrupt signal (INTP20)
1	0	External interrupt signal (INTP21)
1	1	External interrupt signal (INTP0)

Caution 1. During timer operation, overwriting the TKBCTLn0 register is prohibited. However, the TKBCTLn0 register can be refreshed (the same value is written).

Caution 2. Be sure to clear bits 15, 11, 6, and 3 to 0.

Caution 3. For setting of INTP20/INTP21, see Section 22 Comparator Module (CMP).

Remark 1. n = 0 to 2; p = 0, 1

Remark 2. T is the period of the last restart.

15.3.7 16-bit timer KB operation control register n1 (TKBCTLn1) (n = 0 to 2)

The TKBCTLn1 register controls the count operation and sets the count clock of 16-bit timer. The TKBCTLn1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 15 - 8 Format of 16-bit Timer KB Operation Control Register n1 (TKBCTLn1)

Address: F0769H (TKBCTL01), F07A9H (TKBCTL11), F0429H (TKBCTL21)
After reset: 00H
R/W: R/W

Symbol	<7>	6	5	4	3	2	1	0
TKBCTL01	TKBCE0	0	0	TKBCKS0	TKBSCM0	0	TKBMD01	TKBMD00

	<7>	6	5	4	3	2	1	0
TKBCTLm1	TKBCEm	0	0	TKBCKSm	0	0	TKBMDm1	TKBMDm0

TKBCEn	Operation control of the 16-bit timer KB3n
0	Stops timer operation (counter is set to FFFF).
1	Enables timer count operation.

TKBCKSn	Clock selection for 16-bit timer KB3n
0	CK20 clock selected by TPS20[2:0] bits
1	CK21 clock selected by TPS21[2:0] bits

TKBSCM0	Start operation control of 16-bit timer KB30
0	Operates using clock selected by TKBCKS0 bit
1	Counting begins when the CK20 and CK21 clock edges match. After the operation is started, the clock selected by the TKBCKS0 bit is used for operation. **Caution By setting simultaneous start mode to the slave with TKBSCM0 bit, start timing of the slave and master can be matched.**

TKBMDn1	TKBMDn0	Operating mode selection of the 16-bit timer KB3n
0	0	Standalone mode (for the master)
0	1	Simultaneous start/stop mode (for the slaves)
1	0	Simultaneous start/clear mode (for the slaves)
1	1	Interleaved PFC output mode

Caution 1. During timer operation, overwriting the TKBCTLn1 register is prohibited. However, the TKBCTLn1 register can be refreshed (the same value is written).
Caution 2. In TKBCTL01, be sure to clear bits 6, 5, and 2 to 0.
Caution 3. In TKBCTLm1, be sure to clear bits 6, 5, 3, and 2 to 0.

Remark n = 0 to 2; m = 1, 2

15.3.8 16-bit timer KB operation control register n2 (TKBCTLn2) (n = 0 to 2)

The TKBCTLn2 register is used for extended settings of the timer restart trigger. The following is enabled by using this register.

- Additional settings of the trigger to the setting selected by setting the TKBSTSn1 and TKBSTSn0 bits
- Setting of the masking for the source of the external trigger set with an extended setting

The TKBCTLn2 register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 15 - 9 Format of 16-bit Timer KB Operation Control Register n2 (TKBCTLn2) (1/2)

Address: F076AH (TKBCTL02), F07AAH (TKBCTL12), F042AH (TKBCTL22)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TKBCTLn2	0	0	0	0	0	0	TKBMFMn1	TKBMFMn0

7	6	5	4	3	2	1	0
0	TKBINSn2	TKBINSn1	TKBINSn0	0	TKBKCIn2	TKBKCIn1	TKBKCIn0

TKBMFMn1	Control of the source for masking of the external trigger when the maximum frequency limit function is to be used
0	Active TKBOn1Note is not used as a source for masking.
1	Active TKBOn1Note is used as a source for masking.

TKBMFMn0	Control of the source for masking of the external trigger when the maximum frequency limit function is to be used
0	Active TKBOn0Note is not used as a source for masking.
1	Active TKBOn0Note is used as a source for masking.

TKBINSn2	TKBINSn1	TKBINSn0	Selection of the restart trigger for the 16-bit timer KB3n
0	0	0	Does not use trigger input.
0	1	0	Output signal of comparator 0
0	1	1	Output signal of comparator 1
1	0	0	Output signal of comparator 2
1	0	1	Output signal of comparator 3
Other than above			Setting prohibited

TKBKCI02	TKBKCI01	TKBKCI00	Selection of the restart trigger for 16-bit timer KB30
0	0	0	Does not use trigger input.
0	0	1	Compare match 1 signal in timer-KB PWM output gating mode
0	1	0	TRDIOB1 rising edge
0	1	1	TRDIOB1 falling edge
1	0	0	TRDIOC1 rising edge
1	0	1	TRDIOC1 falling edge
Other than above			Setting prohibited

Figure 15 - 9 Format of 16-bit Timer KB Operation Control Register n2 (TKBCTLn2) (2/2)

TKBKCI12	TKBKCI11	TKBKCI10	Selection of the restart trigger for 16-bit timer KB31
0	0	0	Does not use trigger input.
0	0	1	Compare match 1 signal in timer-KB PWM output gating mode
0	1	0	TRDIOD1 rising edge
0	1	1	TRDIOD1 falling edge
1	0	0	TRDIOA1 rising edge
1	0	1	TRDIOA1 falling edge
Other than above			Setting prohibited

TKBKCI22	TKBKCI21	TKBKCI20	Selection of the restart trigger for 16-bit timer KB32
0	0	0	Does not use trigger input.
0	0	1	Compare match 0 signal in timer-KB PWM output gating mode
0	1	0	TRDIOB0 rising edge
0	1	1	TRDIOB0 falling edge
1	0	0	TRDIOD0 rising edge
1	0	1	TRDIOD0 falling edge
Other than above			Setting prohibited

Note An internal signal, which is a TKBOnp output after synchronization with f_{KBKC}, can be a source for masking.

Caution 1. During timer operation, overwriting the TKBCTLn2 register is prohibited. However, the TKBCTLn2 register can be refreshed (the same value is written).

Caution 2. Be sure to clear bits 15 to 10, 7, and 3 to 0.

Caution 3. In the interleaved PFC output mode, the settings of the TKBINSn[2:0] bits and the TKBKCIn[2:0] bits are ignored and an external interrupt signal (INTP20) is selected as the restart trigger.

Remark n = 0 to 2; p = 0, 1

15.3.9 16-bit timer KB output control register n0 (TKBIOCn0) (n = 0 to 2)

The TKBIOCn0 register sets the default level/active level in 16-bit timer KB3n output (TKBOnp). The TKBIOCn0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 15 - 10 Format of 16-bit Timer KB Output Control Register n0 (TKBIOCn0)

Address: F0766H (TKBIOC00), F07A6H (TKBIOC10), F0426H (TKBIOC20)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	<3>	<2>	<1>	<0>
TKBIOCn0	0	0	0	0	TKBTOLn1	TKBTOLn0	TKBTODn1	TKBTODn0

TKBTOLnp	Active level setting of timer output TKBOnp
0	High level
1	Low level

TKBTODnp	Default level setting of timer output TKBOnp
0	Low level
1	High level

Caution 1. During timer operation, overwriting the TKBIOCn0 register is prohibited. However, the TKBIOCn0 register can be refreshed (the same value is written).

Caution 2. Be sure to clear bits 7 to 4 to 0.

Caution 3. Actual TKBOnp pin output is set not only by TKBOnp output but by the port mode registers xx (PMxx) and port registers xx (Pxx) for the shared ports.

Remark n = 0 to 2; p = 0, 1

15.3.10 16-bit timer KB output control register n1 (TKBIOCn1) (n = 0 to 2)

The TKBIOCn1 register controls disable/enable timer control in 16-bit timer KB3n output (TKBOnp). The TKBIOCn1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 15 - 11 Format of 16-bit Timer KB Output Control Register n1 (TKBIOCn1)

Address: F0768H (TKBIOC01), F07A8H (TKBIOC11), F0428H (TKBIOC21)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	<1>	<0>
TKBIOCn1	0	0	0	0	0	0	TKBTOEn1	TKBTOEn0

TKBTOEnp	Timer output TKBOnp output enable/disable
0	Disables timer output (low-level output when TKBTODnp = 0, and high-level output when TKBTODnp = 1.)
1	Enables timer output

Caution 1. The TKBIOCn1 register can be overwritten while the timer is operating.
Caution 2. Be sure to clear bits 7 to 2 to 0.
Caution 3. Actual TKBOnp pin output is set not only by TKBOnp output but by the port mode registers xx (PMxx) and port registers xx (Pxx) for the shared ports.

Remark n = 0 to 2; p = 0, 1

15.3.11 16-bit timer KB flag register n (TKBFLGn) (n = 0 to 2)

The TKBFLGn register indicates status flags for 16-bit timer KB3n. The TKBFLGn register can be read by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 15 - 12 Format of 16-bit Timer KB Flag Register n (TKBFLGn)

Address: F0753H (TKBFLG0), F0793H (TKBFLG1), F0413H (TKBFLG2)
After reset: 00H
R/W: R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
TKBFLGn	TKBSSFn1	TKBSSFn0	TKBSEFn1	TKBSEFn0	TKBIRFn	TKBIEFn	TKBMFFn	TKBRSFn

TKBSSFnp	Status flag for PWM output smooth start function of TKBOnp pin
0	During stop in PWM output smooth start function
1	Executing in PWM output smooth start function

TKBSEFnp	Error flag for PWM output smooth start function of TKBOnp pin
0	No error, or completion of clearing by TKBCLSEnp
1	Error (TKBTRGn.TKBRDTn = 1 occurred during PWM output smooth start execute (TKBSSFnp = 1))

TKBIRFn	Undetected INTP21 trigger error flag for interleaved PFC mode
0	No error, or completion of clearing by TKBCLIRn
1	Error (Undetected INTP21 trigger is in judgment range set by 0 to T/2 and the TKBIRSn[1:0] bits

TKBIEFn	Multiplex detection INTP21 trigger error flag for interleaved PFC mode
0	No error, or completion of clearing by TKBCLIEn
1	Error (Another INTP21 trigger was detected during the TKBOn1 active output)

TKBMFFn	Status flag for maximum frequency limit function
0	Maximum frequency limit function is not occurred, or completion of clearing by the TKBCLRn.TKBCLMFn bit
1	Maximum frequency limit function is occurred

TKBRSFn	Pending status flag for batch overwrite trigger
0	Batch overwrite enabled status or completion of batch overwrite caused by to batch overwrite trigger
1	On hold (waiting for completion) status of batch overwrite due to writing on batch overwrite trigger bit (TKBTRGn.TKBRDTn).

Remark 1. n = 0 to 2; p = 0, 1
Remark 2. T is the period of the last restart.

15.3.12 16-bit timer KB trigger register n (TKBTRGn) (n = 0 to 2)

The TKBTRGn register is a trigger register used for batch overwriting of the compare register for 16-bit timer KB3n. The TKBTRGn register can be written by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 15 - 13 Format of 16-bit Timer KB Trigger Register n (TKBTRGn)

Address: F0752H (TKBTRG0), F0792H (TKBTRG1), F0412H (TKBTRG2)
After reset: 00H
R/W: W

Symbol	7	6	5	4	3	2	1	<0>
TKBTRGn	0	0	0	0	0	0	0	TKBRDTn

TKBRDTn	Trigger for batch overwrite request of compare register
0	Invalid setting
1	Batch overwrite request of compare register

Remark n = 0 to 2

15.3.13 16-bit timer KB flag clear trigger register n (TKBCLRn) (n = 0 to 2)

The TKBCLRn register is used to clear flags in the 16-bit timer KB flag register n (TKBFLGn). The TKBCLRn register can be written by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 15 - 14 Format of 16-bit Timer KB Flag Clear Trigger Register n (TKBCLRn)

Address: F0767H (TKBCLR0), F07A7H (TKBCLR1), F0427H (TKBCLR2)
After reset: 00H
R/W: W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	0
TKBCLRn	0	0	TKBCLSEn1	TKBCLSEn0	TKBCLIRn	TKBCLIEn	TKBCLMFn	0

TKBCLSEnp	Trigger for clearing error flag for PWM output smooth start function of TKBOnp pin
0	Invalid setting
1	Clear the TKBSEFnp flag to 0.

TKBCLIRn	Trigger for clearing undetected INTP21 trigger error flag for interleaved PFC mode
0	Invalid setting
1	Clear the TKBIRFn flag to 0.

TKBCLIEn	Trigger for clearing multiplex detection INTP21 trigger error flag for interleaved PFC mode
0	Invalid setting
1	Clear the TKBIEFn flag to 0.

TKBCLMFn	Trigger for clearing status flag for maximum frequency limit function
0	Invalid setting
1	Clear the TKBMFFn flag to 0.

Remark n = 0 to 2; p = 0, 1

15.3.14 16-bit timer KB dithering count register np (TKBDNRnp) (n = 0 to 2; p = 0, 1)

The TKBDNRnp register is used by the PWM dithering function for TKBOnp output. When the values in this register of the higher 4 bits are N (N = 0H to FH), the active period for N times during each 16-period cycle of PWM output is output to one count clock extended. **Table 15 - 2** shows the relation between the setting of the TKBDNRnp register and the active period for N repetitions of one count clock extended. The TKBDNRnp register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 15 - 15 Format of 16-bit Timer KB Dithering Count Register np (TKBDNRnp)

Address: F074EH (TKBDNR00), F078EH (TKBDNR10), F040EH (TKBDNR20), F0750H (TKBDNR01), F0790H (TKBDNR11), F0410H (TKBDNR21)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TKBDNRnp	TKBDNR3n	TKBDNR2n	TKBDNR1n	TKBDNR0n	0	0	0	0

Caution **Be sure to clear bits 3 to 0 to 0.**

Remark n = 0 to 2; p = 0, 1

Table 15 - 2 16-bit Timer KB Dithering Count Register np (TKBDNRnp) Setting

Repetitions (N) \ Period	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0																
1																
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																
12																
13																
14																
15																

Remark 1. Cell period: Inactive output via settings in the TKBCRn1 and TKBCRn3 registers

Cell period: Inactive output via “settings + 1” in the TKBCRn1 and TKBCRn3 registers

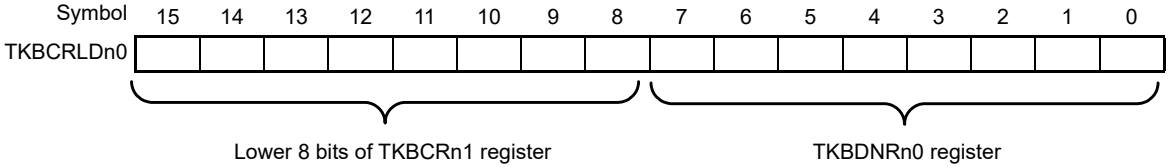
Remark 2. n = 0 to 2; p = 0, 1

15.3.15 16-bit timer KB compare 1L & dithering count register n0 (TKBCRLDn0) (n = 0 to 2)

The TKBCRLDn0 register holds the “lower 8 bits of TKBCRn1 register” values in its higher 8 bits and the “TKBDNRn0 register” values in its lower 8 bits. The TKBCRLDn0 register can be set by a 16-bit memory manipulation instruction.
The value of this register following a reset is 0000H.

Figure 15 - 16 Format of 16-bit Timer KB Compare 1L & Dithering Count Register n0 (TKBCRLDn0)

Address: F0754H (TKBCRLD00), F0794H (TKBCRLD10), F0414H (TKBCRLD20)
After reset: 0000H
R/W: R/W



Caution **Be sure to clear bits 3 to 0 to 0.**

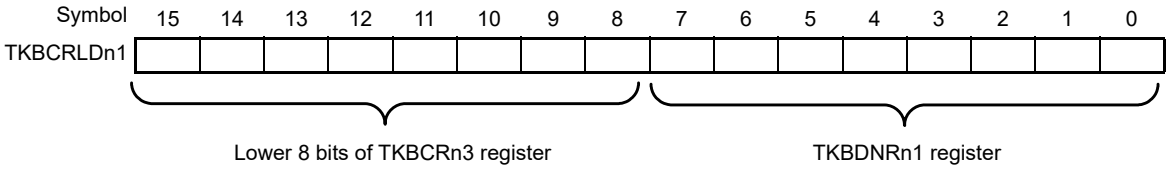
Remark n = 0 to 2

15.3.16 16-bit timer KB compare 3L & dithering count register n1 (TKBCRLDn1) (n = 0 to 2)

The TKBCRLDn1 register holds the “lower 8 bits of TKBCRn3 register” values in its higher 8 bits and the “TKBDNRn1 register” values in its lower 8 bits. The TKBCRLDn1 register can be set by a 16-bit memory manipulation instruction.
The value of this register following a reset is 0000H.

Figure 15 - 17 Format of 16-bit Timer KB Compare 3L & Dithering Count Register n1 (TKBCRLDn1)

Address: F0756H (TKBCRLD01), F0796H (TKBCRLD11), F0416H (TKBCRLD21)
After reset: 0000H
R/W: R/W



Caution **Be sure to clear bits 3 to 0 to 0.**

Remark n = 0 to 2

15.3.17 16-bit timer KB smooth start initial duty register np (TKBSIRnp) (n = 0 to 2; p = 0, 1)

The TKBSIRnp register sets the default duty for the PWM output smooth start function for TKBOnp output. The TKBSIRnp register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 15 - 18 Format of 16-bit Timer KB Smooth Start Initial Duty Register np (TKBSIRnp)

Address: F074AH (TKBSIR00), F078AH (TKBSIR10), F040AH (TKBSIR20), F074CH (TKBSIR01), F078CH (TKBSIR11), F040CH (TKBSIR21)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKBSIRnp																

Remark n = 0 to 2; p = 0, 1

15.3.18 16-bit timer KB smooth start step width register np (TKBSSRnp) (n = 0 to 2; p = 0, 1)

The TKBSSRnp register is used by the PWM output smooth start function for TKBOnp output. When the value of the lower-order four bits of this register is N (N = 0H to FH), output of a PWM waveform with the active output period is continued for N + 1 times by setting the TKBSIRnp register. Afterward, output continues with the (active period + 1 clock) waveform for N + 1 cycles, then with the (active period + 2 clock) waveform for N + 1 cycles, and so on. Finally, when the TKBCRn1 or TKBCRn3 register have the same duty, the PWM output smooth start function is cleared and normal PWM output is set. The TKBSSRnp register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 15 - 19 Format of 16-bit Timer KB Smooth Start Step Width Register np (TKBSSRnp)

Address: F074FH (TKBSSR00), F078FH (TKBSSR10), F040FH (TKBSSR20), F0751H (TKBSSR01), F0791H (TKBSSR11), F0411H (TKBSSR21)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TKBSSRnp	0	0	0	0				

Caution **Be sure to clear bits 7 to 4 to 0.**

Remark n = 0 to 2; p = 0, 1

15.3.19 16-bit timer KB maximum frequency limit setting register n (TKBMFRn) (n = 0 to 2)

The TKBMFRn register sets the minimum period for the timer restart of external trigger. When the value of the 16-bit timer counter KBn (TKBCNTn) is smaller than that of the TKBMFRn register, if trigger input is detected, the trigger is held pending, and TKBCNTn is cleared (restart) after counting to the value set to the TKBMFRn register. The TKBMFRn register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 15 - 20 Format of 16-bit Timer KB Maximum Frequency Limit Setting Register n (TKBMFRn)

Address: F0764H (TKBMFR0), F07A4H (TKBMFR1), F0424H (TKBMFR2)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKBMFRn																

Remark n = 0 to 2

15.3.20 16-bit timer KB skipping control register n (TKBTCTLn) (n = 0 to 2)

The TKBTCTLn register controls the skipping of interrupt signals and the A/D trigger signals. The TKBTCTLn register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 15 - 21 Format of 16-bit Timer KB Skipping Control Register n (TKBTCTLn)

Address: F0490H (TKBTCTL0), F0492H (TKBTCTL1), F0494H (TKBTCTL2)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TKBTCTLn	TKBTKBGENn	0	0	TKBTMDn	0	TKBTADTENn	0	TKBTINTENn

TKBTKBGENn	Enabling or disabling of interlocking with timer RD2 in the timer-KB PWM output gating mode
0	Interlocking is disabled.
1	Interlocking is enabled.

TKBTMDn	Enabling or disabling skipping of the output in the first cycle after the 16-bit timer KB3n starts to operate
0	Output is disabled.
1	Output is enabled.

TKBTADTENn	Enabling or disabling skipping of the A/D trigger signal from the 16-bit timer KB3n
0	Skipping is disabled.
1	Skipping is enabled.

TKBTINTENn	Enabling or disabling skipping of the count end interrupt signal from the 16-bit timer KB3n
0	Skipping is disabled.
1	Skipping is enabled.

Caution 1. Be sure to clear the TKBTMDn bit to 0 when setting the TKBTKBGENn bit to 1.
Caution 2. During timer operation, overwriting the TKBTCTLn register is prohibited. However, the TKBTCTLn register can be refreshed (the same value is written).
Caution 3. Be sure to clear bits 6, 5, 3, and 1 to 0.

Remark n = 0 to 2

15.3.21 16-bit timer KB times-of-skipping setting register n (TKBTCMPn) (n = 0 to 2)

The TKBTCMPn register controls the number of times the interrupt signals and the A/D trigger signals are skipped. The TKBTCMPn register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 15 - 22 Format of 16-bit Timer KB Times-of-Skipping Setting Register n (TKBTCMPn)

Address: F0491H (TKBTCMP0), F0493H (TKBTCMP1), F0495H (TKBTCMP2)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TKBTCMPn	0	0	0	TKBTCMPn[4:0]				

TKBTCMPn[4:0]					Times of skipping
0	0	0	0	0	No skipping
0	0	0	0	1	Skipping once After an interrupt signal or an A/D trigger signal is output, the next signal is not output for one cycle.
⋮					⋮
1	1	1	1	0	Skipping 30 times After an interrupt signal or an A/D trigger signal is output, the next signal is not output for 30 cycles.
1	1	1	1	1	Skipping 31 times After an interrupt signal or an A/D trigger signal is output, the next signal is not output for 31 cycles.

Caution 1. During the operation of 16-bit timer KB30, KB31, or KB32, overwriting the TKBTCMPn register is prohibited.

Caution 2. Be sure to clear bits 7 to 5 to 0.

Remark n = 0 to 2

15.3.22 External interrupt control register n (INTPCTLn) (n = 0 to 2)

The INTPCTLn register controls the external interrupts. The INTPCTLn register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H. The INTPCTL0, INTPCTL1, and INTPCTL2 registers correspond to the INTP20, INTP21, and INTP0 external interrupts, respectively.

Figure 15 - 23 Format of External Interrupt Control Register n (INTPCTLn) (1/2)

Address: F0370H (INTPCTL0), F0371H (INTPCTL1), F0372H (INTPCTL2)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
INTPCTLn	PNFENn	INTFCKn1	INTFCKn0	INTPINVn	INTEGPn	INTEGNn	INTPSTENn	TMRSTENn

TMRSTENn	Enabling or disabling of the timer KB3 functions interlocked for triggering by the INTPx external interrupt
0	Triggering of the timer restart function and forced output stop function of 16-bit timers KB30, KB31, and KB32 by INTPx is disabled.
1	Triggering of the timer restart function and forced output stop function of 16-bit timers KB30, KB31, and KB32 by INTPx is enabled**Note**.

INTPSTENn	Selection of the function of the INTPx external interrupt
0	INTPx pin input is used as the INTPx external interrupt signal.
1	The timer restart signal for 16-bit timers KB30, KB31, and KB32 is used as the INTPx external interrupt signal.

INTEGPn	INTEGNn	Selection of the valid edge or edges of the INTPx external interrupt
0	0	Edge detection disabled (Output of the timer restart signal for 16-bit timers KB30, KB31, and KB32 is disabled and the output signal is fixed to the low level.)
0	1	Falling edge (Output of the timer restart signal for 16-bit timers KB30, KB31, and KB32 is enabled.)
1	0	Rising edge (Output of the timer restart signal for 16-bit timers KB30, KB31, and KB32 is enabled.)
1	1	Both rising and falling edges (Output of the timer restart signal for 16-bit timers KB30, KB31, and KB32 is enabled.)

INTPINVn	Setting of the output inversion of the INTPx external interrupt signal
0	The INTPx signal is not inverted.
1	The INTPx signal is inverted.

Figure 15 - 23 Format of External Interrupt Control Register n (INTPCTLn) (2/2)

INTFCKn1	INTFCKn0	Selection of external interrupt INTPx filter
0	0	No INTPx filter
0	1	INTPx filter enabled, sampling at f_{CLK}, f_{PLL}, or f_{HOCO}
1	0	INTPx filter enabled, sampling at $f_{CLK}/8$, $f_{PLL}/8$, or $f_{HOCO}/8$
1	1	INTPx filter enabled, sampling at TO1

PNFENn	Enabling or disabling the use of the noise filter for the INTPx external interrupt
0	The noise filter is used.
1	The noise filter is not used.

Note When using the timer restart function or forced output stop function 2 of 16-bit timers KB30, KB31, and KB32, clear the ISC0 bit in the ISC register to 0 to use the signal input through the INTP0 pin as an external interrupt signal.

Remark n = 0 to 2; x = 0, 20, 21

15.3.23 Registers for controlling the port functions multiplexed with the pins of 16-bit timers KB30, KB31, and KB32

Set the following registers to control the port functions multiplexed with the outputs of 16-bit timers KB30, KB31, and KB32.

- Port mode registers xx (PMxx)
- Port registers xx (Pxx)
- Port mode control A registers xx (PMCAxx)

For details, see **7.3.1 Port mode registers xx (PMxx)**, **7.3.2 Port registers xx (Pxx)**, and **7.3.7 Port mode control A registers xx (PMCAxx)**.

When the pins multiplexed with TKBO00, TKBO01, TKBO10, TKBO11, TKBO20, and TKBO21 are to be used for outputs of timers, set the port mode register xx (PMxx), port register xx (Pxx), and port mode control A register xx (PMCAxx) bits corresponding to each port to 0.

Remark xx = 1

15.4 Operation of 16-bit Timers KB30, KB31, and KB32

Operation specifications of 16-bit timers KB30, KB31, and KB32 are described below.

- Counter basic operation (see **15.4.1**.)
- Default level and active level (see **15.4.2**.)
- Stop/restart operation (see **15.4.3**.)
- Batch overwrite operation (see **15.4.4**.)

There are 6 different operating modes for 16-bit timers KB30, KB31, and KB32.

- Standalone mode (period controlled by the TKBCRn0 register) (see **15.4.6**.)
- Standalone mode (period controlled by external trigger input) (see **15.4.7**.)
- Simultaneous start/stop mode (period controlled by the TKBCRn0 register) (see **15.4.8**.)
- Simultaneous start/stop mode (period controlled by external trigger input) (see **15.4.8**.)
- Simultaneous start/clear mode (period controlled by master) (see **15.4.9**.)
- Interleaved power factor correction (PFC) output mode (see **15.4.10**.)

Figure 15 - 24 16-bit Timer KB3n Operation Setting Example (Operation Start Flow)

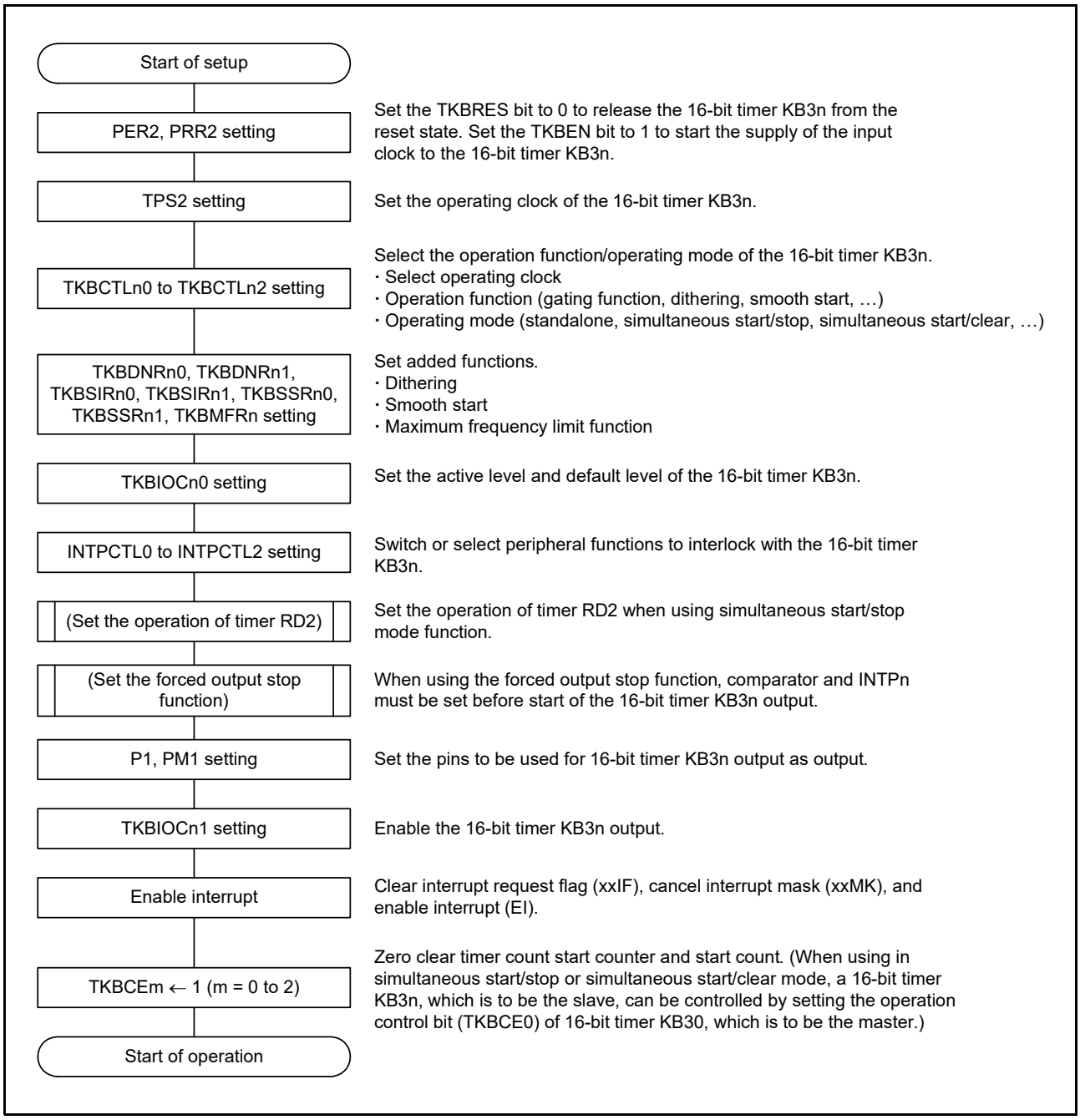


Figure 15 - 25 16-bit Timer KB3n Operation Setting Example (Operation Stop Flow)

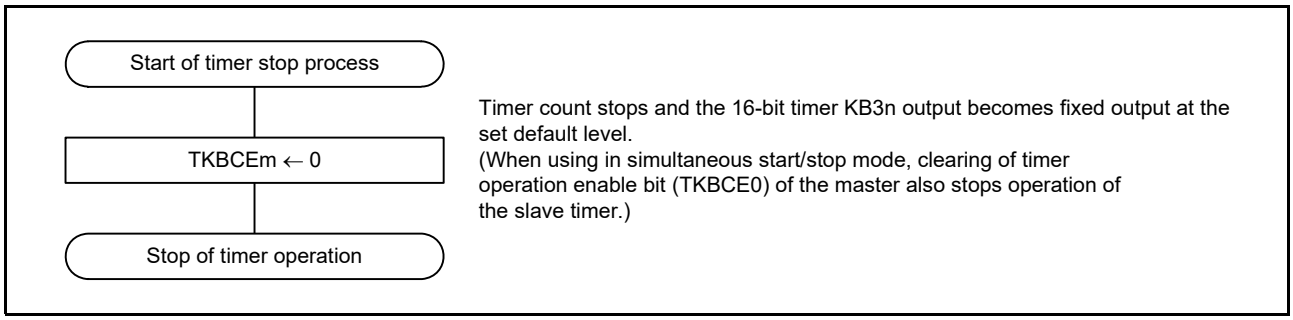
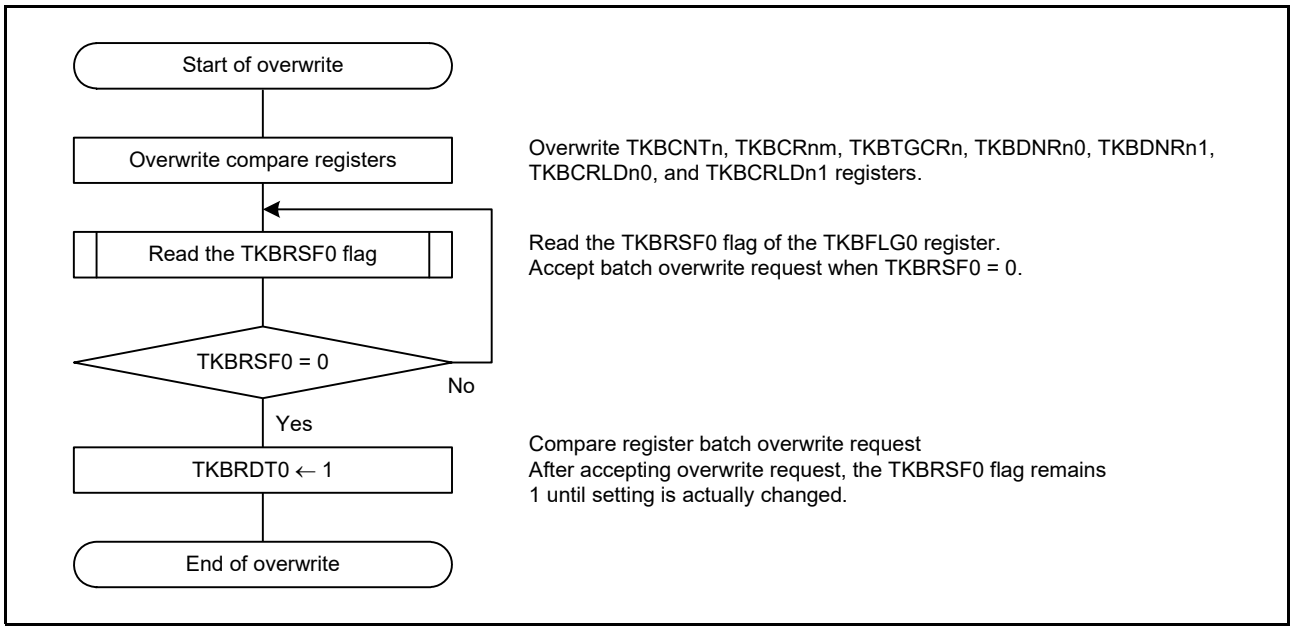


Figure 15 - 26 16-bit Timer KB3n Operation Setting Example (Compare Register Batch Overwrite Flow)



Remark The batch overwrite function is used to change the timer counter operation setting while the 16-bit timer KB3n is operating. The set value is reflected to the operation from the next restart.

15.4.1 Counter basic operation

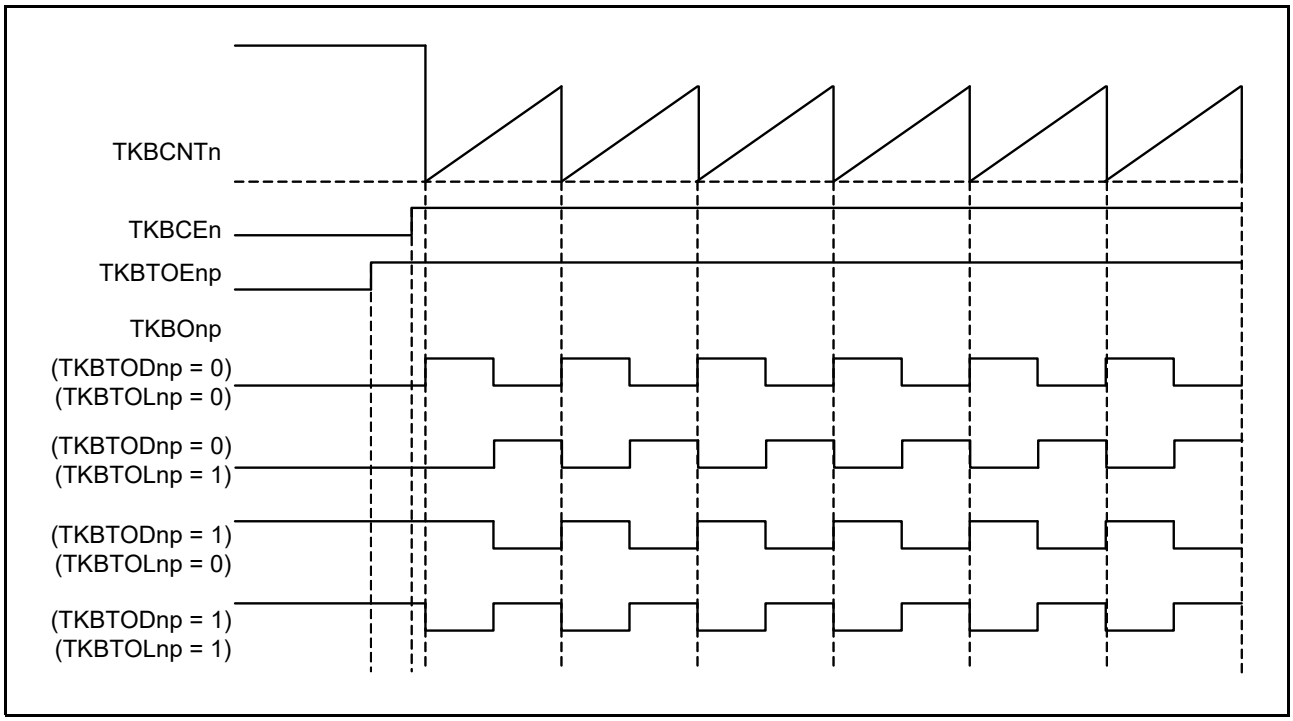
1. Count start operation
   In any mode, the 16-bit counter of the 16-bit timer KB3n starts its counting from the initial value FFFFH when the setting of the TKBCEn bit is modified from 0 to 1. The value of the counter is incremented from FFFFH to 0000H, 0001H, 0002H, 0003H and so on.

2. Clear operation
   The 16-bit counter is reset to 0000H when the 16-bit counter value matches with the value defined in the TKBCRn0 register or an external trigger is in effect if the period is determined by external triggers and an INTTMKBn interrupt occurs.

15.4.2 Default level and active level

1. Basic operation
   Default level and active level settings are available for the 16-bit timer KB3n output by 16-bit timer KB output control register n0 (TKBIOCn0).

Figure 15 - 27 Figure of Timing of Default and Active Level (Basic Operation)



When the TKBTOEnp bit is switched from 0 to 1, PWM waveform is output according to the generation of TKBOnp set condition/reset condition and the setting of the TKBTOLnp bit.
When the TKBTOEnp bit is switched from 1 to 0, default level is output for TKBOnp according to the setting of the TKBTODnp bit.

2. TKBTOEnp switched from 0 to 1
   When the TKBTOEnp bit is changed from 0 to 1 before the value of the 16-bit timer counter KBn (TKBCNTn) matches with the value of the 16-bit timer KB compare register np (TKBCRnp), while the timer counter is in operation, the timer output generated becomes the PWM waveform in accordance with the TKBTOLnp setting at the timing when it matches.
   If the TKBTOEnp bit is changed from 0 to 1 after the value of the TKBCNTn counter matches with the value of the TKBCRnp register, the timer output remains its initial setting level until the next timing of match occurs.

Figure 15 - 28 Figure of Timing of Default and Active Level (the TKBTOEnp Bit Switched from 0 to 1 Before Matching the TKBCNTn Counter and the TKBCRn1 to TKBCRn3 Registers)

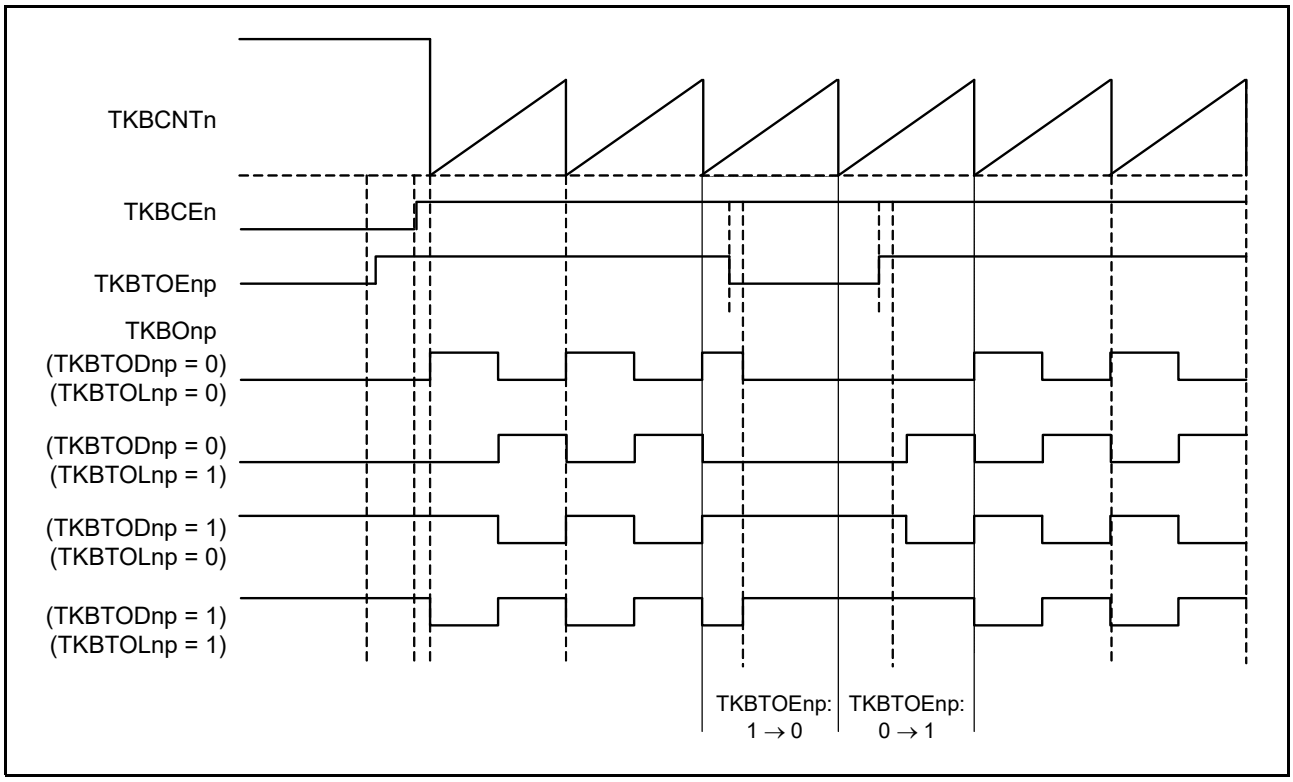
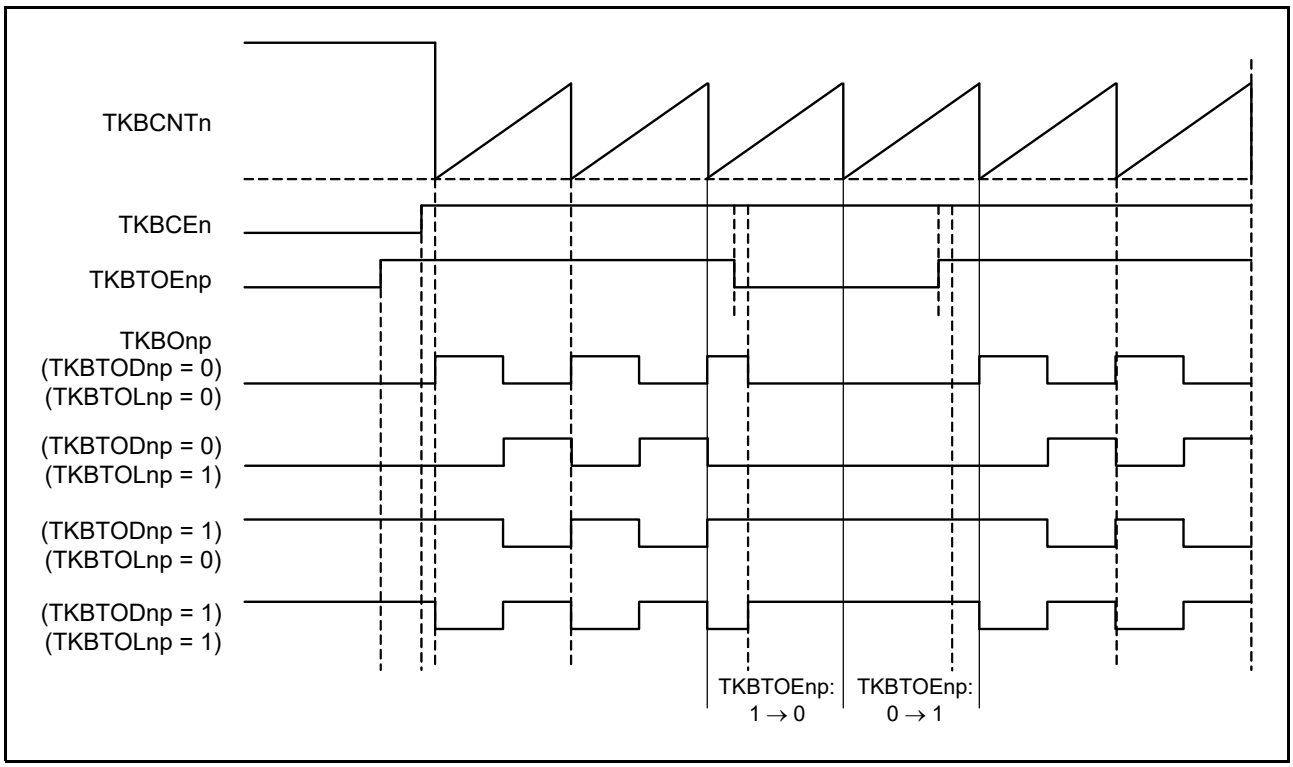


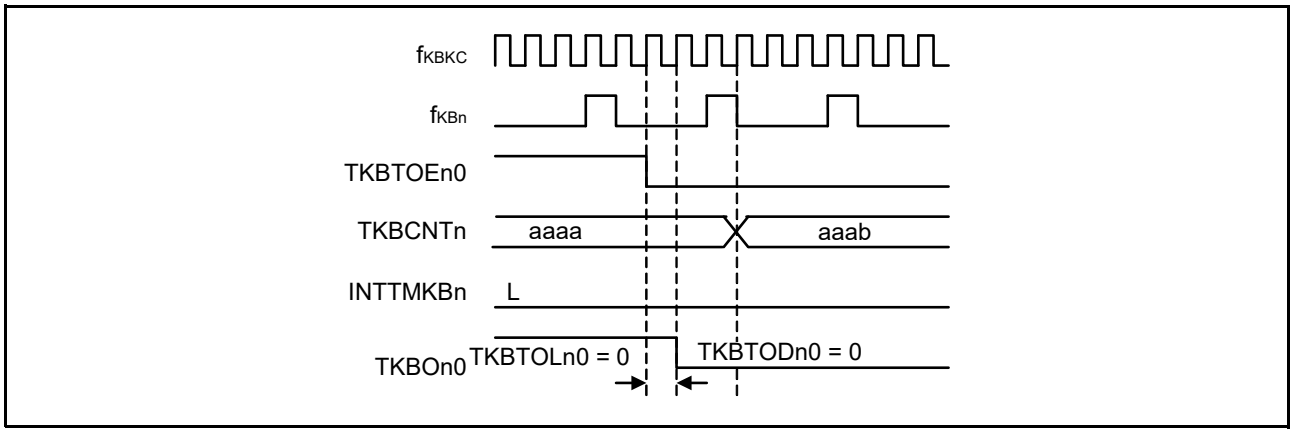
Figure 15 - 29 Figure of Timing of Default and Active Level (the TKBTOEnp Bit Switched from 0 to 1 After Matching the TKBCNTn Counter and the TKBCRn1 to TKBCRn3 Registers)



3. The TKBTOEnp bit switched from 1 to 0

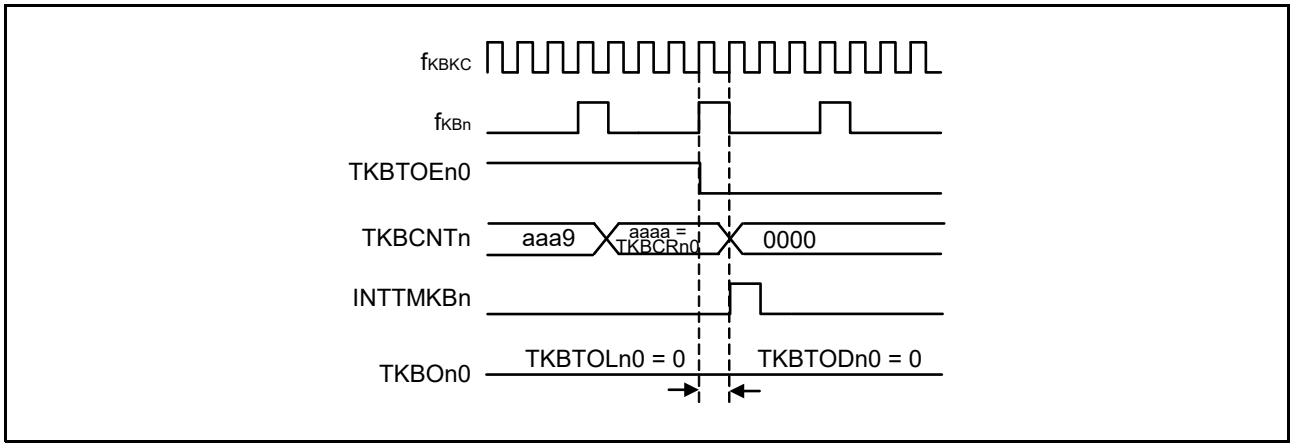
a) Basic timing
TKBOnp is default level set by the TKBTODnp bit after 1 f_{KBKC} when the TKBTOEnp bit is switched from 1 to 0.

Figure 15 - 30 Figure of Timing of Default and Active Level (the TKBTOEn0 Bit Switched from 1 to 0)



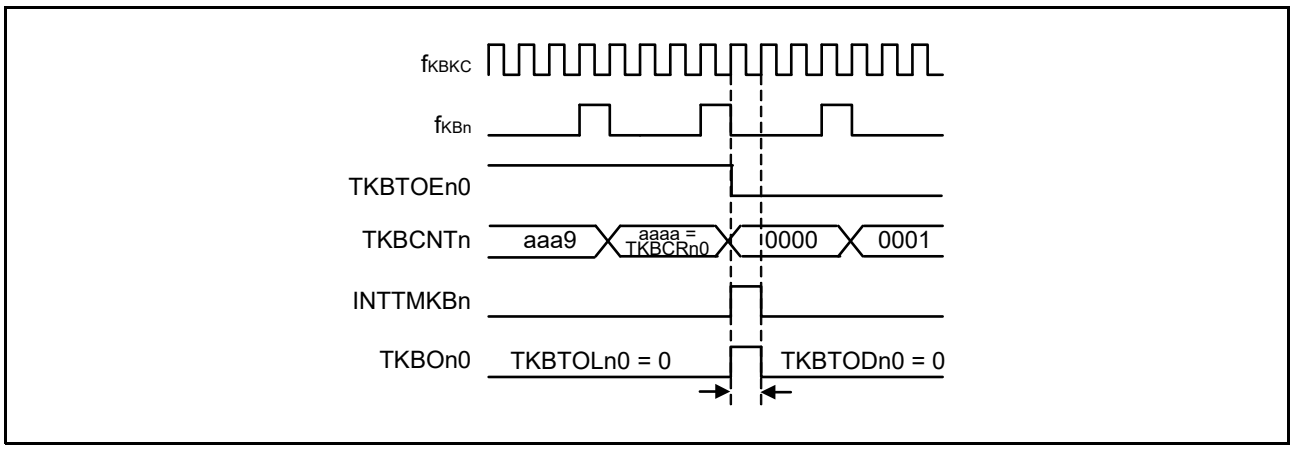
b) When the setting due to the matched value of the TKBCRn0 register and the event that the TKBTOEnp bit is cleared occur at the same instant:
When the set timing of the TKBTOEnp bit (Low to High) is simultaneous with the matching between the TKBCNTn counter and the TKBCRnm register, the change of the TKBTOEnp bit is given priority to become default level set for the TKBTODnp bit.

Figure 15 - 31 Figure of Timing for Default and Active Level (the Set Timing of the TKBTOEnp Bit (Low to High) Is Simultaneous with the Matching between the TKBCNTn Counter and the TKBCRnm Register)



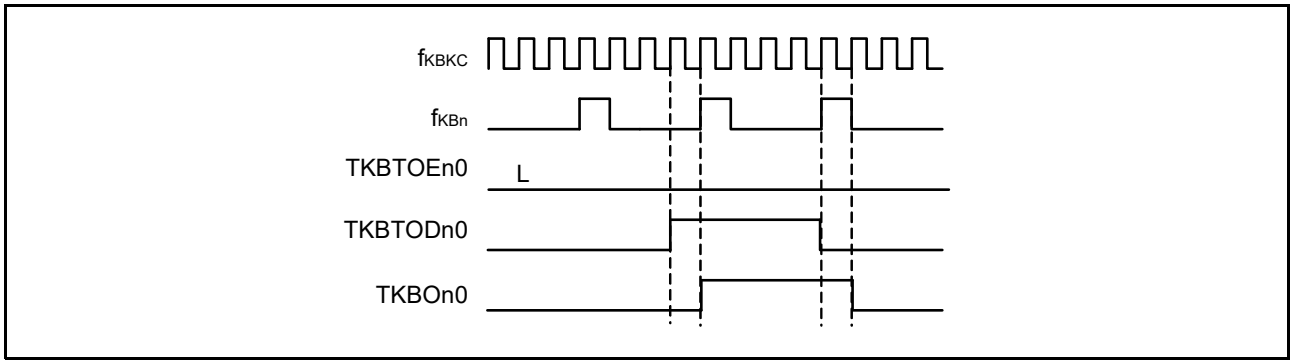
c) When the operation of the TKBTOEnp bit is simultaneous with generation of timer count clock
TKBOnp is set by the matching of TKBCNTn = TKBCRnm in case when the operation of the TKBTOEnp bit is simultaneous with generation of timer count clock.
After 1 f_{KBKC}, TKBOnp is default level which is set with the TKBTODnp bit.

Figure 15 - 32 Figure of Timing of Default and Active Level (Operation of the TKBTOEn0 Bit Is Simultaneous with Generation of Timer Count Clock.)



4. Change the TKBTODnp bit at TKBTOEnp = 0
   When the TKBTODnp bit being changed at TKBTOEnp = 0, after 1 f_{KBKC}, TKBOnp is default level which is set with the TKBTODnp bit.

Figure 15 - 33 Figure of Timing of Default and Active Level (Change the TKBTODn0 Bit at TKBTOEn0 = 0)



15.4.3 Stop/restart operation

Stop and start of operation of 16-bit timer KB3n are controlled by the setting of the TKBCEn bit.

16-bit timer KB3n is reset and stops operation by changing the TKBCEn bit from 1 to 0.

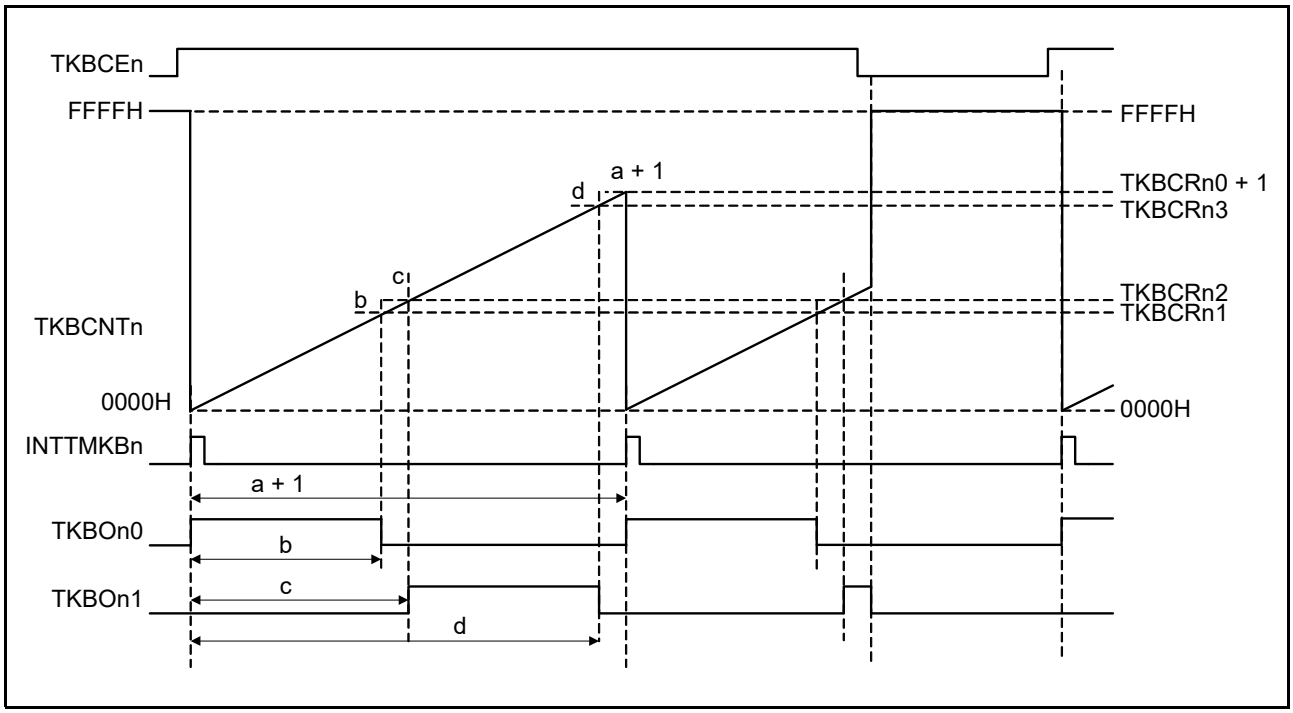
The TKBCNTn counter is reset to FFFFH and stops operation at this time.

The TKBOnp pin outputs the default level set by the TKBTODnp bit.

16-bit timer KB3n starts operation by changing the TKBCEn bit from 0 to 1.

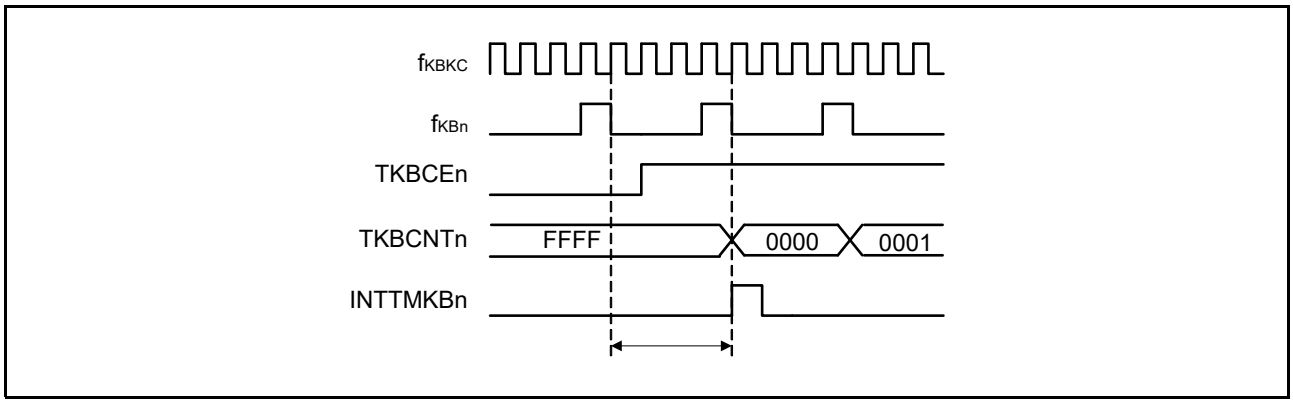
The TKBCNTn counter maintains FFFFH when TKBCEn = 0 and starts counting up by changing the TKBCEn bit from 0 to 1.

Figure 15 - 34 Figure of Timing of Stop Operation (TKBTOLnp = 0, TKBTODnp = 0)



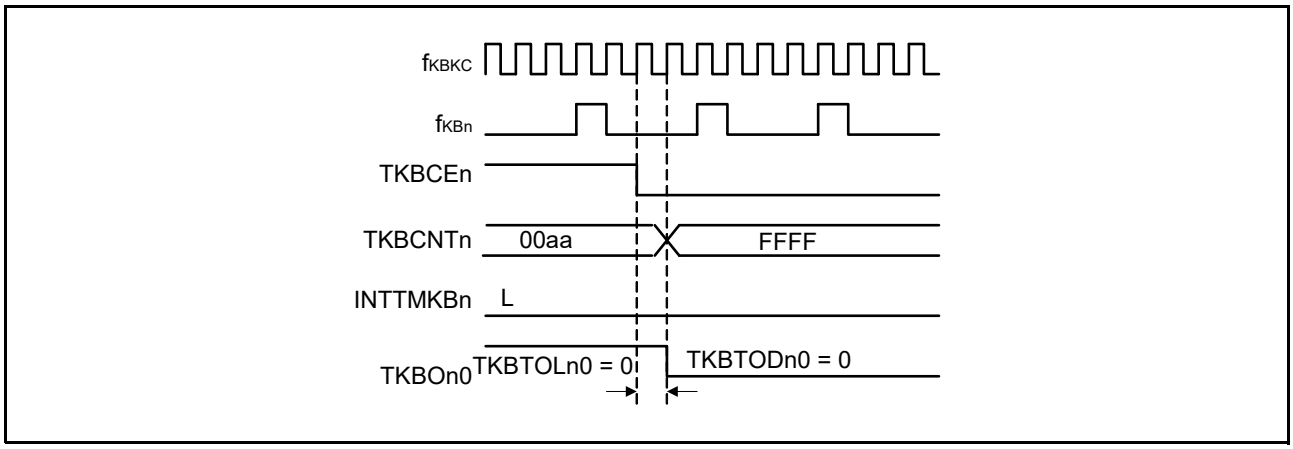
1. Count operation start timing
   When the TKBCEn bit is switched from 0 to 1 counting operation starts after the progress of the minimum 1 f_{KBKC} to the maximum 1 f_{KBn}.
   INTTMKBn is output at counting operation start timing.

Figure 15 - 35 Figure of Timing of Start Operation (the TKBCEn Bit Switched from 0 to 1)



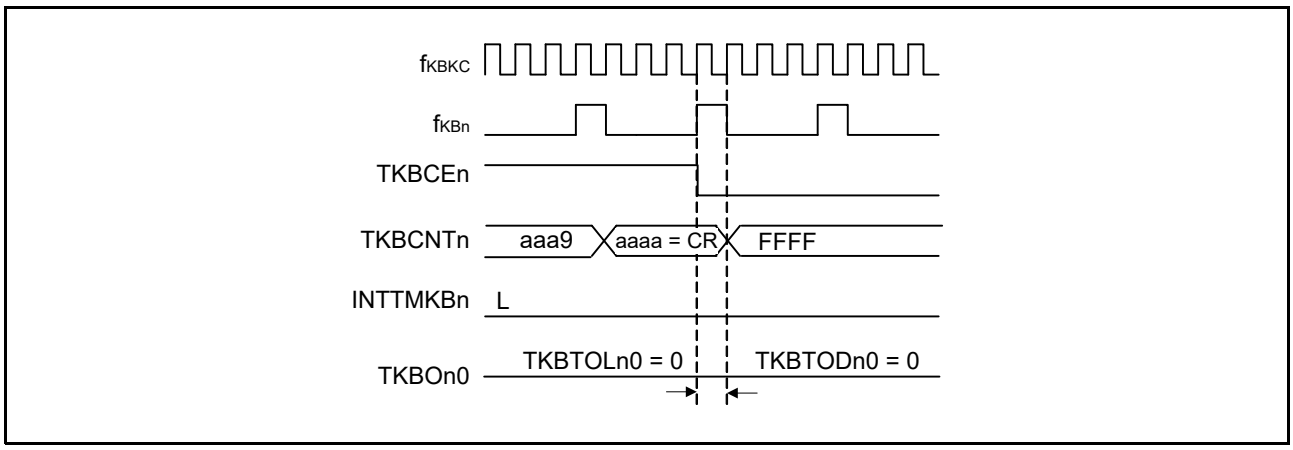
2. Count operation stop timing
   When the TKBCEn bit is switched from 1 to 0 counting operation is stopped after the progress of minimum 1 f_{KBKC}.
   The TKBCNTn counter is reset to FFFFH and TKBOnp is default level set by the TKBTODnp bit.

Figure 15 - 36 Figure of Timing of Stop Operation (the TKBCEn Bit Switched from 1 to 0)



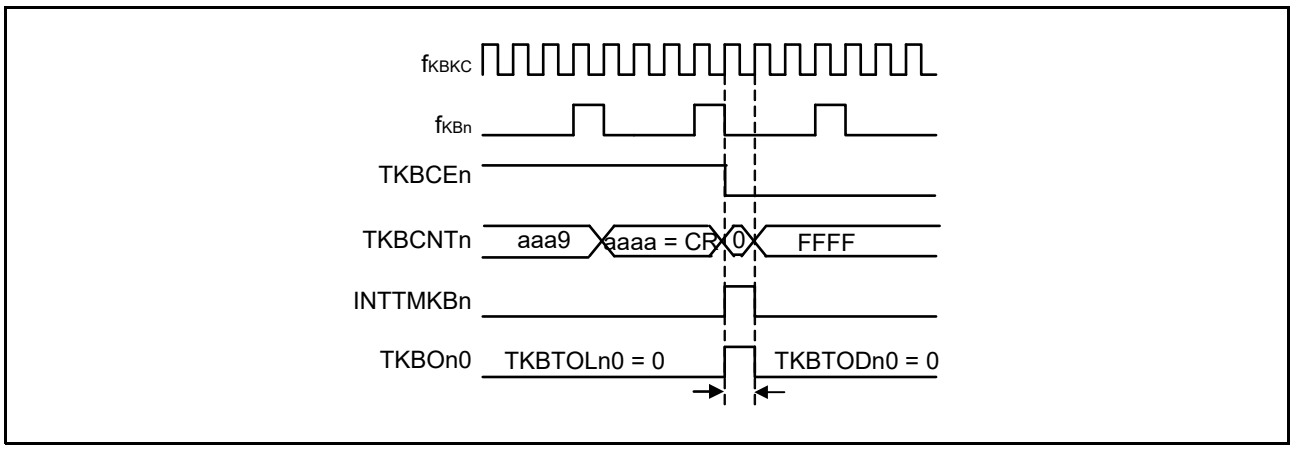
Before the generation of 1 f_{KBn}, INTTMKBn is not output even matching of TKBCNTn = TKBCRn0 being generated.

Figure 15 - 37 Figure of Timing of Stop Operation (Operation of TKBCEn Is Before the Generation of Timer Count Clock.)



TKBOnp is set and INTTMKBn being output via the matching of TKBCNTn = TKBCRn0 generated in case when the operation of the TKBCEn bit is simultaneous with the generation of 1 fKBKC.
The TKBCNTn counter is reset to FFFFH after the progress of 1 fKBKC and TKBOnp is default level set by the TKBTODnp bit.

Figure 15 - 38 Figure of Timing of Stop Operation (Operation of TKBCEn Is Simultaneous with the Generation of Timer Count Clock.)

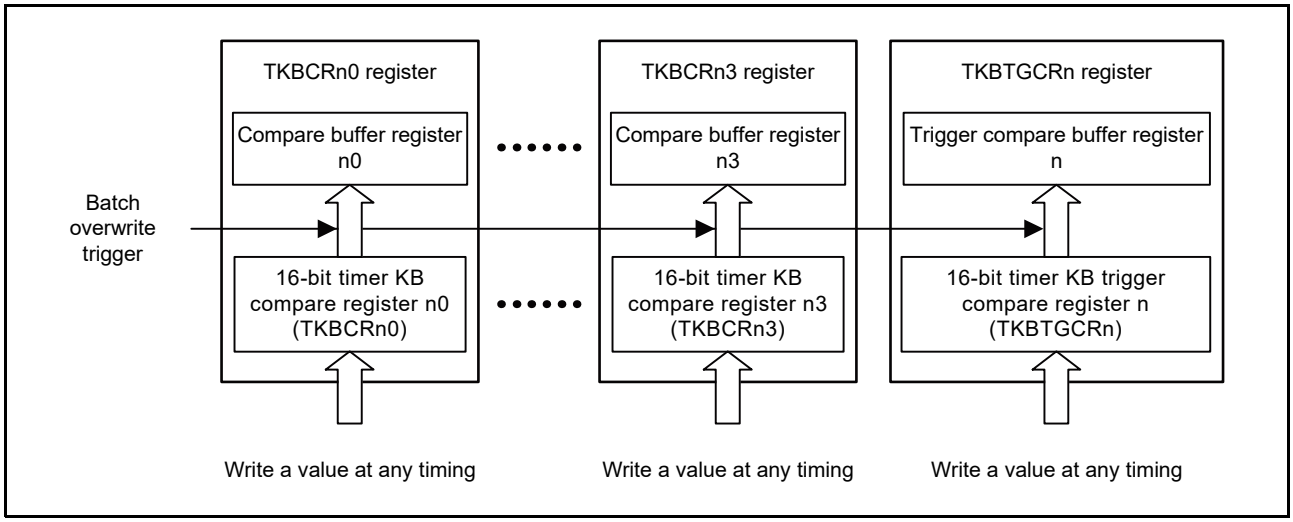


15.4.4 Batch overwrite operation

The 16-bit timer KB compare register np (TKBCRnp) for the 16-bit timer KB3n has, as shown in **Figure 15 - 39**, two stages.

Therefore, its value does not become effective immediately even if any value is set to the TKBCRnp register by a program. The value set to the TKBCRnp register at any timing is transferred at once to buffer registers at the time when the counter starts running or when transfer trigger occurs, and it is actually used for any comparison operation. This enables multiple compare registers to be set with each value at different timing.

Figure 15 - 39 Compare Register Batch Overwrite Function



Remark As shown above, the 16-bit timer KB compare register np (TKBCRnp) has two-stage structure and the stages are treated as a single register except when values are written to them. The values of the buffer registers cannot be read directly.

1. Procedure of batch overwrite
   a) Update the 16-bit timer KB compare register (TKBCRnp).
   b) Check that the TKBRSFn flag in the 16-bit timer KB flag register (TKBFLGn) is 0.
   c) Set the TKBRDTn bit in the 16-bit timer KB trigger register (TKBTRGn) to 1.
   d) An overwrite trigger occurs (any of the three conditions shown in **2. Timing of batch overwrite trigger** matches).
2. Timing of batch overwrite trigger
   There are three cases when the compare registers are written all together. Among these, c) can be controlled by register settings.

   a) When starting count operation of the 16-bit timer KB3n
   b) Count value of the 16-bit timer counter KBn and the value that is set to the 16-bit timer KB compare register n0 (TKBCRn0) matches.
   c) An external trigger occurs, while batch overwrite with an external trigger is permitted.

15.4.5 Skipping of interrupt requests and A/D conversion triggers

16-bit timers KB30, KB31, KB32 can be used to control skipping of the output of interrupt request and A/D conversion trigger signals. Whether to enable or disable the output of interrupt request and A/D conversion trigger signals in the first cycle after the start of 16-bit timer KB3n operation can be specified by the TKBTMDn bit in the TKBTCTLn register. When the TKBTMDn bit is set to 1, the interrupt and trigger signals are output in the first cycle.

Figure 15 - 40 shows an example of operation when the output of interrupt request and A/D conversion trigger signals is disabled in the first cycle after the start of 16-bit timer KB3n operation (TKBTCTLn.TKBTMDn = 0) in standalone mode and **Figure 15 - 41** shows an example of operation when the output is enabled (TKBTCTLn.TKBTMDn = 1).

Figure 15 - 40 Example of Operation When Output Is Disabled in the First Cycle after the Start of 16-bit Timer KB3n Operation (TKBTCTLn.TKBTMDn = 0) in Standalone Mode

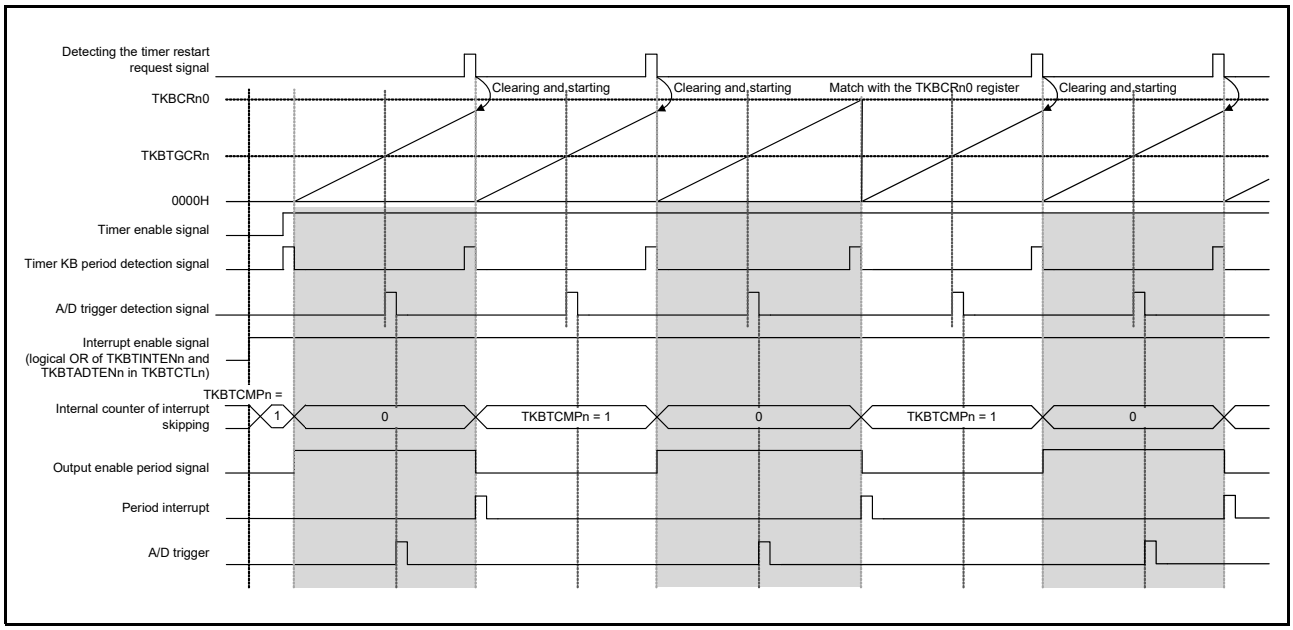
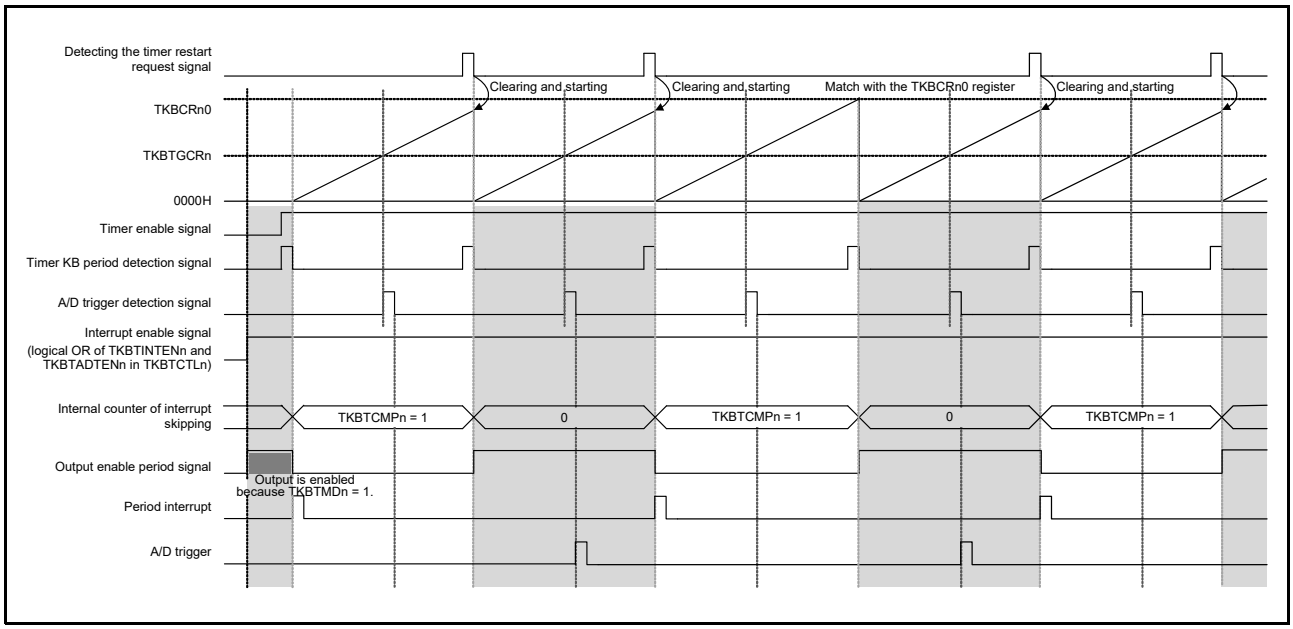


Figure 15 - 41 Example of Operation When Output Is Enabled in the First Cycle after the Start of 16-bit Timer KB3n Operation (TKBTCTLn.TKBTMDn = 1) in Standalone Mode



When a 16-bit timer KB3n is operating in simultaneous start/clear mode, counter operation is the same for both the master and slave. Note that since skipping control can be set for each of the timers, control over skipping period interrupts and A/D triggers can be set up individually per timer. **Figure 15 - 42** shows an example of operation when output is disabled in the first cycle after the start of 16-bit timer KB3n operation (TKBTCTLn.TKBTMDn = 0) in simultaneous start/clear mode and **Figure 15 - 43** shows an example of operation when the output is enabled (TKBTCTLn.TKBTMDn = 1).

Figure 15 - 42 Example of Operation When Output Is Disabled in the First Cycle after the Start of 16-bit Timer KB3n Operation (TKBTCTLn.TKBTMDn = 0) in Simultaneous Start/Clear Mode (Master: Skipping of One Output Pulse; Slave: Skipping of Three Output Pulses)

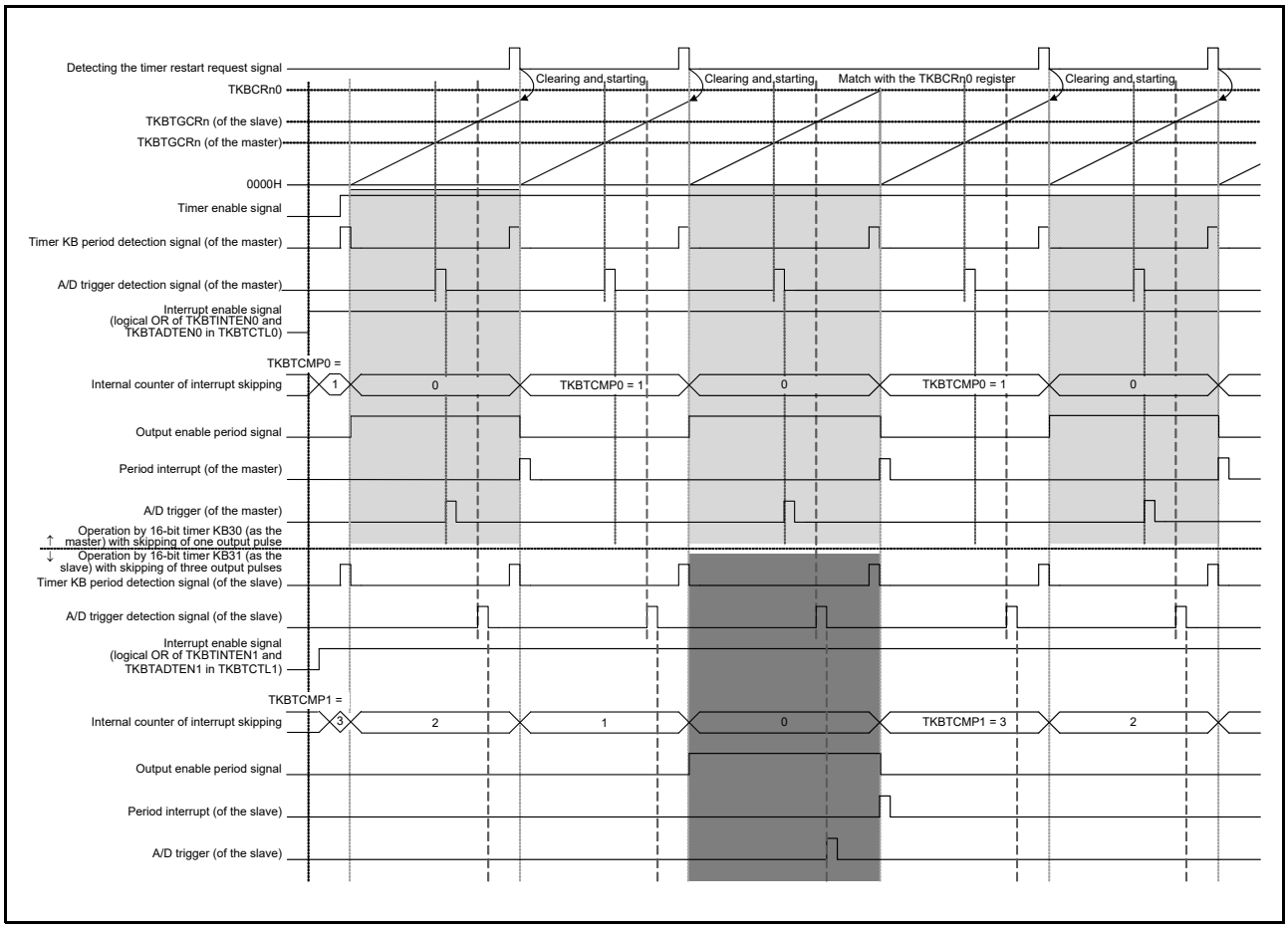
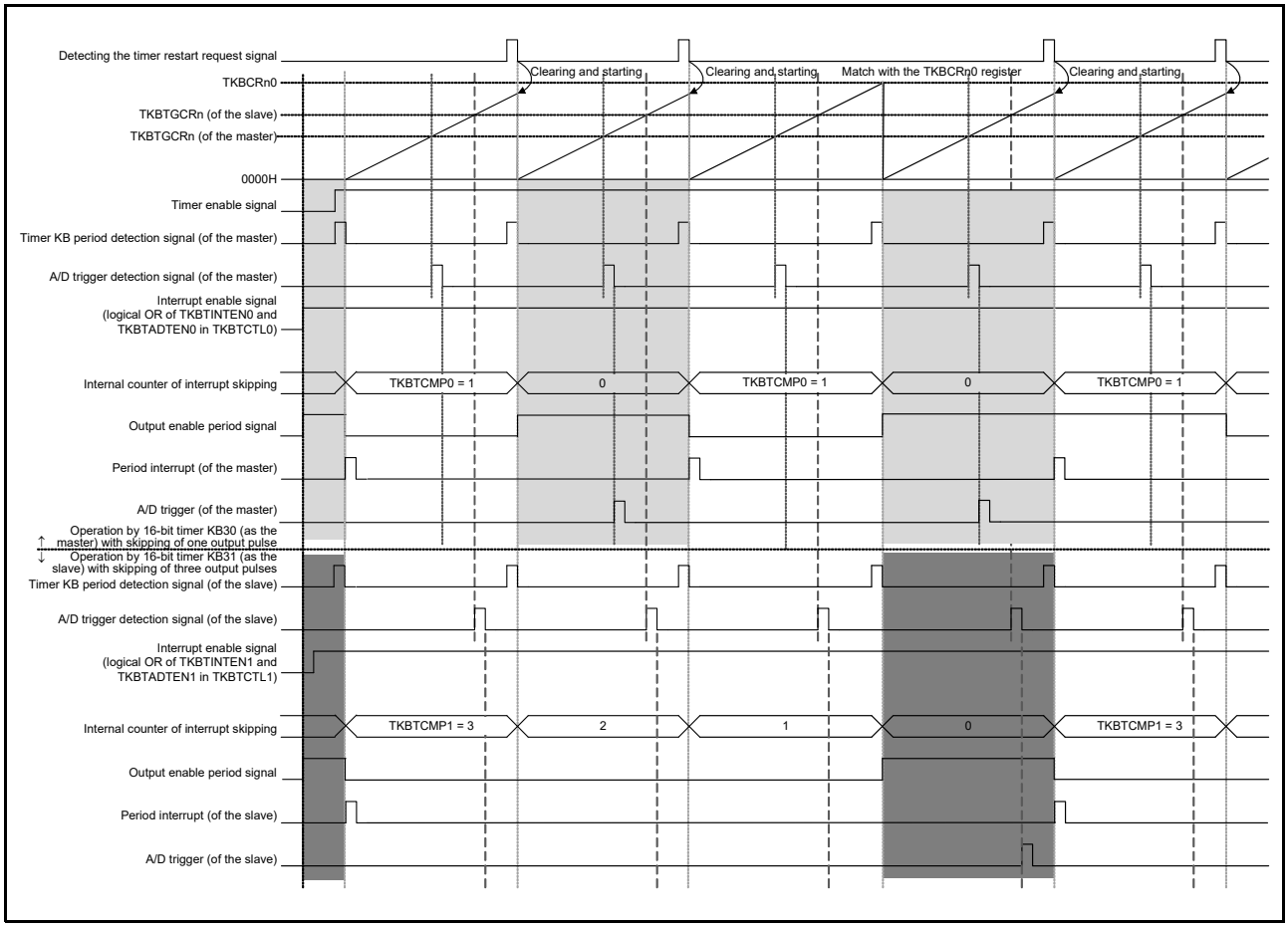


Figure 15 - 43 Example of Operation When Output Is Enabled in the First Cycle after the Start of 16-bit Timer KB3n Operation (TKBTCTLn.TKBTMDn = 1) in Simultaneous Start/Clear Mode (Master: Skipping of One Output Pulse; Slave: Skipping of Three Output Pulses)



When a 16-bit timer KB3n is set up for interlocked operation with timer RD2 and the PWM output gating function of the 16-bit timer KB3n is to be used, skipping control interlocked with the PWM output gating function is possible. To use this control interlocked with the gating mode, set TKBTCTLn.TKBTKBGENn to 1 and TKBTCTLn.TKBTMDn to 0. When both gate control signals output from timer RD2 are at the low level, skipping control is stopped. The gate control signals then go to the high from the low level, and the internal 5-bit counter is cleared and restarted. The interrupt request and A/D conversion trigger signals are not generated while the gate control signals are at the low level.

Figure 15 - 44 shows the configuration of control over counting during control interlocked with the gating mode and **Figure 15 - 45** shows an example of operation during control interlocked with the gating mode.

Figure 15 - 44 Configuration of Control over Counting during Control Interlocked with the Gating Mode

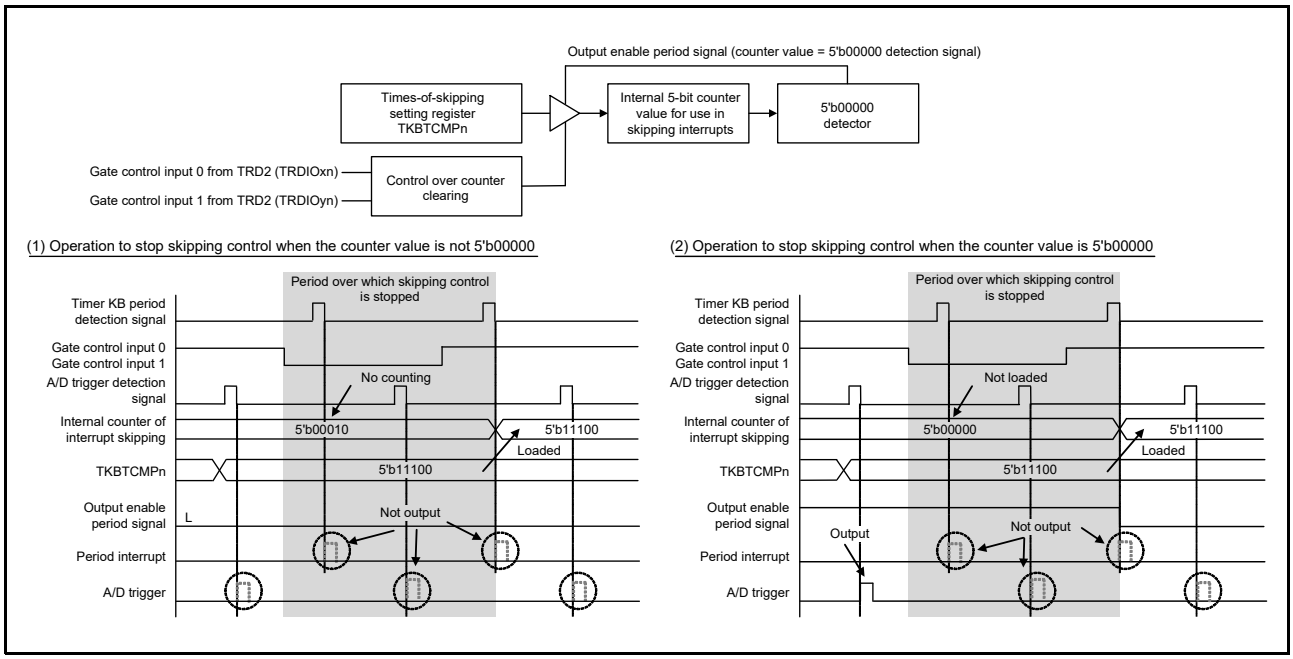
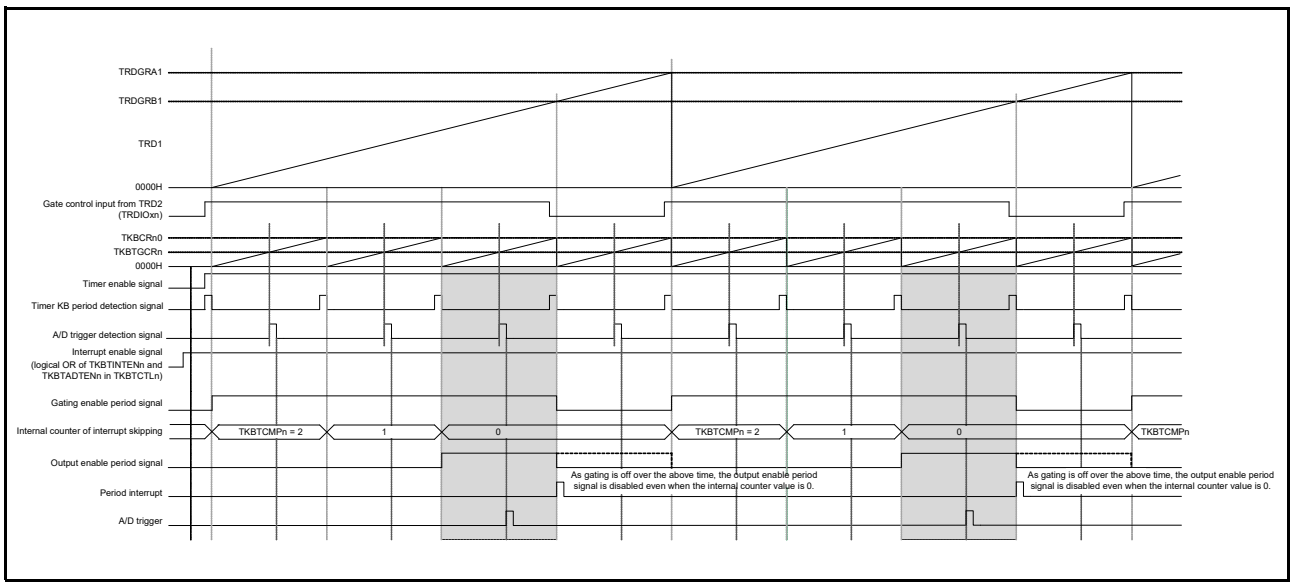


Figure 15 - 45 Example of Operation during Control Interlocked with the Gating Mode



15.4.6 Standalone mode (period controlled by the TKBCRn0 register)

1. Outline of functions

In standalone operating mode, the period is defined by setting value of the TKBCRn0 register, then TKBOn0 is generated by the TKBCRn0 and TKBCRn1 registers, and then TKBOn1 is generated by the TKBCRn2 and TKBCRn3 registers.

Duty can be set within range of 0% to 100% and the period and Duty can be calculated using the following formula.

[Calculation formula for TKBOn0 output]

Pulse period = (TKBCRn0 setting + 1) × Counter clock period

Duty [%] = (TKBCRn1 setting/(TKBCRn0 setting + 1)) × 100

0% Output: TKBCRn1 setting = 0000H

100% Output: TKBCRn1 setting ≥ TKBCRn0 setting + 1

[Calculation formula for TKBOn1 output]

Duty [%] = ((TKBCRn3 setting – TKBCRn2 setting)/(TKBCRn0 setting + 1)) × 100

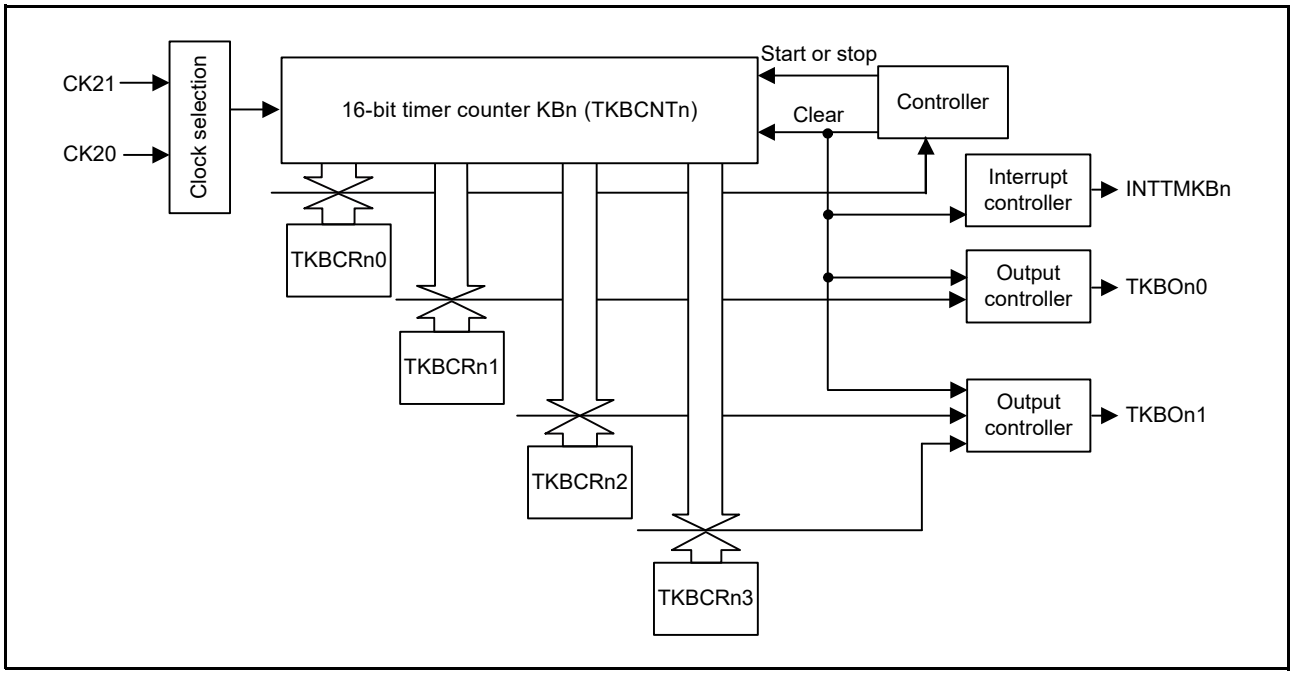
0% Output: TKBCRn3 setting = TKBCRn2 setting

100% Output: TKBCRn2 setting = 0000H, TKBCRn3 setting ≥ TKBCRn0 setting + 1

Caution It should always be: TKBCRn2 setting ≤ TKBCRn3 setting.

Figure 15 - 46 shows the configuration in standalone mode.

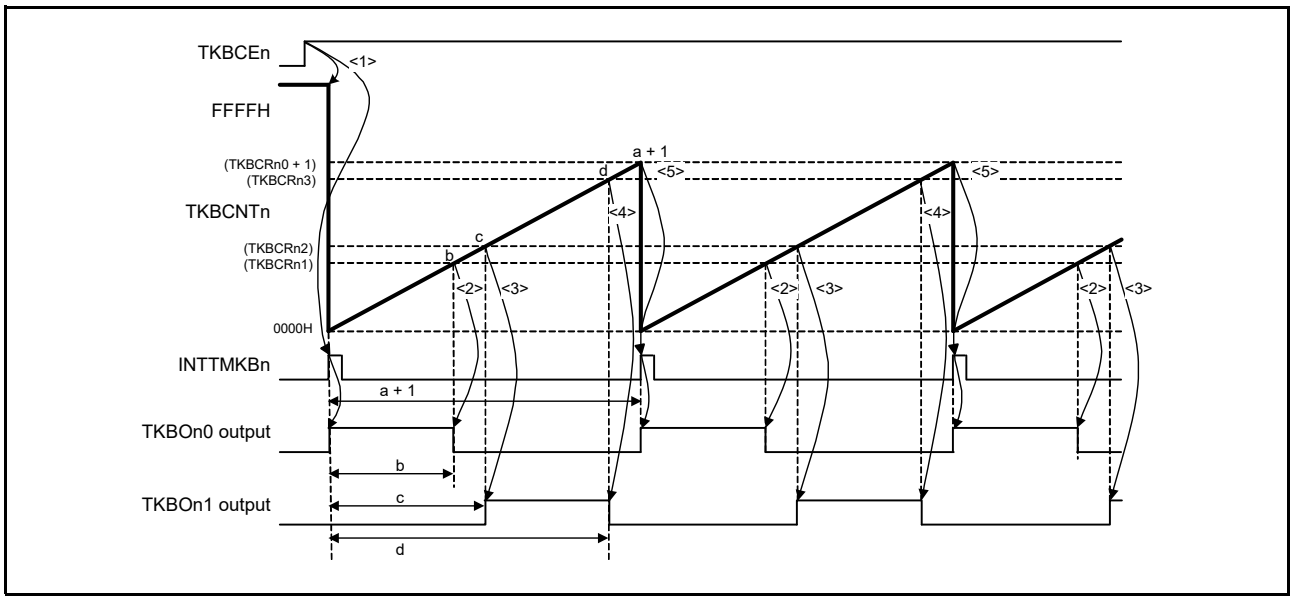
Figure 15 - 46 Configuration in Standalone Mode (Period Controlled by the TKBCRn0 Register)



2. Outline of operation

Figure 15 - 47 shows an example of the operation timing in standalone mode.

Figure 15 - 47 Example of Timing in Standalone Mode (Period Controlled by the TKBCRn0 Register) (at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



This section describes an example of the standalone operation (periodic control by TKBCRn0). <1> to <5> in **Figure 15 - 47** are described below.

<1> When the TKBCEn bit is set with a value of 1, the 16-bit timer counter KBn (TKBCNTn) changes from FFFFH to 0000H in synchronizing with the count clock, then it starts counting up. At the same time, INTTMKBn output is generated and TKBOn0 output changes from its initial value specified with the TKBTODn0 bit of the TKBIOCn0 register to its active value (high level in this example) specified with the TKBTOLn0 bit (TKBOn1 output hold its initial value specified with the TKBTODn1 bit).

<2> When the TKBCNTn counter is counted up and its value matches with the value specified in the 16-bit timer KB compare register n1 (TKBCRn1), TKBOn0 output becomes inactive level.

<3> When the TKBCNTn counter is counted up and its value matches with the value specified in the 16-bit timer KB compare register n2 (TKBCRn2), TKBOn1 output becomes active level.

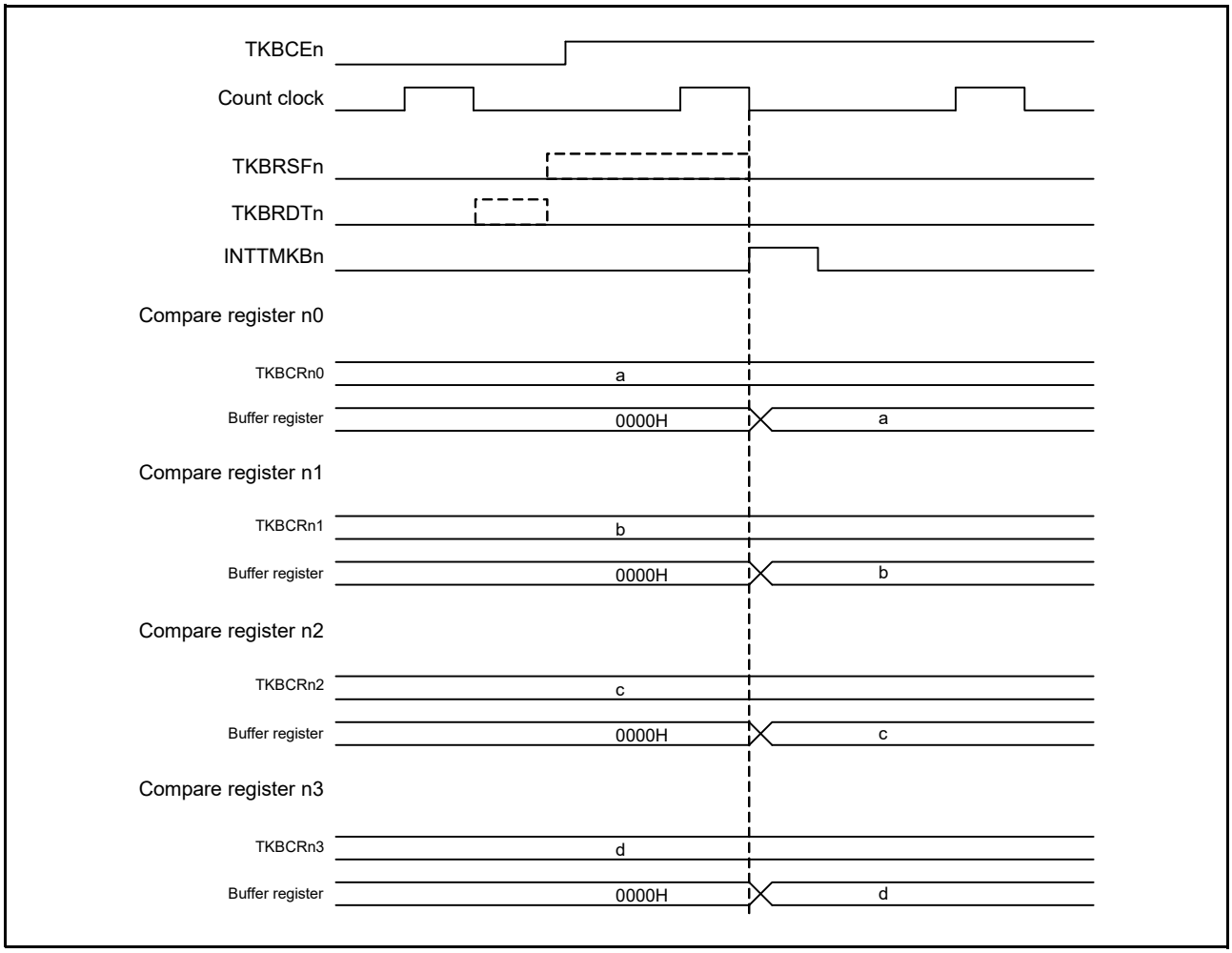
<4> When the TKBCNTn counter is counted up and its value matches with the value specified in the 16-bit timer KB compare register n3 (TKBCRn3), TKBOn1 output becomes inactive level.

<5> When the TKBCNTn counter is counted up and its value matches with the value specified in the 16-bit timer KB compare register n0 (TKBCRn0), INTTMKBn output is generated at the next count clock and TKBOn0 output becomes active level. The TKBCNTn counter starts its upward counting from 0000H.

<6> Repeats <2> through <5>.

3. Operation of batch overwrite (at starting the counting operation)
   Compare registers of the 16-bit timer KB3n have function which updates internal buffer register simultaneously at the starting of counter operation caused by count clock which is generated after overwriting 1 to the TKBCEn bit. Batch overwrite is generated without writing 1 on the TKBRDTn bit only in case of counting operation start timing (see **Figure 15 - 48**).

Figure 15 - 48 Batch Overwrite Function: Figure of Buffer Updating Timing at Counting Operation Start

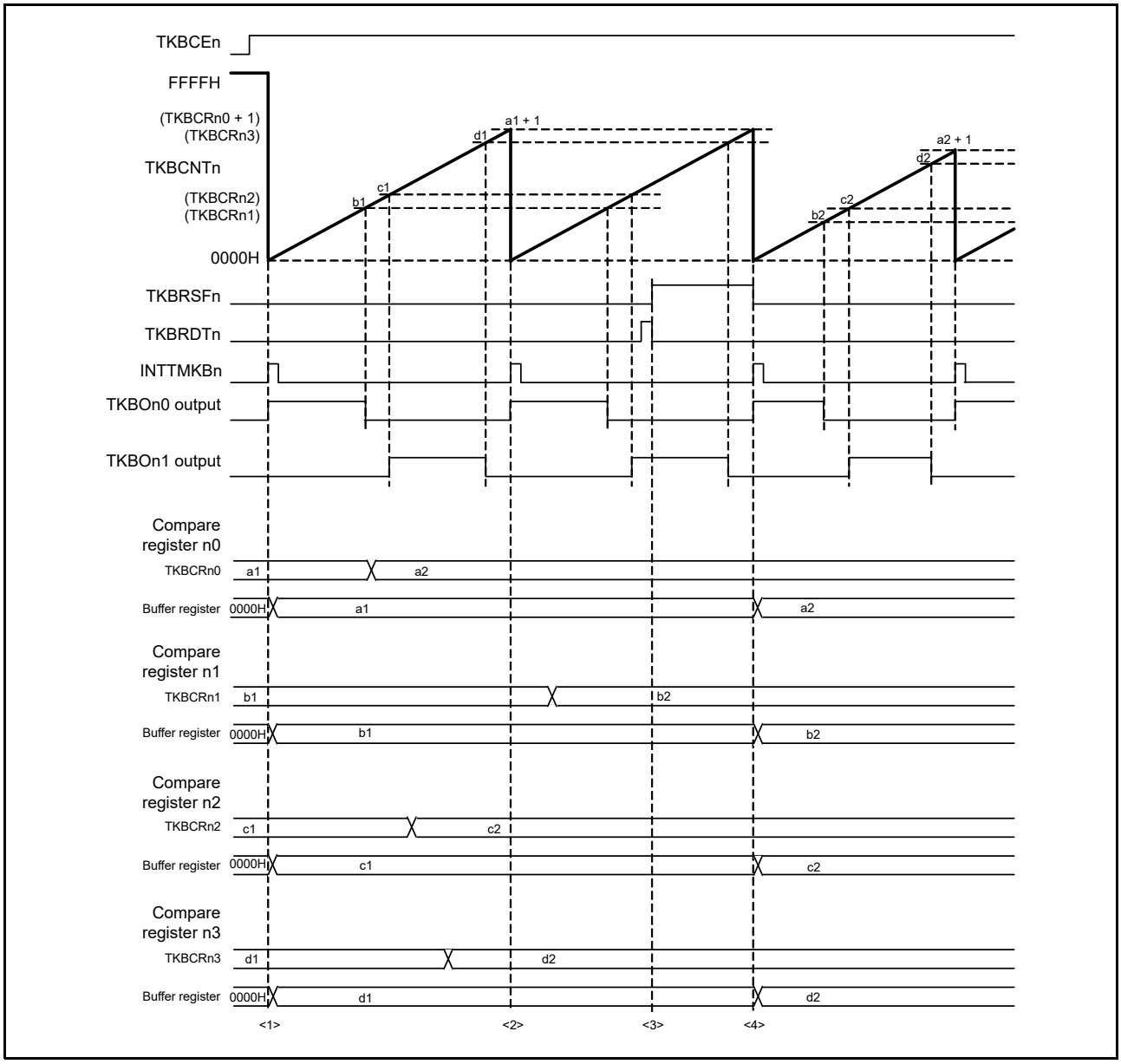


Remark When TKBCEn = 0, the TKBRSFn flag is set to 1 at writing 1 to the TKBRDTn bit.
The TKBRSFn flag is cleared to 0 at counting operation start timing (counter start trigger generated).

4. Batch overwrite function: Update buffer during counting operation
   Compare registers of the 16-bit timer KB3n have function which updates internal buffer register simultaneously at the next counter clear (the TKBCNTn counter and the TKBCRn0 register matched), identifying the writing 1 to the TKBRDTn bit as batch overwriting trigger. The TKBRSFn flag is provided as flag to indicate from writing of 1 to the TKBRDTn bit until the completion of batch overwrite (see **Figure 15 - 49**).

   <1> Compare register setting is transferred to buffer register at the timing when the TKBCEn bit is set from 0 to 1 and the TKBCNTn counter starts counting operation.
   <2> Overwriting is not generated if writing of 1 to the TKBRDTn bit is not implemented even counter clear is generated after the TKBCRnm and TKBTGCRn registers are overwritten.
   <3> Batch overwrite pending flag (the TKBRSFn flag) is 1 by writing 1 to the TKBRDTn bit.
   <4> Compare register setting is transferred to buffer register by counter clear generated when the TKBRSFn flag is 1. The TKBRSFn flag becomes 0 simultaneously.

Figure 15 - 49 Batch Overwrite Function: Figure of the Timing of Buffer Updating During Counting Operation



5. Sample of register settings in standalone mode (period controlled by the TKBCRn0 register)

TKBCTLn0

15	14	13	12	11	10	9	8
—	TKBGTEn1	TKBSSEn1	TKBDIEn1	—	TKBGTEn0	TKBSSEn0	TKBDIEn0
0	1/0	1/0	1/0	0	1/0	1/0	1/0

7	6	5	4	3	2	1	0
TKBMFEn	—	TKBIRSn1	TKBIRSn0	—	TKBTSEn	TKBSTSn1	TKBSTSn0
0	0	0	0	0	0	0	0

TKBCTLn1

7	6	5	4	3	2	1	0
TKBCEn	—	—	TKBCKSn	TKBSCMn	—	TKBMDn1	TKBMDn0
1/0	0	0	1/0	0	0	0	0

TKBIOCn0

7	6	5	4	3	2	1	0
—	—	—	—	TKBTOLn1	TKBTOLn0	TKBSSEn0	TKBDIEn0
0	0	0	0	1/0	1/0	1/0	1/0

TKBIOCn1

7	6	5	4	3	2	1	0
—	—	—	—	—	—	TKBTOEn1	TKBTOEn0
0	0	0	0	0	0	1/0	1/0

TKBCTLn2

15	14	13	12	11	10	9	8
—	—	—	—	—	—	TKBMFMn 1	TKBMFMn 0
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
—	TKBINSn2	TKBINSn1	TKBINSn0	—	TKBKCIn2	TKBKCIn1	TKBKCIn0
0	0	0	0	0	0	0	0

TKBCRn0	0000H to FFFFH
TKBCRn1	0000H to FFFFH
TKBCRn2	0000H to FFFFH
TKBCRn3	0000H to FFFFH
TKBTGCRn	0000H to FFFFH
TKBSIRn0	0000H to FFFFH
TKBSIRn1	0000H to FFFFH
TKBSSRn0	00H to 0FH
TKBSSRn1	00H to 0FH
TKBDNRn0	00H to F0H
TKBDNRn1	00H to F0H
TKBMFRn	0000H

: Setting is fixed for this mode : Setting is not needed (default setting)

15.4.7 Standalone mode (period controlled by external trigger input)

1. Outline of functions

By standalone mode, period can be controlled not only by the TKBCRn0 register but also by external trigger input (timer restart function). By using this function, PFC control in the critical conduction mode can be implemented, for example.

For external trigger input detection, the input signal selected by setting the following bits is used.

- The TKBSTSn[1:0] bits in the 16-bit timer KB operation control register n0
- The TKBINSn[2:0] bits in the 16-bit timer KB operation control register n2
- The TKBKCIn[2:0] bits in the 16-bit timer KB operation control register n2

When the external trigger input is detected, the TKBCNTn counter is cleared with 0000H and TKBOn0/TKBOn1 output is respectively set to active level and inactive level. When setting value of the TKBCRn0 register and the TKBCNTn counter match is generated before external trigger input detection, the TKBCNTn counter is cleared to 0000H and the operation is continued.

For the formula to calculate TKBOn0/TKBOn1 output in case external trigger input not yet detected and the period is controlled by the TKBCRn0 register, see **15.4.6 Standalone mode (period controlled by the TKBCRn0 register)**.

Calculation formula for TKBOn0/TKBOn1 output in case of period to be controlled by external trigger input detection is as follows:

[Calculation formula for TKBOn0 output]

Pulse period = (Counter value at external trigger input detection + 1) × Count clock period

Duty [%] = (Setting value of TKBCRn1/(Counter value of external trigger input detection + 1)) × 100

0% output: TKBCRn1 setting = 0000H

100% output: TKBCRn1 setting ≥ Counter value at external trigger input detection + 1

[Calculation formula for TKBOn1 output]

Pulse period = (Counter value at external trigger input detection + 1) × Count clock period

Duty [%] = ((Setting value of TKBCRn3 – Setting value of TKBCRn2)/(Counter value of external trigger input detection +1)) × 100

0% output: TKBCRn3 setting = TKBCRn2 setting

100% output: TKBCRn2 setting = 0000H, TKBCRn3 setting ≥ Counter value at external trigger input detection + 1

Caution It should always be: TKBCRn2 setting ≤ TKBCRn3 setting.

Figure 15 - 50 shows the configuration in standalone mode (period controlled by external trigger input).

Figure 15 - 50 Configuration in Standalone Mode (Period Controlled by External Trigger Input)

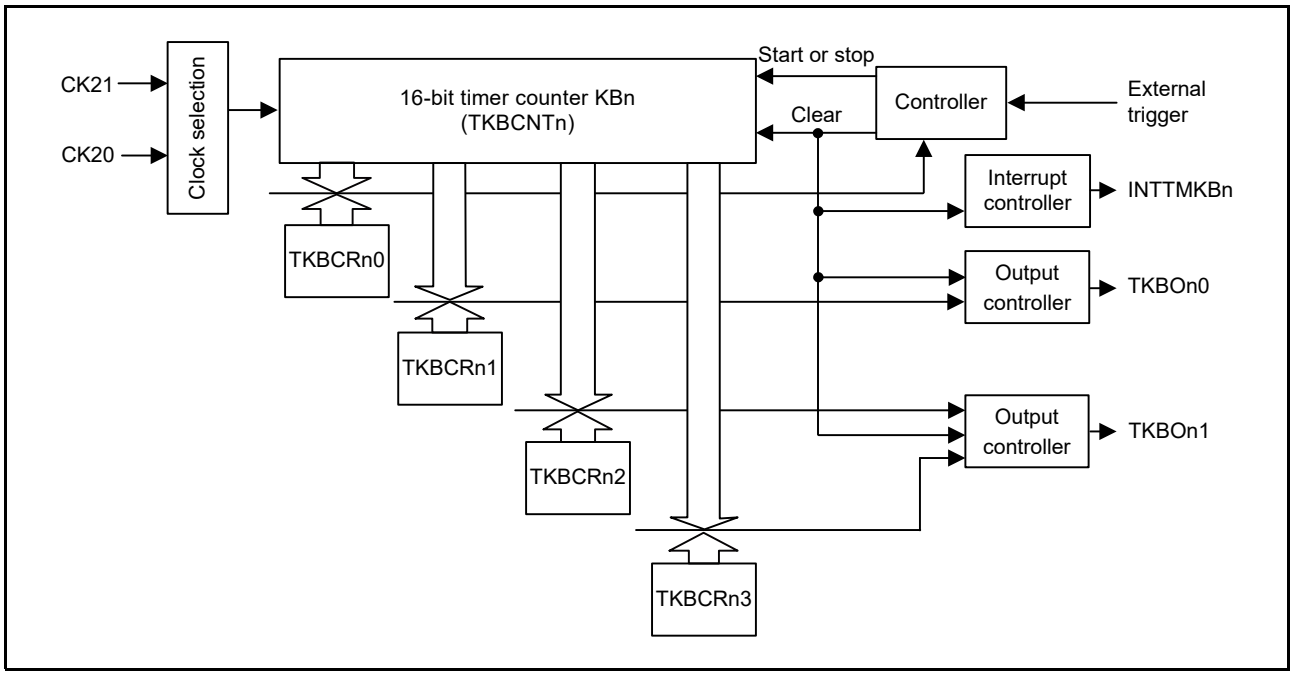


Table 15 - 3 External Trigger Assignment List of Standalone Operation (Period Controlled by External Trigger Input)

Item	16-bit Timer KB30	16-bit Timer KB31	16-bit Timer KB32
Comparator 0	✓	✓	✓
Comparator 1	✓	✓	✓
Comparator 2	✓	✓	✓
Comparator 3	✓	✓	✓
INTP20	✓	✓	✓
INTP21	✓	✓	✓
INTP0	✓	✓	✓

2. Batch overwrite function (period controlled by external trigger input, buffer updating during counting operation (TKBTSEn bit set to 1))

   In standalone mode of period controlled by external trigger input, counter clear and compare register batch overwrite is implemented at the timing when external trigger input is detected after writing 1 to the TKBRDTn bit and through setting the TKBTSEn bit of the TKBCTLn0 register in 1.

   Same as in counter clear, batch overwriting is implemented as well in case when the TKBCRn0 register and the TKBCNTn counter being matched before the detection of external trigger input after writing 1 to the TKBRDTn bit.

   The source of the external trigger input is selected by setting the following bits.

   - The TKBSTSn[1:0] bits in the TKBCTLn0 register
   - The TKBINSn[2:0] bits in the TKBCTLn2 register
   - The TKBKCIn[2:0] bits in the TKBCTLn2 register

   Figure 15 - 51 shows an example of the timing of the batch overwrite operation with the TKBTSEn bit set to 1.

   <1> Compare register setting is transferred to buffer register at the timing when the TKBCEn bit is set from 0 to 1 and the TKBCNTn counter starts counting operation.

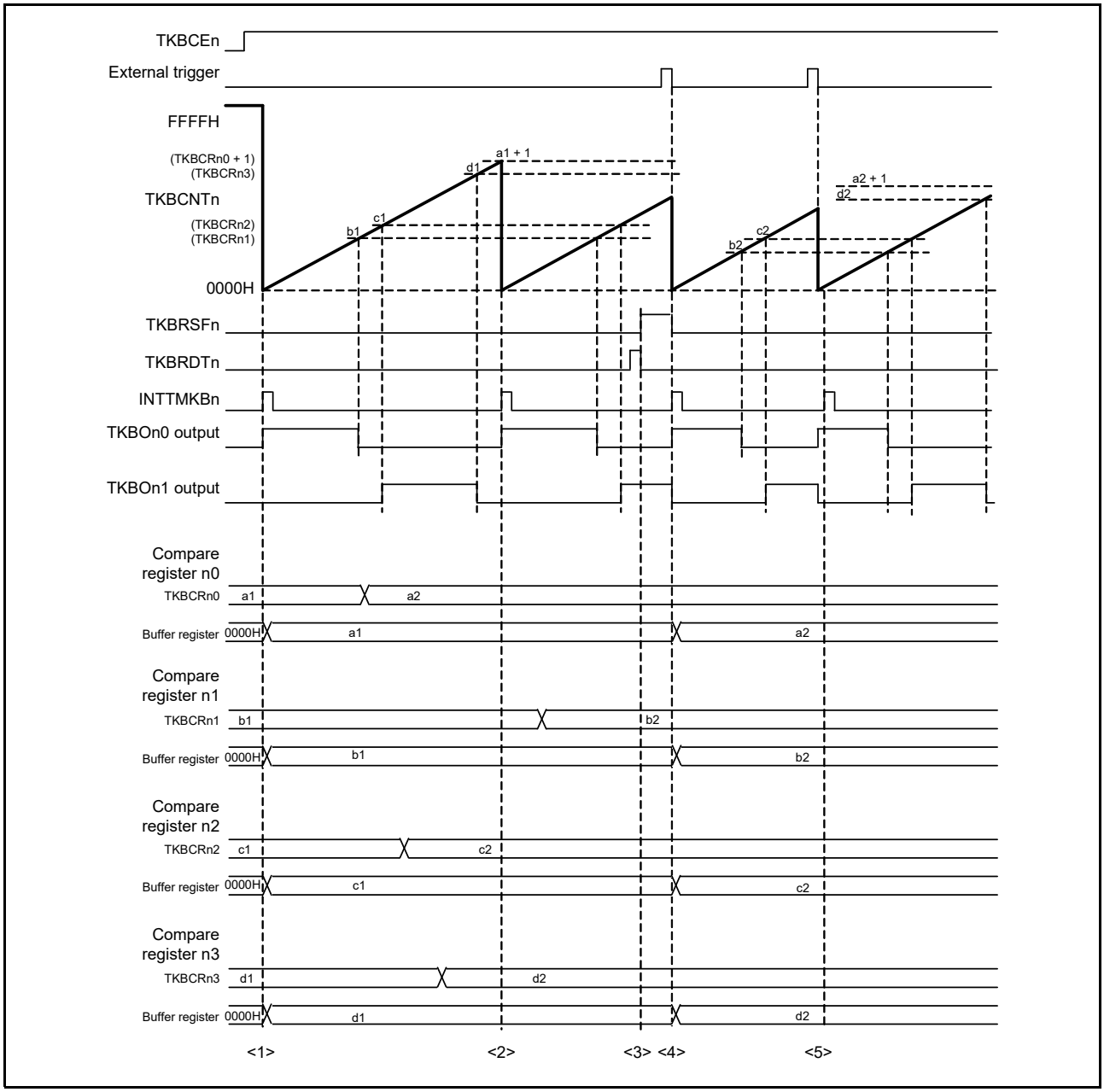
   <2> Overwriting is not generated if writing of 1 to the TKBRDTn bit is not implemented even counter clear is generated after the TKBCRnm and TKBTGCRn registers are overwritten.

   <3> Batch overwrite pending flag (the TKBRSFn flag) is 1 by writing 1 to the TKBRDTn bit.

   <4> When a counter clear is generated by an external trigger input while the TKBTSEn bit is set to 1 and the TKBRSFn flag is 1, the setting value in the compare register is transferred to the buffer register. At the same time, the TKBRSFn flag becomes 0.

   <5> Even if the counter clear event is generated by the external trigger input, the batch overwrite does not occur unless a value 1 is written in the TKBRDTn bit.

Figure 15 - 51 Batch Overwrite Function: Figure of Standalone for External Trigger Input Factor and the Timing of Buffer Updating During Counting Operation (TKBTSEn Bit Set to 1)



3. Batch overwrite function (period controlled by external trigger input, buffer updating during counting operation (TKBTSEn bit clear to 0))

   This is an example of the case where the TKBTSEn bit in the TKBCTLn0 register is set to 0 during standalone operation under the periodic control by external trigger input. In this case, when a value 1 is written in the TKBRDTn bit and the external trigger input is detected while batch overwrite suspension flag (the TKBRSFn flag) is 1, the counter is cleared whereas compare register batch overwrite is not implemented.

   The source of the external trigger input is selected by setting the following bits.

   - The TKBSTSn[1:0] bits in the TKBCTLn0 register
   - The TKBINSn[2:0] bits in the TKBCTLn2 register
   - The TKBKCIn[2:0] bits in the TKBCTLn2 register

   Figure 15 - 52 shows an example of the batch overwrite operation timing when the TKBTSEn bit is set to 0.

   <1> Compare register setting is transferred to buffer register at the timing when the TKBCEn bit is set from 0 to 1 and TKBCNTn starts counting operation.

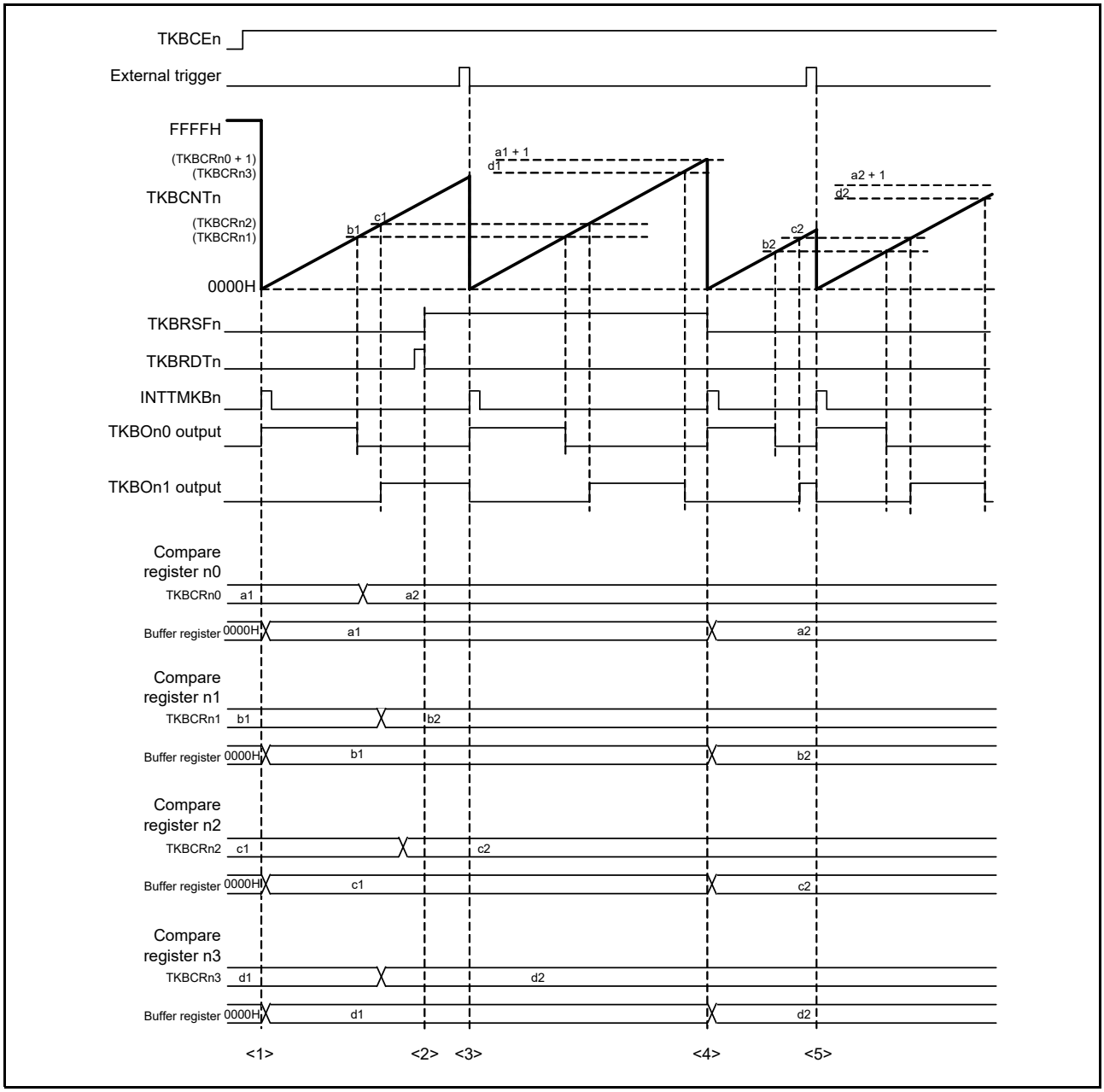
   <2> After rewriting the TKBCRmn and TKBTGCRn registers, batch overwrite pending flag (the TKBRSFn flag) becomes 1 by writing 1 to the TKBRDTn bit.

   <3> Even if the counter clear event is generated by the external trigger input, the batch overwrite does not occur unless the TKBTSEn bit is 1.

   <4> When the counter clear event (the TKBCNTn counter matches with the TKBCRn0 register) occurs under the status of the TKBRSFn flag is 1, the value set to the compare register is transferred to the buffer register. At the same time, the TKBRSFn flag becomes 0.

   <5> Even if the counter clear event is generated by the external trigger input, the batch overwrite does not occur unless the TKBTSEn bit and the TKBRSFn flag are both 1.

Figure 15 - 52 Batch Overwrite Function: Figure of Standalone for External Trigger Input Factor and the Timing of Buffer Updating During Counting Operation (TKBTSEn Bit Clear to 0)



4. Sample of register settings in standalone mode (period controlled by external trigger input)

TKBCTLn0

15	14	13	12	11	10	9	8
—	TKBGTEn1	TKBSSEn1	TKBDIEn1	—	TKBGTEn0	TKBSSEn0	TKBDIEn0
0	1/0	1/0	1/0	0	1/0	1/0	1/0

7	6	5	4	3	2	1	0
TKBMFEn	—	TKBIRSn1	TKBIRSn0	—	TKBTSEn	TKBSTSn1	TKBSTSn0
1/0	0	0	0	0	1/0	1/0	1/0

TKBCTLn1

7	6	5	4	3	2	1	0
TKBCEn	—	—	TKBCKSn	TKBSCMn	—	TKBMDn1	TKBMDn0
1/0	0	0	1/0	0	0	0	0

TKBIOCn0

7	6	5	4	3	2	1	0
—	—	—	—	TKBTOLn1	TKBTOLn0	TKBTODn1	TKBTODn0
0	0	0	0	1/0	1/0	1/0	1/0

TKBIOCn1

7	6	5	4	3	2	1	0
—	—	—	—	—	—	TKBTOEn1	TKBTOEn0
0	0	0	0	0	0	1/0	1/0

TKBCTLn2

15	14	13	12	11	10	9	8
—	—	—	—	—	—	TKBMFMn1	TKBMFMn0
0	0	0	0	0	0	1/0	1/0

7	6	5	4	3	2	1	0
—	TKBINSn2	TKBINSn1	TKBINSn0	—	TKBKCIn2	TKBKCIn1	TKBKCIn0
0	1/0	1/0	1/0	0	1/0	1/0	1/0

TKBCRn0	0000H to FFFFH
TKBCRn1	0000H to FFFFH
TKBCRn2	0000H to FFFFH
TKBCRn3	0000H to FFFFH
TKBTGCRn	0000H to FFFFH
TKBSIRn0	0000H to FFFFH
TKBSIRn1	0000H to FFFFH
TKBSSRn0	00H to 0FH
TKBSSRn1	00H to 0FH
TKBDNRn0	00H to F0H
TKBDNRn1	00H to F0H
TKBMFRn	0000H to FFFFH

: Setting is fixed for this mode : Setting is not needed (default setting)

15.4.8 Simultaneous start/stop mode

1. Outline of functions

Slave 16-bit timer KB3m can be start/stop simultaneously by synchronization with count start/stop of master 16-bit timer KB30 when master/slave is configured using multiple 16-bit timers KB3n.
Select “Standalone Mode (TKBMD0[1:0] = 00B)” for master and “Simultaneous Start/Stop Mode (TKBMD0[1:0] = 01B)” for slave in such case.
Only the start/stop timing of master and slave is synchronized in case of simultaneous start/stop mode.
When different count clock (CK0/CK1) is selected between master and slave, counting operation start timing for master and slave can be arranged through setting master TKBSCM0 bit to 1.
Each timer operates separately after the timing for counting operation to be started.
The TKBSCM0 bit is set for only master.

Caution 1. Master is 16-bit timer KB30 only.
Caution 2. Master selecting clock must be faster or with the same speed as slave selecting clock.

Relationship of Selected Clock between Master and Slave	Relationship between CK0 and CK1	TKBSCM0 Bit of Master	Available
Selecting the same clock for master and slave	—	0	✓
Selecting different clocks for master and slave	When master selected clock is faster than slave selected clock	1	✓
Selecting different clocks for master and slave	When master selected clock is slower than slave selected clock	—	×

For the formula to calculate TKBOn0/TKBOn1 outputs in case of simultaneous start/stop mode see **15.4.6 Standalone mode (period controlled by the TKBCRn0 register)** and **15.4.7 Standalone mode (period controlled by external trigger input)**.

2. Simultaneous start/stop mode

Master: Sample of register settings in standalone mode

TKBCTL00

15	14	13	12	11	10	9	8
—	TKBGTE01	TKBSSE01	TKBDIE01	—	TKBGTE00	TKBSSE00	TKBDIE00
0	1/0	1/0	1/0	0	1/0	1/0	1/0

7	6	5	4	3	2	1	0
TKBMFE0	—	TKBIRS01	TKBIRS00	—	TKBTSE0	TKBSTS01	TKBSTS00
1/0	0	0	0	0	1/0	1/0	1/0

TKBCTL01

7	6	5	4	3	2	1	0
TKBCE0	—	—	TKBCKS0	TKBSCM0	—	TKBMD01	TKBMD00
1/0	0	0	1/0	1/0	0	0	0

TKBIOC00

7	6	5	4	3	2	1	0
—	—	—	—	TKBTOL01	TKBTOL00	TKBTOD01	TKBTOD00
0	0	0	0	1/0	1/0	1/0	1/0

TKBIOC01

7	6	5	4	3	2	1	0
—	—	—	—	—	—	TKBTOE01	TKBTOE00
0	0	0	0	0	0	1/0	1/0

TKBCTL02

15	14	13	12	11	10	9	8
—	—	—	—	—	—	TKBMFM01	TKBMFM00
0	0	0	0	0	0	1/0	1/0

7	6	5	4	3	2	1	0
—	TKBINS02	TKBINS01	TKBINS00	—	TKBKCI02	TKBKCI01	TKBKCI00
0	1/0	1/0	1/0	0	1/0	1/0	1/0

TKBCR00	0000H to FFFFH
TKBCR01	0000H to FFFFH
TKBCR02	0000H to FFFFH
TKBCR03	0000H to FFFFH
TKBTGCR0	0000H to FFFFH
TKBSIR00	0000H to FFFFH
TKBSIR01	0000H to FFFFH
TKBSSR00	00H to 0FH
TKBSSR01	00H to 0FH
TKBDNR00	00H to F0H
TKBDNR01	00H to F0H
TKBMFR0	0000H to FFFFH

: Setting is fixed for this mode : Setting is not needed (default setting)

3. Simultaneous start/stop mode

Slave: Sample of register settings in simultaneous start/stop mode

TKBCTLk0

15	14	13	12	11	10	9	8
—	TKBGTEk1	TKBSSEk1	TKBDIEk1	—	TKBGTEk0	TKBSSEk0	TKBDIEk0
0	1/0	1/0	1/0	0	1/0	1/0	1/0

7	6	5	4	3	2	1	0
TKBMFEk	—	TKBIRSk1	TKBIRSk0	—	TKBTSEk	TKBSTSk1	TKBSTSk0
1/0	0	0	0	0	1/0	1/0	1/0

TKBCTLk1

7	6	5	4	3	2	1	0
TKBCEk	—	—	TKBCKSk	TKBSCMk	—	TKBMDk1	TKBMDk0
0	0	0	1/0	0	0	0	1

TKBIOCk0

7	6	5	4	3	2	1	0
—	—	—	—	TKBTOLk1	TKBTOLk0	TKBTODk1	TKBTODk0
0	0	0	0	1/0	1/0	1/0	1/0

TKBIOCk1

7	6	5	4	3	2	1	0
—	—	—	—	—	—	TKBTOEk1	TKBTOEk0
0	0	0	0	0	0	1/0	1/0

TKBCTLk2

15	14	13	12	11	10	9	8
—	—	—	—	—	—	TKBMFMk1	TKBMFMk0
0	0	0	0	0	0	1/0	1/0

7	6	5	4	3	2	1	0
—	TKBINSk2	TKBINSk1	TKBINSk0	—	TKBKCIk2	TKBKCIk1	TKBKCIk0
0	1/0	1/0	1/0	0	1/0	1/0	1/0

TKBCRk0	0000H to FFFFH
TKBCRk1	0000H to FFFFH
TKBCRk2	0000H to FFFFH
TKBCRk3	0000H to FFFFH
TKBTGCRk	0000H to FFFFH
TKBSIRk0	0000H to FFFFH
TKBSIRk1	0000H to FFFFH
TKBSSRk0	00H to 0FH
TKBSSRk1	00H to 0FH
TKBDNRk0	00H to F0H
TKBDNRk1	00H to F0H
TKBMFRk	0000H to FFFFH

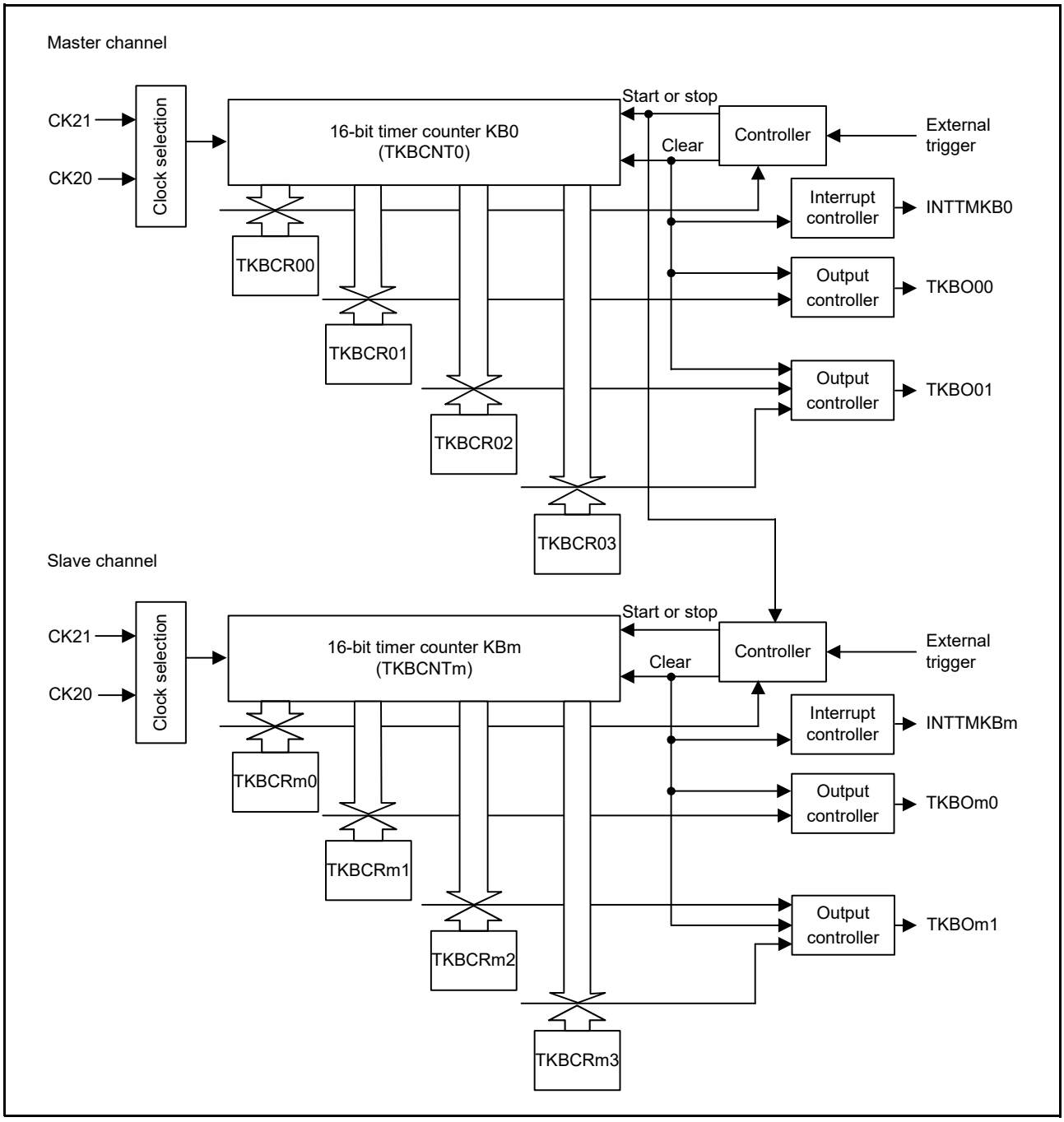
: Setting is fixed for this mode : Setting is not needed (default setting)

Remark k = 1, 2

4. Configuration in simultaneous start/stop mode

Figure 15 - 53 shows configuration in simultaneous start/stop mode.

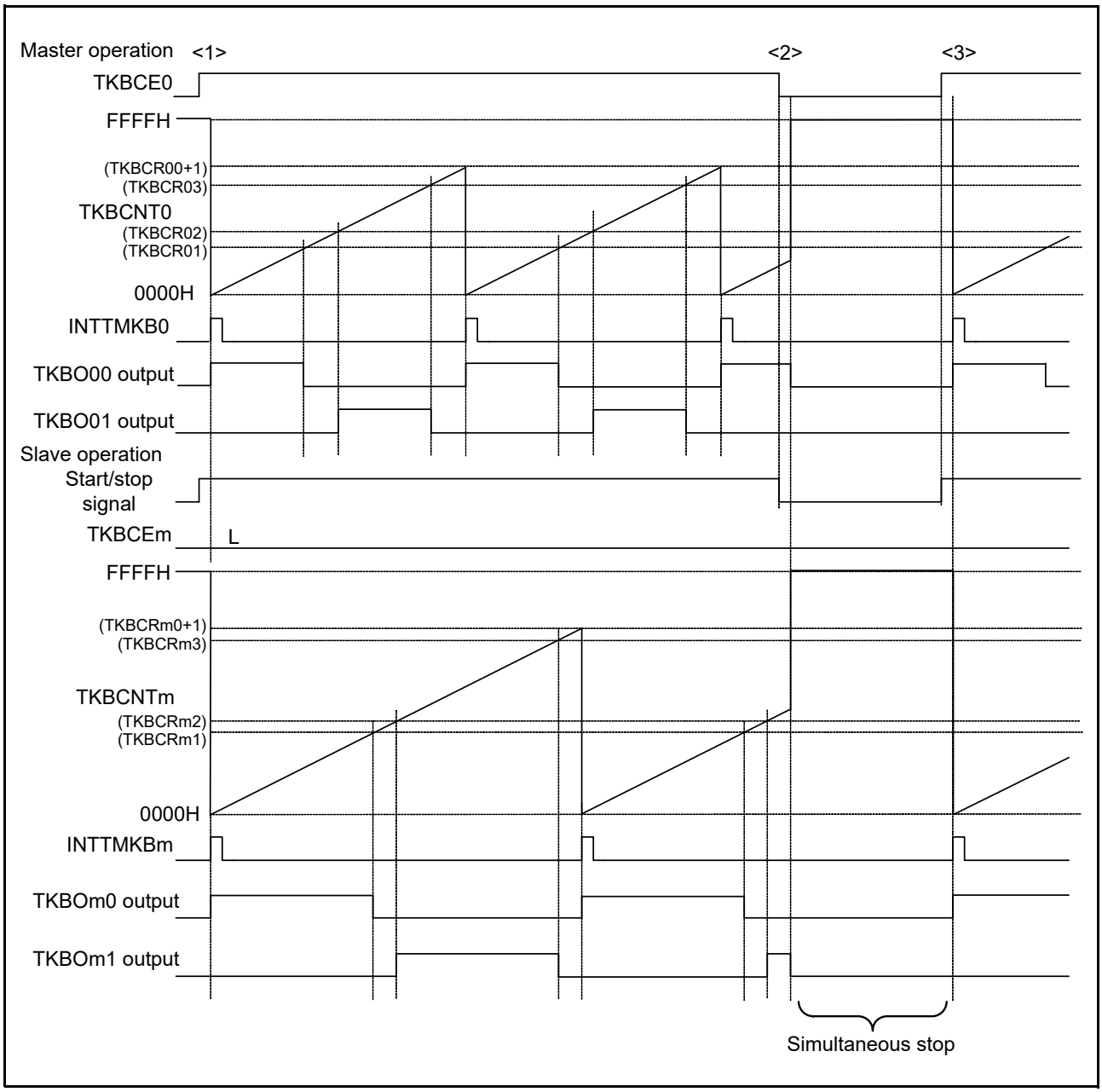
Figure 15 - 53 Configuration in Simultaneous Start/Stop Mode



5. Outline of operation

Figure 15 - 54 shows an example of the operation timing in simultaneous start/stop mode.

Figure 15 - 54 Sample Timing in Simultaneous Start/Stop Mode (Period Controlled by TKBCR00 register) (at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



The following describes an operational example of simultaneous start/stop mode. <1> to <3> in **Figure 15 - 54** are described below.

<1> When the master TKBCE0 bit is set to 1, the master 16-bit timer counter KB0 (TKBCNT0) and the slave 16-bit timer counter KBm (TKBCNTm) change from FFFFH to 0000H upon synching with the count clock and they start upward counting. At the same time, the master and slave generate INTTMKB0 and INTTMKBm respectively and TKBO00 and TKBOm0 output change from their initial value to active value (in this example, it's high level). For further detailed operation, see **Figure 15 - 47 Example of Timing in Standalone Mode**.

<2> If TKBCE0 is set to 0, synching with the input clock of the 16-bit timer KB3n, TKBCNT0 counter of the master and the TKBCNTm counter of the slave stop its upward count and sets to FFFFH. At the same time, the output of both master and slave change to their default levels. This status is maintained until TKBCE0 bit of the master is set as 1.

<3> If TKBCE0 bit of the master is set as 1, the same sequence of operation starting from <1> is repeated.

15.4.9 Simultaneous start/clear mode

1. Outline of functions

Slave 16-bit timer KB3m can be started and cleared simultaneously by synchronization with timings of start/stop of counting by master 16-bit timer KB30, counter clearing and batch overwriting when master/slave is configured using multiple 16-bit timers KB3n.
Operate the master in standalone mode (TKBMD0[1:0] = 00B) and the slave in simultaneous start/clear mode (TKBMDm[1:0] = 10B) in such case.
Select the same division clock for Master/Slave in TKBCKS0 and TKBCKSm bits.

See **15.4.6 Standalone mode (period controlled by the TKBCRn0 register)** for the calculation of master TKBO00/TKBO01 output.
Batch overwriting is controlled by writing 1 to master TKBRDT0 bit.
Verifying of master TKBRSF0 flag is needed to read TKBRSF0 flag.

Slave TKBCNTm counter is cleared at the same timing for clearing of master TKBCNT0 counter.
Batch overwriting for Slave compare register is executed at the same timing for master batch overwriting.

The role of slave TKBCRm0 register is shifted to register which sets TKBOm0 active timing as slave operates according to the period generated by master TKBCR00 register.
The INTTMKBm signal are generated when the TKBCNT0 counter starts counting and when it is cleared. When skipping control is not to be used, the other INTTMKBm signals have the same output timing as the INTTMKB0 signal.

The slave duty is calculated by following formula and able to be set within range of 0% to 100%.

[Calculation formula for slave TKBOm0 output]

Pulse period = (Master setting TKBCR00 + 1) × Count clock period
Duty [%] = ((Setting value of TKBCRm1 – Setting value of TKBCRm0)/(Setting value of master TKBCR00 + 1)) × 100
0% output: TKBCRm1 setting = TKBCRm0 setting
100% output: TKBCRm0 setting = 0000H, TKBCRm1 setting ≥ Master TKBCR00 setting + 1

Caution Be sure to set a value of TKBCRm0 ≤ set value of TKBCRm1.

[Calculation formula for slave TKBOm1 output]

Pulse period = (Master TKBCR00 setting + 1) × Count clock period
Duty [%] = ((Setting value of TKBCRm3 – Setting value of TKBCRm2)/(Setting value of master TKBCR00 + 1)) × 100
0% output: TKBCRm3 setting = TKBCRm2 setting
100% output: TKBCRm2 setting = 0000H, Setting value of TKBCRm3 ≥ Setting value of master TKBCR00 + 1

Caution Be sure to set value of TKBCRm2 ≤ set value of TKBCRm3

Remark m = 1, 2

2. Simultaneous start/clear mode

   Master: Sample register settings in standalone mode

TKBCTL00

15	14	13	12	11	10	9	8
—	TKBGTE01	TKBSSE01	TKBDIE01	—	TKBGTE00	TKBSSE00	TKBDIE00
0	1/0	1/0	1/0	0	1/0	1/0	1/0

7	6	5	4	3	2	1	0
TKBMFE0	—	TKBIRS01	TKBIRS00	—	TKBTSE0	TKBSTS01	TKBSTS00
1/0	0	0	0	0	1/0	1/0	1/0

TKBCTL01

7	6	5	4	3	2	1	0
TKBCE0	—	—	TKBCKS0	TKBSCM0	—	TKBMD01	TKBMD00
1/0	0	0	1/0	0	0	0	0

TKBIOC00

7	6	5	4	3	2	1	0
—	—	—	—	TKBTOL01	TKBTOL00	TKBTOD01	TKBTOD00
0	0	0	0	1/0	1/0	1/0	1/0

TKBIOC01

7	6	5	4	3	2	1	0
—	—	—	—	—	—	TKBTOE01	TKBTOE00
0	0	0	0	0	0	1/0	1/0

TKBCTL02

15	14	13	12	11	10	9	8
—	—	—	—	—	—	TKBMFM01	TKBMFM00
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
—	TKBINS02	TKBINS01	TKBINS00	—	TKBKCI02	TKBKCI01	TKBKCI00
0	1/0	1/0	1/0	0	1/0	1/0	1/0

TKBCR00	0000H to FFFFH
TKBCR01	0000H to FFFFH
TKBCR02	0000H to FFFFH
TKBCR03	0000H to FFFFH
TKBTGCR0	0000H to FFFFH
TKBSIR00	0000H to FFFFH
TKBSIR01	0000H to FFFFH
TKBSSR00	00H to 0FH
TKBSSR01	00H to 0FH
TKBDNR00	00H to F0H
TKBDNR01	00H to F0H
TKBMFR0	0000H to FFFFH

: Setting is fixed for this mode : Setting is not needed (default setting)

3. Simultaneous start/clear mode

Slave: Sample register settings in simultaneous start/clear mode

TKBCTLk0

15	14	13	12	11	10	9	8
—	TKBGTE01	TKBSSEk1	TKBDIEk1	—	TKBGTE00	TKBSSEk0	TKBDIEk0
0	1/0	1/0	1/0	0	1/0	1/0	1/0

7	6	5	4	3	2	1	0
TKBMFEk	—	TKBIRSk1	TKBIRSk0	—	TKBTSEk	TKBSTSk1	TKBSTSk0
0	0	0	0	0	0	0	0

TKBCTLk1

7	6	5	4	3	2	1	0
TKBCEk	—	—	TKBCKSk	TKBSCMk	—	TKBMDk1	TKBMDk0
0	0	0	1/0	0	0	1	0

TKBIOCk0

7	6	5	4	3	2	1	0
—	—	—	—	TKBTOLk1	TKBTOLk0	TKBTODk1	TKBTODk0
0	0	0	0	1/0	1/0	1/0	1/0

TKBIOCk1

7	6	5	4	3	2	1	0
—	—	—	—	—	—	TKBTOEk1	TKBTOEk0
0	0	0	0	0	0	1/0	1/0

TKBCTLk2

15	14	13	12	11	10	9	8
—	—	—	—	—	—	TKBMFMk1	TKBMFMk0
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
—	TKBINSk2	TKBINSk1	TKBINSk0	—	TKBKCIk2	TKBKCIk1	TKBKCIk0
0	0	0	0	0	0	0	0

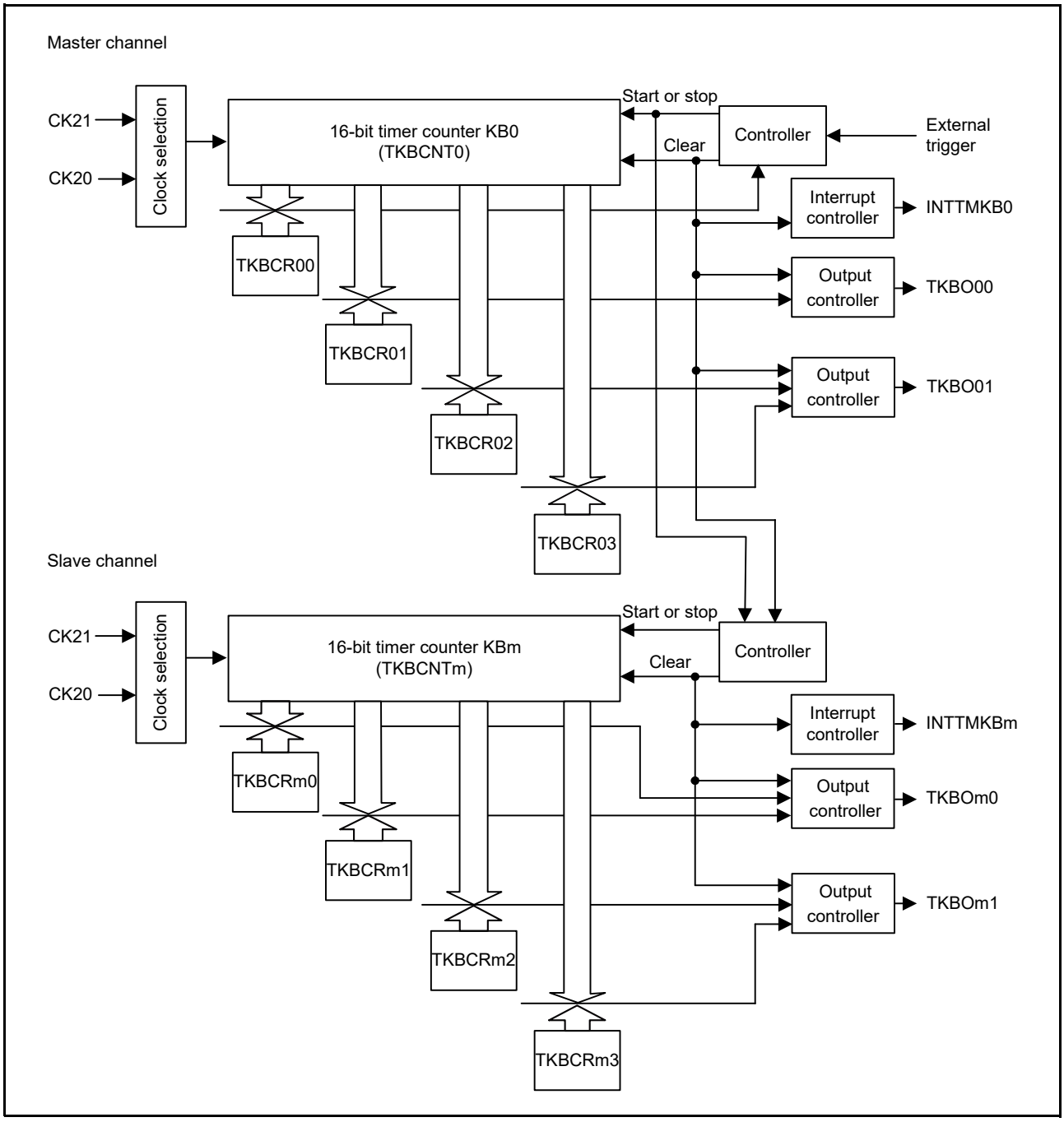
TKBCRk0	0000H to FFFFH
TKBCRk1	0000H to FFFFH
TKBCRk2	0000H to FFFFH
TKBCRk3	0000H to FFFFH
TKBTGCRk	0000H to FFFFH
TKBSIRk0	0000H to FFFFH
TKBSIRk1	0000H to FFFFH
TKBSSRk0	00H to 0FH
TKBSSRk1	00H to 0FH
TKBDNRk0	00H to F0H
TKBDNRk1	00H to F0H
TKBMFRk	0000H

: Setting is fixed for this mode : Setting is not needed (default setting)

Remark k = 1, 2

4. Configuration in simultaneous start/clear mode (period controlled by master)
   Figure 15 - 55 shows configuration in simultaneous start/clear mode.

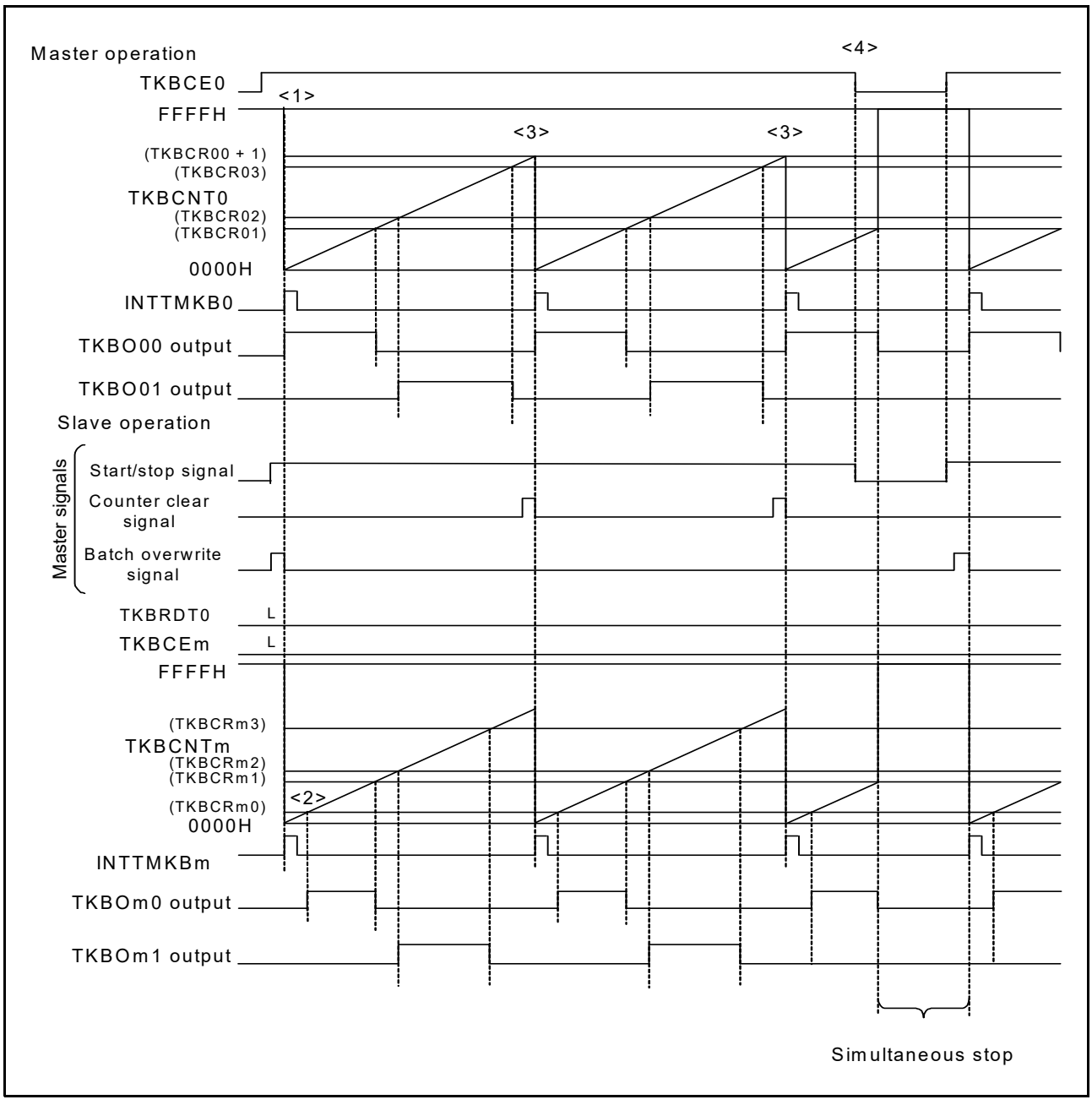
Figure 15 - 55 Configuration in Simultaneous Start/Clear Mode (Period Controlled by Master)



5. Outline of operation

Figure 15 - 56 shows an example of the operation timing in simultaneous start/clear mode.

Figure 15 - 56 Sample Timing in Simultaneous Start/Clear Mode (Period Controlled by Master) (at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



The following describes an operational example of simultaneous start/clear mode. <1> to <4> in **Figure 15 - 56** are described below.

<1> When the master TKBCE0 bit is set to 1, the master 16-bit timer counter KB0 (TKBCNT0) and the slave 16-bit timer counter KBm (TKBCNTm) change from FFFFH to 0000H upon synching with the count clock and they start upward counting. At the same time, the master generate INTTMKB0 respectively and TKBO00 output change from their initial value to active value (in this example, it's high level).

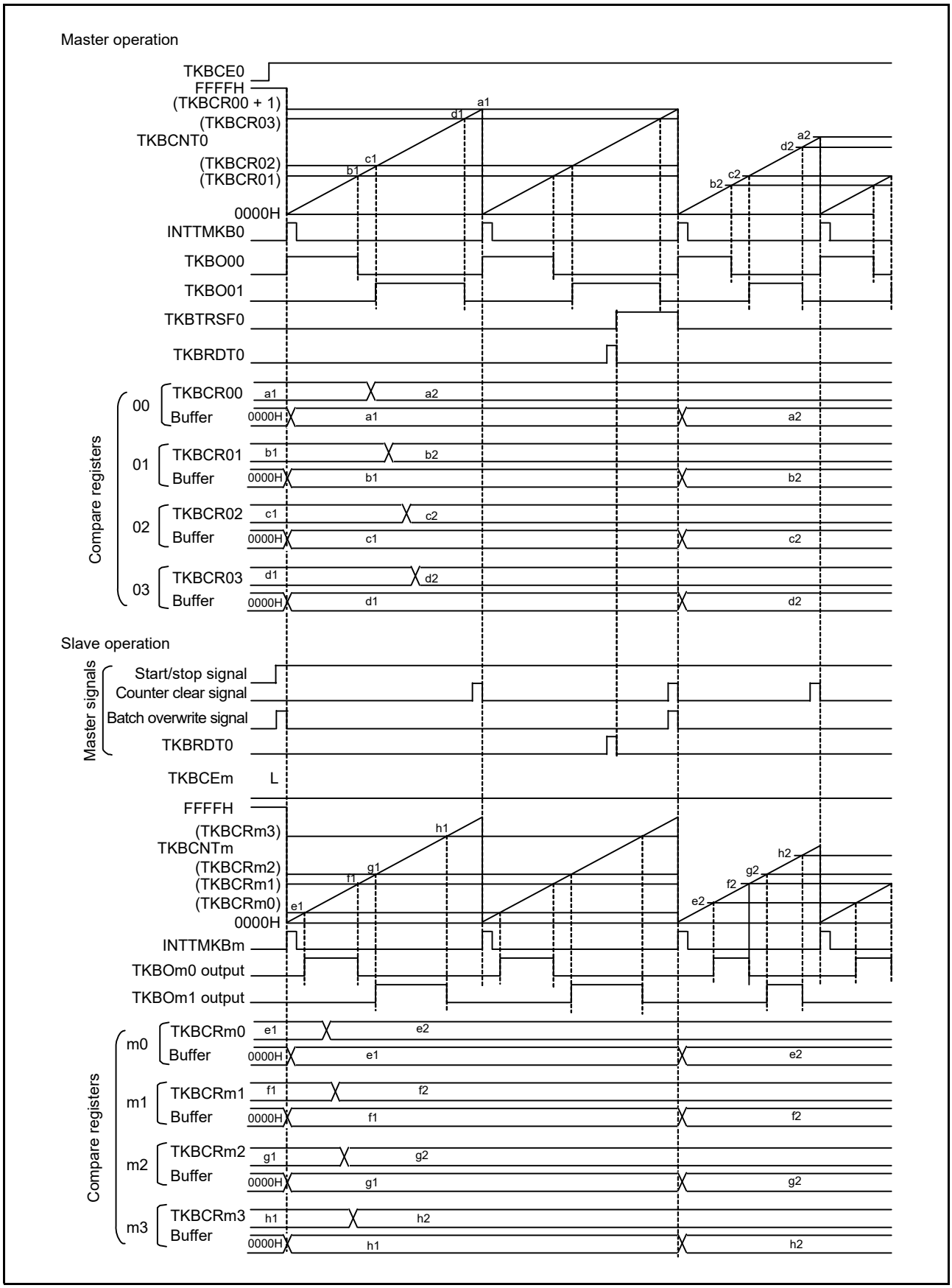
<2> When count value of the TKBCNTm counter matches with the value specified in the TKBCRm0 register, TKBOm0 output of slave becomes active level. For further detailed operation, see **Figure 15 - 47 Example of Timing in Standalone Mode**.

<3> When count value of TKBCNT0 counter matches with the value specified in TKBCR00 register, clear signal of master is output. At the same time, 16-bit timer counter (TKBCNT0, TKBCNTm) for master and slave is cleared.

<4> If TKBCE0 bit is set to 0, synching with the input clock of the 16-bit timer KB3n, TKBCNT0 counter of the master and the TKBCNTm counter of the slave stop its upward count and sets to FFFFH. At the same time, the output of both master and slave change to their default levels. This status is maintained until TKBCE0 bit of the master is set as 1.

Figure 15 - 57 shows an example of the operation timing when batch overwriting in simultaneous start/clear mode. In this case, TKBRDT0 bit for master set to 1, at the same time batch overwriting by slave in next clear timing.

Figure 15 - 57 Sample Timing in Simultaneous Start/Clear Mode (Period Controlled by Master) (at Batch Overwrite)



15.4.10 Interleaved power factor correction (PFC) output mode

1. Outline of functions

This is the mode that can generate a signal as interleaved output that controls PFC circuit which regulates the harmonic current of the power source.

As interleaved PFC circuit can regulate peak input current at greater extent than single PFC circuit, it can make parts smaller and implement high powered power source units. This mode supports critical conduction type PFC circuit.

Interleaved PFC control requires two inputs for zero current detection and two PWM outputs for switching. 16-bit timers KB30, KB31, and KB32 implement the interleaved PFC control by combining external interrupt input INTP20 and timer output TKBOn0, and interrupt input INTP21 and timer output TKBOn1.

TKBOn0 generates pulse output based on the signal input of INTP20, and TKBOn1 generates pulse output based on the signal input of INTP21.

In this case, it controls TKBOn1 output to be 180 degree of phase shift based on the output timing of TKBOn0.

Remark Single PFC control can be implemented in the standalone mode (period controlled by external trigger input). For more detail, see **15.4.7 Standalone mode (period controlled by external trigger input)**. PFC in the continuous current mode can be implemented in the standalone mode (period controlled by the TKBCRn0 register). For more detail, see **15.4.6 Standalone mode (period controlled by the TKBCRn0 register)**.

The restart period of the 16-bit timer KB3n is set by the TKBCRn0 register for cases in which external input INTP20 not being detected.

Active width for TKBOn0 output is set by the TKBCRn1 register.

Active width for TKBOn1 output is set by the TKBCRn3 register.

Remark Interleaved power factor correction (PFC) output mode does not use TKBCRn2.

The setting value for TKBTOLn0 bit and TKBTODn0 bit, and TKBTOLn1 bit and TKBTODn1 bit must be the same value. This makes that when the default level is low (high) level, the active level becomes high (low) level.

[Calculation formula for TKBOn0 output & TKBOn1 output]

Pulse period (max.)**Note 1** = (TKBCRn0 setting + 1) × Count clock period

Active width for TKBOn0 output = TKBCRn1 setting × Count clock period

Active width for TKBOn1 output = TKBCRn3 setting × Count clock period

Width of the cycle-to-cycle phase difference during TKBOn1 output**Note 2** = INT[(width of the previous period – 1)**Note 3**/2 + 1] × Count clock period

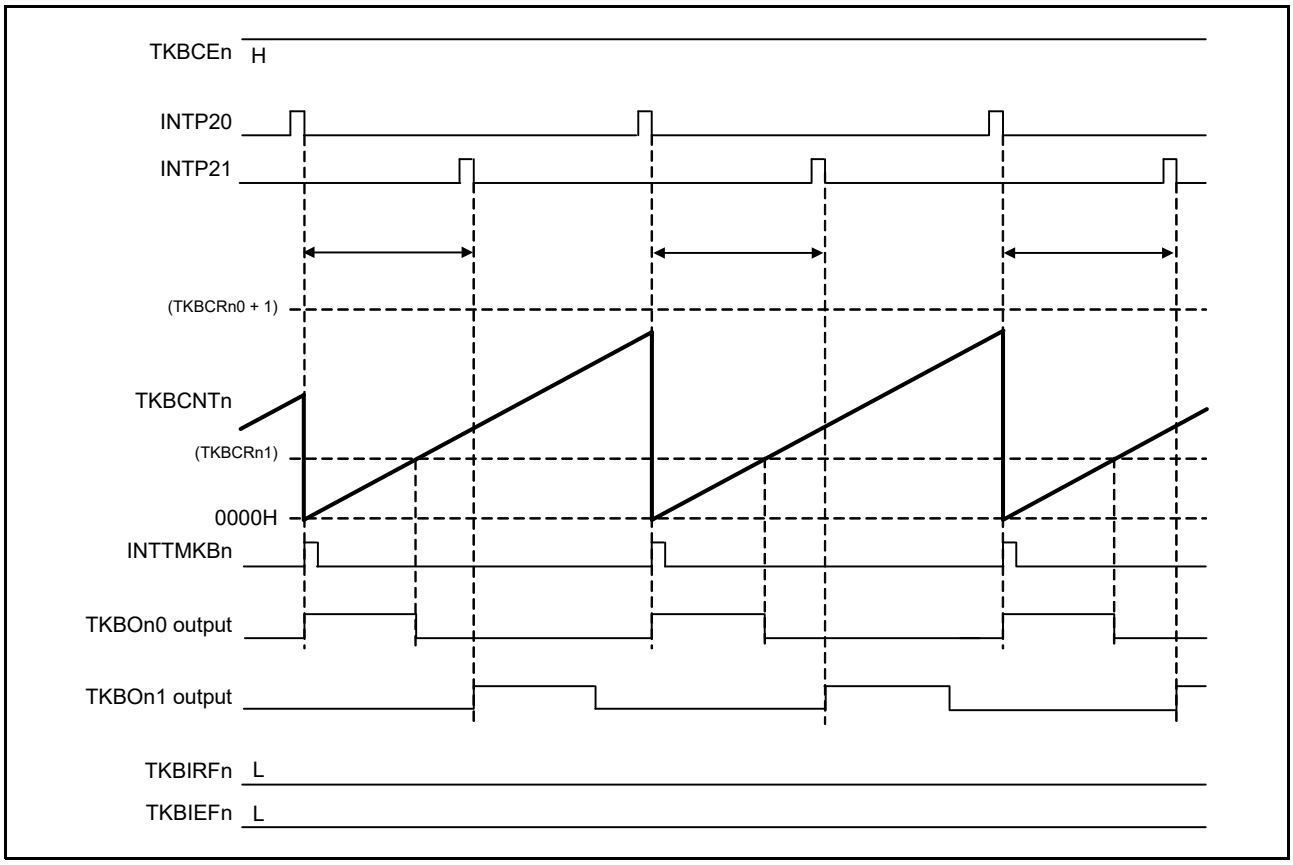
Note 1. This is the restart period of the 16-bit timer KB3n in case when external interrupt input INTP20 not being detected.

Note 2. Except when condition No.7 holds.

Note 3. When condition No.1 holds, this is determined by the setting of the TKBCRn0 register.

Figure 15 - 58 shows overview of the basic operation in interleaved PFC mode. In the basic operation in interleaved PFC mode, the TKBCNTn counter is incremented from 0000H by using INTP20 as a trigger. In this case, TKBOn0 becomes active level, then becomes inactive level when it matches with the setting value of TKBCRn1 register. TKBOn1 becomes active level by being triggered by INTP21 which has a phase shifted from the one of INTP20, and becomes inactive level when it matches with the setting value of TKBCRn3 register.
Another INTP20 comes in before the TKBCNTn counter matches with the setting value of TKBCRn0 register, then the above operation is repeated.

Figure 15 - 58 Outline of Basic Operation in Interleaved PFC Mode (at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



2. Output condition of TKBOn1 in interleaved PFC

There are output conditions for TKBOn1 output which are controlled according to the table below.

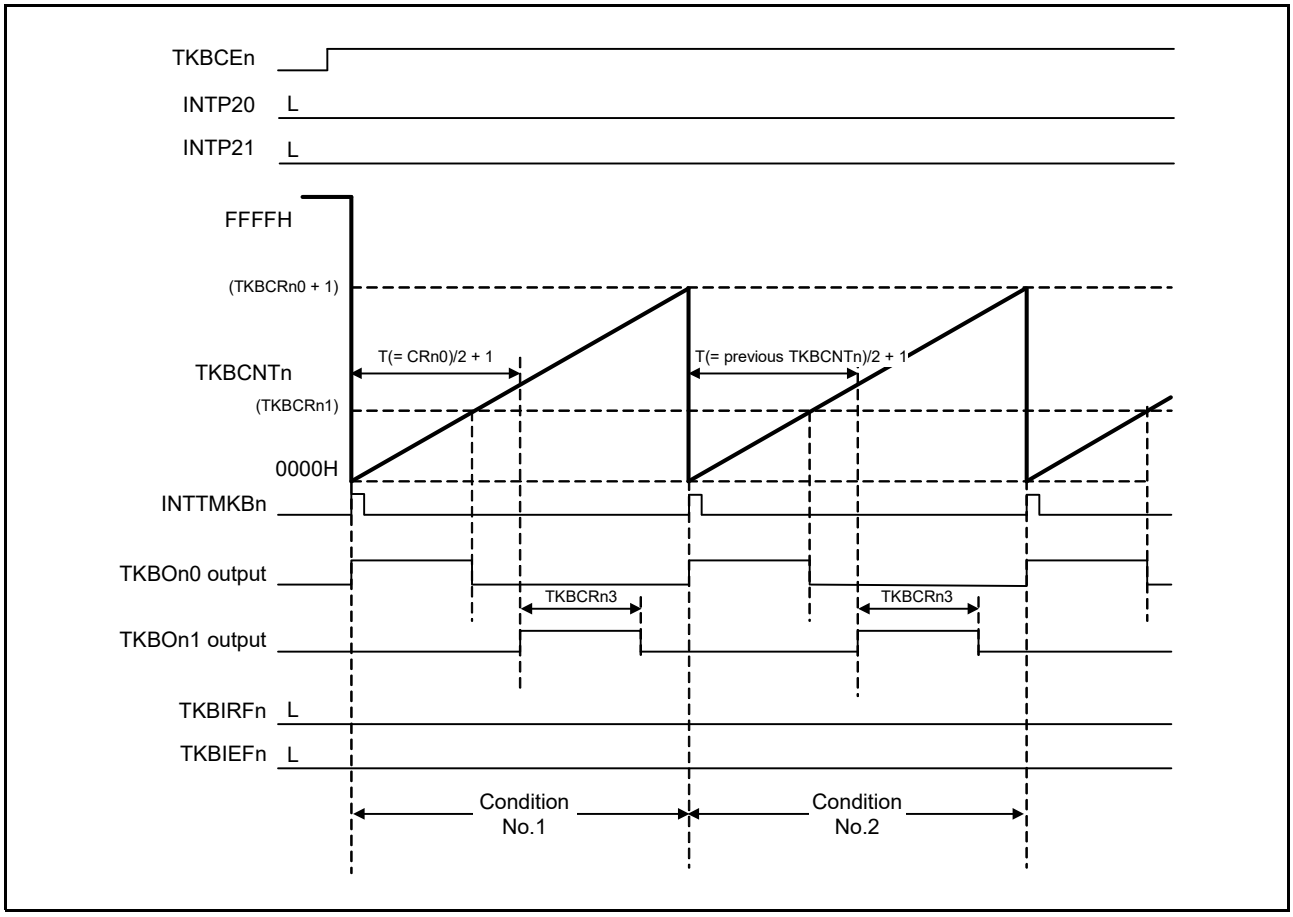
Condition No.	Judgment Status 1 (INTP20 Input)	Judgment Status 2 (Matching with CR0/INTP21 Input)	Judgment Status 3 (Period Width)	Output Status
1	First period (Generate a wave form setting T for CR0)	—	—	Output by T/2
2	INTP20 input not detected	matching of CNTn and CRn0 (Ignore INTP21 input detection)	Subsequent period is over T/2	Output by T/2
3	↑	↑	Succeeding period is below T/2	Maintain the status
4	Subsequent period of No.3	—	—	Output by T/2
5	INTP20 input detected (for the first time)**Note 1**	—	—	Output by T/2
6	INTP20 input detected (from the second time)**Note 2**	INTP21 detected (within the range from previous TKBOn1 falling edge to T/2)	—	Output by T/2
7	INTP20 input detected (from the second time)**Note 2**	INTP21 detected (T/2 to T/2+T/(TKBIRSn1 and TKBIRSn0 setting) range)	—	Output by Trigger Input
8	INTP20 input detected (from the second time)**Note 2**	INTP21 detected after the range (T/2+T/(TKBIRSn1 and TKBIRSn0 setting))	—	Maintain the status
9	Subsequent period of No.8	—	—	Output by T/2
10	INTP20 input detected	—	Succeeding period is below T/2	Maintain the status
11	Subsequent period of No.10	—	—	Output by T/2

Note 1. INTP20 input detected (for the first time) means that the previous period wasn't cleared for INTP20 input being detected.

Note 2. INTP20 input detected (from the second time) means that the previous period being cleared for INTP20 input being detected.

See the following figures of wave form corresponding to each “Condition No.”

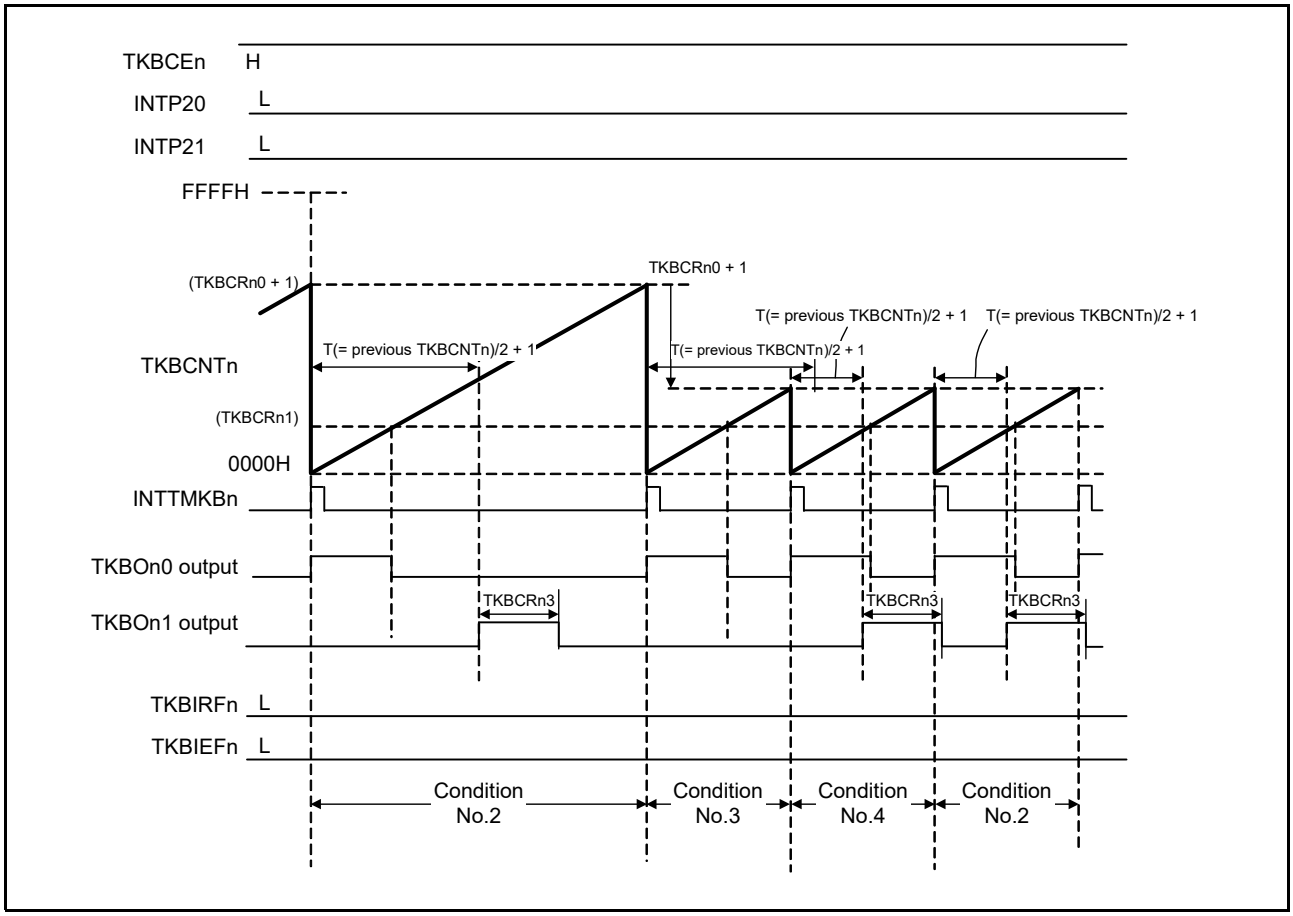
Figure 15 - 59 Figure of Timing of Interleaved PFC Mode (Operation for Conditions No.1 and No.2) (at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



Condition No.1 Only for the first period after TKBCEn = 1 setting, TKBOn1 with setting width of the TKBCRn3 register is output setting “T” as the TKBCRn0 register.

Condition No.2 In the second period, TKBOn1 with setting width of the TKBCRn3 register is output at T/2 of previous period.

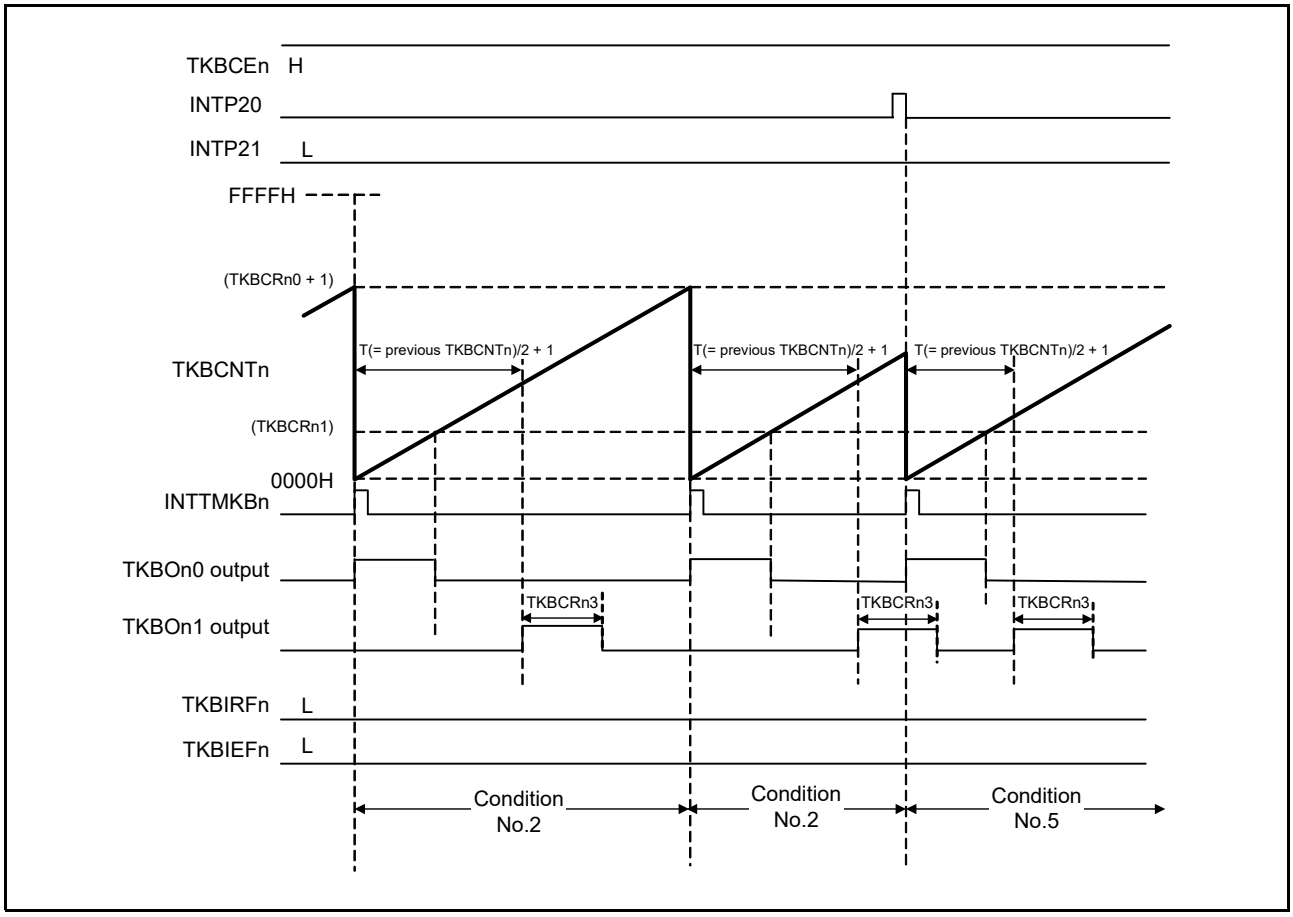
Figure 15 - 60 Figure of Timing of Interleaved PFC Mode (Operation for Conditions No.3 and No.4) (at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



Condition No.3 TKBOn1 keeps the status and T/2 of the previous period not ensured.

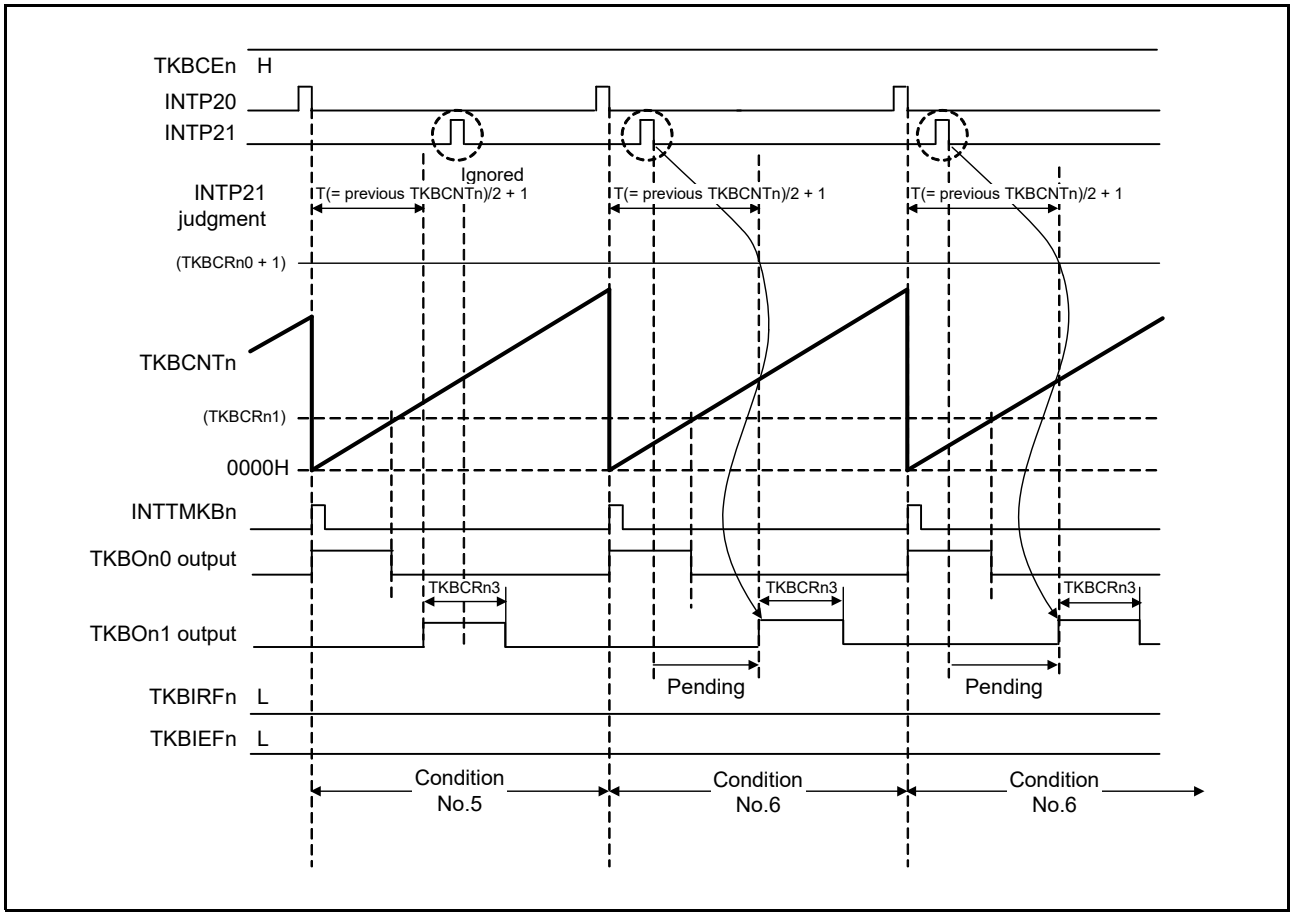
Condition No.4 TKBOn1 with setting width of the TKBCRn3 register is output at T/2 of previous period.

Figure 15 - 61 Figure of Timing of Interleaved PFC Mode (Operation for Condition No.5) (at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



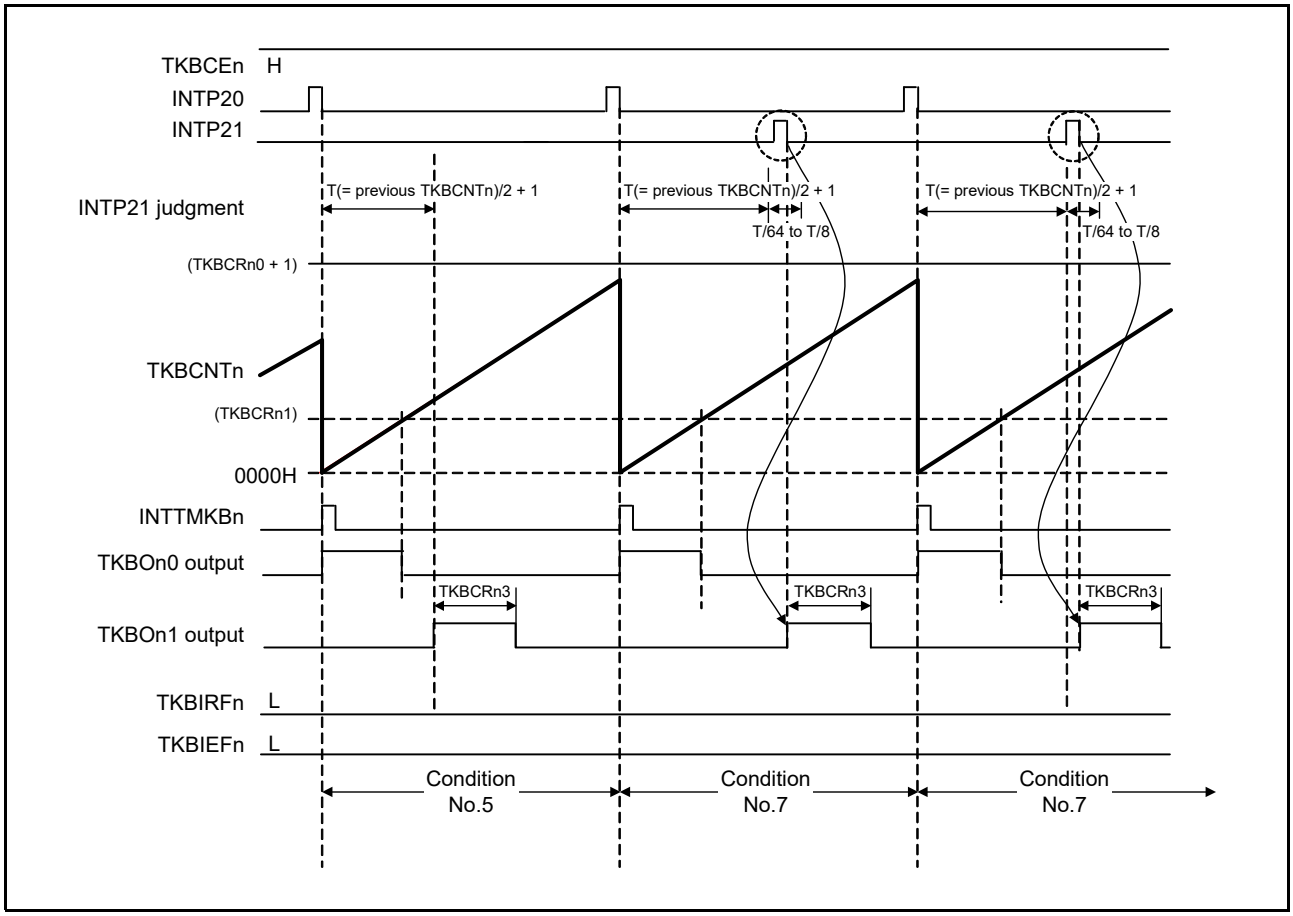
Condition No.5 INTP20 which was first detected after setting TKBCEn = 1 outputs TKBOn1 with setting width of the TKBCRn3 register.

Figure 15 - 62 Figure of Timing of Interleaved PFC Mode (Operation for Condition No.6) (at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



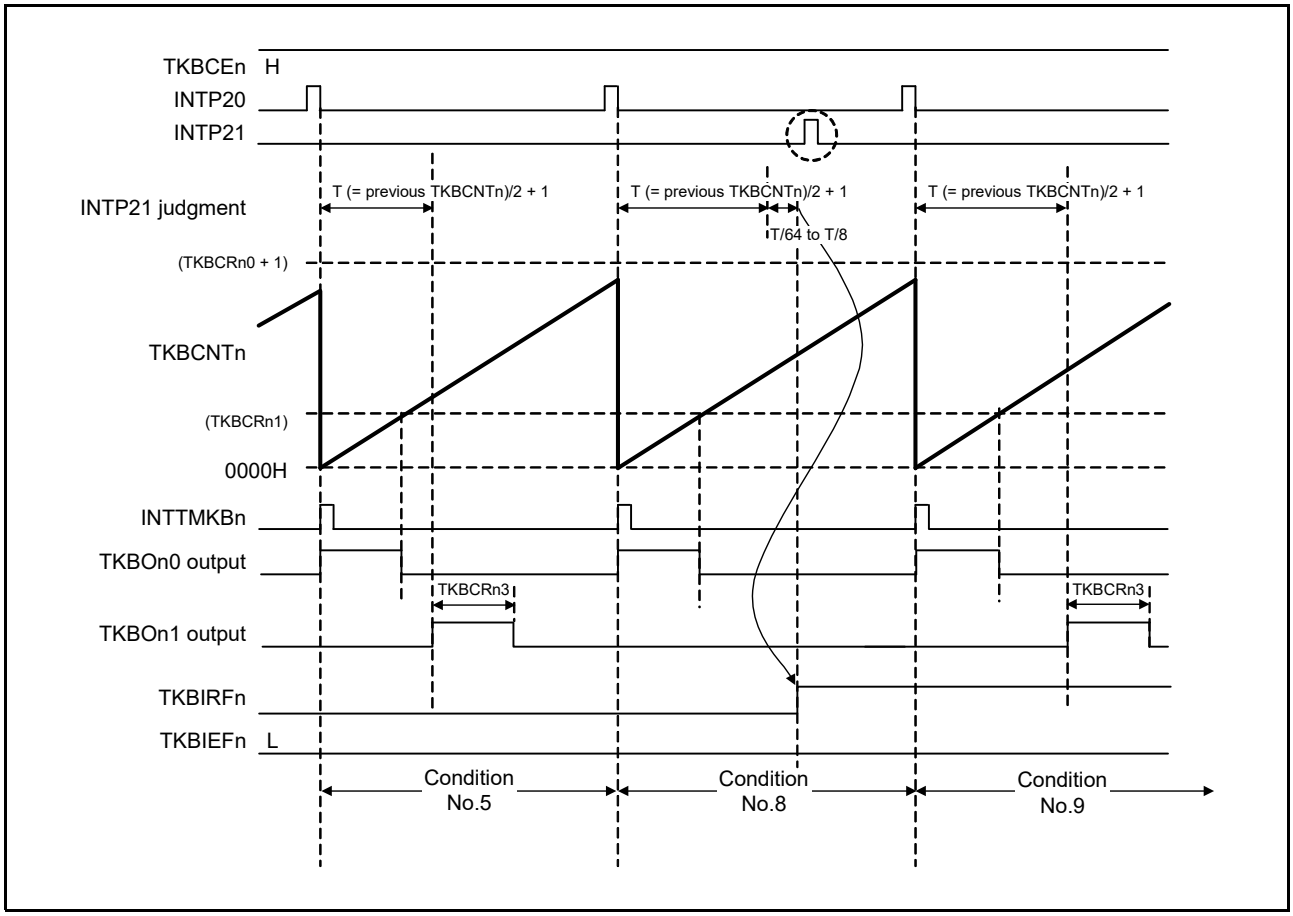
Condition No.6 TKBOn1 with setting width of the TKBCRn3 register is output at T/2 of previous period as INTP21 input is below T/2 of the previous period.

Figure 15 - 63 Figure of Timing of Interleaved PFC Mode (Operation for Condition No.7) (at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



Condition No.7 After the detection of INTP20 when INTP21 is detected over T/2 of the previous period and within T/2 + T/m (m stands for 8/16/32/64; set by the TKBIRSn[1:0] bits), TKBOn1 is output by setting width of the TKBCRn3 register.

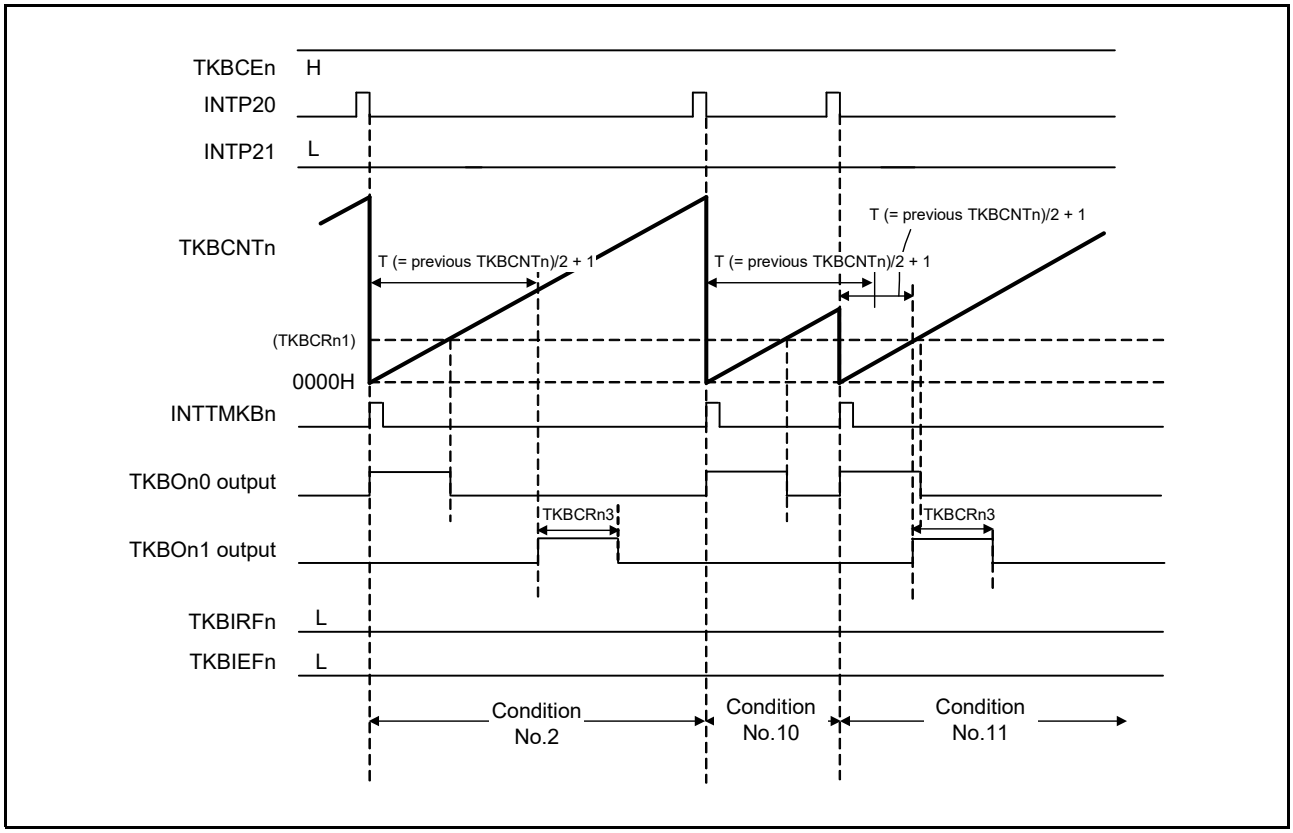
Figure 15 - 64 Figure of Timing of Interleaved PFC Mode (Operation for Conditions No.8 and No.9) (at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



Condition No.8 INTP21 wasn't detected within T/2 + T/m (m stands for 8/16/32/64; set at the TKBIRSn[1:0] bits) of the previous period and TKBOn1 maintains the status. Then the TKBIRFn flag is set by 1.

Condition No.9 TKBOn1 with setting width of the TKBCRn3 register is output at T/2 of previous period.

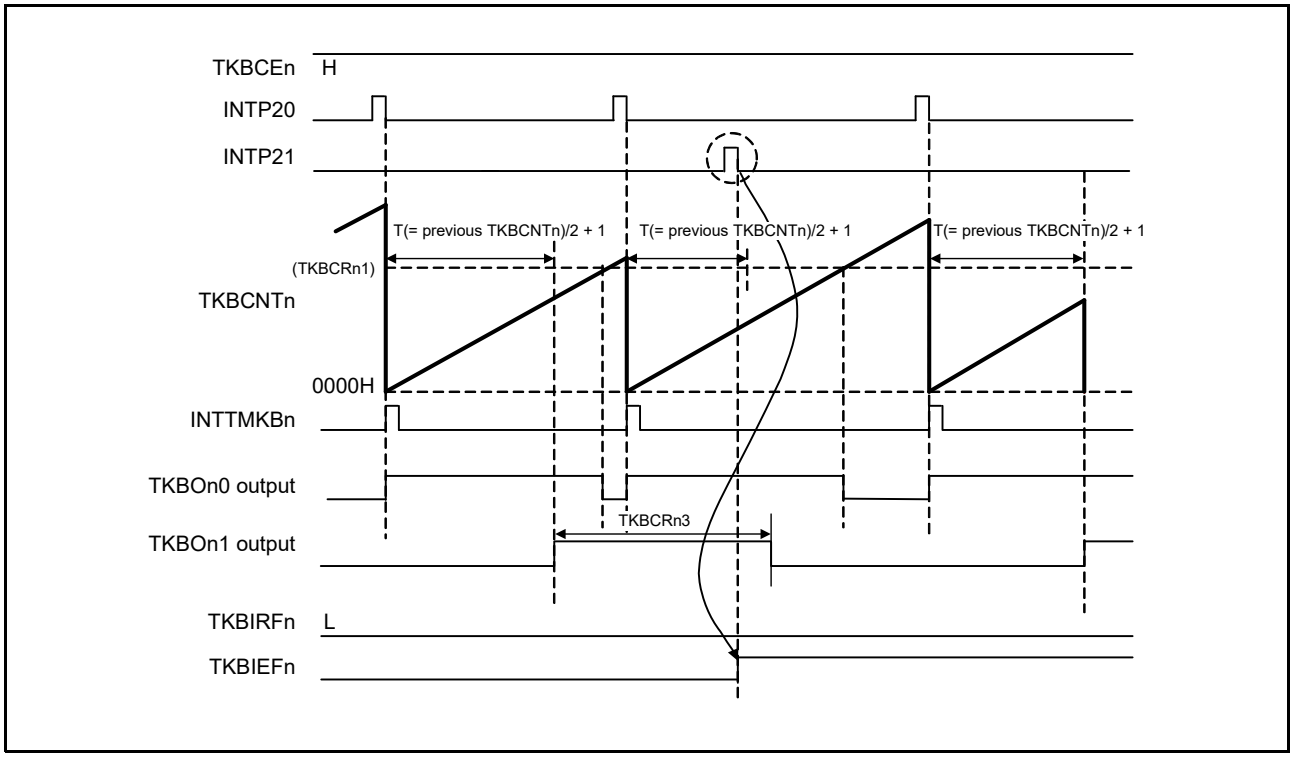
Figure 15 - 65 Figure of Timing of Interleaved PFC Mode (Operation for Conditions No.10 and No.11) (at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



Condition No.10 TKBOn1 keeps the status and T/2 of the previous period not ensured.

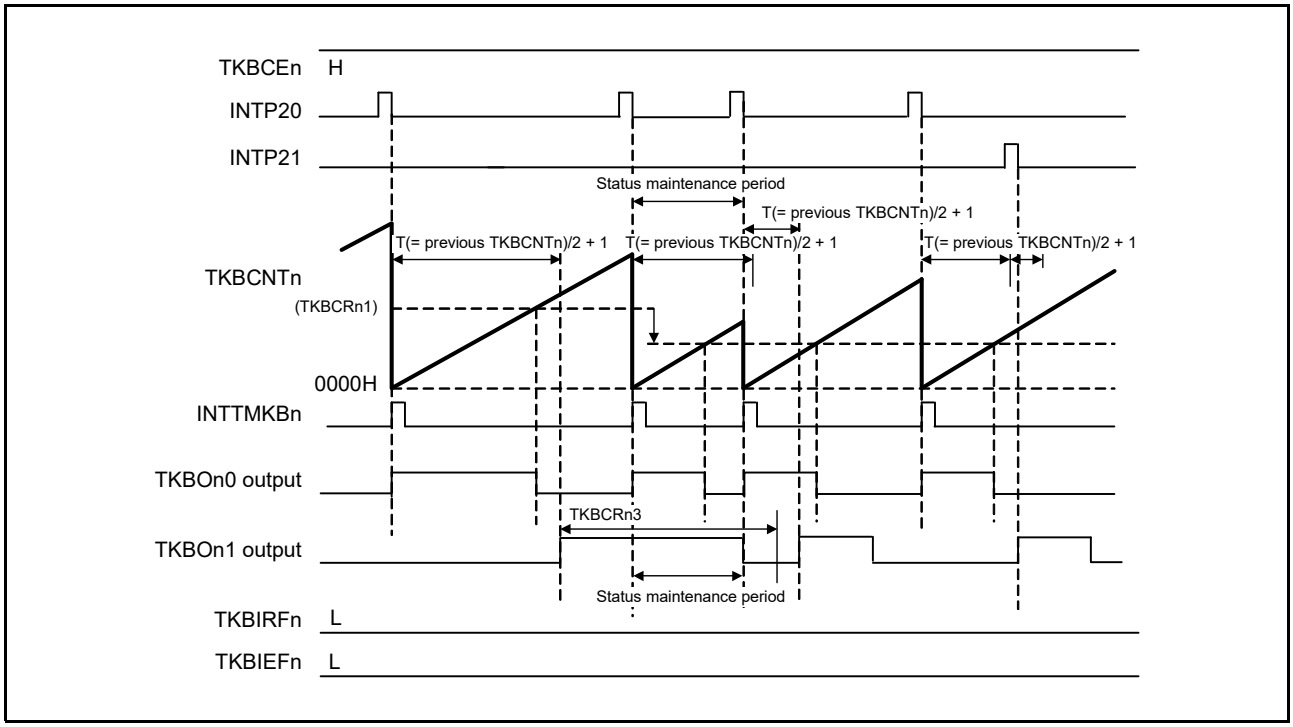
Condition No.11 TKBOn1 with setting width of the TKBCRn3 register is output at T/2 of previous period.

Figure 15 - 66 Figure of Timing of Interleaved PFC Mode (In Case When INTP21 Input Was Detected During TKBOn1 Output)



When INTP21 input is detected during TKBOn1 output of the previous period, this trigger is ignored. This is when the TKBIEFn flag is set by 1.

Figure 15 - 67 Figure of Timing of Interleaved PFC Mode (Output of TKBOn1 Is at the Width of the Previous Output Width and Exceeds Period of Status Maintenance)



When TKBOn1 output of the previous output width is long which exceeds status maintenance period, it is default output compulsively at the starting timing of the subsequent period following the completion of the status maintenance period.

3. List of register settings in interleaved PFC output mode

TKBCTLn0

15	14	13	12	11	10	9	8
—	TKBGTEn1	TKBSSEn1	TKBDIEn1	—	TKBGTEn0	TKBSSEn0	TKBDIEn0
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TKBMFEn	—	TKBIRSn1	TKBIRSn0	—	TKBTSEn	TKBSTSn1	TKBSTSn0
1/0	0	1/0	1/0	0	1	0	0

TKBCTLn1

7	6	5	4	3	2	1	0
TKBCEn	—	—	TKBCKSn	TKBSCMn	—	TKBMDn1	TKBMDn0
1/0	0	0	1/0	0	0	1	1

TKBIOCn0

7	6	5	4	3	2	1	0
—	—	—	—	TKBTOLn1	TKBTOLn0	TKBTODn1	TKBTODn0
0	0	0	0	1/0	1/0	1/0	1/0

TKBIOCn1

7	6	5	4	3	2	1	0
—	—	—	—	—	—	TKBTOEn1	TKBTOEn0
0	0	0	0	0	0	1/0	1/0

TKBCTLn2

15	14	13	12	11	10	9	8
—	—	—	—	—	—	TKBMFMn1	TKBMFMn0
0	0	0	0	0	0	1/0	1/0

7	6	5	4	3	2	1	0
—	TKBINSn2	TKBINSn1	TKBINSn0	—	TKBKCIn2	TKBKCIn1	TKBKCIn0
0	0	0	0	0	0	0	0

TKBCRn0	0000H to FFFFH
TKBCRn1	0000H to FFFFH
TKBCRn2	0000H
TKBCRn3	0000H to FFFFH
TKBTGCRn	0000H to FFFFH
TKBSIRn0	0000H
TKBSIRn1	0000H
TKBSSRn0	00H
TKBSSRn1	00H
TKBDNRn0	00H
TKBDNRn1	00H
TKBMFRn	0000H to FFFFH

: Setting is fixed for this mode : Setting is not needed (default setting)

15.5 Optional Functions of 16-bit Timers KB30, KB31, and KB32

The optional functions can be added to 16-bit timers KB30, KB31, and KB32.

The following table shows available optional functions for each operating mode for 16-bit timers KB30, KB31, and KB32.

Operating Mode		Standalone Mode		Simultaneous Start/Stop Mode		Simultaneous Start/Clear Mode	Interleaved PFC Output Mode
Period Controlling Method for Operating Mode		Period Controlled by CR0	Period Controlled by Trigger	Period Controlled by CR0	Period Controlled by Trigger	Period Controlled by Master	Period Controlled by INTP20/CR0
Optional Function	A/D Conversion Start Timing Signal Output Function	✓	✓	✓	✓	✓	✓
	PWM Output Dithering Function	✓	✓	✓	✓	✓	×
	PWM Output Smooth Start Function	✓	✓	✓	✓	✓	×
	PWM Output Gating Function	✓	✓	✓	✓	✓	×
	Maximum Frequency Limit Function	×	✓	×	✓	×	✓

Remark For details of the operation specifications, see **15.4.2 Default level and active level** and **15.4.3 Stop/restart operation**. The triggers referred to by “Period Controlled by Trigger” in the table above represent external triggers selected by TKBSTSn1 and TKBSTSn0, TKBINSn2 to TKBINSn0, or TKBKCIn2 to TKBKCIn0.

15.5.1 A/D conversion start timing signal output function

An A/D conversion start timing signal output can be generated by setting the 16-bit timer KB trigger compare register n (TKBTGCRn). Thereby, the 16-bit timer KB3n and A/D conversion start timing can be synchronized.
The trigger output signal of the 16-bit timer KB3n is output by detecting the match between TKBCNTn and the TKBTGCRn register which makes trigger output available at any timing corresponding to set period of the TKBCRnm register. Output width of the trigger output signal of the 16-bit timer KB3n is the width of 1 clock of timer clock. Trigger output timing from PWM output period start can be calculated by following formula;

Trigger output timing = Setting of the TKBTGCRn register × Count clock period

Caution The trigger output signal of the 16-bit timer KB3n is not output when TKBCRn0 < TKBTGCRn.

Figure 15 - 68 A/D Conversion Start Timing Signal Output Function for Standalone Mode (Period Controlled by TKBCRn0)

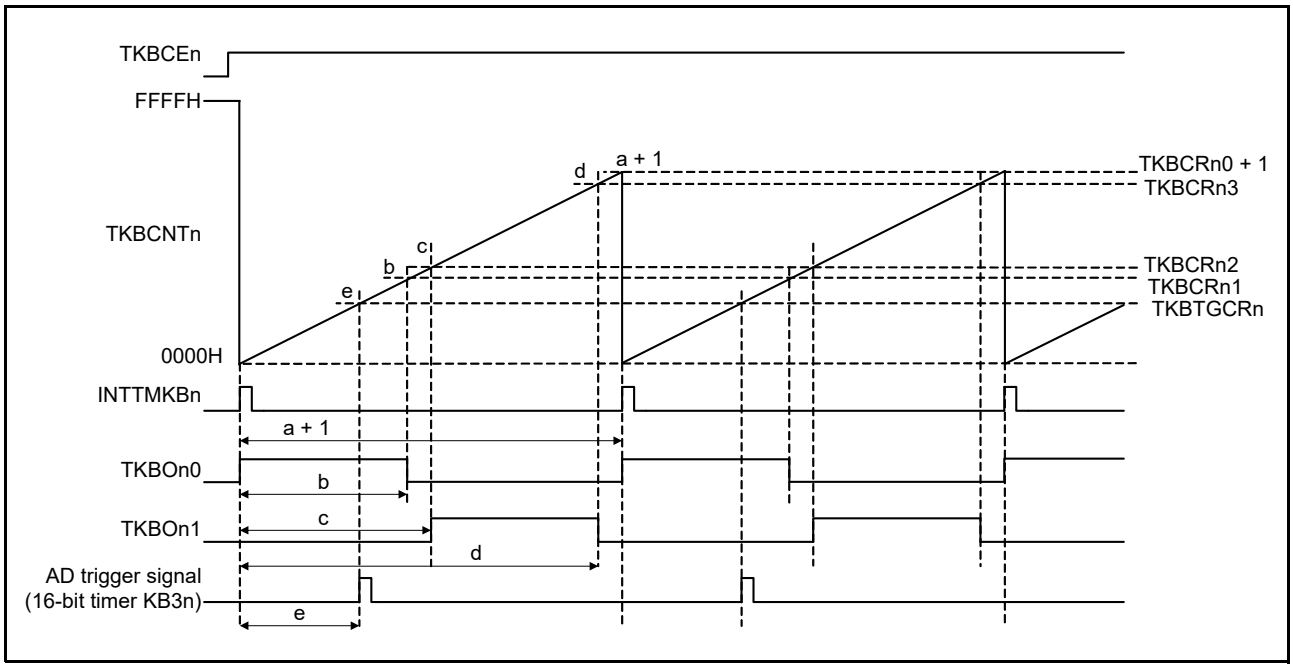
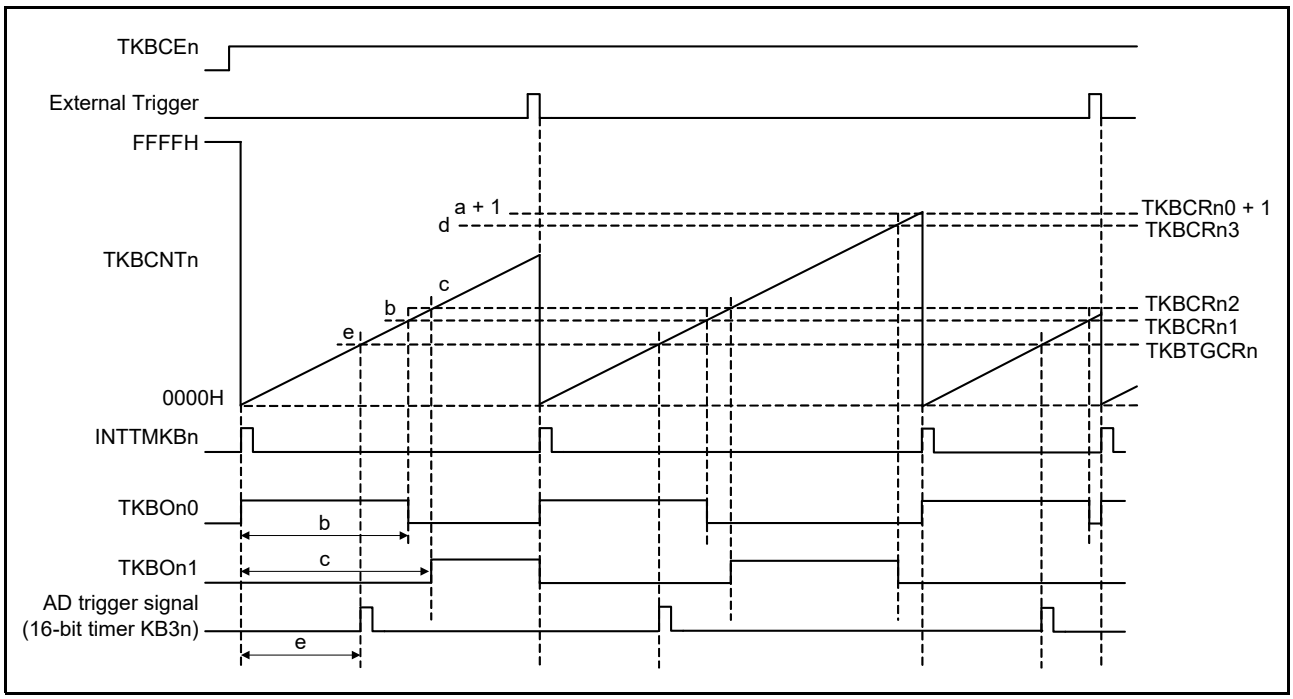


Figure 15 - 69 A/D Conversion Start Timing Signal Output Function for Standalone Mode (Period Controlled by External Trigger Input)



15.5.2 PWM output dithering function

16-bit timers KB30, KB31, and KB32 are capable of producing high resolution PWM output by using the PWM output dithering function.

Having 16 periods of PWM period as standard, 16 times higher PWM output is available for average resolution through extension of active period by 1 count clock at n period (n = 0 to 15) during 16 periods.

The period extending active period during 16 periods by 1 count clock is defined by the TKBDNRnp bit.

The relationship between the TKBDNRnp bit and the period extending active period for 1 count clock is as follows:

Table 15 - 4 Relationship between TKBDNRnp and the Period which Extends Active Period for 1 Count Clock

Repetitions (N) \ Period	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0																
1	■															
2	■								■							
3	■				■				■							
4	■				■				■				■			
5	■		■		■				■				■			
6	■		■		■				■		■		■			
7	■		■		■		■		■		■		■			
8	■		■		■		■		■		■		■		■	
9	■	■	■		■		■		■		■		■		■	
10	■	■	■		■		■		■	■	■		■		■	
11	■	■	■		■	■	■		■	■	■		■		■	
12	■	■	■		■	■	■		■	■	■		■	■	■	
13	■	■	■	■	■	■	■		■	■	■		■	■	■	
14	■	■	■	■	■	■	■		■	■	■	■	■	■	■	
15	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	

Remark 1. □ Cell period: Reset output waveform via settings for TKBCRn1 and TKBCRn3 registers

■ Cell period: Reset output waveform via settings +1 for TKBCRn1 and TKBCRn3 registers

Remark 2. n = 0 to 2; p = 0, 1

Figure 15 - 70 Figure of Waveform at Dithering Operation

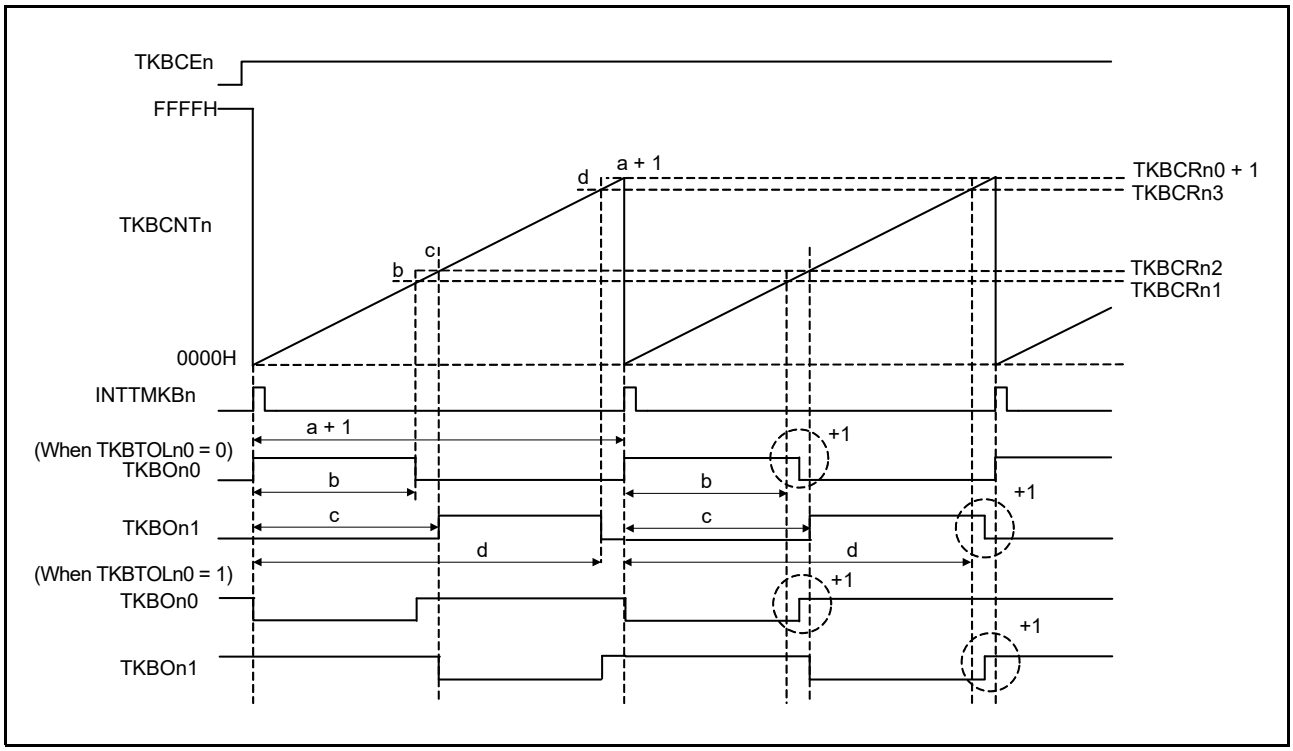


Figure 15 - 71 Figure of Waveform at Dithering Operation (When TKBCRn1 = TKBCRn0 (100% Nearest Neighbor), TKBCRn2 = TKBCRn3(0% Nearest Neighbor)

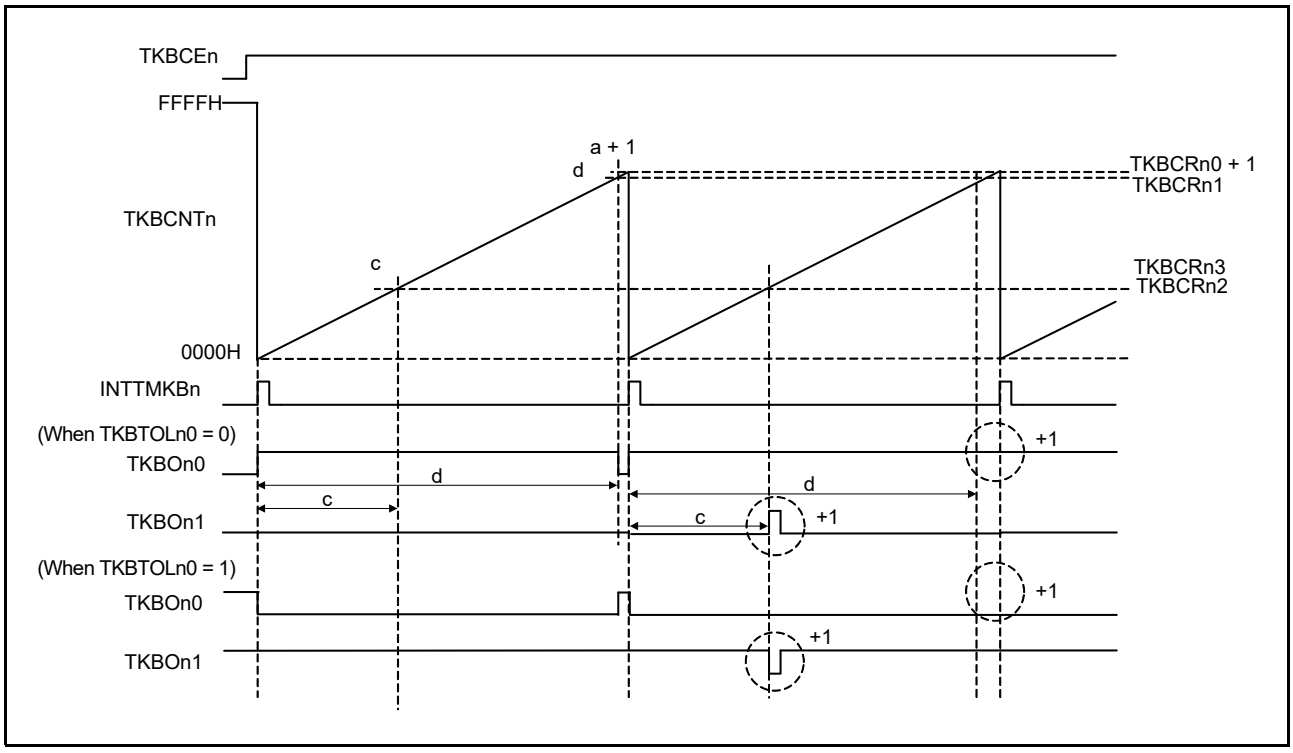
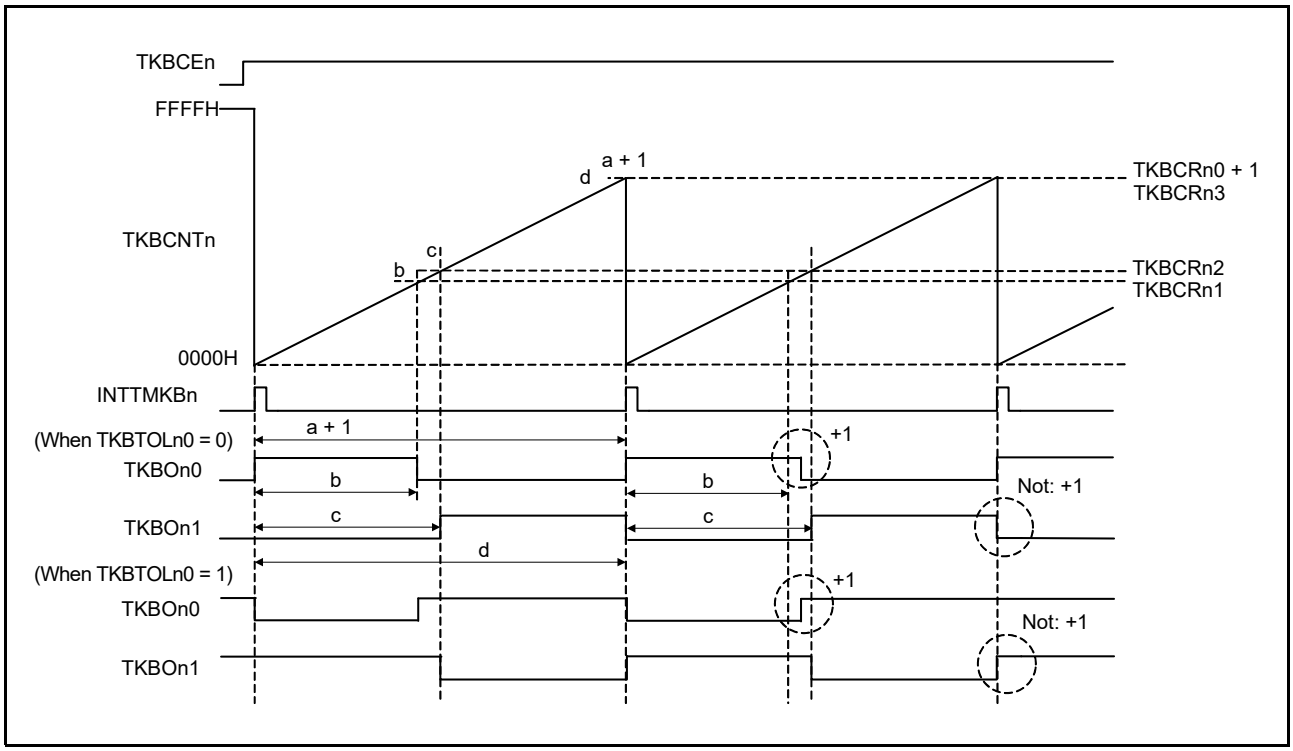


Figure 15 - 72 Figure of Waveform at Dithering Operation (When TKBCRn3 = TKBCRn0 + 1)



1. Available operating mode
   This shows enable or disable status under each mode that is specified by TKBCTLn0 register (the TKBSTSn[1:0] bits) and TKBCTLn1 register (the TKBMDn[1:0] bits).

Operating Mode	TKBMDn[1:0]	Setting Available
Standalone mode (period controlled by the TKBCRn0 register)	00B	✓
Standalone mode (period controlled by external trigger input)	00B	✓
Simultaneous start/stop mode (period controlled by the TKBCRn0 register)	01B	✓
Simultaneous start/stop mode (period controlled by external trigger input)	01B	✓
Simultaneous start/clear mode (period controlled by master)	10B	✓
Interleaved PFC output mode	11B	×

The TKBDNRn0/TKBDNRn1 registers control PWM output dithering function of relative TKBOn0/TKBOn1.

Caution 1. [Overwrite during the operation (TKBCEn = 1) of TKBDNRn0/TKBDNRn1 register]
Regarding TKBDNRn0/TKBDNRn1 owns buffer, overwrite during the operation (TKBCEn = 1) is available.
At this time, batch overwriting is available via writing 1 to TKBRDTn bit.

Caution 2. [Access by TKBCRLDn0/TKBCRLDn1 register]

- **The TKBCRLDn0 register is a 16-bit register mapping lower 8 bit TKBCRn1 and the TKBDNRn0 register.**
- **The TKBCRLDn1 register is a 16-bit register mapping lower 8 bit TKBCRn3 and the TKBDNRn1 register.**
- **Value of the TKBDNRn0/TKBDNRn1 registers is changed even in case that they have accessed the TKBCRLDn0/TKBCRLDn1 registers.**
- **Value of the TKBCRn1/TKBCRn3 registers is changed even in case that they have accessed the TKBCRLDn0/TKBCRLDn1 registers.**
  Note that only the lower 8 bits of the TKBCRn1/TKBCRn3 registers is changed when it is accessed to the TKBCRLDn0/TKBCRLDn1 registers.

Caution 3. [To Combine PWM Output Smooth Start Function with PWM Output Dithering Function]

- **PWM output dithering function is invalid during the execution of PWM output smooth start function (TKBSSFnp = 1).**
- **PWM output dithering function is valid when PWM output smooth start function is stopped (TKBSSFnp = 0).**

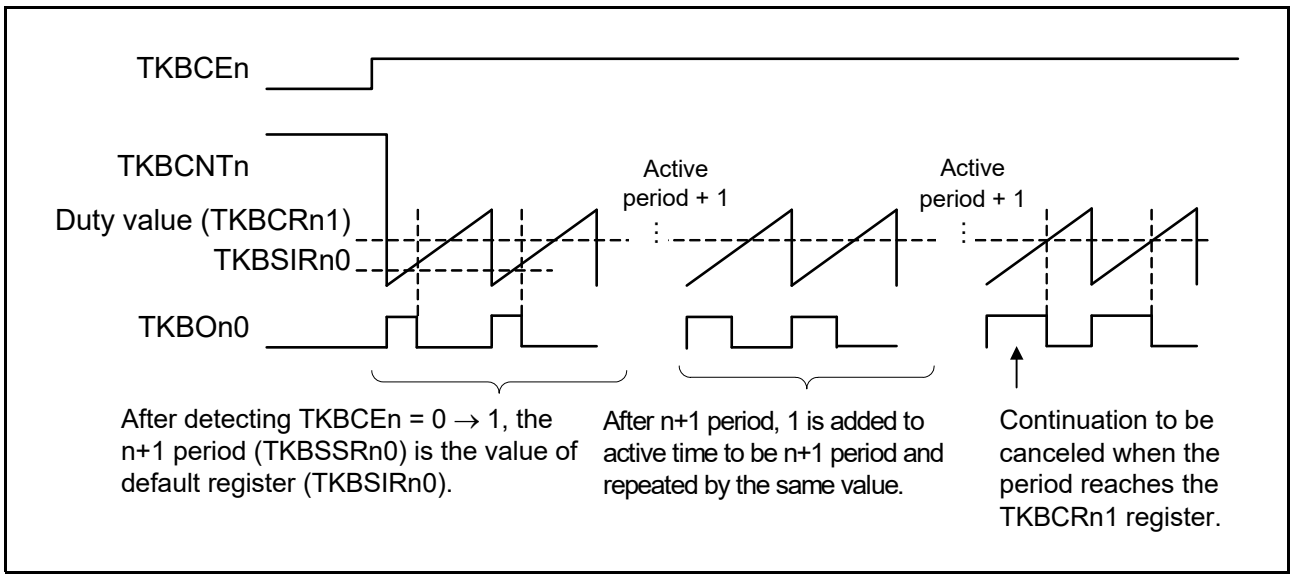
15.5.3 PWM output smooth start function

16-bit timers KB30, KB31, and KB32 own PWM output smooth start function corresponding to rush current control and over-voltage prevention. PWM output smooth start function begins at the start timing of a timer. The process that a user has performed with software in the past can be easily accomplished with the optional function of the hardware. It generates PWM waveform setting the default duty register (TKBSIRnp) of 16-bit timer KB smooth start as 1 period active period. After outputting PWM waveform of the same active period adding 1 to the value of repetition assigned by 16-bit timer KB smooth start step width register (TKBSSRnp), outputs the same TKBSSRnp + 1 period waveform again, executing “active period + 1”.
After repeating the action, PWM output smooth start function is canceled when the same active period defined by the TKBCRn1 and TKBCRn3 registers is reached.

16-bit timer KB smooth start default duty register should be set according to following condition;
0000H ≤ TKBSIRn0 < TKBCRn1 ≤ TKBCRn0 + 1
TKBCRn2 ≤ TKBSIRn1 < TKBCRn3 ≤ TKBCRn0 + 1

It should be set according to following condition when simultaneous start/clear mode is applied;
TKBCRn0 ≤ TKBSIRn0 < TKBCRn1 ≤ TKBCR00 + 1 of Master

Figure 15 - 73 PWM Output Smooth Start Function



1. Operating mode available for PWM output smooth start function

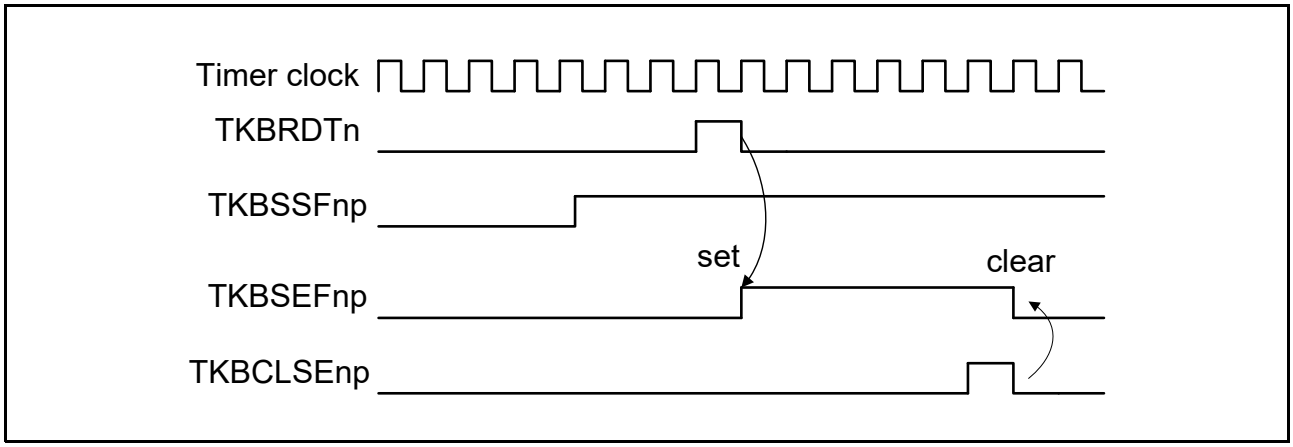
Operating Mode	TKBMDn[1:0]	Setting Available
Standalone mode (period controlled by the TKBCRn0 register)	00B	✓
Standalone mode (period controlled by external trigger input)	00B	✓
Simultaneous start/stop mode (period controlled by the TKBCRn0 register)	01B	✓
Simultaneous start/stop mode (period controlled by external trigger input)	01B	✓
Simultaneous start/clear mode (period controlled by master)	10B	✓
Interleaved PFC output mode	11B	×

The TKBSIRn0 and TKBSSRn0 registers and TKBSIRn1 and TKBSSRn1 registers are respectively used to control the PWM output smooth start function of TKBOn0 and TKBOn1.

2. Overwrite during the operation (TKBCEn = 1) of TKBSIRn0/TKBSIRn1/TKBSSRn0/TKBSSRn1 registers
Overwrite during the operation (TKBCEn = 1) is available for TKBSIRn0/TKBSIRn1/TKBSSRn0/TKBSSRn1. TKBSIRn0/TKBSIRn1/TKBSSRn0/TKBSSRn1 own buffer and batch overwriting is available via writing 1 to TKBRDTn bit. In TKBSIRn0/TKBSIRn1, the buffer value at starting PWM output smooth start function is duty default, and in TKBSSRn0/TKBSSRn1, it is comparison value of internal 4-bit counter.
The internal 4-bit counter is incremented upward using the period of the TKBCNTn counter as a count clock and it becomes 0H when it matches with TKBSSRn0/TKBSSRn1, then continues its counting operation.

3. Overwrite during the operation (TKBCEn = 1) of TKBCRn0/TKBCRn1/TKBCRn2/TKBCRn3/TKBSIRn0/TKBSIRn1/TKBSSRn0/TKBSSRn1 registers
When the TKBRDTn bit is set to 1 during the period of PWM output smooth start (TKBSSFn0 = 1 and TKBSSFn1 = 1), batch overwrite is masked and TKBSEFnp flag is set. In order to perform batch overwrite, clear the TKBSEFnp flag and confirm the TKBSSFnp flag becomes 0, then set 1 to the TKBRDTn bit.

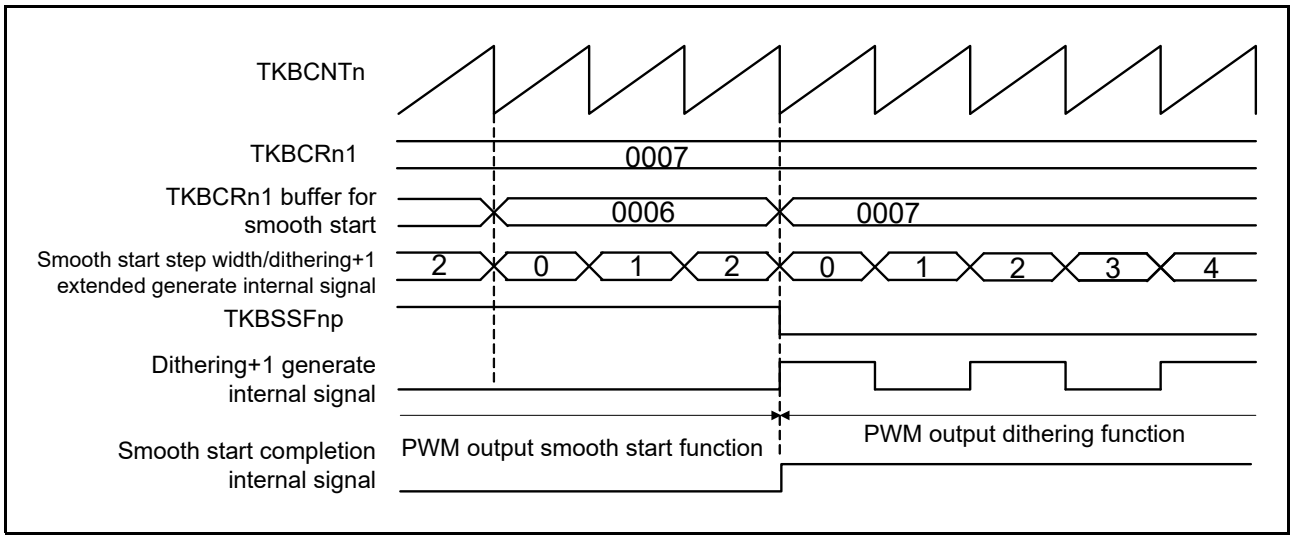
Figure 15 - 74 Overwrite During the Smooth Start Function Operation (TKBSSFnp = 1) of TKBCRn0/TKBCRn1/TKBCRn2/TKBCRn3/TKBSIRn0/TKBSIRn1/TKBSSRn0/TKBSSRn1 Registers



4. To Combine PWM output smooth start function with PWM output dithering function
PWM output dithering function is invalid during the execution of PWM output smooth start function (TKBSSFnp = 1). PWM output dithering function will be valid when PWM output smooth start function is stopped (TKBSSFnp = 0).

5. Completion of PWM output smooth start function and operation of TKBSSFnp
   Figure 15 - 75 shows when the TKBCRn1, TKBDNRnp, and TKBSSRnp registers are set to 0007H, 70H, and 02H, respectively. At the timing that TKBCRn1 = 0007H and the value of TKBCRn1 buffer for internal smooth start match, the TKBSSFnp flag is cleared then dithering function begins.

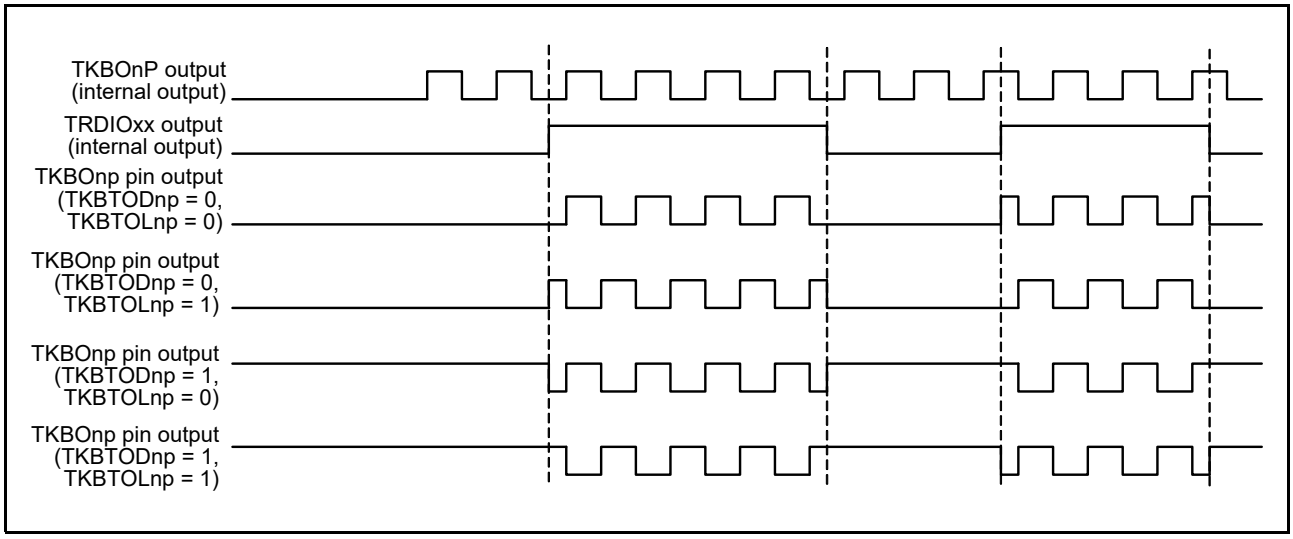
Figure 15 - 75 Completion of PWM Output Smooth Start Function and Operation of TKBSSFnp



15.5.4 PWM output gating function (without combining with PWM output smooth start function)

With this function, during high-level period of timer RD2 output (TRDIOxx), PWM pulse is output from the TKBOnp output pins of 16-bit timers KB30, KB31, and KB32. During low-level period of timer RD2 output (TRDIOxx), the default level (set by TKBTODnp) is output from the TKBOnp output pins of 16-bit timers KB30, KB31, and KB32.

Figure 15 - 76 PWM Output Gating Function



Corresponding relationship between TRDIOxx gating output and TKBOnp output to be gated is 1:1; please refer the following.

- TRDIOB1: TKBO00
- TRDIOC1: TKBO01
- TRDIOD1: TKBO10
- TRDIOA1: TKBO11
- TRDIOB0: TKBO20
- TRDIOD0: TKBO21

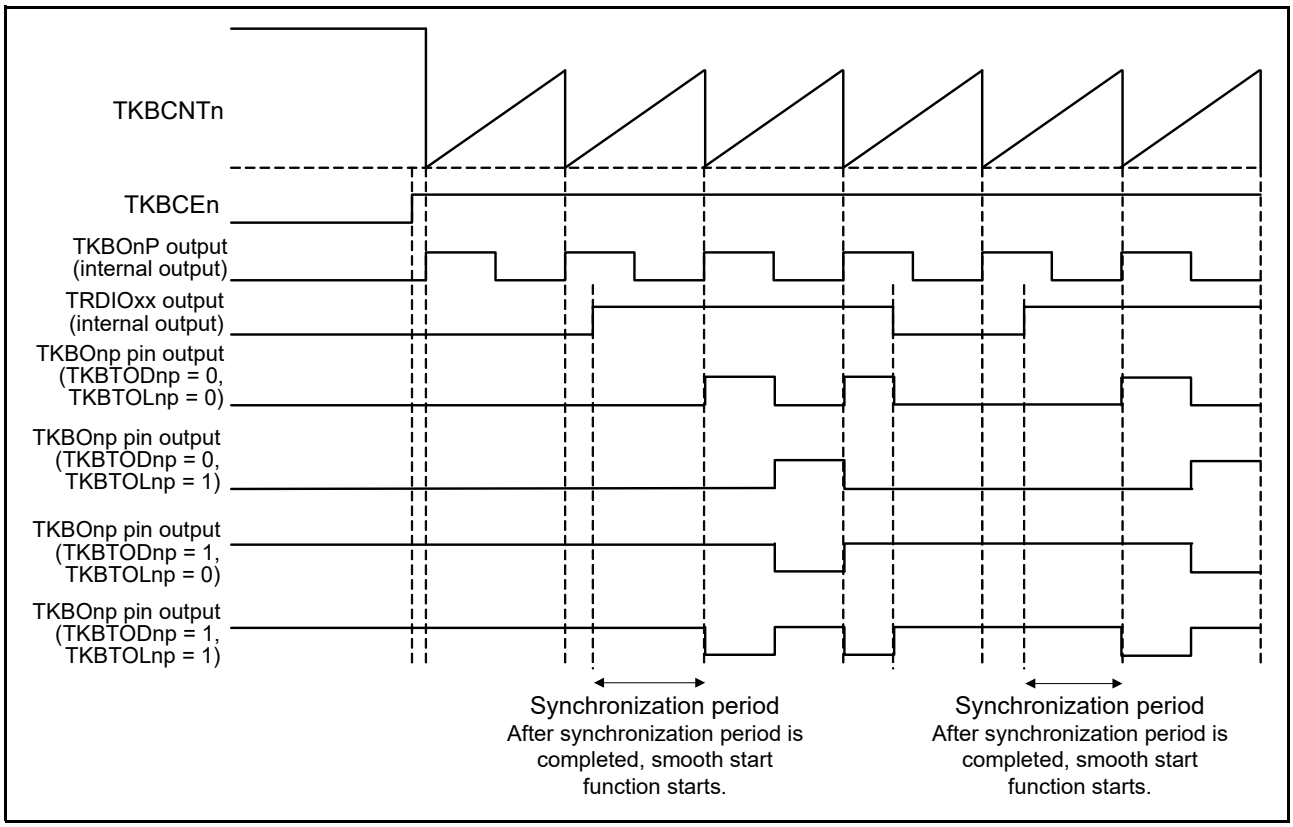
1. Operating mode available for PWM output gating function
   The output gating function is applicable in the following operating modes.

Operating Mode	TKBMDn[1:0]	Setting Available
Standalone mode (period controlled by the TKBCRn0 register)	00B	✓
Standalone mode (period controlled by external trigger input)	00B	✓
Simultaneous start/stop mode (period controlled by the TKBCRn0 register)	01B	✓
Simultaneous start/stop mode (period controlled by external trigger input)	01B	✓
Simultaneous start/clear mode (period controlled by master)	10B	✓
Interleaved PFC output mode	11B	×

15.5.5 PWM output gating function (combining with PWM output smooth start function)

The functions PWM output gating and PWM output smooth start can be combined.
When the smooth start function is also used at the same time, PWM pulse is generated from TKBOnp output pin of 16-bit timers KB30, KB31, and KB32 synching with the period of 16-bit timers KB30, KB31, and KB32 after detecting rising edge of timer RD2 output (TRDIOxx). Through the detection of falling edge of timer RD2 output (TRDIOxx), the default level (set by TKBTODnp) is output from the TKBOnp output pins of 16-bit timers KB30, KB31, and KB32.

Figure 15 - 77 Sample of TKBOnp Output Synchronization Waveform at Starting TRDIOxx Output of PWM Output Gating Function (Combining with PWM Output Smooth Start Function)



1. Operating mode available for PWM output gating function
   The output gating function is applicable in the following operating modes.

Operating Mode	TKBMDn[1:0]	Setting Available
Standalone mode (period controlled by the TKBCRn0 register)	00B	✓
Standalone mode (period controlled by external trigger input)	00B	✓
Simultaneous start/stop mode (period controlled by the TKBCRn0 register)	01B	✓
Simultaneous start/stop mode (period controlled by external trigger input)	01B	✓
Simultaneous start/clear mode (period controlled by master)	10B	✓
Interleaved PFC output mode	11B	×

See **15.5.3 PWM output smooth start function** for the details of PWM output smooth start function.

15.5.6 Maximum frequency limit function

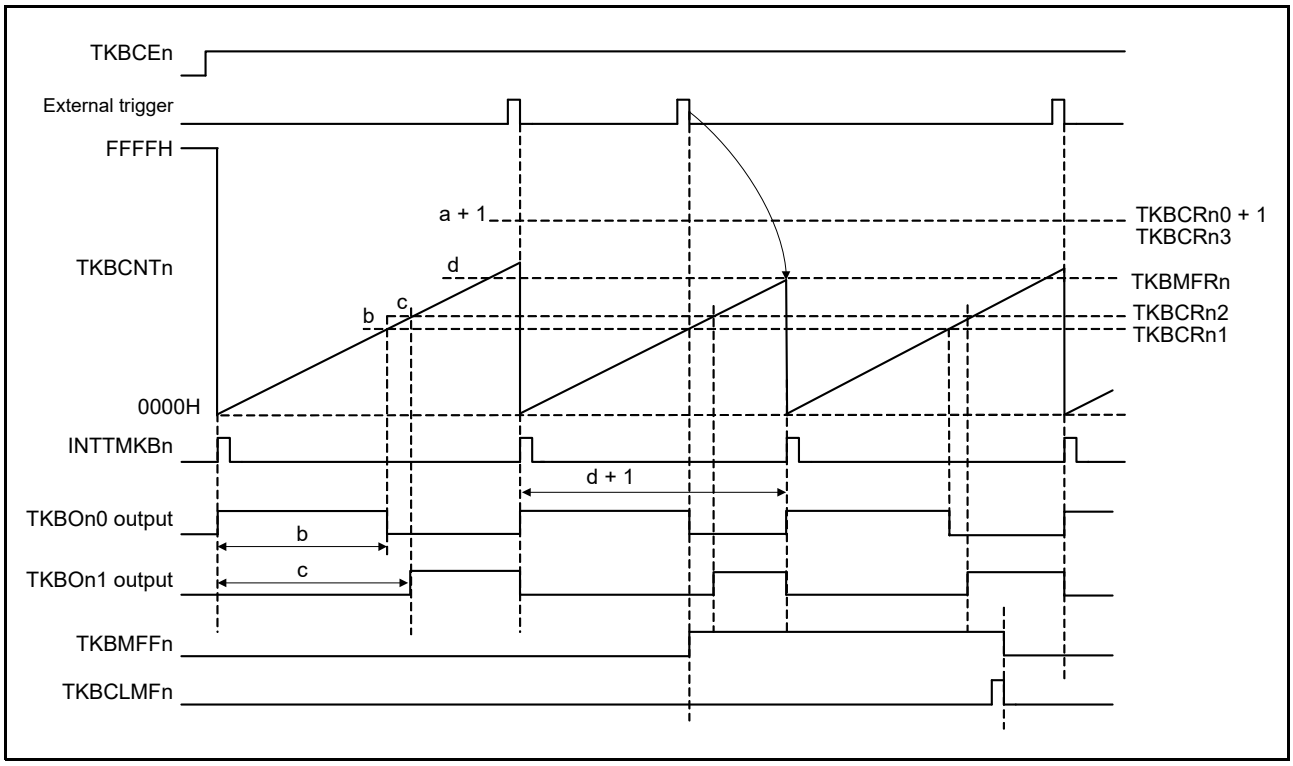
16-bit timers KB30, KB31, and KB32 are a function that regulates the minimum period of the counter clear (maximum frequency) in the periodic control by external trigger or interleaved PFC output mode.
When this function is used, if external trigger input which performs the counter clear occurs while the counter value is less than the setting value of maximum frequency limit register (TKBMFRn), it performs the counter clear after it continues counting until it reaches the setting value of TKBMFRn.

1. Formula for maximum frequency limit (= 1/Minimum period)
   Minimum period (= 1/Maximum frequency limit) = (TKBMFRn setting + 1) × Count clock period

   Caution The following condition need to be satisfied: TKBMFRn setting ≤ TKBCRn0 setting

   When the counter value is smaller than the value in the TKBMFRn register at the timing of external trigger input detection, the TKBMFFn flag is set to 1. The TKBMFFn flag is cleared to 0 by writing 1 to the TKBCLMFn bit.

Figure 15 - 78 Maximum Frequency Limit Function



Remark Period controlled by external trigger input.

2. Operating mode available for maximum frequency limit function
   The maximum frequency limit is applicable in the following operating modes.

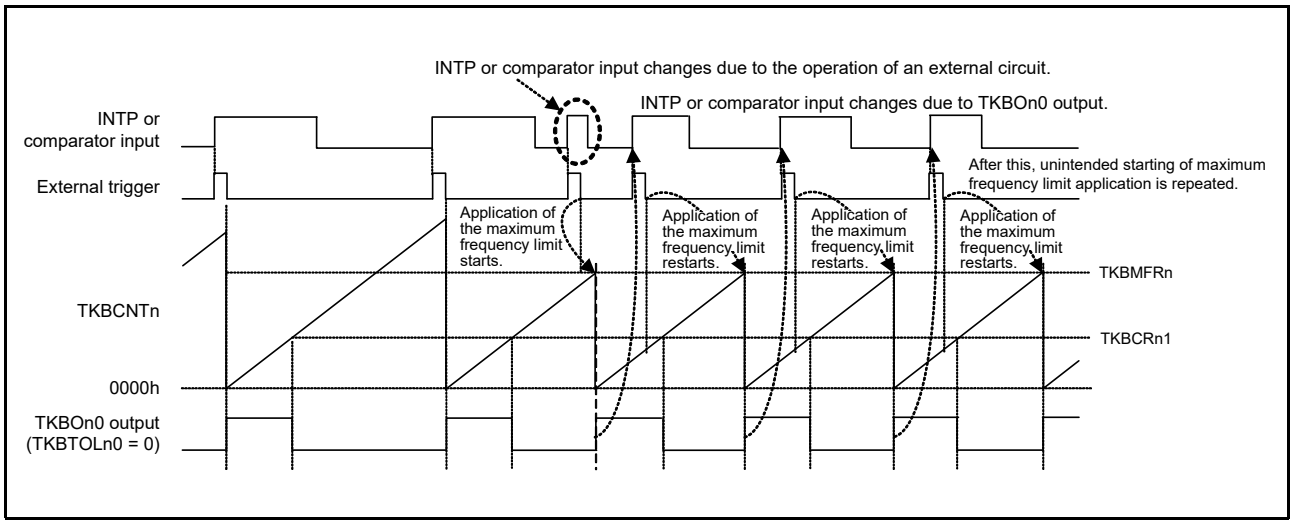
Operating Mode	TKBMDn[1:0]	Setting Available
Standalone mode (period controlled by the TKBCRn0 register)	00B	×
Standalone mode (period controlled by external trigger input)	00B	✓
Simultaneous start/stop mode (period controlled by the TKBCRn0 register)	01B	×
Simultaneous start/stop mode (period controlled by external trigger input)	01B	✓
Simultaneous start/clear mode (period controlled by master)	10B	×
Interleaved PFC output mode	11B	✓

Remark The maximum frequency limit is available when the period is controlled by external trigger input. Period control for the slaves in simultaneous start/clear mode is applied in the same way as that for the master. When the standalone mode (period controlled by external trigger input) is selected for the master, the maximum frequency limit can be applied to the master. Set the TKBCTLn0.TKBMFEn bit to 0 for the slaves so that the maximum frequency limit is not applied to them at this time.

3. Masking control when the maximum frequency limit is in use
When the maximum frequency limit is to be used (TKBCTLn0.TKBMFEn = 1), setting the TKBCTLn2.TKBMFMnp bits, which control use or non-use of the sources for masking of the external trigger when the maximum frequency limit is to be applied, enables masking the detection of external triggers input while TKBOnp output is active.
For the detection of external triggers, the triggers to be masked are only the INTP20 signal in interleaved PFC mode and the restart triggers selected by the TKBCTLn0.TKBSTSn1 and TKBSTSn0 and TKBCTLn2.TKBINSn2 to TKBINSn0 bits in standalone mode or simultaneous start/stop mode. This does not apply to the restart triggers selected by the TKBCTLn2.TKBKCIn2 to TKBKCIn0 bits. When a TKBOnp output is in the high impedance state due to forced output stop control, the output is treated as inactive. When the maximum frequency limit is not to be applied (TKBCTLn0.TKBMFEn = 0), the settings of TKBCTLn2.TKBMFMn1 and TKBMFMn0 bits are invalid and masking control is not applied. When the simultaneous start/clear mode is to be used, masking control with the maximum frequency limit being applied is not usable for master operation in standalone mode. Set the TKBCTLn2.TKBMFMn1 and TKBMFMn0 bits for the master to 2'b00.

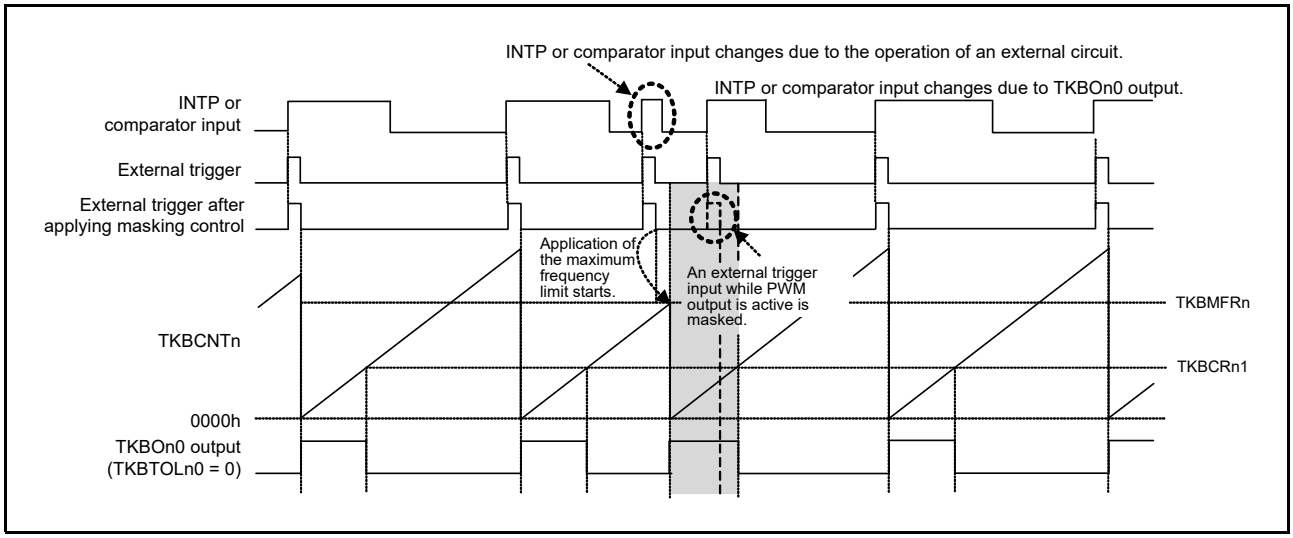
Figure 15 - 79 shows an example of operation when application of the maximum frequency limit is started successively due to the operation of an external circuit and **Figure 15 - 80** shows an example of operation when masking control with the maximum frequency limit being applied (TKBCTLn2.TKBMFMn0 = 1) is used to suppress the successive starting of the maximum frequency limit application.

Figure 15 - 79 Example of Operation When Application of the Maximum Frequency Limit Is Started Successively Due to the Operation of an External Circuit



An external trigger input is generated due to the operation of an external circuit and application of the maximum frequency limit starts. When the values of the TKBCNTn counter and maximum frequency limit setting register n (TKBMFRn) match, the TKBCNTn counter is cleared and TKBOn0 output becomes active. After that, repeatedly generated external trigger inputs on changes to the TKBOn0 output may lead to continuing operation with the setting of the maximum frequency limit setting register n (TKBMFRn) used as the period.

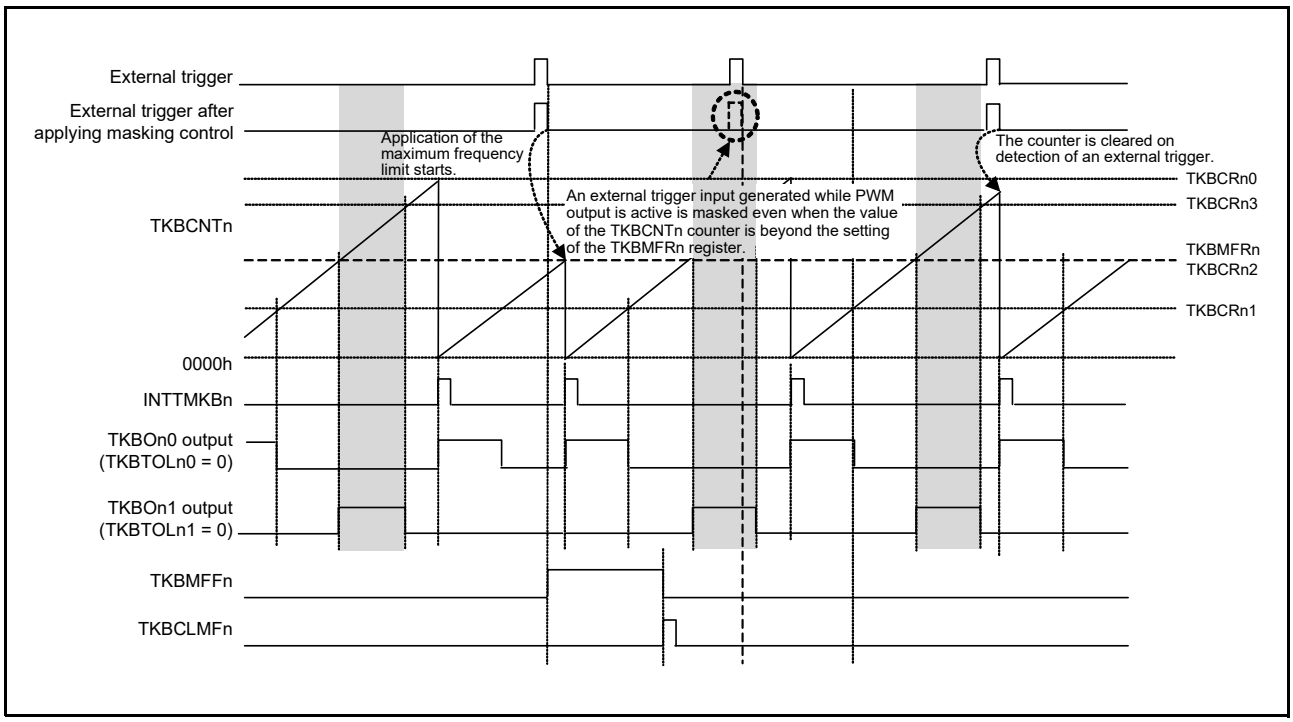
Figure 15 - 80 Example of Operation When Masking Control with the Maximum Frequency Limit Being Applied (TKBCTLn2.TKBMFMn0 = 1) Is Used to Suppress the Successive Starting of Maximum Frequency Limit Application



Setting the TKBCTLn2.TKBMFMn0 bit to 1 masks detection of an external trigger input generated while TKBOn0 output is active, which enables suppression of the successive starting of maximum frequency limit application shown in **Figure 15 - 79**.

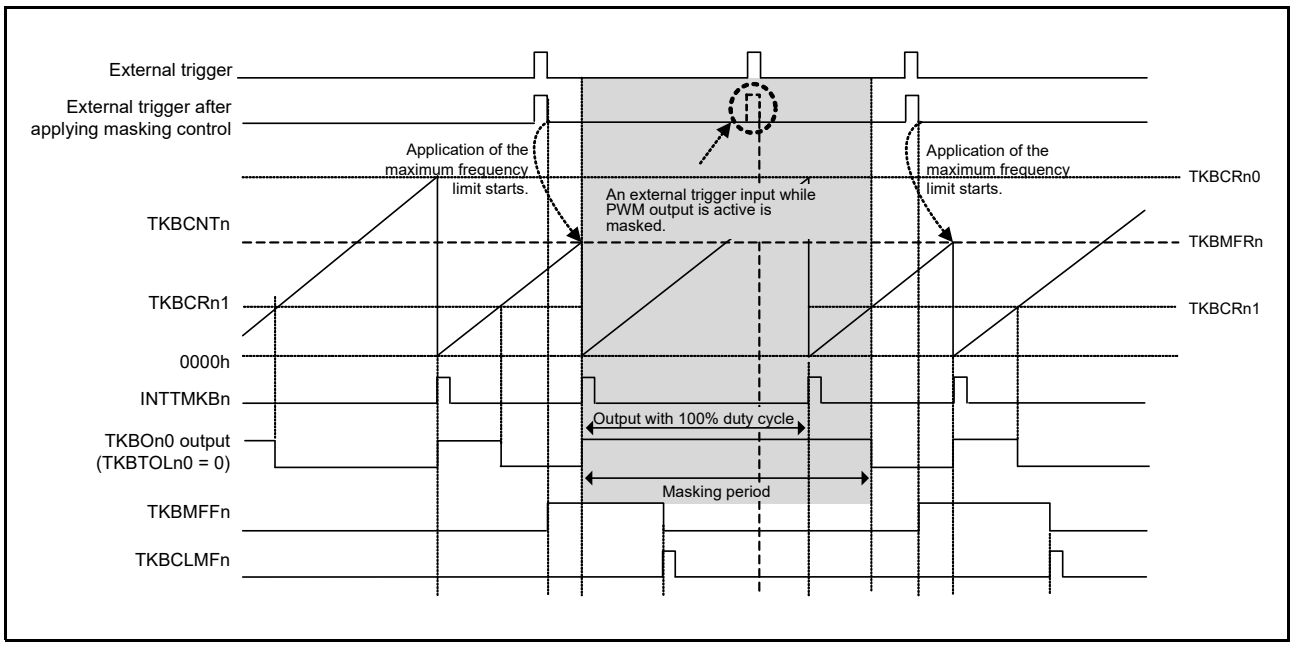
Under masking control with the maximum frequency limit being applied, detection of an external trigger input generated while TKBOnp output is active is masked regardless of the value of the TKBCNTn counter and the setting of the maximum frequency limit setting register n (TKBMFRn). **Figure 15 - 81** shows an example of operation under masking control when the value of the TKBCNTn counter is no less than the setting of the TKBMFRn register.

Figure 15 - 81 Example of Operation under Masking Control When the Value of the TKBCNTn Counter Is No Less Than the Setting of the TKBMFRn Register (When TKBCTLn2.TKBMFMn1 and TKBMFMn0 = 2'b10)



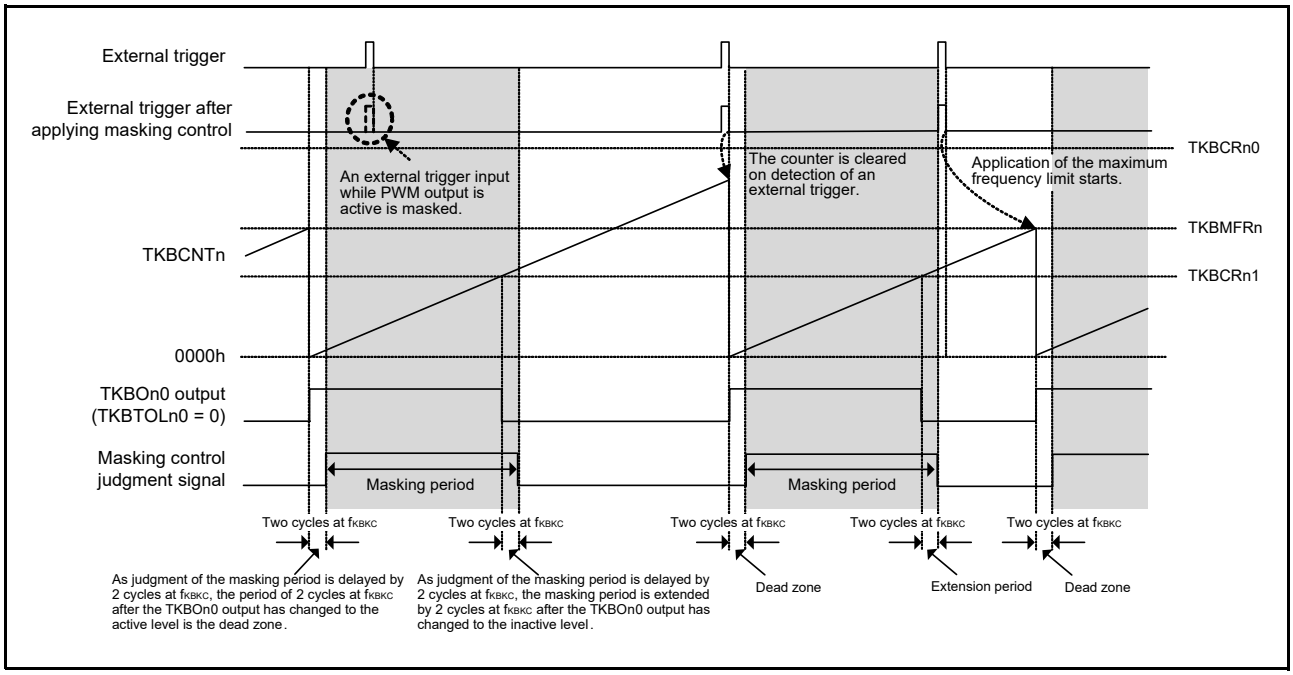
When the duty cycle for the TKBOn0 output is changed during operation or is switched to 100% by the forced output stop function, the PWM period is used as the masking period and so the TKBCNTn counter is not cleared until the value of the TKBCNTn counter and the setting of the TKBCRn0 register match. **Figure 15 - 82** shows an example of operation under masking control for TKBOn0 output with a duty cycle of 100%.

Figure 15 - 82 Example of Operation under Masking Control for TKBOn0 Output with a Duty Cycle of 100% (When TKBCTLn2.TKBMFMn0 = 1)



4. Notes on using masking control with the maximum frequency limit being applied
   The 16-bit timer KB3n uses the TKBOnp output after having applied the forced output stop function to judge the masking period, that is, it controls the masking of external triggers by using an internally synchronized signal. Therefore, masking control is delayed by the two cycles at f_{KBKC} that are used in synchronization. **Figure 15 - 83** shows the period over which masking control with the maximum frequency limit being applied is enabled.

Figure 15 - 83 Period over Which Masking Control with the Maximum Frequency Limit Being Applied Is Enabled (When TKBCTLn2.TKBMFMn0 = 1)



15.5.7 Multi-phase function

The 16-bit timer KB3n can be set up for interlocked operation with timer RD2 by the settings of the TKBKCIn2 to TKBKCIn0 bits. Timer RD2 generates restart triggers by using the compare registers (TRDGRji and TRDCMPji) to set phase differences between 16-bit timer KB3n output signals in the timer-KB PWM output gating mode. The restart triggers generated by timer RD2 are used to delay the start of operations by 16-bit timers KB31 and KB32 relative to 16-bit timer KB30 operation as the basis to realize a multi-phase function.

Remark i = 0, 1; j = A to D

Setting the TKBCEn bit to 1 starts counting by 16-bit timers KB3n. 16-bit timers KB3n start operation to generate PWM waveforms with the duty cycle set to 0%. 16-bit timers KB3n are sequentially restarted by resetting the duty cycles to be used immediately after the timers have started and then starting timer RD2 operation. The result allows the output of PWM waveforms with different phases. In addition, using the simultaneous start/stop mode of the 16-bit timer KB3n enables simultaneous control over starting and stopping operations by 16-bit timers KB3n. **Figure 15 - 84** shows an example of multi-phase operation to produce three phases when 16-bit timers KB30 to KB32 and timer RD2 are in use.

The tables below list the register settings other than the initial values following a reset which are used in the example of operation. For further details on the settings, see **12.3 Registers for Controlling Timer RD2** and **15.3 Registers Controlling 16-bit Timers KB30, KB31, and KB32**.

Table 15 - 5 Timer RD2 Settings

Register	Bit	Setting
TRDBCR	GCE	1: Timer-KB PWM output gating mode is enabled.
TRDPMR	TRDPWMB1	1: TRDTKBOUT0 output is enabled.
	TRDPWMD1	1: TRDTKBOUT2 output is enabled.
	TRDPWMB0	1: TRDTKBOUT4 output is enabled.
TRDGRC0/TRDGRA0	—	Set the PWM period for TRD0.
TRDGRC1/TRDGRA1	—	Set the PWM period for TRD1.
TRDGRD1/TRDGRB1	—	Set the timing of toggling the PWM output level of the TRDTKBOUT0 or TRDIOB1 signal (for control of 16-bit timer KB30).
TRDCMPD1/TRDCMPB1	—	Set the timing of toggling the PWM output level of the TRDTKBOUT2 or TRDIOD1 signal (for control of 16-bit timer KB31).
TRDGRD0/TRDGRB0	—	Set the timing of toggling the PWM output level of the TRDTKBOUT4 or TRDIOB0 signal (for control of 16-bit timer KB32).

Table 15 - 6 16-bit Timer KB30 Settings

Register	Bit	Setting
TKBCTL00	TKBTSE0	1: Uses the compare register batch overwrite function set by an external trigger.
TKBCTL02	TKBKCI02 to TKBKCI00	010: TRDIOB1 rising edge
TKBIOC01	TKBTOE01 and TKBTOE00	11: Timer outputs on the TKBO01 and TKBO00 pins are enabled.
TKBCR00	—	Set this register to FFFFH.
TKBCR01	—	PWM setting for TKBO00 output
TKBCR02, TKBCR03	—	PWM setting for TKBO01 output

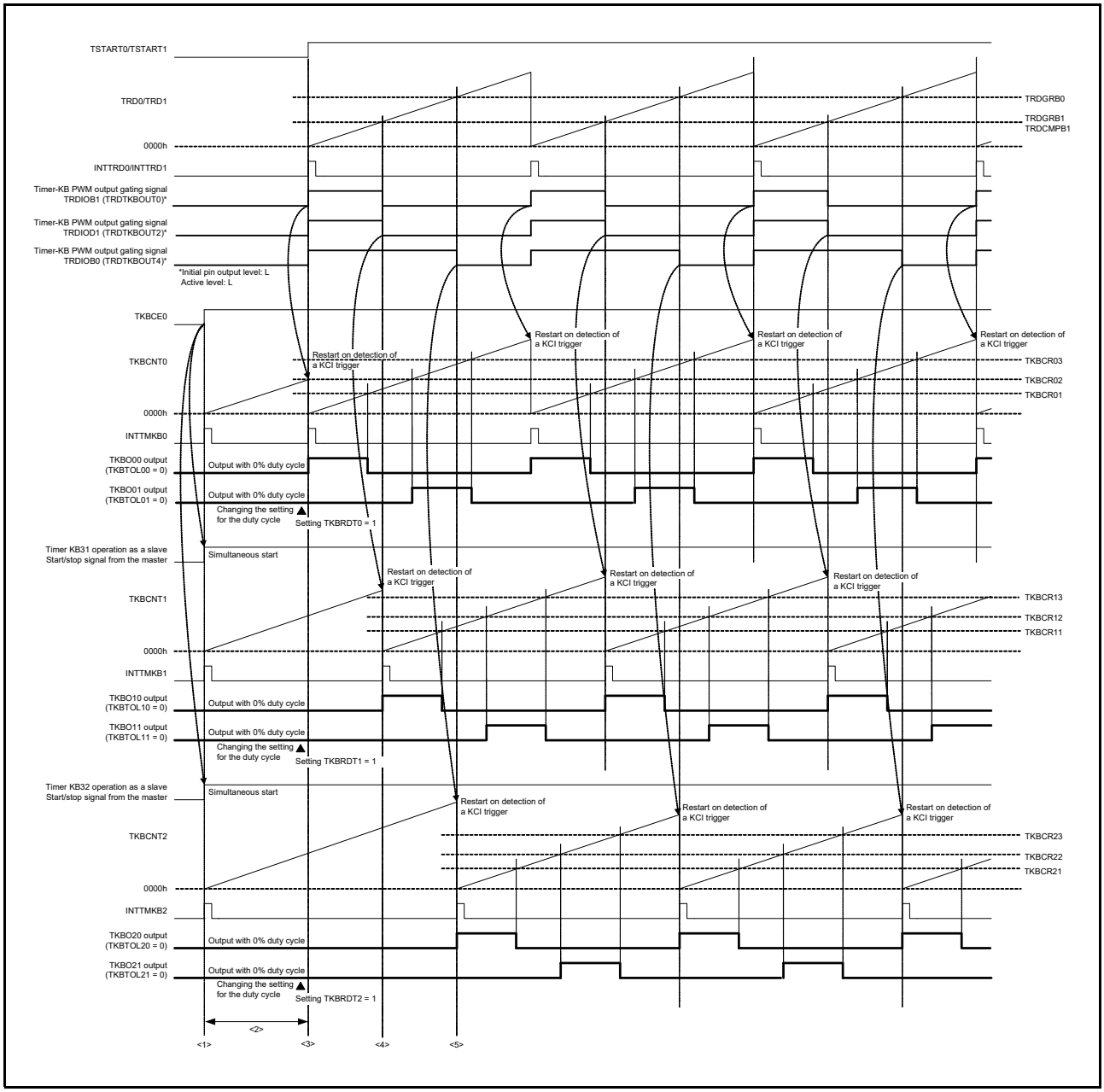
Table 15 - 7 16-bit Timer KB31 Settings

Register	Bit	Setting
TKBCTL10	TKBTSE1	1: Uses the compare register batch overwrite function set by an external trigger.
TKBCTL11	TKBMD11 and TKBMD10	01: Simultaneous start/stop mode (for the slaves)
TKBCTL12	TKBKCI12 to TKBKCI10	011: TRDIOD1 falling edge
TKBIOC11	TKBTOE11 and TKBTOE10	11: Timer outputs on the TKBO11 and TKBO10 pins are enabled.
TKBCR10	—	Set this register to FFFFH.
TKBCR11	—	PWM setting for TKBO10 output
TKBCR12, TKBCR13	—	PWM setting for TKBO11 output

Table 15 - 8 16-bit Timer KB32 Settings

Register	Bit	Setting
TKBCTL20	TKBTSE2	1: Uses the compare register batch overwrite function set by an external trigger.
TKBCTL21	TKBMD21 and TKBMD20	01: Simultaneous start/stop mode (for the slaves)
TKBCTL22	TKBKCI22 to TKBKCI20	011: TRDIOB0 falling edge
TKBIOC21	TKBTOE21 and TKBTOE20	11: Timer outputs on the TKBO21 and TKBO20 pins are enabled.
TKBCR20	—	Set this register to FFFFH.
TKBCR21	—	PWM setting for TKBO20 output
TKBCR22, TKBCR23	—	PWM setting for TKBO21 output

Figure 15 - 84 Example of Multi-phase Operation to Produce Three Phases When 16-bit Timers KB30 to KB32 and Timer RD2 Are in Use



<1> The duty cycle of the 16-bit timer KB3n outputs is set to 0%. Setting the TKBCE0 bit to 1 starts counting by 16-bit timers KB3n at the same time.

<2> The setting for the duty cycle is changed and the TKBRDTn bit is set to 1 for each of 16-bit timers KB3n. The TKBRSF0, TKBRSF1, and TKBRSF2 flags have the value 1 at this time.

<3> Setting the TSTART1 and TSTART0 bits to 11 after the initial settings of timer RD2 have been made starts counting by timer RD2.
The counter value is initialized in response to detection of the restart trigger for 16-bit timer KB30 selected by the TKBKCI02 to TKBKCI00 bits and a PWM waveform with the changed duty cycle is output. After that, timer KB30 continues to operate.

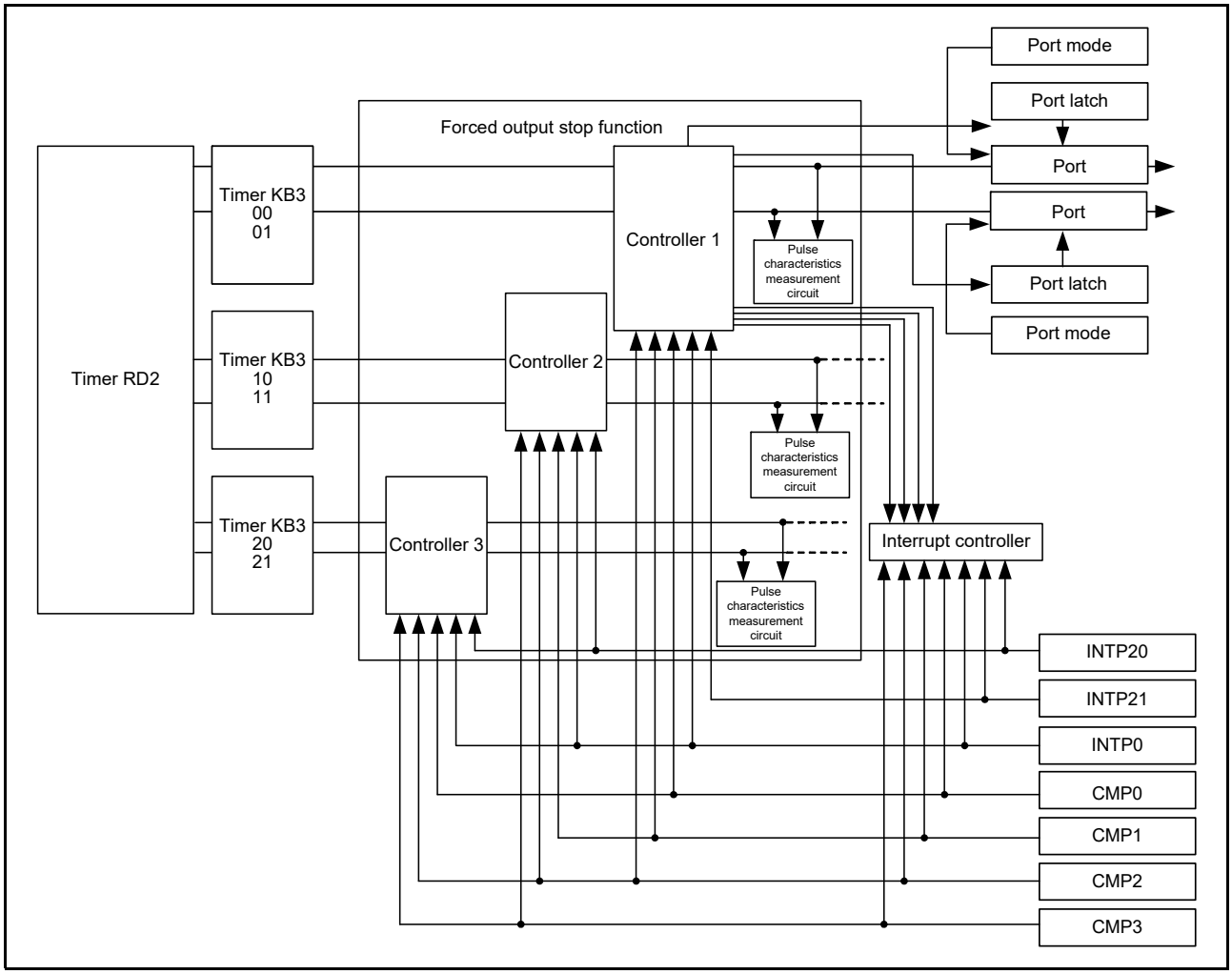
<4> The counter value is initialized in response to detection of the restart trigger for 16-bit timer KB31 selected by the TKBKCI12 to TKBKCI10 bits and a PWM waveform with the changed duty cycle is output. After that, timer KB31 continues to operate.

<5> The counter value is initialized in response to detection of the restart trigger for 16-bit timer KB32 selected by the TKBKCI22 to TKBKCI20 bits and a PWM waveform with the changed duty cycle is output. After that, timer KB32 continues to operate.

15.6 Forced Output Stop Function

Forced output stop function is a function to protect power supply, etc.
If any abnormal situation that occurs in a power circuit configured external to the microcontroller leads to an over-voltage or over-current, using an external interrupt or a comparator output as a trigger source can protect the circuit by maintaining the timer output in the Hi-Z or a fixed-output state without mediation by the CPU under program control.
This function is activated or terminated when an edge of an input signal is detected.
The following figure shows the system structure of forced output stop function.

Figure 15 - 85 System Structure of Forced Output Stop Function



15.6.1 Forced output stop functions 1 and 2

There are two ways of controlling the forced output stop function. Forced output stop function 1 can select fixed level output or Hi-Z output, and forced output stop function 2 can only set fixed level output. The tables below show the differences in control between the functions.

1. Selectable output levels for forced output stop functions 1 and 2

Selectable Output Levels	Forced Output Stop	
	Function 1	Function 2
Hi-Z output	✓	×
Fixed low-level output	✓	✓
Fixed high-level output	✓	✓

2. Start of or release from forced output stop functions 1 and 2

Details of Function/Operation (Start of a Forced Output Stop)	Forced Output Stop	
	Function 1	Function 2
Forced output stop is started following the detection of the rising edge of comparator output.	✓	✓
Forced output stop is started following the detection of the rising or falling edge of external interruption input.	✓	✓
Forced output stop is started by setting the TKBPAHTSnp bit.	✓	×

Details of Function/Operation (Release from a Forced Output Stop)	Forced Output Stop	
	Function 1	Function 2
Release from a forced output stop proceeds by setting the TKBPAHTTnp bit.	✓	×
Release from a forced output stop proceeds by synchronization with a period of 16-bit timer KB30, KB31, or KB32 after the TKBPAHTTnp bit is set.	✓	×
Release from a forced output stop proceeds at the next counter period after the starting of a forced output stop.	×	✓
Release from a forced output stop proceeds at the next counter period after the detection of a falling edge of the trigger signal for forced output stop.	×	✓
Release from a forced output stop proceeds following synchronization with f_{KBKC} after the detection of a falling edge of the trigger signal for forced output stop (the fixed off function).	×	✓

3. Selectable trigger signals and available trigger bits for forced output stop functions 1 and 2

Selectable Trigger Signals	Forced Output Stop	
	Function 1	Function 2
Comparators 0 to 3	✓	✓
External interrupt inputs (INTP20, INTP21, and INTP0)	✓	✓

Available Trigger Bits	Forced Output Stop	
	Function 1	Function 2
TKBPAHTSnp (trigger bit which starts forced output stop for TKBOnp output)	✓	×
TKBPAHTTnp (trigger bit which cancels forced output stop for TKBOnp output)	✓	×

Remark n = 0 to 2; p = 0, 1

Table 15 - 9 External Trigger Assignment List of Forced Output Stop Function 1

Item	TKBO00	TKBO01	TKBO10	TKBO11	TKBO20	TKBO21
Comparator 0	✓	✓	—	—	✓	✓
Comparator 1	✓	✓	✓	✓	—	—
Comparator 2	✓	✓	✓	✓	✓	✓
Comparator 3	—	—	✓	✓	✓	✓
INTP20	—	—	✓	✓	✓	✓
INTP21	✓	✓	—	—	—	—
INTP0	✓	✓	✓	✓	✓	✓

Table 15 - 10 External Trigger Assignment List of Forced Output Stop Function 2

Item	TKBO00	TKBO01	TKBO10	TKBO11	TKBO20	TKBO21
Comparator 0	✓	✓	—	—	✓	✓
Comparator 1	✓	✓	✓	✓	—	—
Comparator 2	✓	✓	✓	✓	✓	✓
Comparator 3	—	—	✓	✓	✓	✓
INTP20	—	—	✓	✓	✓	✓
INTP21	✓	✓	—	—	—	—
INTP0	✓	✓	✓	✓	✓	✓

Caution For setting of INTP20, INTP21, and INTP0, see Section 22 Comparator Module (CMP).

15.6.2 Configuration of hardware for the forced output stop function

The forced output stop function is implemented by the following hardware.

Table 15 - 11 Configuration of Hardware for the Forced Output Stop Function

Item	Configuration
Control registers	• Forced output stop function control register np (TKBPACTLnp) (n = 0 to 2; p = 0, 1) • Forced output stop function control register n2 (TKBPACTLn2) (n = 0 to 2) • Forced output stop function control register n3 (TKBPACTLn3) (n = 0 to 2) • Forced output stop function control register n4 (TKBPACTLn4) (n = 0 to 2) • Pulse characteristics measurement capture register np (TKBPAPLSnp) (n = 0 to 2; p = 0, 1) • Pulse characteristics measurement capture register npL (TKBPAPLSnpL) (n = 0 to 2; p = 0, 1) • Forced output stop function flag register n (TKBPAFLGn) (n = 0 to 2) • Forced output stop function 1 start trigger register n (TKBPAHFSn) (n = 0 to 2) • Forced output stop function 1 cancel trigger register n (TKBPAHFTn) (n = 0 to 2)

15.6.3 Registers controlling forced output stop function

Forced output stop function is controlled by the following registers.

- Forced output stop function control register np (TKBPACTLnp) (n = 0 to 2; p = 0, 1)
- Forced output stop function control register n2 (TKBPACTLn2) (n = 0 to 2)
- Forced output stop function control register n3 (TKBPACTLn3) (n = 0 to 2)
- Forced output stop function control register n4 (TKBPACTLn4) (n = 0 to 2)
- Pulse characteristics measurement capture register np (TKBPAPLSnp) (n = 0 to 2; p = 0, 1)
- Pulse characteristics measurement capture register npL (TKBPAPLSnpL) (n = 0 to 2; p = 0, 1)
- Forced output stop function flag register n (TKBPAFLGn) (n = 0 to 2)
- Forced output stop function 1 start trigger register n (TKBPAHFSn) (n = 0 to 2)
- Forced output stop function 1 cancel trigger register n (TKBPAHFTn) (n = 0 to 2)

15.6.3.1 Forced output stop function control register np (TKBPACTLnp) (n = 0 to 2; p = 0, 1)

The TKBPACTLnp register is a register that selects the signal to be used as the trigger to control the forced output stop function of the TKBOnp pin, and to select the pin for setting forced output stop mode. The TKBPACTLnp register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 15 - 86 Format of Forced Output Stop Function Control Register 0p (TKBPACTL0p) (1/3)

Address: F0770H (TKBPACTL00), F0772H (TKBPACTL01)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TKBPACTL0p	TKBPAFXS0p4	TKBPAFXS0p3	TKBPAFXS0p2	TKBPAFXS0p1	TKBPAFXS0p0	TKBPAHZS0p4	TKBPAHZS0p3	TKBPAFCM0p

7	6	5	4	3	2	1	0
0	TKBPAHZS0p2	TKBPAHZS0p1	TKBPAHZS0p0	TKBPAHCM0p1	TKBPAHCM0p0	TKBPAMD0p1	TKBPAMD0p0

TKBPAFXS0p4	External interruption trigger selection for forced output stop function 2
0	INTP0 can not be used as a trigger.
1	INTP0 can be used as a trigger[Note 1].

TKBPAFXS0p3	External interruption trigger selection for forced output stop function 2
0	INTP21 can not be used as a trigger.
1	INTP21 can be used as a trigger[Note 1].

TKBPAFXS0p2	Comparator trigger selection for forced output stop function 2
0	Comparator 2 can not be used as a trigger.
1	Comparator 2 can be used as a trigger[Note 2].

TKBPAFXS0p1	Comparator trigger selection for forced output stop function 2
0	Comparator 1 can not be used as a trigger.
1	Comparator 1 can be used as a trigger[Note 2].

TKBPAFXS0p0	Comparator trigger selection for forced output stop function 2
0	Comparator 0 can not be used as a trigger.
1	Comparator 0 can be used as a trigger[Note 2].

Figure 15 - 86 Format of Forced Output Stop Function Control Register 0p (TKBPACTL0p) (2/3)

TKBPAFCM0p	Operating mode selection for forced output stop function 2
0	Forced output stop function 2 starts with trigger input, and forced output stop function 2 is cleared at the next counter period**Note 3**.
1	Forced output stop function 2 starts with trigger input, and forced output stop function 2 is cleared at the next counter period following detection of the reverse edge of the trigger**Note 3**.

TKBPAHZS0p4	External interruption trigger selection for forced output stop function 1
0	INTP0 can not be used as a trigger.
1	INTP0 can be used as a trigger**Note 1**.

TKBPAHZS0p3	External interruption trigger selection for forced output stop function 1
0	INTP21 can not be used as a trigger.
1	INTP21 can be used as a trigger**Note 1**.

TKBPAHZS0p2	Comparator trigger selection for forced output stop function 1
0	Comparator 2 can not be used as a trigger.
1	Comparator 2 can be used as a trigger**Note 2**.

TKBPAHZS0p1	Comparator trigger selection for forced output stop function 1
0	Comparator 1 can not be used as a trigger.
1	Comparator 1 can be used as a trigger**Note 2**.

TKBPAHZS0p0	Comparator trigger selection for forced output stop function 1
0	Comparator 0 can not be used as a trigger.
1	Comparator 0 can be used as a trigger**Note 2**.

Figure 15 - 86 Format of Forced Output Stop Function Control Register 0p (TKBPACTL0p) (3/3)

TKBPAHCM0p1	TKBPAHCM0p0	Clear condition selection for forced output stop function 1
0	0	Forced output stop function 1 starts with trigger input, and forced output stop function 1 is cleared when forced output stop function release trigger (TKBPAHTT0p) = 1 is written, regardless of the trigger signal level.
0	1	Forced output stop function 1 starts with trigger input, and when the trigger signal is in its high-level period, writing “forced output stop function release trigger (TKBPAHTT0p) = 1” is invalid. Forced output stop function 1 is cleared when forced output stop function release trigger (TKBPAHTT0p) = 1 is written while the trigger signal is in its low-level period.
1	0	Forced output stop function 1 starts with trigger input, and forced output stop function 1 is cleared at the next counter period after forced output stop function release trigger (TKBPAHTT0p) = 1 is written, regardless of the trigger signal level**Note 3**.
1	1	Forced output stop function 1 starts with trigger input, and when the trigger signal is in its high-level period, writing “forced output stop function release trigger (TKBPAHTT0p) = 1” is invalid. Forced output stop function 1 is cleared at the next counter period after forced output stop function release trigger (TKBPAHTT0p) = 1 is written when the trigger signal is in its low-level period **Note 3**.

TKBPAMD0p1	TKBPAMD0p0	Output status selection when executing forced output stop function	
		Forced output stop function 1	Forced output stop function 2
0	0	Hi-Z output	Output fixed at low level
0	1	Hi-Z output	Output fixed at high level
1	0	Output fixed at low level	Output fixed at low level
1	1	Output fixed at high level	Output fixed at high level

Note 1. When an external interrupt is used as the forced output stop function, see **22.4.6 Notes on using interlocking of the comparators with the 16-bit timers KB30, KB31, and KB32**.

Note 2. When a comparator is used as the forced output stop function of the 16-bit timer KB3n, set the CnFCK[1:0] bits to 00B. See **22.4.6 Notes on using interlocking of the comparators with the 16-bit timers KB30, KB31, and KB32**.

Note 3. When the 16-bit timer KB3n is stopped (TKBCEn = 0) without waiting for the next counter period, the forced output stop function is kept on until the 16-bit timer KB3n is restarted (TKBCEn = 1).

Caution 1. During timer operation, overwriting the TKBPACTL0p register is prohibited. However, the TKBPACTL0p register can be refreshed (the same value is written).

Caution 2. Be sure to clear bit 7 to 0.

Remark n = 0 to 2; p = 0, 1

Figure 15 - 87 Format of Forced Output Stop Function Control Register 1p (TKBPACTL1p) (1/3)

Address: F07B0H (TKBPACTL10), F07B2H (TKBPACTL11)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TKBPACTL1p	TKBPAFXS1p4	TKBPAFXS1p3	TKBPAFXS1p2	TKBPAFXS1p1	TKBPAFXS1p0	TKBPAHZS1p4	TKBPAHZS1p3	TKBPAFCM1p

7	6	5	4	3	2	1	0
0	TKBPAHZS1p2	TKBPAHZS1p1	TKBPAHZS1p0	TKBPAHCM1p1	TKBPAHCM1p0	TKBPAMD1p1	TKBPAMD1p0

TKBPAFXS1p4	External interruption trigger selection for forced output stop function 2
0	INTP0 can not be used as a trigger.
1	INTP0 can be used as a trigger**Note 1**.

TKBPAFXS1p3	External interruption trigger selection for forced output stop function 2
0	INTP20 can not be used as a trigger.
1	INTP20 can be used as a trigger**Note 1**.

TKBPAFXS1p2	Comparator trigger selection for forced output stop function 2
0	Comparator 3 can not be used as a trigger.
1	Comparator 3 can be used as a trigger**Note 2**.

TKBPAFXS1p1	Comparator trigger selection for forced output stop function 2
0	Comparator 2 can not be used as a trigger.
1	Comparator 2 can be used as a trigger**Note 2**.

TKBPAFXS1p0	Comparator trigger selection for forced output stop function 2
0	Comparator 1 can not be used as a trigger.
1	Comparator 1 can be used as a trigger**Note 2**.

Figure 15 - 87 Format of Forced Output Stop Function Control Register 1p (TKBPACTL1p) (2/3)

TKBPAFCM1p	Operating mode selection for forced output stop function 2
0	Forced output stop function 2 starts with trigger input, and forced output stop function 2 is cleared at the next counter period**Note 3**.
1	Forced output stop function 2 starts with trigger input, and forced output stop function 2 is cleared at the next counter period following detection of the reverse edge of the trigger**Note 3**.

TKBPAHZS1p4	External interruption trigger selection for forced output stop function 1
0	INTP0 can not be used as a trigger.
1	INTP0 can be used as a trigger**Note 1**.

TKBPAHZS1p3	External interruption trigger selection for forced output stop function 1
0	INTP20 can not be used as a trigger.
1	INTP20 can be used as a trigger**Note 1**.

TKBPAHZS1p2	Comparator trigger selection for forced output stop function 1
0	Comparator 3 can not be used as a trigger.
1	Comparator 3 can be used as a trigger**Note 2**.

TKBPAHZS1p1	Comparator trigger selection for forced output stop function 1
0	Comparator 2 can not be used as a trigger.
1	Comparator 2 can be used as a trigger**Note 2**.

TKBPAHZS1p0	Comparator trigger selection for forced output stop function 1
0	Comparator 1 can not be used as a trigger.
1	Comparator 1 can be used as a trigger**Note 2**.

Figure 15 - 87 Format of Forced Output Stop Function Control Register 1p (TKBPACTL1p) (3/3)

TKBPAHCM1p1	TKBPAHCM1p0	Clear condition selection for forced output stop function 1
0	0	Forced output stop function 1 starts with trigger input, and forced output stop function 1 is cleared when forced output stop function release trigger (TKBPAHTT1p) = 1 is written, regardless of the trigger signal level.
0	1	Forced output stop function 1 starts with trigger input, and when the trigger signal is in its high-level period, writing “forced output stop function release trigger (TKBPAHTT1p) = 1” is invalid. Forced output stop function 1 is cleared when forced output stop function release trigger (TKBPAHTT1p) = 1 is written while the trigger signal is in its low-level period.
1	0	Forced output stop function 1 starts with trigger input, and forced output stop function 1 is cleared at the next counter period after forced output stop function release trigger (TKBPAHTT1p) = 1 is written, regardless of the trigger signal level**Note 3**.
1	1	Forced output stop function 1 starts with trigger input, and when the trigger signal is in its high-level period, writing “forced output stop function release trigger (TKBPAHTT1p) = 1” is invalid. Forced output stop function 1 is cleared at the next counter period after forced output stop function release trigger (TKBPAHTT1p) = 1 is written when the trigger signal is in its low-level period **Note 3**.

TKBPAMD1p1	TKBPAMD1p0	Output status selection when executing forced output stop function	
		Forced output stop function 1	Forced output stop function 2
0	0	Hi-Z output	Output fixed at low level
0	1	Hi-Z output	Output fixed at high level
1	0	Output fixed at low level	Output fixed at low level
1	1	Output fixed at high level	Output fixed at high level

Note 1. When an external interrupt is used as the forced output stop function, see **22.4.6 Notes on using interlocking of the comparators with the 16-bit timers KB30, KB31, and KB32**.

Note 2. When a comparator is used as the forced output stop function of the 16-bit timer KB3n, set the CnFCK[1:0] bits to 00B. See **22.4.6 Notes on using interlocking of the comparators with the 16-bit timers KB30, KB31, and KB32**.

Note 3. When the 16-bit timer KB3n is stopped (TKBCEn = 0) without waiting for the next counter period, the forced output stop function is kept on until the 16-bit timer KB3n is restarted (TKBCEn = 1).

Caution 1. During timer operation, overwriting the TKBPACTL1p register is prohibited. However, the TKBPACTL1p register can be refreshed (the same value is written).

Caution 2. Be sure to clear bit 7 to 0.

Remark n = 0 to 2; p = 0, 1

Figure 15 - 88 Format of Forced Output Stop Function Control Register 2p (TKBPACTL2p) (1/3)

Address: F0430H (TKBPACTL20), F0432H (TKBPACTL21)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TKBPACTL2p	TKBPAFXS2p4	TKBPAFXS2p3	TKBPAFXS2p2	TKBPAFXS2p1	TKBPAFXS2p0	TKBPAHZS2p4	TKBPAHZS2p3	TKBPAFCM2p

7	6	5	4	3	2	1	0
0	TKBPAHZS2p2	TKBPAHZS2p1	TKBPAHZS2p0	TKBPAHCM2p1	TKBPAHCM2p0	TKBPAMD2p1	TKBPAMD2p0

TKBPAFXS2p4	External interruption trigger selection for forced output stop function 2
0	INTP0 can not be used as a trigger.
1	INTP0 can be used as a trigger**Note 1**.

TKBPAFXS2p3	External interruption trigger selection for forced output stop function 2
0	INTP20 can not be used as a trigger.
1	INTP20 can be used as a trigger**Note 1**.

TKBPAFXS2p2	Comparator trigger selection for forced output stop function 2
0	Comparator 3 can not be used as a trigger.
1	Comparator 3 can be used as a trigger**Note 2**.

TKBPAFXS2p1	Comparator trigger selection for forced output stop function 2
0	Comparator 2 can not be used as a trigger.
1	Comparator 2 can be used as a trigger**Note 2**.

TKBPAFXS2p0	Comparator trigger selection for forced output stop function 2
0	Comparator 0 can not be used as a trigger.
1	Comparator 0 can be used as a trigger**Note 2**.

Figure 15 - 88 Format of Forced Output Stop Function Control Register 2p (TKBPACTL2p) (2/3)

TKBPAFCM2p	Operating mode selection for forced output stop function 2
0	Forced output stop function 2 starts with trigger input, and forced output stop function 2 is cleared at the next counter period**Note 3**.
1	Forced output stop function 2 starts with trigger input, and forced output stop function 2 is cleared at the next counter period following detection of the reverse edge of the trigger**Note 3**.

TKBPAHZS2p4	External interruption trigger selection for forced output stop function 1
0	INTP0 can not be used as a trigger.
1	INTP0 can be used as a trigger**Note 1**.

TKBPAHZS2p3	External interruption trigger selection for forced output stop function 1
0	INTP20 can not be used as a trigger.
1	INTP20 can be used as a trigger**Note 1**.

TKBPAHZS2p2	Comparator trigger selection for forced output stop function 1
0	Comparator 3 can not be used as a trigger.
1	Comparator 3 can be used as a trigger**Note 2**.

TKBPAHZS2p1	Comparator trigger selection for forced output stop function 1
0	Comparator 2 can not be used as a trigger.
1	Comparator 2 can be used as a trigger**Note 2**.

TKBPAHZS2p0	Comparator trigger selection for forced output stop function 1
0	Comparator 0 can not be used as a trigger.
1	Comparator 0 can be used as a trigger**Note 2**.

Figure 15 - 88 Format of Forced Output Stop Function Control Register 2p (TKBPACTL2p) (3/3)

TKBPAHCM2p1	TKBPAHCM2p0	Clear condition selection for forced output stop function 1
0	0	Forced output stop function 1 starts with trigger input, and forced output stop function 1 is cleared when forced output stop function release trigger (TKBPAHTT2p) = 1 is written, regardless of the trigger signal level.
0	1	Forced output stop function 1 starts with trigger input, and when the trigger signal is in its high-level period, writing “forced output stop function release trigger (TKBPAHTT2p) = 1” is invalid. Forced output stop function 1 is cleared when forced output stop function release trigger (TKBPAHTT2p) = 1 is written while the trigger signal is in its low-level period.
1	0	Forced output stop function 1 starts with trigger input, and forced output stop function 1 is cleared at the next counter period after forced output stop function release trigger (TKBPAHTT2p) = 1 is written, regardless of the trigger signal level**Note 3**.
1	1	Forced output stop function 1 starts with trigger input, and when the trigger signal is in its high-level period, writing “forced output stop function release trigger (TKBPAHTT2p) = 1” is invalid. Forced output stop function 1 is cleared at the next counter period after forced output stop function release trigger (TKBPAHTT2p) = 1 is written when the trigger signal is in its low-level period **Note 3**.

TKBPAMD2p1	TKBPAMD2p0	Output status selection when executing forced output stop function	
		Forced output stop function 1	Forced output stop function 2
0	0	Hi-Z output	Output fixed at low level
0	1	Hi-Z output	Output fixed at high level
1	0	Output fixed at low level	Output fixed at low level
1	1	Output fixed at high level	Output fixed at high level

Note 1. When an external interrupt is used as the forced output stop function, see **22.4.6 Notes on using interlocking of the comparators with the 16-bit timers KB30, KB31, and KB32**.

Note 2. When a comparator is used as the forced output stop function of the 16-bit timer KB3n, set the CnFCK[1:0] bits to 00B. See **22.4.6 Notes on using interlocking of the comparators with the 16-bit timers KB30, KB31, and KB32**.

Note 3. When the 16-bit timer KB3n is stopped (TKBCEn = 0) without waiting for the next counter period, the forced output stop function is kept on until the 16-bit timer KB3n is restarted (TKBCEn = 1).

Caution 1. During timer operation, overwriting the TKBPACTL2p register is prohibited. However, the TKBPACTL2p register can be refreshed (the same value is written).

Caution 2. Be sure to clear bit 7 to 0.

Remark n = 0 to 2; p = 0, 1

15.6.3.2 Forced output stop function control register n2 (TKBPACTLn2) (n = 0 to 2)

The TKBPACTLn2 register is a register that enables or disables the forced output stop function of the TKBOnp pin. The TKBPACTLn2 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 15 - 89 Format of Forced Output Stop Function Control Register n2 (TKBPACTLn2)

Address: F0777H (TKBPACTL02), F07B7H (TKBPACTL12), F0437H (TKBPACTL22)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	<1>	<0>
TKBPACTLn2	0	0	0	0	0	0	TKBPACEn1	TKBPACEn0

TKBPACEnp	Input control of trigger signal used for forced output stop function of the TKBOnp pin.
0	Disable operation of forced output stop function
1	Enable operation of forced output stop function

Caution 1. The TKBPACTLn2 register can be overwritten while the timer is operating.
Caution 2. Be sure to clear bits 7 to 2 to 0.

Remark n = 0 to 2; p = 0, 1

15.6.3.3 Forced output stop function control register n3 (TKBPACTLn3) (n = 0 to 2)

The TKBPACTLn3 register controls the following operations for the TKBOnp pins.

- Interrupt output when the forced output stop function is terminated or activated
- Fixed off function for forced output stop function 2

The TKBPACTLn3 register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 15 - 90 Format of Forced Output Stop Function Control Register n3 (TKBPACTLn3)

Address: F0778H (TKBPACTL03), F07B8H (TKBPACTL13), F0438H (TKBPACTL23)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TKBPACTLn3	0	TKBPASTPIn1	TKBPASTAIn1	TKBPAFIXn1	0	TKBPASTPIn0	TKBPASTAIn0	TKBPAFIXn0

TKBPASTPIn1	Control of the interrupt when forced stopping of the output from the TKBOn1 pin is terminated
0	An interrupt is not generated when forced stopping of the output is terminated.
1	An interrupt is generated when forced stopping of the output is terminated.

TKBPASTAIn1	Control of the interrupt when forced stopping of the output from the TKBOn1 pin is activated
0	An interrupt is not generated when forced stopping of the output is activated.
1	An interrupt is generated when forced stopping of the output is activated.

TKBPAFIXn1	Control of the fixed off function for forced output stop function 2 for the TKBOn1 pin
0	The fixed off function is not used.
1	The fixed off function is used.

TKBPASTPIn0	Control of the interrupt when forced stopping of the output from the TKBOn0 pin is terminated
0	An interrupt is not generated when forced stopping of the output is terminated.
1	An interrupt is generated when forced stopping of the output is terminated.

TKBPASTAIn0	Control of the interrupt when forced stopping of the output from the TKBOn0 pin is activated
0	An interrupt is not generated when forced stopping of the output is activated.
1	An interrupt is generated when forced stopping of the output is activated.

TKBPAFIXn0	Control of the fixed off function for forced output stop function 2 for the TKBOn0 pin
0	The fixed off function is not used.
1	The fixed off function is used.

(Cautions and Remark are listed on the next page.)

Caution 1. During timer operation, overwriting the TKBPACTLn3 register is prohibited. However, the TKBPACTLn3 register can be refreshed (the same value is written).

Caution 2. Be sure to clear bits 7 and 3 to 0.

Remark n = 0 to 2; p = 0, 1

15.6.3.4 Forced output stop function control register n4 (TKBPACTLn4) (n = 0 to 2)

The TKBPACTLn4 register controls the pulse characteristics measurement function for the TKBOnp pin. The TKBPACTLn4 register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 15 - 91 Format of Forced Output Stop Function Control Register n4 (TKBPACTLn4) (1/2)

Address: F0779H (TKBPACTL04), F07B9H (TKBPACTL14), F0439H (TKBPACTL24)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TKBPACTLn4	TKBPACTENn11	TKBPACTENn10	TKBPACTLVn11	TKBPACTLVn10	TKBPACTENn01	TKBPACTENn00	TKBPACTLVn01	TKBPACTLVn00

TKBPACTENn11	TKBPACTENn10	Control of the operation of the forced output stop function pulse count register n1
0	0	The pulse characteristics measurement function is disabled.
0	1	The pulse characteristics measurement function for the TKBOn0 pin is enabled.
1	0	The pulse characteristics measurement function for the TKBOn1 pin is enabled.
1	1	Setting prohibited

TKBPACTLVn11	TKBPACTLVn10	Setting of the function of the forced output stop function pulse count register n1
0	0	Counting proceeds while the pulse is at the high level to measure the width and the result is stored in the corresponding TKBPAPLSn1 register.
0	1	Counting proceeds while the pulse is at the low level to measure the width and the result is stored in the corresponding TKBPAPLSn1 register.
1	0	Counting proceeds between two consecutive rising edges to measure the pulse interval and the result is stored in the corresponding TKBPAPLSn1 register.
1	1	Counting proceeds between two consecutive falling edges to measure the pulse interval and the result is stored in the corresponding TKBPAPLSn1 register.

TKBPACTENn01	TKBPACTENn00	Control of the operation of the forced output stop function pulse count register n0
0	0	The pulse characteristics measurement function is disabled.
0	1	The pulse characteristics measurement function for the TKBOn0 pin is enabled.
1	0	The pulse characteristics measurement function for the TKBOn1 pin is enabled.
1	1	Setting prohibited

Figure 15 - 91 Format of Forced Output Stop Function Control Register n4 (TKBPACTLn4) (2/2)

TKBPACTL Vn01	TKBPACTL Vn00	Setting of the function of the forced output stop function pulse count register n0
0	0	Counting proceeds while the pulse is at the high level to measure the width and the result is stored in the corresponding TKBPAPLSn0 register.
0	1	Counting proceeds while the pulse is at the low level to measure the width and the result is stored in the corresponding TKBPAPLSn0 register.
1	0	Counting proceeds between two consecutive rising edges to measure the pulse interval and the result is stored in the corresponding TKBPAPLSn0 register.
1	1	Counting proceeds between two consecutive falling edges to measure the pulse interval and the result is stored in the corresponding TKBPAPLSn0 register.

Remark n = 0 to 2; p = 0, 1

15.6.3.5 Pulse characteristics measurement capture register np (TKBPAPLSnp) (n = 0 to 2; p = 0, 1)

The TKBPAPLSnp register holds the counted pulse characteristics. The TKBPAPLSnp register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 15 - 92 Format of Pulse Characteristics Measurement Capture Register np (TKBPAPLSnp)

Address: F077AH (TKBPAPLS00), F07BAH (TKBPAPLS10), F043AH (TKBPAPLS20), F077CH (TKBPAPLS01), F07BCH (TKBPAPLS11), F043CH (TKBPAPLS21)

After reset: 0000H

R/W: R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKBPAPLSnp																

Remark n = 0 to 2; p = 0, 1

15.6.3.6 Pulse characteristics measurement capture register npL (TKBPAPLSnpL) (n = 0 to 2; p = 0, 1)

The TKBPAPLSnpL register reads and holds the lower-order 8 bits of the counted pulse characteristics stored in the TKBPAPLSnp register. The TKBPAPLSnpL register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 15 - 93 Format of Pulse Characteristics Measurement Capture Register npL (TKBPAPLSnpL)

Address: F077EH (TKBPAPLS00L), F07BEH (TKBPAPLS10L), F043EH (TKBPAPLS20L), F077FH (TKBPAPLS01L), F07BFH (TKBPAPLS11L), F043FH (TKBPAPLS21L)

After reset: 00H

R/W: R

Symbol	7	6	5	4	3	2	1	0
TKBPAPLSnpL								

Remark n = 0 to 2; p = 0, 1

15.6.3.7 Forced output stop function flag register n (TKBPAFLGn) (n = 0 to 2)

The TKBPAFLGn register indicates status flags for forced output stop function of the TKBOnp pin. The TKBPAFLGn register can be read by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 15 - 94 Format of Forced Output Stop Function Flag Register n (TKBPAFLGn)

Address: F0776H (TKBPAFLG0), F07B6H (TKBPAFLG1), F0436H (TKBPAFLG2)
After reset: 00H
R/W: R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
TKBPAFLGn	TKBPAFSFn1	TKBPAHSFn1	TKBPAFSFn0	TKBPAHSFn0	TKBPAFIFn1	TKBPAHIFn1	TKBPAFIFn0	TKBPAHIFn0

TKBPAFSFnp	Status flag of forced output stop function 2 for TKBOnp pin
0	Forced output stop function clear status
1	Forced output stop function status

TKBPAHSFnp	Status flag of forced output stop function 1 for TKBOnp pin
0	Forced output stop function clear status
1	Forced output stop function status

TKBPAFIFnp	Input monitor bit of forced output stop function 2 for TKBOnp pin
0	Forced output stop function 2 trigger signal is at low level (inactive)
1	Forced output stop function 2 trigger signal is at high level (active)

TKBPAHIFnp	Input monitor bit of forced output stop function 1 for TKBOnp pin
0	Forced output stop function 1 trigger signal is at low level (inactive)
1	Forced output stop function 1 trigger signal is at high level (active)

Caution **The timing to cancel the forced output stop function 1 depends on the setting. For details, see 15.7.2 Software cancel operation for forced output stop function 1.**

Remark n = 0 to 2; p = 0, 1

15.6.3.8 Forced output stop function 1 start trigger register n (TKBPAHFSn) (n = 0 to 2)

The TKBPAHFSn register is the start trigger register used by forced output stop function 1 of the TKBOnp pin. The TKBPAHFSn register can be written by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 15 - 95 Format of Forced Output Stop Function 1 Start Trigger Register n (TKBPAHFSn)

Address: F0774H (TKBPAHFS0), F07B4H (TKBPAHFS1), F0434H (TKBPAHFS2)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	<1>	<0>
TKBPAHFSn	0	0	0	0	0	0	TKBPAHTSn1	TKBPAHTSn0

TKBPAHTSnp	Start trigger of forced output stop function 1 for TKBOnp output
0	Invalid setting
1	Starts forced output stop function 1 for TKBOnp output

Caution 1. The TKBPAHFSn register can be overwritten while the timer is operating.
Caution 2. Be sure to clear bits 7 to 2 to 0.
Caution 3. When TKBPAHFSn register is read, 0 is read.

15.6.3.9 Forced output stop function 1 cancel trigger register n (TKBPAHFTn) (n = 0 to 2)

The TKBPAHFTn register is the cancel trigger register used by forced output stop function 1 of the TKBOnp pin. The TKBPAHFTn register can be written by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 15 - 96 Format of Forced Output Stop Function 1 Cancel Trigger Register n (TKBPAHFTn)

Address: F0775H (TKBPAHFT0), F07B5H (TKBPAHFT1), F0435H (TKBPAHFT2)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	<1>	<0>
TKBPAHFTn	0	0	0	0	0	0	TKBPAHTTn1	TKBPAHTTn0

TKBPAHTTnp	Cancel trigger of forced output stop function for TKBOnp output
0	Invalid setting
1	Clears forced output stop function 1 for TKBOnp output

Caution 1. The TKBPAHFTn register can be overwritten while the timer is operating.
Caution 2. Be sure to clear bits 7 to 2 to 0.
Caution 3. When TKBPAHFSn register is read, 0 is read.
Caution 4. The timing to cancel the forced output stop function 1 depends on the setting. For details, see 15.7.2 Software cancel operation for forced output stop function 1.

Remark n = 0 to 2; p = 0, 1

15.7 Operation of Forced Output Stop Function 1

Timer output can be fixed to Hi-Z, high, or low level directly (not via the CPU) and asynchronously with the operating clock f_{KBKC} of the 16-bit timer KB3n and timer RD2 when a trigger source occurs (outputs of comparators 0 to 3, INTP20, INTP21, and INTP0). The forced output stop status is canceled synchronously with the operating clock f_{KBKC} of the 16-bit timer KB3n and timer RD2 by setting the stop trigger of forced output stop function 1.

15.7.1 Summary for forced output stop function 1

In this function, comparator output signal or external interrupt signal, and software trigger are used as trigger signals for forced output stop function 1.
The output level selectable at forced output stop is controlled by the TKBPAMDnp[1:0] bits of the TKBPACTLnp register.
The following table shows the relationship of forced output stop function 1 of the p output pins (TKBOnp) of the 16-bit timer KB3n.

Table 15 - 12 Relationship of Forced Output Stop Function 1 of the TKBOnp Pin

TKBPAMDnp1	TKBPAMDnp0	Output Level Selection at Forced Output Stop Function 1 Execution
0	0	Hi-Z output
0	1	Hi-Z output
1	0	Fixed low-level output
1	1	Fixed high-level output

The selection for comparator output or external interrupt signal being used is controlled by the TKBPAHZSnp4 to TKBPAHZSnp0 bits of the forced output stop function control register np (TKBPACTLnp).
The following table shows trigger selection for forced output stop function 1 of the p output pins (TKBOnp) of the 16-bit timer KB3n.

Table 15 - 13 Trigger Selection for Forced Output Stop Function 1 of the TKBOnp Pin

Bit	Selectable Trigger Signals		
	16-bit Timer KB30	16-bit Timer KB31	16-bit Timer KB32
TKBPAHZSnp0	Comparator 0	Comparator 1	Comparator 0
TKBPAHZSnp1	Comparator 1	Comparator 2	Comparator 2
TKBPAHZSnp2	Comparator 2	Comparator 3	Comparator 3
TKBPAHZSnp3	INTP21	INTP20	INTP20
TKBPAHZSnp4	INTP0	INTP0	INTP0

Remark n = 0 to 2; p = 0, 1

15.7.2 Software cancel operation for forced output stop function 1

The table below shows the start trigger (TKBPAHTSnp bit for TKBPAHFSn register) setting to start forced output stop function 1.

Table 15 - 14 Operation of Start Trigger (TKBPAHTSnp Bit) of Forced Output Stop Function 1

TKBPAHTSnp	Start of Forced Output Stop Function by Software
0	Invalid setting
1	Writing 1 initiates the fixed output control of Hi-Z/low-level/high-level for TKBOnp output (the same function with rising edge detection of trigger signal by forced output stop function 1).

The table below shows the cancel trigger (TKBPAHTTnp bit for TKBPAHFTn register) setting to cancel forced output stop function 1.

Table 15 - 15 Operation of Cancel Trigger (TKBPAHTTnp Bit) at Forced Output Stop Function 1

TKBPACTLnp Register		Cancel of Forced Output Stop Function 1 by Software
TKBPAHCMnp1	TKBPAHCMnp0	
0	0	Forced output stop function 1 starts with trigger input, and forced output stop function 1 is cleared when forced output stop function release trigger (TKBPAHTTnp) = 1 is written, regardless of the trigger signal level.
0	1	Forced output stop function 1 starts with trigger input, and when the trigger signal is in its high-level period, writing “forced output stop function release trigger (TKBPAHTTnp) = 1” is invalid. Forced output stop function 1 is cleared when forced output stop function release trigger (TKBPAHTTnp) = 1 is written while the trigger signal is in its low-level period.
1	0	Forced output stop function 1 starts with trigger input, and forced output stop function 1 is cleared at the next counter period after forced output stop function release trigger (TKBPAHTTnp) = 1 is written, regardless of the trigger signal level**Note**.
1	1	Forced output stop function 1 starts with trigger input, and when the trigger signal is in its high-level period, writing “forced output stop function release trigger (TKBPAHTTnp) = 1” is invalid. Forced output stop function 1 is cleared at the next counter period after forced output stop function release trigger (TKBPAHTTnp) = 1 is written when the trigger signal is in its low-level period**Note**.

Note When the 16-bit timer KB3n is stopped (TKBCEn = 0) without waiting for the next counter period, the forced output stop function is kept on until the 16-bit timer KB3n is restarted (TKBCEn = 1).

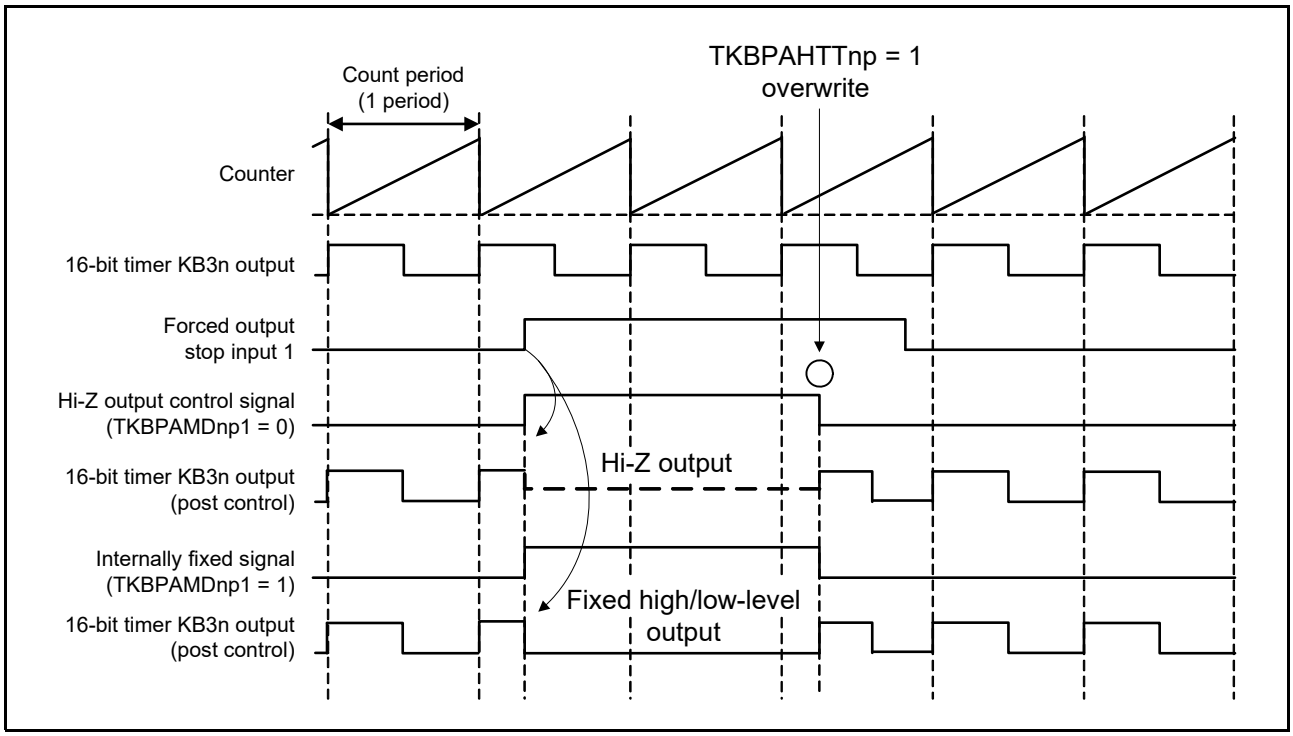
Remark n = 0 to 2; p = 0, 1

15.7.3 Basic operation of forced output stop function 1

This shows the operations of forced output function 1 with different setting of the TKBPAHCMnp[1:0] bits.
The trigger signal that initiates the forced output stop function 1 (forced output stop input 1) is an OR output of the trigger signal selected by the TKBPAHZSnp4 to TKBPAHZSnp0 bits of the forced output stop function control register np (TKBPACTLnp) and the TKBPAHTSnp bit of the trigger register n (TKBPAHFSn) that initiates forced output stop function 1.

1. Forced output stop function 1 at TKBPAHCMnp[1:0] = 00B

Figure 15 - 97 Forced Output Stop Function 1 at TKBPAHCMnp[1:0] = 00B



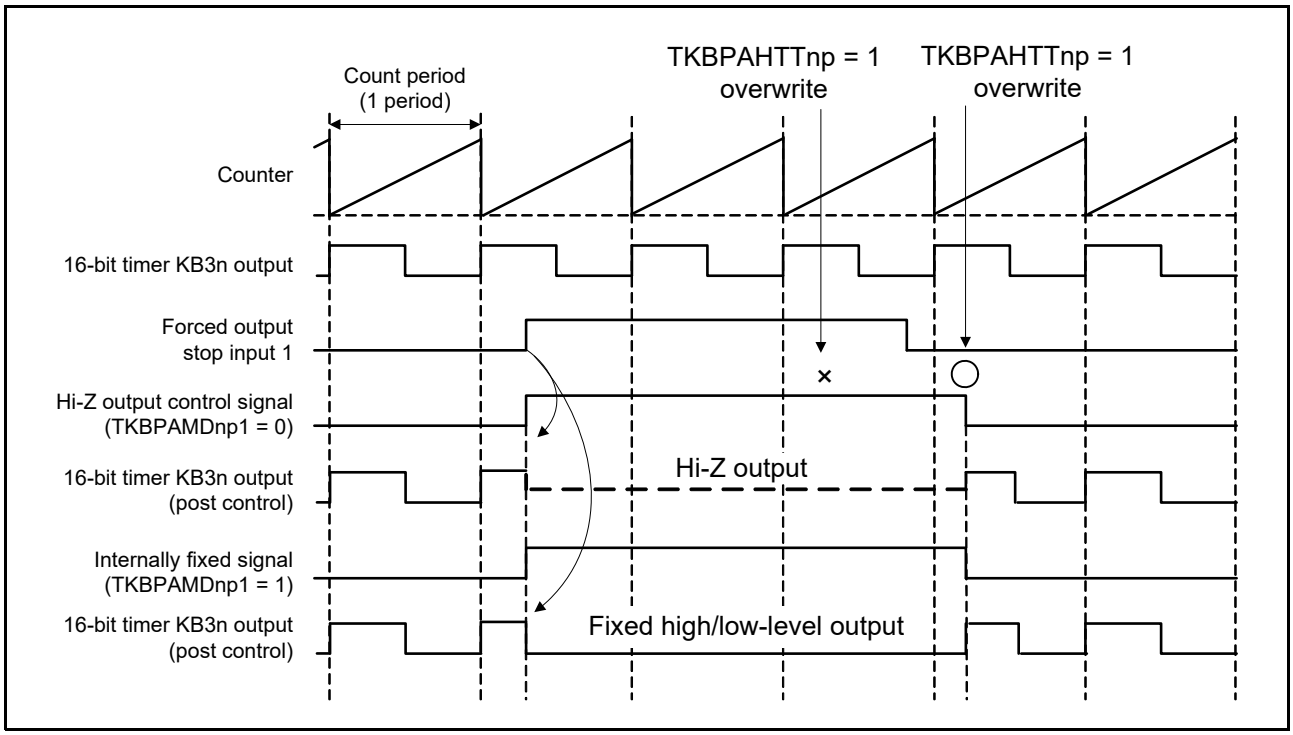
a) TKBPAMDnp1 = 0 (Hi-Z output)
Hi-Z output is realized via the detection of rising edge of forced output stop input 1.
Regardless of input level of forced output stop input 1, it returns to timer output via writing of 1 to cancel trigger (TKBPAHTTnp bit).
High-level period of Hi-Z output control signal is the period for forced output stop 1 (Hi-Z output).

b) TKBPAMDnp1 = 1 (fixed output)
Fixed output is realized in low-level/high-level according to the setting of the TKBPAMDnp0 bit at the detection of rising edge of forced output stop input 1.
Regardless of input level of forced output stop input 1, output level fixing is canceled and returned to timer output via writing of 1 to cancel trigger (TKBPAHTTnp bit).
High-level period of internal fixed signal is the period for forced output stop 1 (low-level/high-level output fixing).

Remark n = 0 to 2; p = 0, 1

2. Forced output stop function 1 at TKBPAHCMnp[1:0] = 01B

Figure 15 - 98 Forced Output Stop Function 1 at TKBPAHCMnp[1:0] = 01B



a) TKBPAMDnp1 = 0 (Hi-Z output)

Hi-Z output is realized via the detection of rising edge of forced output stop input 1.

During the high-level period of forced output stop input 1, writing 1 to cancel trigger (TKBPAHTTnp bit) is invalid.

After forced output stop input 1 turned into the low-level, it returns to timer output via writing of 1 to cancel trigger (TKBPAHTTnp bit).

High-level period of Hi-Z output control signal is the period for forced output stop 1 (Hi-Z output).

b) TKBPAMDnp1 = 1 (fixed output)

Fixed output is realized in low-level/high-level according to the setting of the TKBPAMDnp0 bit at the detection of rising edge of forced output stop input 1.

During the high-level period of forced output stop input 1, writing 1 to cancel trigger (TKBPAHTTnp bit) is invalid.

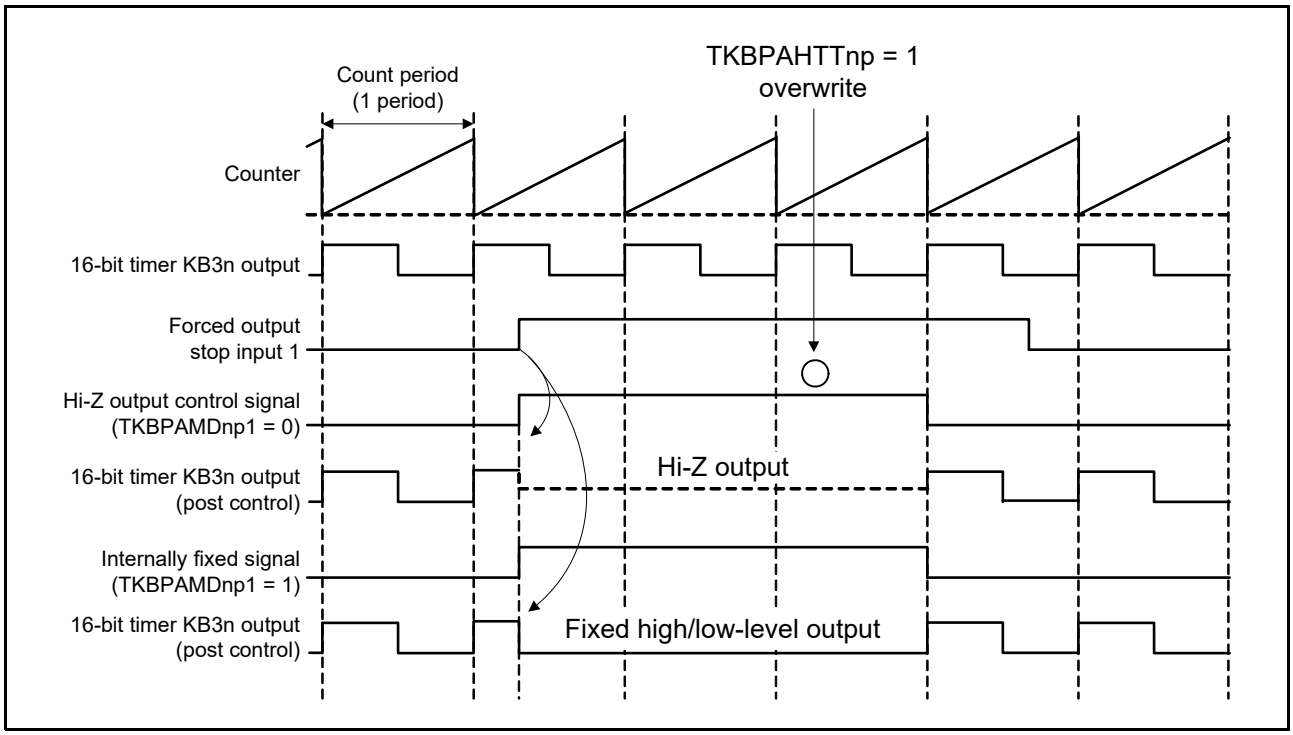
After the forced output stop input 1 turned into the low-level, output level fixing is canceled and returned to timer output via writing of 1 to cancel trigger (TKBPAHTTnp bit).

High-level period of internal fixed signal is the period for forced output stop 1 (low-level/high-level output fixing).

Remark n = 0 to 2; p = 0, 1

3. Forced output stop function 1 at TKBPAHCMnp[1:0] = 10B

Figure 15 - 99 Forced Output Stop Function 1 at TKBPAHCMnp[1:0] = 10B



a) TKBPAMDnp1 = 0 (Hi-Z output)

Hi-Z output is realized via the detection of rising edge of forced output stop input 1.

Regardless of the input level of forced output stop input 1, it returns to timer output after writing of 1 to cancel trigger (TKBPAHTTnp bit), in the next period.

High-level period of Hi-Z output control signal is the period for forced output stop 1 (Hi-Z output).

b) TKBPAMDnp1 = 1 (fixed output)

Fixed output is realized in low-level/high-level according to the setting of the TKBPAMDnp0 bit at the detection of rising edge of forced output stop input 1.

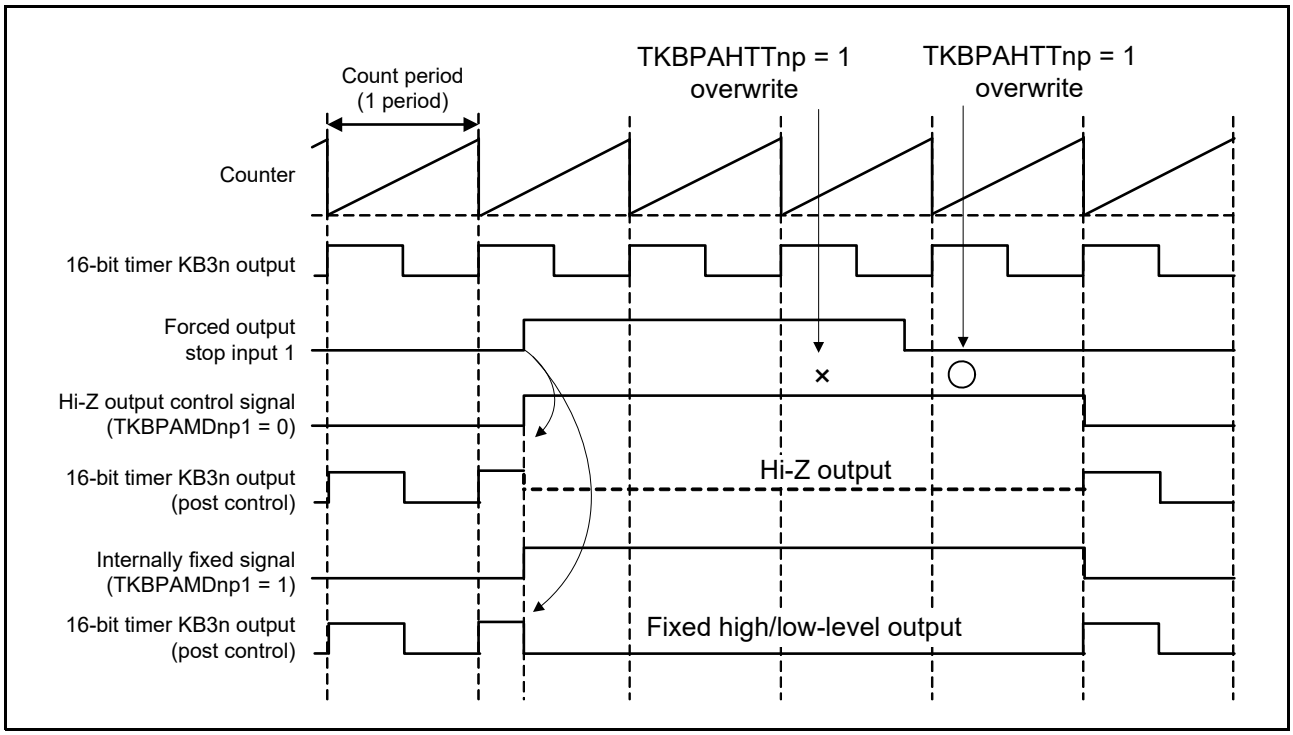
Regardless of input level of forced output stop input 1, output level fixing is canceled and returned to timer output after writing of 1 to cancel trigger (TKBPAHTTnp bit) in the next counter period.

High-level period of internal fixed signal is the period for forced output stop 1 (low-level/high-level output fixing).

Remark n = 0 to 2; p = 0, 1

4. Forced output stop function 1 at TKBPAHCMnp[1:0] = 11B

Figure 15 - 100 Forced Output Stop Function 1 at TKBPAHCMnp[1:0] = 11B



a) TKBPAMDnp1 = 0 (Hi-Z output)

Hi-Z output is realized via the detection of rising edge of forced output stop input 1.

During the high-level period of forced output stop input 1, writing 1 to cancel trigger (TKBPAHTTnp bit) is invalid.

It returns to timer output after writing of 1 to cancel trigger (TKBPAHTTnp bit) during the low-level period of forced output stop input 1, in the next period.

High-level period of Hi-Z output control signal is the period for forced output stop 1 (Hi-Z output).

b) TKBPAMDnp1 = 1 (fixed output)

Fixed output is realized in low-level/high-level according to the setting of the TKBPAMDnp0 bit at the detection of rising edge of forced output stop input 1.

During the high-level period of forced output stop input 1, writing 1 to cancel trigger (TKBPAHTTnp bit) is invalid.

During the low-level period of forced output stop input 1, output level fixing is canceled and returned to timer output in the next counter period after writing 1 to cancel trigger (TKBPAHTTnp bit).

High-level period of internal fixed signal is the period for forced output stop 1 (low-level/high-level output fixing).

Remark n = 0 to 2; p = 0, 1

15.8 Operation of Forced Output Stop Function 2

Timer output can be fixed to high or low level directly (not via the CPU) and asynchronously with the operating clock f_{KBKC} of the 16-bit timer KB3n and timer RD2 when a trigger source occurs (outputs of comparators 0 to 3, INTP20, INTP21, and INTP0). The forced output stop status is canceled at the beginning of the next counter cycle after the trigger source occurs or after the trigger source signal changes to inactive level.

15.8.1 Summary for forced output stop function 2

In this function, comparator output signal and external interrupt are used as trigger signals for forced output stop function 2.
The output level selectable at forced output stop is controlled by the TKBPAMDnp[1:0] bits of the TKBPACTLnp register.
The following table shows the relationship of forced output stop function 2 of the p output pins (TKBOnp) of the 16-bit timer KB3n.

Table 15 - 16 Relationship of Forced Output Stop Function 2 of the TKBOnp Pin

TKBPAMDnp1	TKBPAMDnp0	Output Level Selection at Forced Output Stop Function 2 Execution
0	0	Fixed low-level output
0	1	Fixed high-level output
1	0	Fixed low-level output
1	1	Fixed high-level output

The selection for comparator output or external interrupt signal being used is controlled by the TKBPAHZSnp4 to TKBPAHZSnp0 bits of the forced output stop function control register np (TKBPACTLnp).
The following table shows trigger selection for forced output stop function 2 of the p output pins (TKBOnp) of the 16-bit timer KB3n.

Table 15 - 17 Trigger Selection for Forced Output Stop Function 2 of the TKBOnp Pin

Bit	Selectable Trigger Signals		
	16-bit Timer KB30	16-bit Timer KB31	16-bit Timer KB32
TKBPAFXSnp0	Comparator 0	Comparator 1	Comparator 0
TKBPAFXSnp1	Comparator 1	Comparator 2	Comparator 2
TKBPAFXSnp2	Comparator 2	Comparator 3	Comparator 3
TKBPAFXSnp3	INTP21	INTP20	INTP20
TKBPAFXSnp4	INTP0	INTP0	INTP0

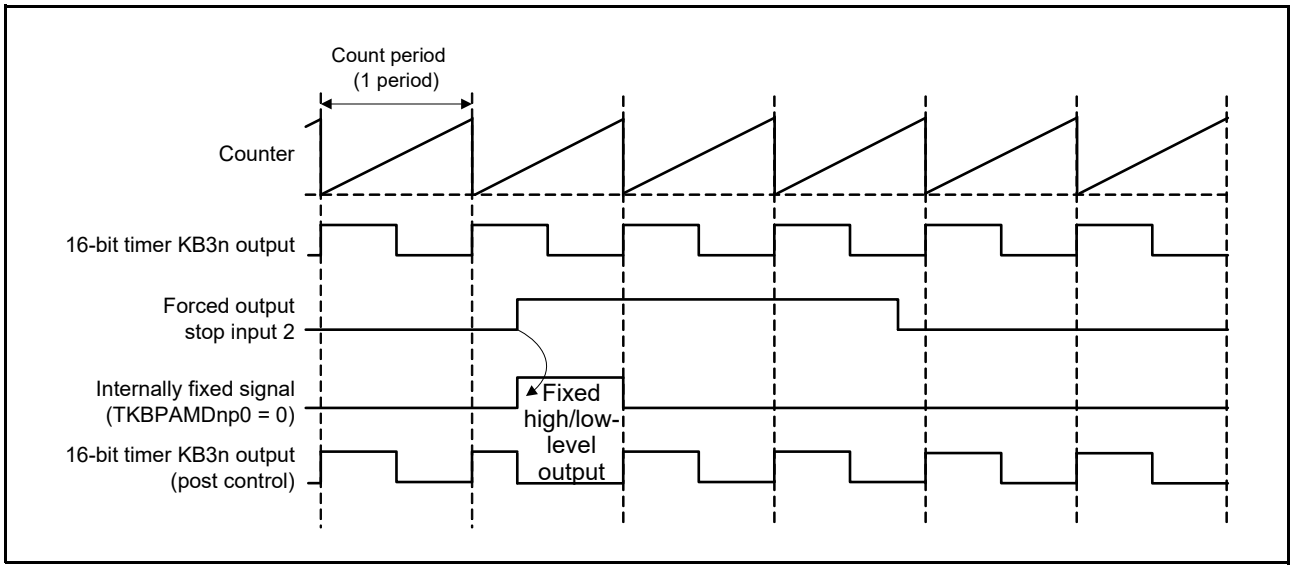
Remark n = 0 to 2; p = 0, 1

15.8.2 Basic operation of forced output stop function 2

This shows the operations of forced output function 2 with different setting of TKBPAFCMnp bits.
The trigger signal that initiates the forced output stop function 2 (forced output stop input 2) is the trigger signal selected by the TKBPAFXSnp0 to TKBPAFXSnp4 bits of the forced output stop function control register np (TKBPACTLnp).

1. Forced output stop function 2 at TKBPAFCMnp = 0

Figure 15 - 101 Forced Output Stop Function 2 at TKBPAFCMnp = 0

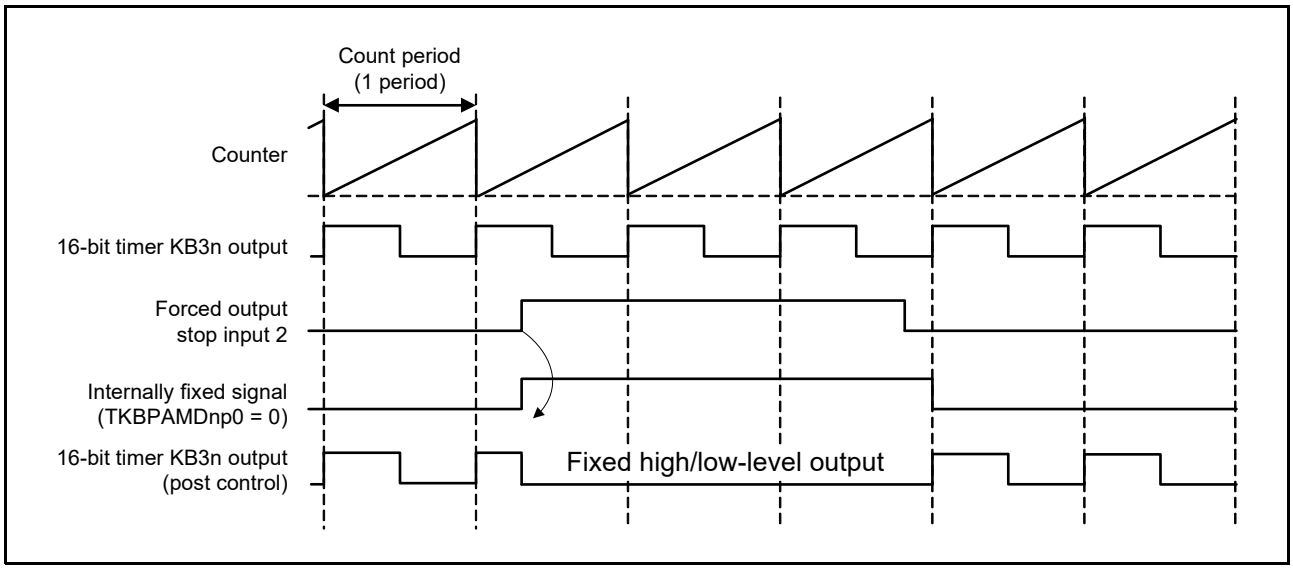


Fixed output is realized in low-level/high-level according to TKBPAMDnp0 setting at the detection of rising edge of forced output stop input 2.
Regardless of the input level of the forced output stop input 2, the fixing of output level is canceled at the next counter cycle and returned to timer output.
High-level period of internal fixed signal is the period for forced output stop 2 (low-level/high-level output fixing).

Remark n = 0 to 2; p = 0, 1

2. Forced output stop function 2 at TKBPAFCMnp = 1

Figure 15 - 102 Forced Output Stop Function 2 at TKBPAFCMnp = 1



Fixed output is realized in low-level/high-level according to TKBPAMDnp0 setting at the detection of rising edge of forced output stop input 2.

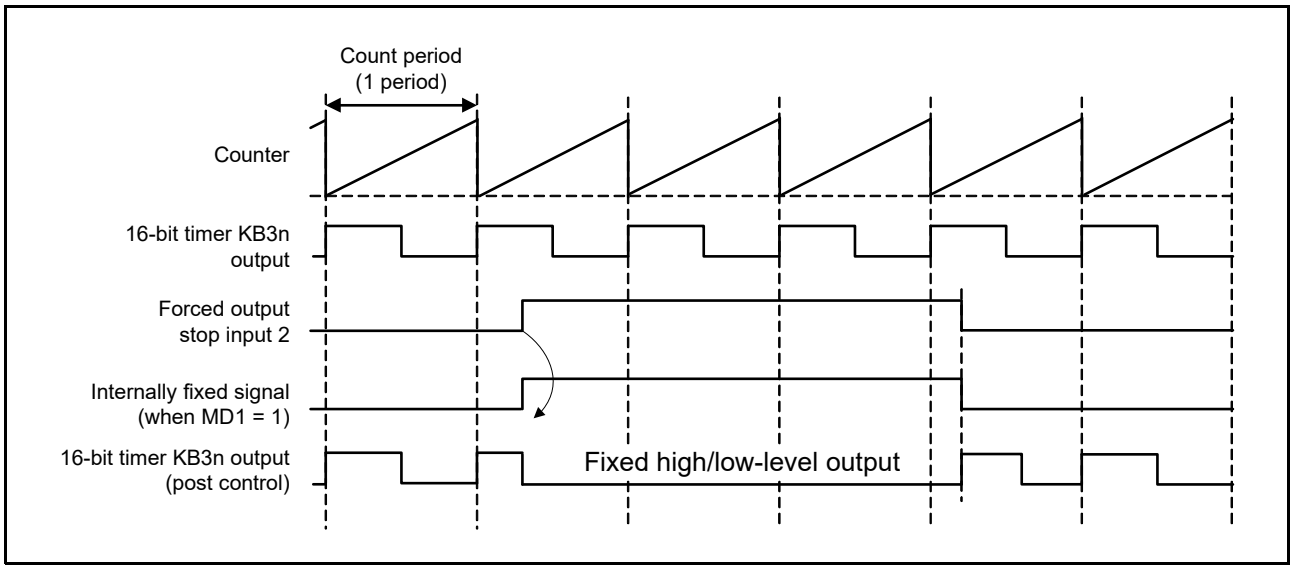
After the falling edge of forced output stop input 2, the fixing of output level is canceled at the next counter cycle and returned to timer output.

High-level period of internal fixed signal is the period for forced output stop 2 (low-level/high-level output fixing).

Remark n = 0 to 2; p = 0, 1

3. Forced output stop function 2 when TKBPAFIXnp = 1 (that is, the fixed off function)

Figure 15 - 103 Forced Output Stop Function 2 When TKBPAFIXnp = 1 (That is, the Fixed Off Function)



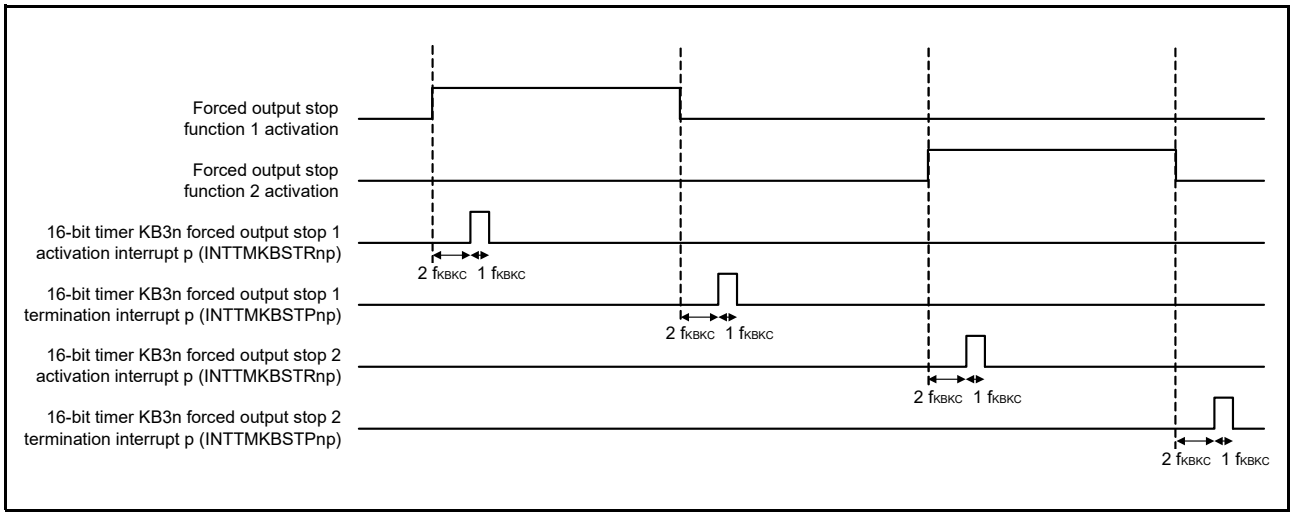
Fixed output is realized in low-level/high-level according to TKBPAMDnp0 setting at the detection of rising edge of forced output stop input 2.
After the falling edge of forced output stop input 2, the fixing of output level is canceled synchronously with a detection signal and returned to timer output.
High-level period of internal fixed signal is the period of forced output stop 2 (low-level/high-level output fixing).
Activation of forced stopping of the output is not synchronized with f_{KBKC} so it proceeds on a rising edge of forced output stop input 2.
Termination of forced stopping of the output is synchronized with f_{KBKC}, so it proceeds when two to three cycles of f_{KBKC} have elapsed after a falling edge of forced output stop input 2.
If a further rising edge of forced output stop input 2 is detected while synchronization processing is in progress, release from the forced output stop state does not proceed so the fixed output state is retained.

Remark n = 0 to 2; p = 0, 1

15.8.3 Interrupt output to indicate activation and termination of forced output stopping

Interrupt signals can be output to indicate the activation or termination of forced stopping of an output.
Interrupt output is controlled by setting forced output stop function control register n3 (TKBPACTLn3).

Figure 15 - 104 Interrupt Output to Indicate Activation and Termination of Forced Output Stopping



INTTMKBSTRn1, INTTMKBSTPn1, INTTMKBSTRn2, and INTTMKBSTPn2 are output for one clock period of f_{KBKC} to indicate the activation and termination of forced output stop function 1 and the activation and termination of forced output stop function 2, respectively, except in the following situation. If both forced output stop functions 1 and 2 are activated or terminated simultaneously, the signals are output for two clock periods of f_{KBKC}.
Since activation and termination of forced output stop functions 1 and 2 are asynchronous with respect to f_{KBKC}, interrupt outputs are delayed for two to three cycles of f_{KBKC} from the activation or termination.

Remark n = 0 to 2; p = 0, 1

15.9 Operation of the Pulse Characteristics Measurement Function

15.9.1 Overview of the pulse characteristics measurement function

The characteristics of the pulses in the PWM output signal actually output to a TKBOnp pin can be measured. To use the pulse characteristics measurement function, use forced output stop function control register n4 (TKBPACTLn4) to select the TKBOnp pin for which measurement is to proceed and the targets for measurement (widths at high or low level or the intervals between rising or falling edges). The two results of measurement are stored in pulse characteristics measurement capture registers n0 and n1 (TKBPAPLSn0 and TKBPAPLSn1). Pulse characteristics measurement capture registers n0L and n1L (TKBPAPLSn0L and TKBPAPLSn1L) can be used to read the values of the 8 lower-order bits of pulse characteristics measurement capture registers n0 and n1 (TKBPAPLSn0 and TKBPAPLSn1).

Measured result for pulse characteristic n0 = Setting of pulse characteristics measurement capture register n0 (TKBPAPLSn0) × period of the 16-bit timer KB3n counter clock
Measured result for pulse characteristic n1 = Setting of pulse characteristics measurement capture register n1 (TKBPAPLSn1) × period of the 16-bit timer KB3n counter clock

Caution When a forced output stop is activated or terminated during measurement, the measured results may include an error of one or two cycles of the counter clock.

When measured results for pulse characteristics n0 and n1 exceed FFFFH cycles, FFFFH and FFH are stored in pulse characteristics measurement capture registers n0 and n1 (TKBPAPLSn0 and TKBPAPLSn1) and pulse characteristics measurement capture registers n0L and n1L (TKBPAPLSn0L and TKBPAPLSn1L), respectively.

Remark n = 0 to 2; p = 0, 1

15.9.2 Basic operation of the pulse characteristics measurement function

The dedicated internal 16-bit counter is used in pulse characteristics measurement. The results for pulse characteristics measured by the internal 16-bit counter are stored in pulse characteristics measurement capture registers n0 and n1 (TKBPAPLSn0 and TKBPAPLSn1). Pulse characteristics measurement capture registers n0 and n1 (TKBPAPLSn0 and TKBPAPLSn1) are read in response to any of the following three types of interrupt: 16-bit timer KB3n interrupt (INTTMKBn), 16-bit timer KB3n forced output stop 1 or 2 activation interrupt p (INTTMKBSTRnp), and 16-bit timer KB3n forced output stop 1 or 2 termination interrupt p (INTTMKBSTPnp).
The operations of the pulse characteristics measurement function, corresponding to the respective settings of the TKBPACTLVnp[1:0] bits, are shown on the following pages.

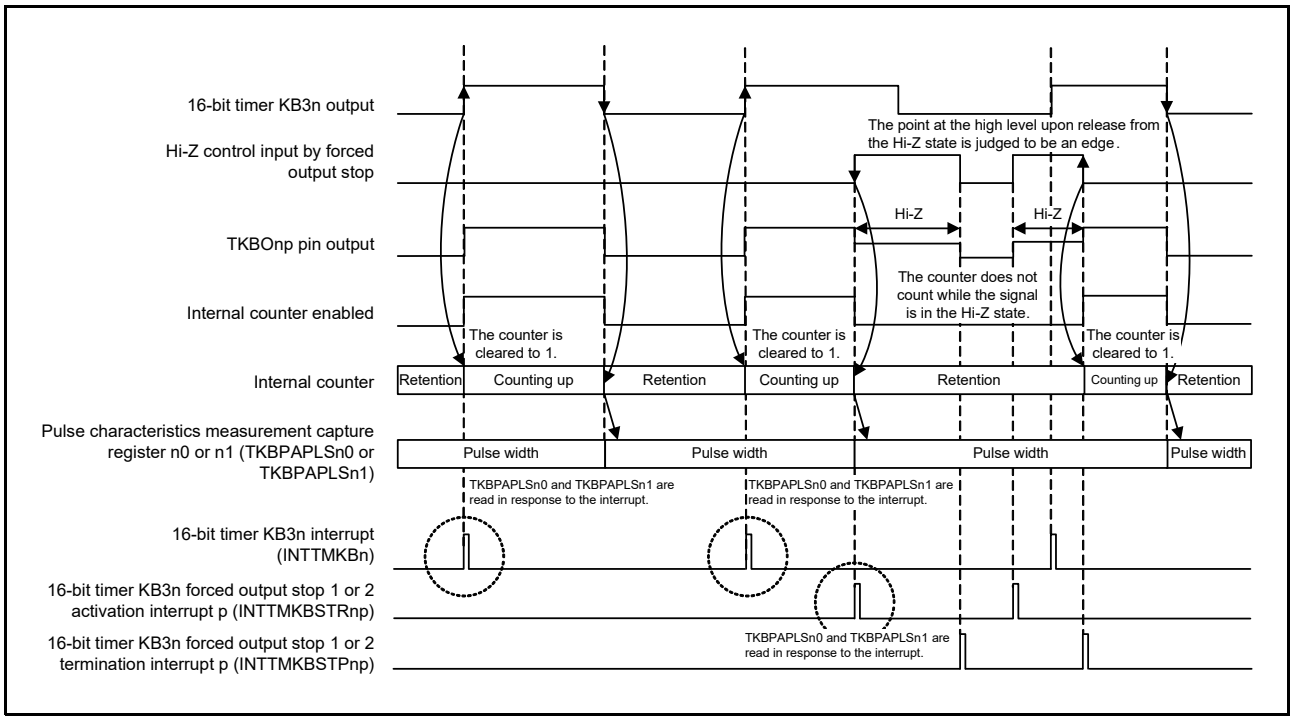
Remark n = 0 to 2; p = 0, 1

1. Measurement of widths at high level (selected when the setting of the TKBPACTLVnp[1:0] bits is 00B)
   For measurement of a width at high level, the internal 16-bit counter starts counting from 0001H in response to detecting a rising edge on the TKBOnp pin and stops counting in response to a falling edge on the TKBOnp pin while counting is in progress. Pulse characteristics measurement capture register n0 or n1 (TKBPAPLSn0 or TKBPAPLSn1) holds the value of the internal 16-bit counter at the time of detecting a falling edge on the TKBOnp pin while counting was in progress. **Figure 15 - 105** to **Figure 15 - 107** show examples of operation when forced output stop (Hi-Z, fixed low-level, or fixed high-level output) is applied.

Figure 15 - 105 shows an example of operation when a TKBOnp pin is placed in the high-Z state in response to the activation of a forced output stop before or after the period over which PWM output is at the high level. **Figure 15 - 106** shows an example of operation when a TKBOnp output goes low in response to the activation of a forced output stop before or after the period over which PWM output is at the high level.

- The width at high level measured following detection of a falling edge that occurred in response to the activation of a forced output stop is stored in pulse characteristics measurement capture register n0 or n1 (TKBPAPLSn0 or TKBPAPLSn1) and read from the given register in response to the 16-bit timer KB3n forced output stop 1 or 2 activation interrupt p (INTTMKBSTRnp).
- The width at high level measured following detection of a falling edge of PWM output while forced output stop is inactive is stored in pulse characteristics measurement capture register n0 or n1 (TKBPAPLSn0 or TKBPAPLSn1) and read from the given register in response to the 16-bit timer KB3n interrupt (INTTMKBn).

Figure 15 - 105 Pulse Characteristics Measurement Function (Timing for Measurement of Widths at High Level: Hi-Z Output during a Forced Output Stop, TKBIOCn0.TKBTOLnp = 0)



The timing with which TKBOnp output changes from the high level to the Hi-Z state in response to the activation of a forced output stop is treated as detection of a falling edge, and the timing with which TKBOnp output changes from the Hi-Z state to the high level in response to the termination of a forced output stop is treated as detection of a rising edge.

Figure 15 - 106 Pulse Characteristics Measurement Function (Timing for Measurement of Widths at High Level: Fixed Low-level Output during a Forced Output Stop, TKBIOCn0.TKBTOLnp = 0)

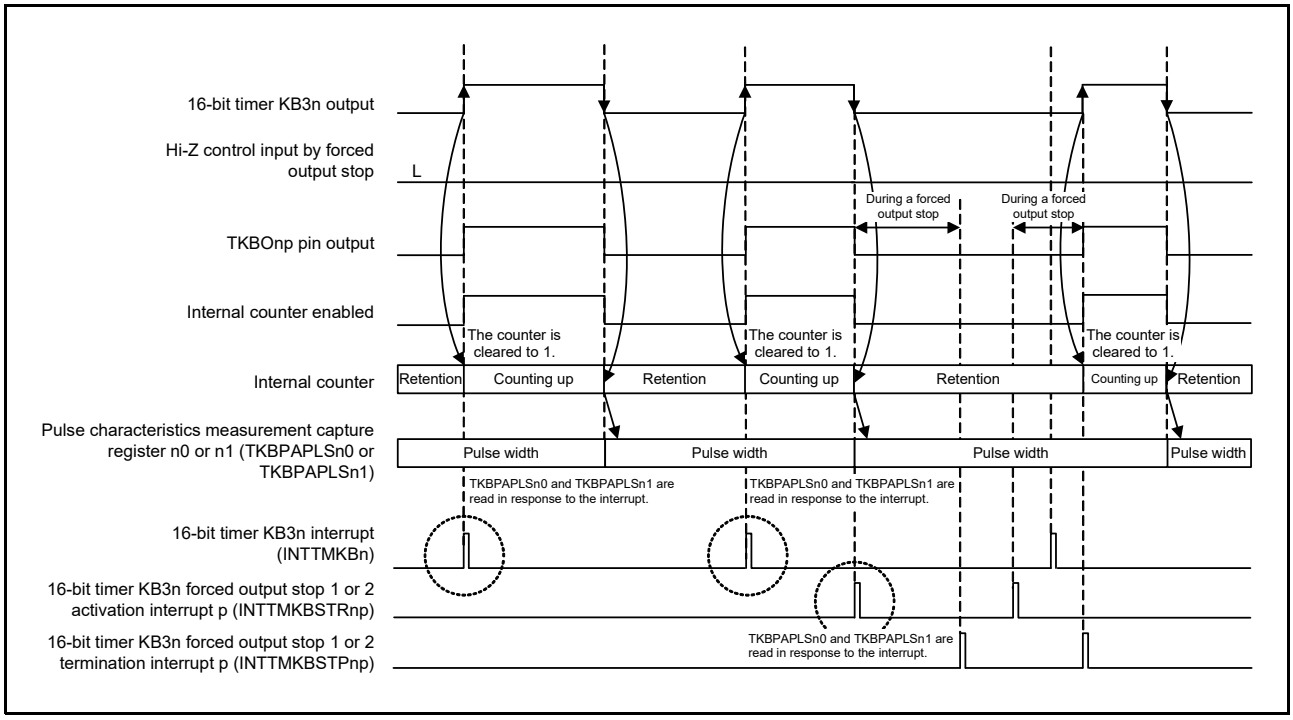
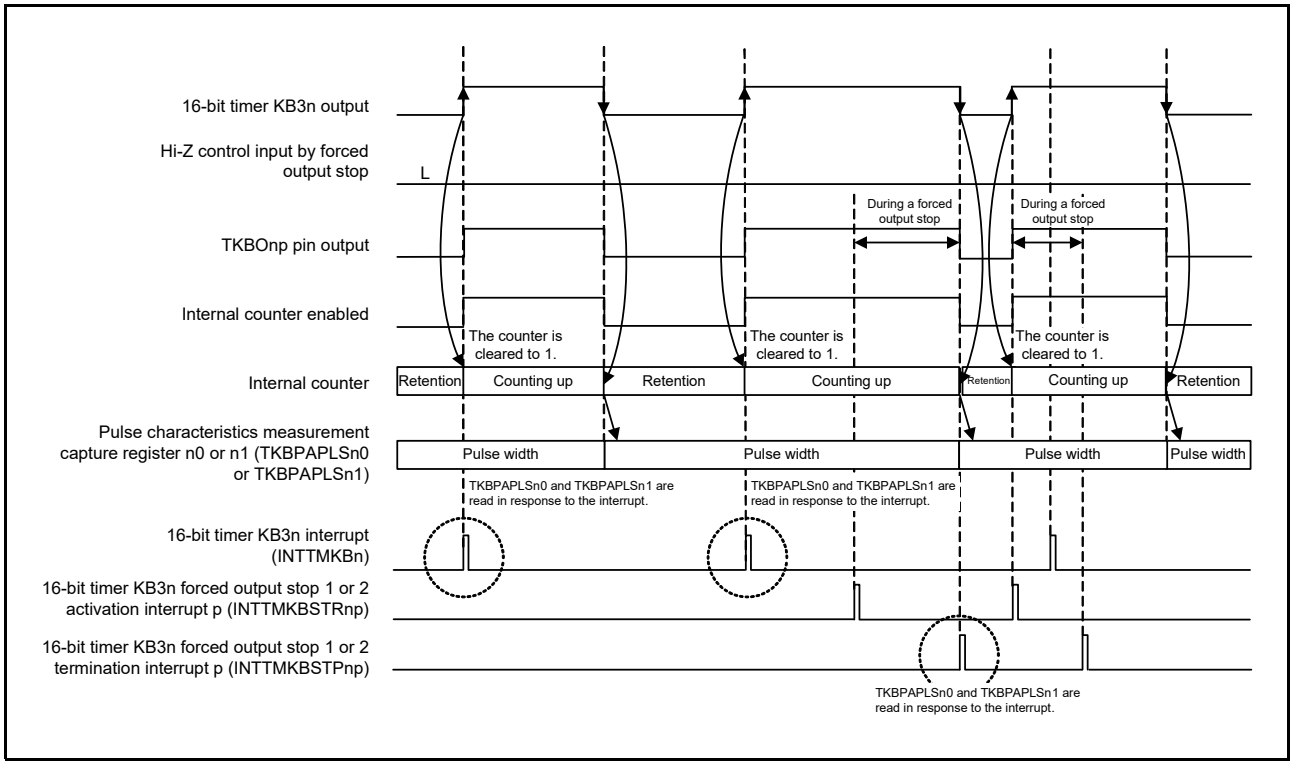


Figure 15 - 107 shows an example of operation when a TKBOnp output goes high in response to the activation of a forced output stop before or after the period over which PWM output is at the high level.

- The width at high level measured following detection of a falling edge that occurred in response to the termination of a forced output stop is stored in pulse characteristics measurement capture register n0 or n1 (TKBPAPLSn0 or TKBPAPLSn1) and read from the given register in response to the 16-bit timer KB3n forced output stop 1 or 2 termination interrupt p (INTTMKBSTPnp).
- The width at high level measured following detection of a falling edge of PWM output while forced output stop is inactive is stored in pulse characteristics measurement capture register n0 or n1 (TKBPAPLSn0 or TKBPAPLSn1) and read from the given register in response to the 16-bit timer KB3n interrupt (INTTMKBn).

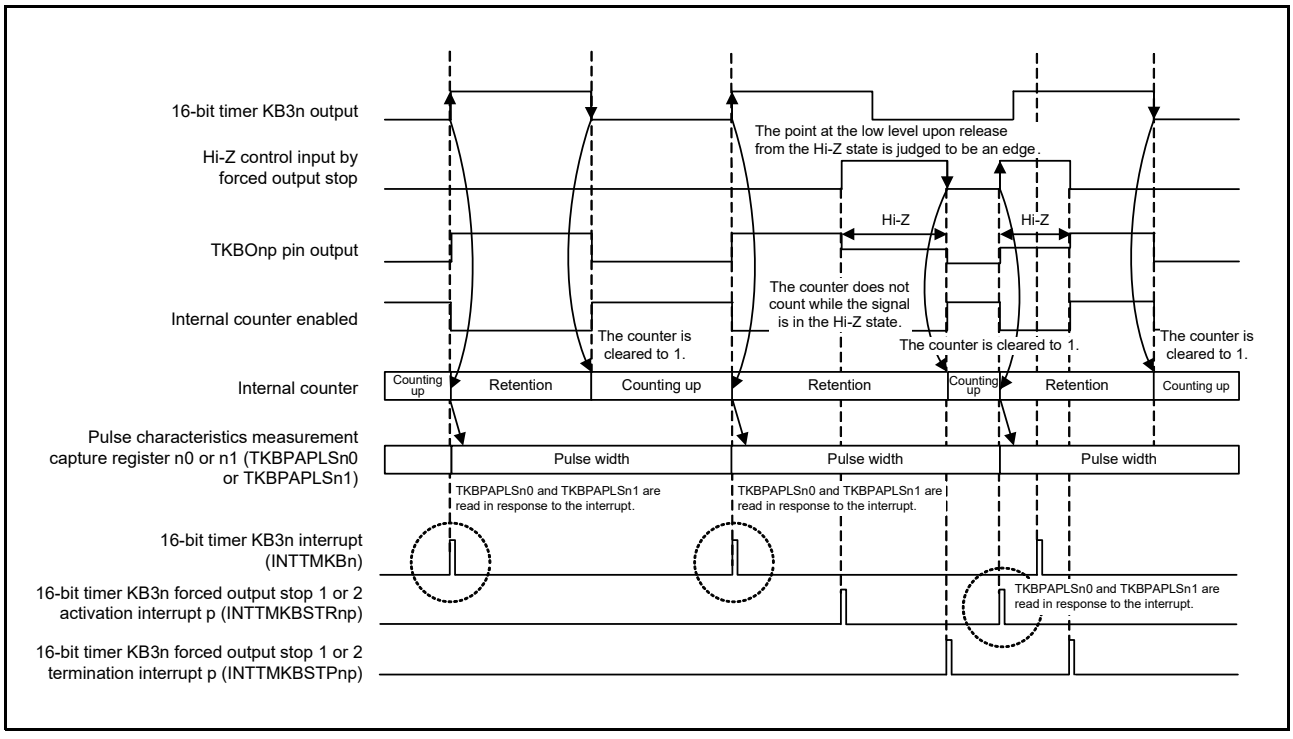
Figure 15 - 107 Pulse Characteristics Measurement Function (Timing for Measurement of Widths at High Level: Fixed High-level Output during a Forced Output Stop, TKBIOCn0.TKBTOLnp = 0)



2. Measurement of widths at low level (selected when the setting of the TKBPACTLVnp[1:0] bits is 01B)
   For measurement of a width at low level, the internal 16-bit counter starts counting from 0001H in response to detecting a falling edge on the TKBOnp pin and stops counting in response to a rising edge on the TKBOnp pin while counting is in progress. Pulse characteristics measurement capture register n0 or n1 (TKBPAPLSn0 or TKBPAPLSn1) holds the value of the internal 16-bit counter at the time of detecting a rising edge on the TKBOnp pin while counting was in progress. **Figure 15 - 108** to **15 - 110** show examples of operation when forced output stop (Hi-Z, fixed high-level, or fixed low-level output) is applied.

   Figure 15 - 108 shows an example of operation when a TKBOnp pin is placed in the high-Z state in response to the activation of a forced output stop before or after the period over which PWM output is at the high level. **Figure 15 - 109** shows an example of operation when a TKBOnp output goes high in response to the activation of a forced output stop before or after the period over which PWM output is at the high level.
   - The width at low level measured following detection of a rising edge that occurred in response to the activation of a forced output stop is stored in pulse characteristics measurement capture register n0 or n1 (TKBPAPLSn0 or TKBPAPLSn1) and read from the given register in response to the 16-bit timer KB3n forced output stop 1 or 2 activation interrupt p (INTTMKBSTRnp).
   - The width at low level measured following detection of a rising edge of PWM output while forced output stop is inactive is stored in pulse characteristics measurement capture register n0 or n1 (TKBPAPLSn0 or TKBPAPLSn1) and read from the given register in response to the 16-bit timer KB3n interrupt (INTTMKBn).

Figure 15 - 108 Pulse Characteristics Measurement Function (Timing for Measurement of Widths at Low Level: Hi-Z Output during a Forced Output Stop, TKBIOCn0.TKBTOLnp = 0)



The timing with which TKBOnp output changes from the low level to the Hi-Z state in response to the activation of a forced output stop is treated as detection of a rising edge, and the timing with which TKBOnp output changes from the Hi-Z state to the low level in response to the termination of a forced output stop is treated as detection of a falling edge.

Figure 15 - 109 Pulse Characteristics Measurement Function (Timing for Measurement of Widths at Low Level: Fixed High-level Output during a Forced Output Stop, TKBIOCn0.TKBTOLnp = 0)

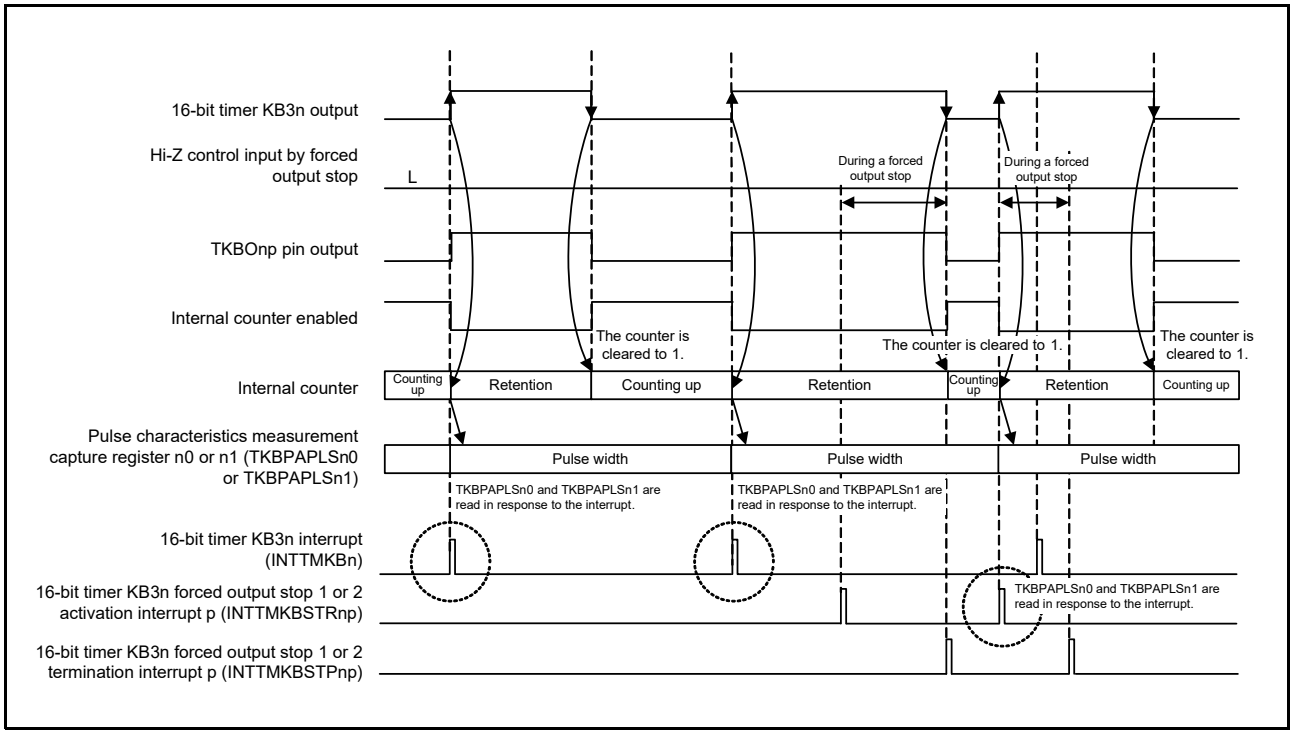
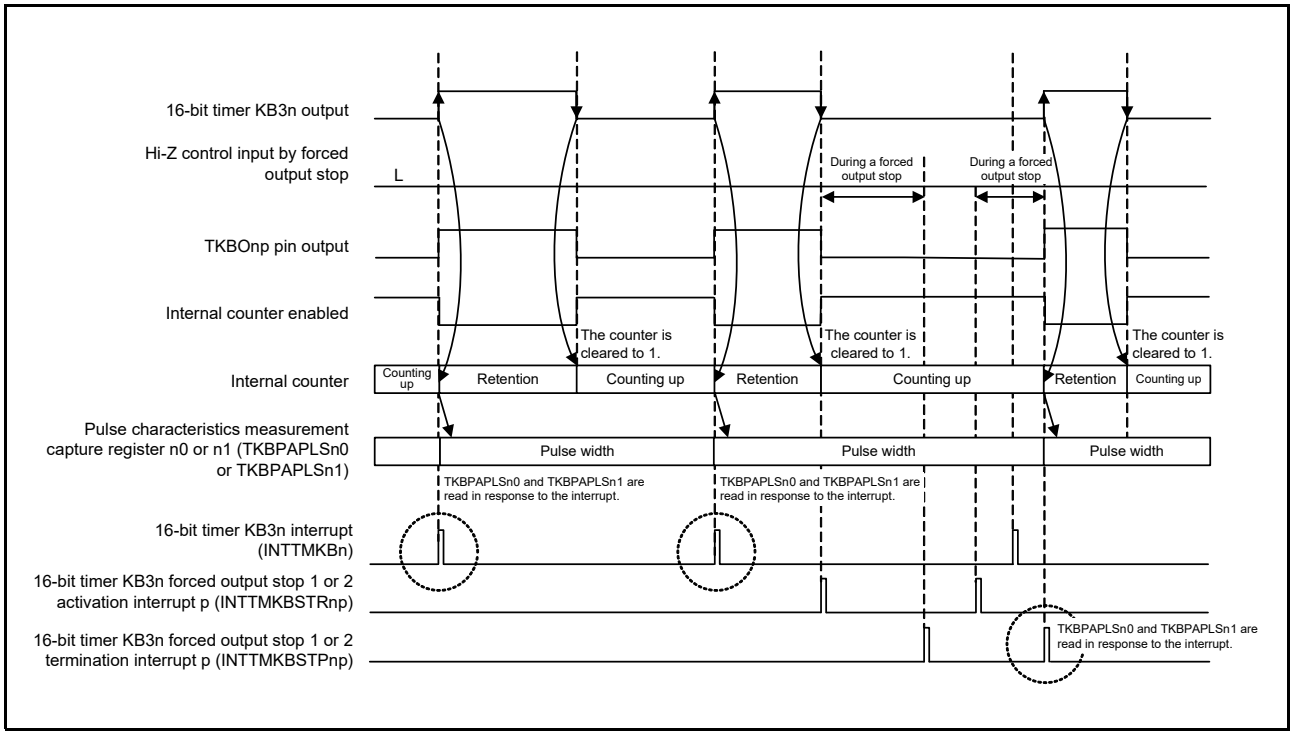


Figure 15 - 110 shows an example of operation when a TKBOnp output goes low in response to the activation of a forced output stop before or after the period over which PWM output is at the high level.

- The width at low level measured following detection of a rising edge that occurred in response to the termination of a forced output stop is stored in pulse characteristics measurement capture register n0 or n1 (TKBPAPLSn0 or TKBPAPLSn1) and read from the given register in response to the 16-bit timer KB3n forced output stop 1 or 2 termination interrupt p (INTTMKBSTPnp).
- The width at low level measured following detection of a rising edge of PWM output while forced output stop is inactive is stored in pulse characteristics measurement capture register n0 or n1 (TKBPAPLSn0 or TKBPAPLSn1) and read from the given register in response to the 16-bit timer KB3n interrupt (INTTMKBn).

Figure 15 - 110 Pulse Characteristics Measurement Function (Timing for Measurement of Widths at Low Level: Fixed Low-level Output during a Forced Output Stop, TKBIOCn0.TKBTOLnp = 0)



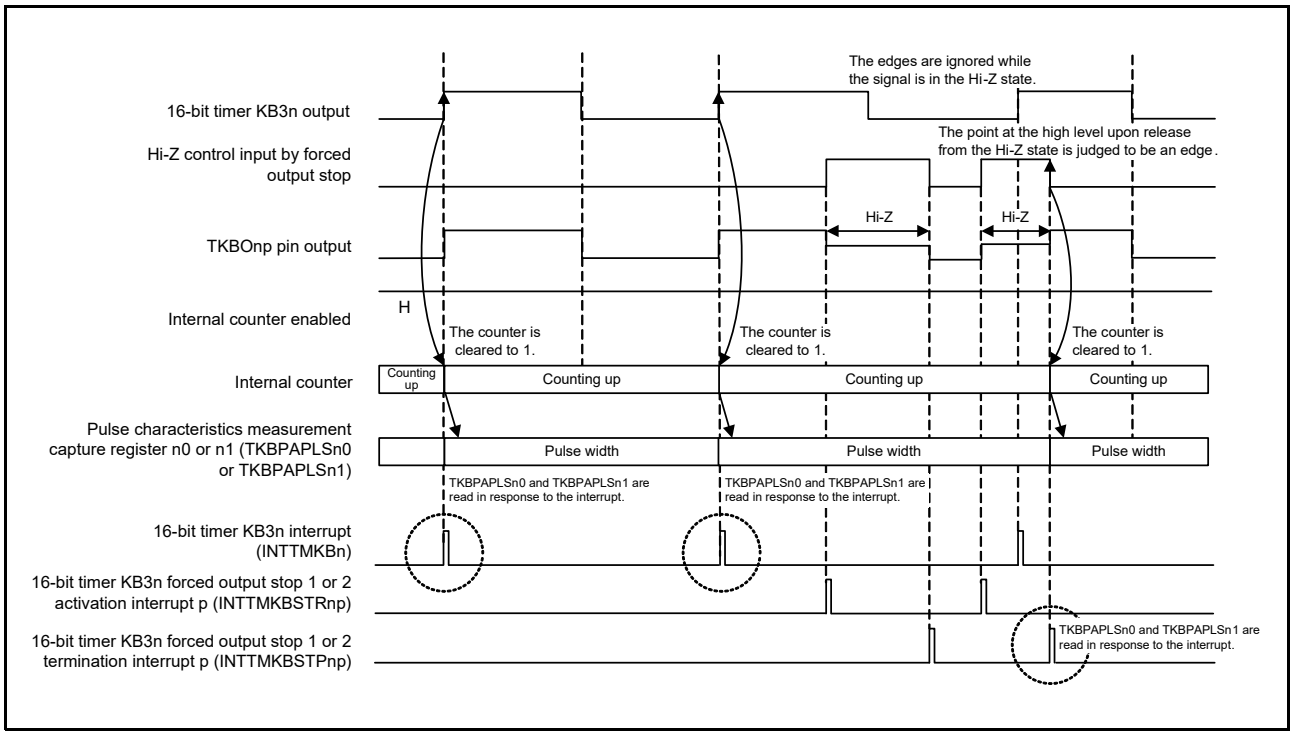
3. Measurement of pulse intervals between two consecutive rising edges (selected when the setting of the TKBPACTLVnp[1:0] bits is 10B)

   For measurement of a pulse interval between two consecutive rising edges, the internal 16-bit counter starts counting from 0001H in response to detecting a rising edge on the TKBOnp pin. Pulse characteristics measurement capture register n0 or n1 (TKBPAPLSn0 or TKBPAPLSn1) holds the value of the internal 16-bit counter at the time of detecting a rising edge on the TKBOnp pin while counting was in progress. **Figure 15 - 111** to **15 - 113** show examples of operation when forced output stop (Hi-Z, fixed low-level, or fixed high-level output) is applied.

   Figure 15 - 111 shows an example of operation when a TKBOnp pin is placed in the high-Z state in response to the activation of a forced output stop before or after the period between two consecutive rising edges of PWM output. **Figure 15 - 112** shows an example of operation when a TKBOnp output goes low in response to the activation of a forced output stop before or after the period between two consecutive rising edges of PWM output.

   - The pulse interval measured following detection of a rising edge that occurred in response to the termination of a forced output stop is stored in pulse characteristics measurement capture register n0 or n1 (TKBPAPLSn0 or TKBPAPLSn1) and read from the given register in response to the 16-bit timer KB3n forced output stop 1 or 2 termination interrupt p (INTTMKBSTPnp).
   - The pulse interval measured following detection of a rising edge of PWM output while forced output stop is inactive is stored in pulse characteristics measurement capture register n0 or n1 (TKBPAPLSn0 or TKBPAPLSn1) and read from the given register in response to the 16-bit timer KB3n interrupt (INTTMKBn).

Figure 15 - 111 Pulse Characteristics Measurement Function (Timing for Measurement of Pulse Intervals between Two Consecutive Rising Edges: Hi-Z Output during a Forced Output Stop, TKBIOCn0.TKBTOLnp = 0)



The timing with which TKBOnp output changes from the Hi-Z state to the high level in response to the termination of a forced output stop is treated as detection of a rising edge.

Figure 15 - 112 Pulse Characteristics Measurement Function (Timing for Measurement of Pulse Intervals between Two Consecutive Rising Edges: Fixed Low-level Output during a Forced Output Stop, TKBIOCn0.TKBTOLnp = 0)

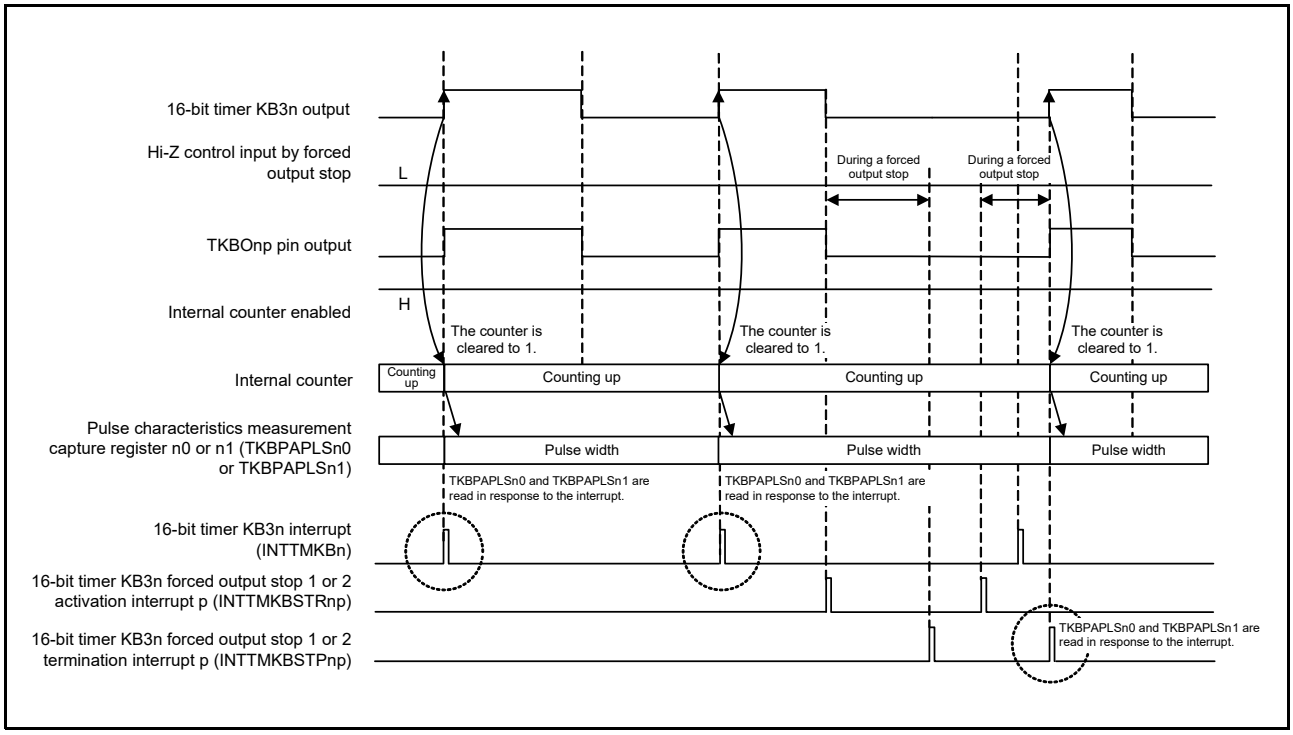
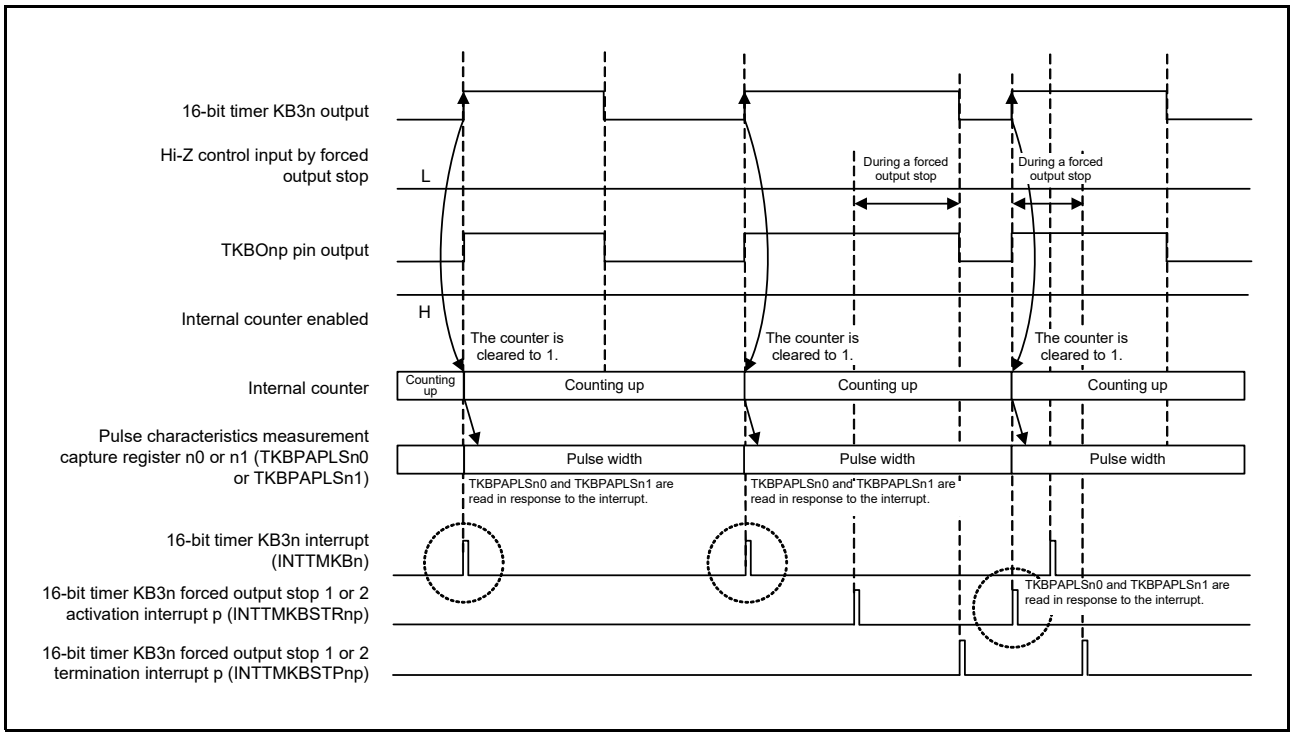


Figure 15 - 113 shows an example of operation when a TKBOnp output goes high in response to the activation of a forced output stop before or after the period between two consecutive rising edges of PWM output.

- The pulse interval measured following detection of a rising edge that occurred in response to the activation of a forced output stop is stored in pulse characteristics measurement capture register n0 or n1 (TKBPAPLSn0 or TKBPAPLSn1) and read from the given register in response to the 16-bit timer KB3n forced output stop 1 or 2 activation interrupt p (INTTMKBSTRnp).
- The pulse interval measured following detection of a rising edge of PWM output while forced output stop is inactive is stored in pulse characteristics measurement capture register n0 or n1 (TKBPAPLSn0 or TKBPAPLSn1) and read from the given register in response to the 16-bit timer KB3n interrupt (INTTMKBn).

Figure 15 - 113 Pulse Characteristics Measurement Function (Timing for Measurement of Pulse Intervals between Two Consecutive Rising Edges: Fixed High-level Output during a Forced Output Stop, TKBIOCn0.TKBTOLnp = 0)



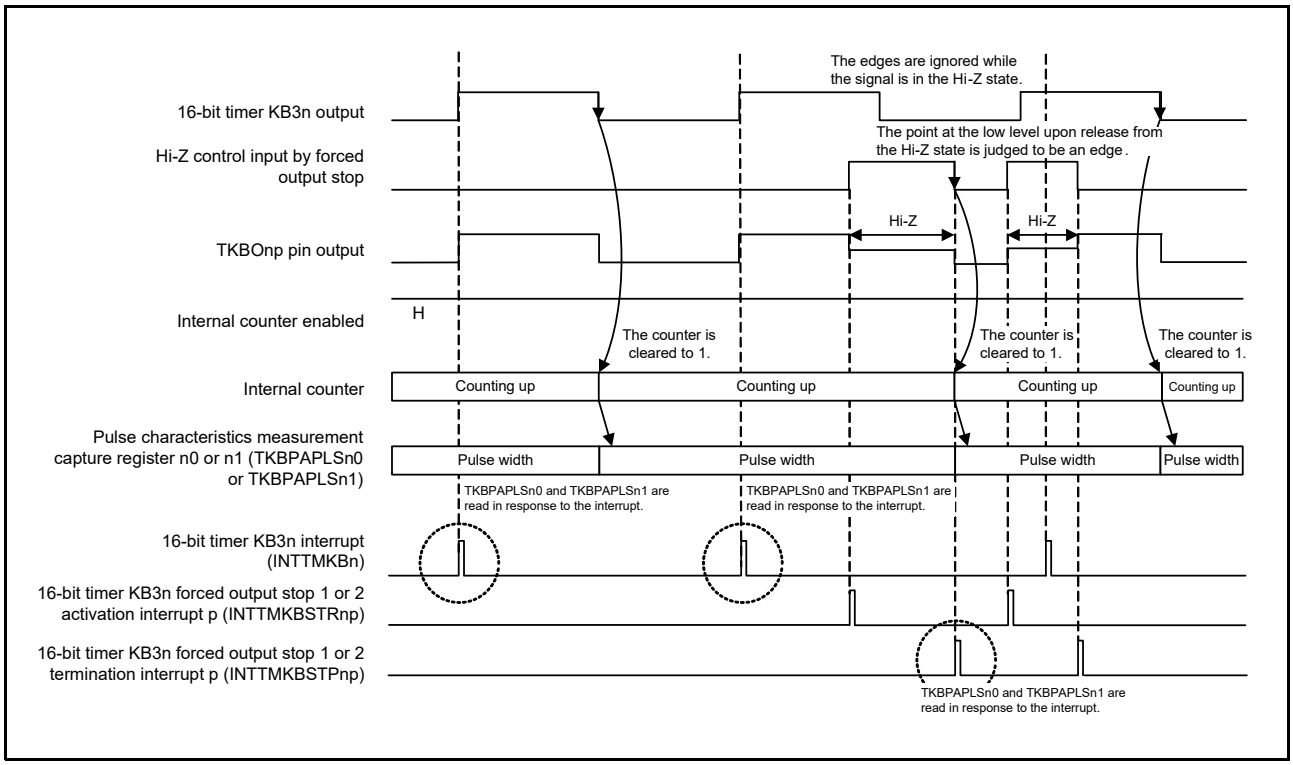
4. Measurement of pulse intervals between two consecutive falling edges (selected when the setting of the TKBPACTLVnp[1:0] bits is 11B)

   For measurement of a pulse interval between two consecutive falling edges, the internal 16-bit counter starts counting from 0001H in response to detecting a falling edge on the TKBOnp pin. Pulse characteristics measurement capture register n0 or n1 (TKBPAPLSn0 or TKBPAPLSn1) holds the value of the internal 16-bit counter at the time of detecting a falling edge on the TKBOnp pin while counting was in progress. **Figure 15 - 114** to **15 - 116** show examples of operation when forced output stop (Hi-Z, fixed high-level, or fixed low-level output) is applied.

   Figure 15 - 114 shows an example of operation when a TKBOnp pin is placed in the high-Z state in response to the activation of a forced output stop before or after the period between two consecutive falling edges of PWM output. **Figure 15 - 115** shows an example of operation when a TKBOnp output goes high in response to the activation of a forced output stop before or after the period between two consecutive falling edges of PWM output.

   - The pulse interval measured following detection of a falling edge that occurred in response to the termination of a forced output stop is stored in pulse characteristics measurement capture register n0 or n1 (TKBPAPLSn0 or TKBPAPLSn1) and read from the given register in response to the 16-bit timer KB3n forced output stop 1 or 2 termination interrupt p (INTTMKBSTPnp).
   - The pulse interval measured following detection of a falling edge of PWM output while forced output stop is inactive is stored in pulse characteristics measurement capture register n0 or n1 (TKBPAPLSn0 or TKBPAPLSn1) and read from the given register in response to the 16-bit timer KB3n interrupt (INTTMKBn).

Figure 15 - 114 Pulse Characteristics Measurement Function (Timing for Measurement of Pulse Intervals between Two Consecutive Falling Edges: Hi-Z Output during a Forced Output Stop, TKBIOCn0.TKBTOLnp = 0)



The timing with which TKBOnp output changes from the Hi-Z state to the low level in response to the termination of a forced output stop is treated as detection of a falling edge.

Figure 15 - 115 Pulse Characteristics Measurement Function (Timing for Measurement of Pulse Intervals between Two Consecutive Falling Edges: Fixed High-level Output during a Forced Output Stop, TKBIOCn0.TKBTOLnp = 0)

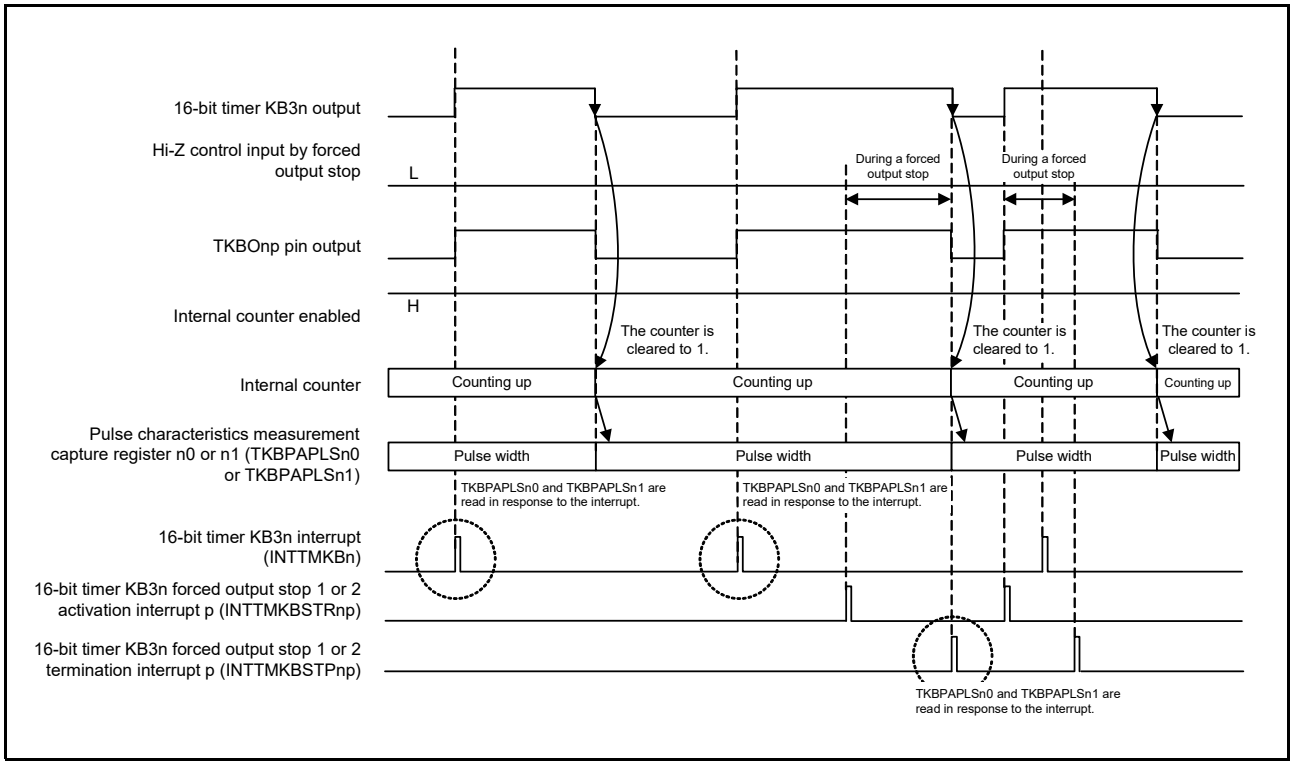
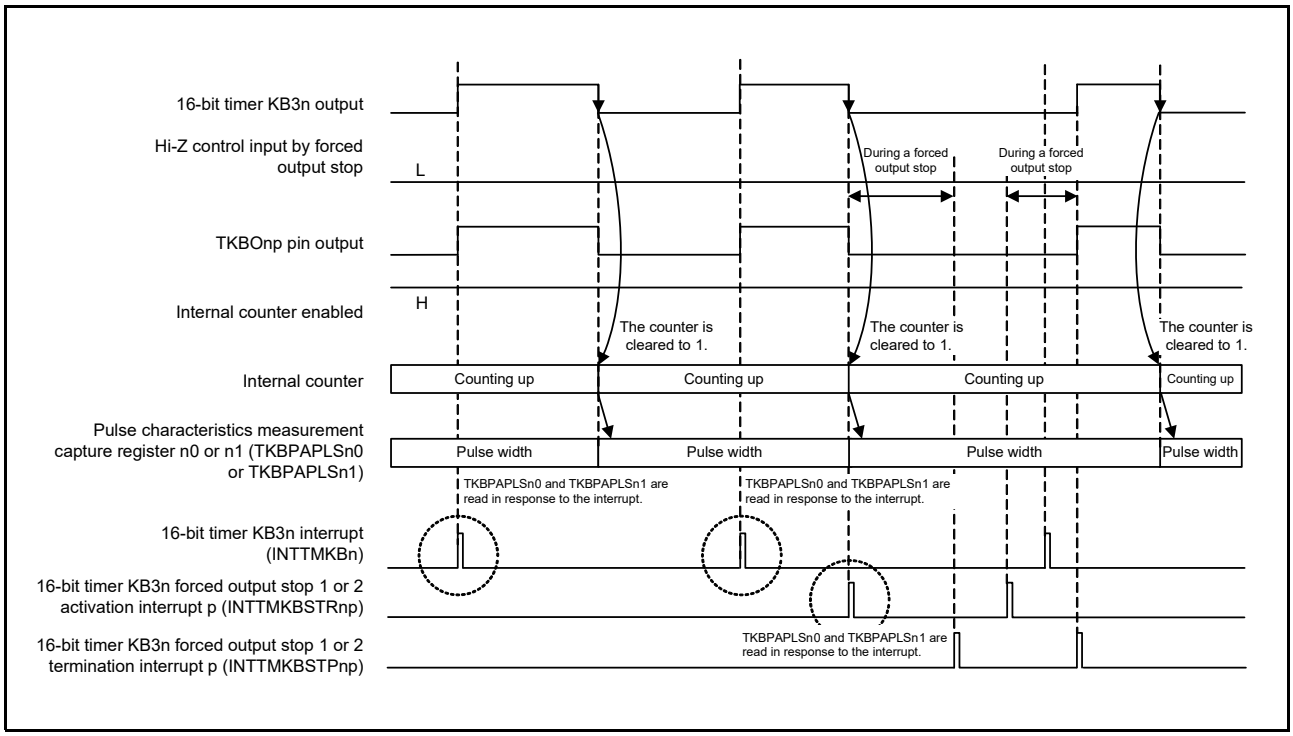


Figure 15 - 116 shows an example of operation when a TKBOnp output goes low in response to the activation of a forced output stop before or after the period between two consecutive falling edges of PWM output.

- The pulse interval measured following detection of a falling edge that occurred in response to the activation of a forced output stop is stored in pulse characteristics measurement capture register n0 or n1 (TKBPAPLSn0 or TKBPAPLSn1) and read from the given register in response to the 16-bit timer KB3n forced output stop 1 or 2 activation interrupt p (INTTMKBSTRnp).
- The pulse interval measured following detection of a falling edge of PWM output while forced output stop is inactive is stored in pulse characteristics measurement capture register n0 or n1 (TKBPAPLSn0 or TKBPAPLSn1) and read from the given register in response to the 16-bit timer KB3n interrupt (INTTMKBn).

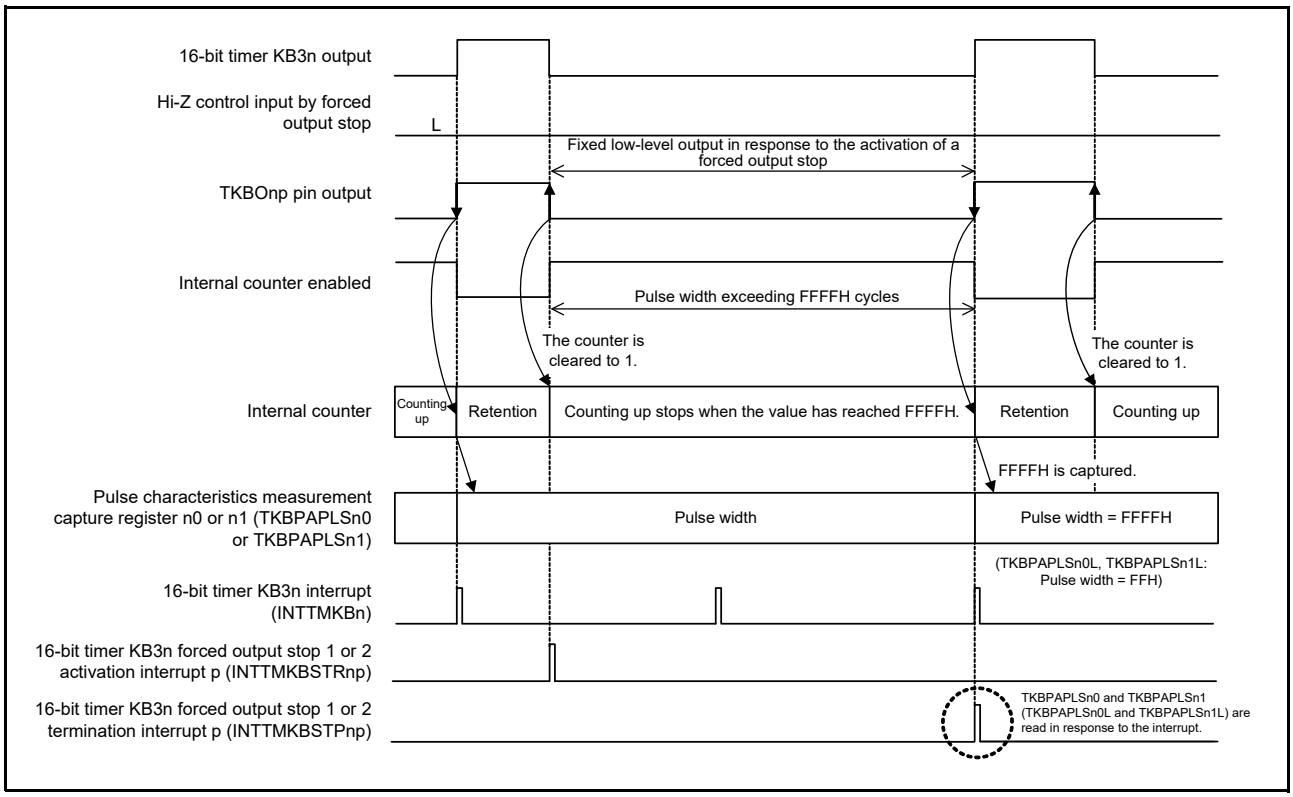
Figure 15 - 116 Pulse Characteristics Measurement Function (Timing for Measurement of Pulse Intervals between Two Consecutive Falling Edges: Fixed Low-level Output during a Forced Output Stop, TKBIOCn0.TKBTOLnp = 0)



15.9.3 Overflow at the time of measurement of a pulse characteristic

When measured results for pulse characteristics n0 and n1 exceed FFFFH cycles, FFFFH and FFH are stored in pulse characteristics measurement capture registers n0 and n1 (TKBPAPLSn0 and TKBPAPLSn1) and pulse characteristics measurement capture registers n0L and n1L (TKBPAPLSn0L and TKBPAPLSn1L), respectively. **Figure 15 - 117** shows an example of operation when the PWM output is at the low level for over FFFFH cycles of counting in measuring width at low level.

Figure 15 - 117 Example of Operation When the PWM Output Is at the Low Level for Over FFFFH Cycles of Counting in Measuring Width at Low Level



15.9.4 Changing the conditions of pulse characteristics measurement during operation

The settings of TKBPACTLn4 (n = 0 to 2) may be changed during the operation of a 16-bit timer KB3n. Since changing the settings means dynamically changing the conditions of pulse characteristics measurement, the values read from pulse characteristics measurement capture registers n0 and n1 should be treated as undefined for one or two periods of the signal being measured after the settings have been changed.

Figure 15 - 118 shows an example of operation in case of changing the target for measurement on the TKBOn0 pin from widths at high level to pulse intervals (between two consecutive falling edges) during operation of a 16-bit timer KB3n. **Figure 15 - 119** shows an example of operation in case of changing the pin on which measurement of widths at high level is to proceed from TKBOn0 to TKBOn1 during operation of a 16-bit timer KB3n.

Figure 15 - 118 Example of Operation in Case of Changing the Target for Measurement on the TKBOn0 Pin from Widths at High Level to Pulse Intervals (between Two Consecutive Falling Edges) during Operation of a 16-bit Timer KB3n

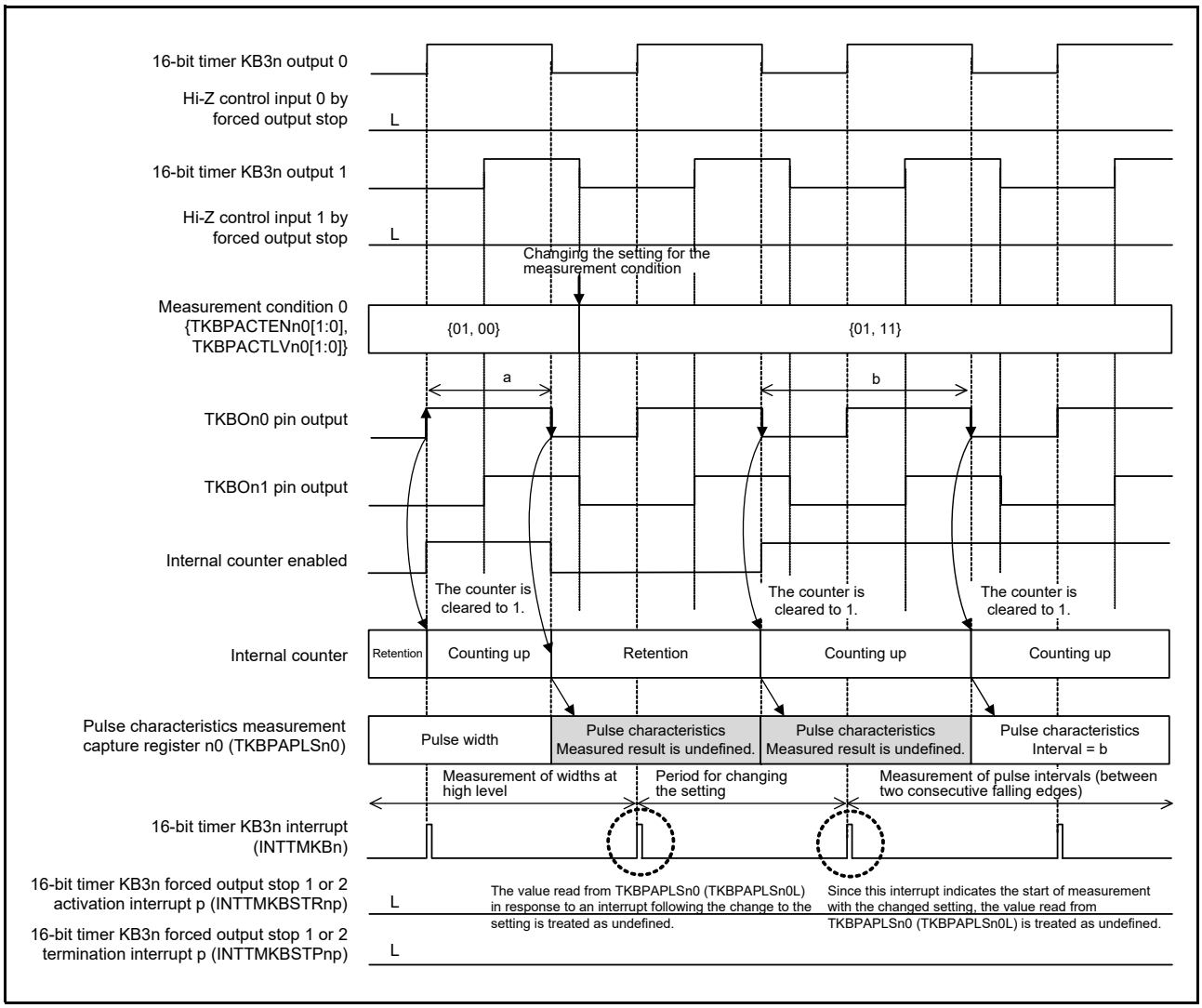
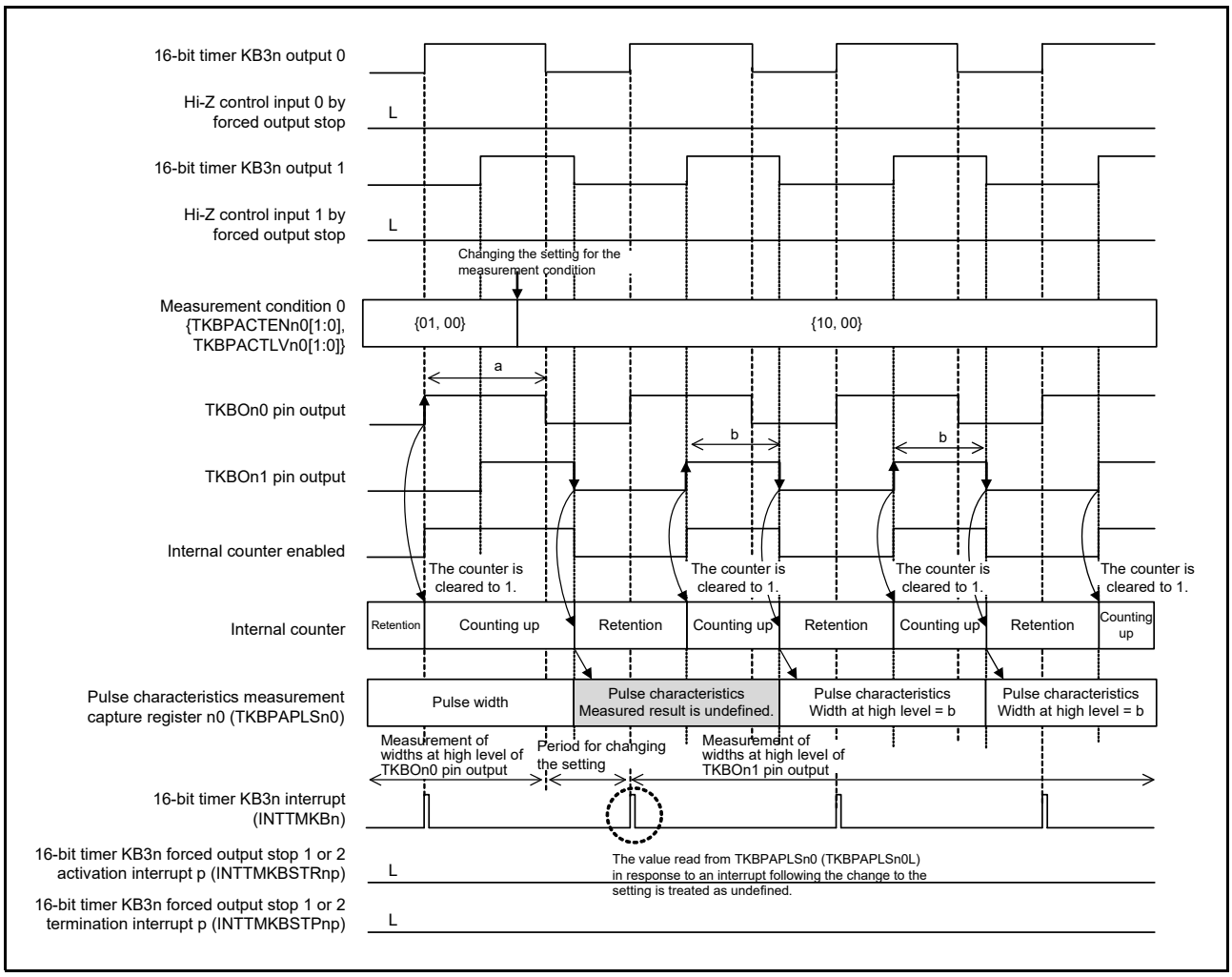


Figure 15 - 119 Example of Operation in Case of Changing the Pin on Which Measurement of Widths at High Level Is to Proceed from TKBOn0 to TKBOn1 during Operation of a 16-bit Timer KB3n



15.10 Notes on Using Interlocking of External Interrupts INTPm with 16-bit Timers KB30, KB31, and KB32

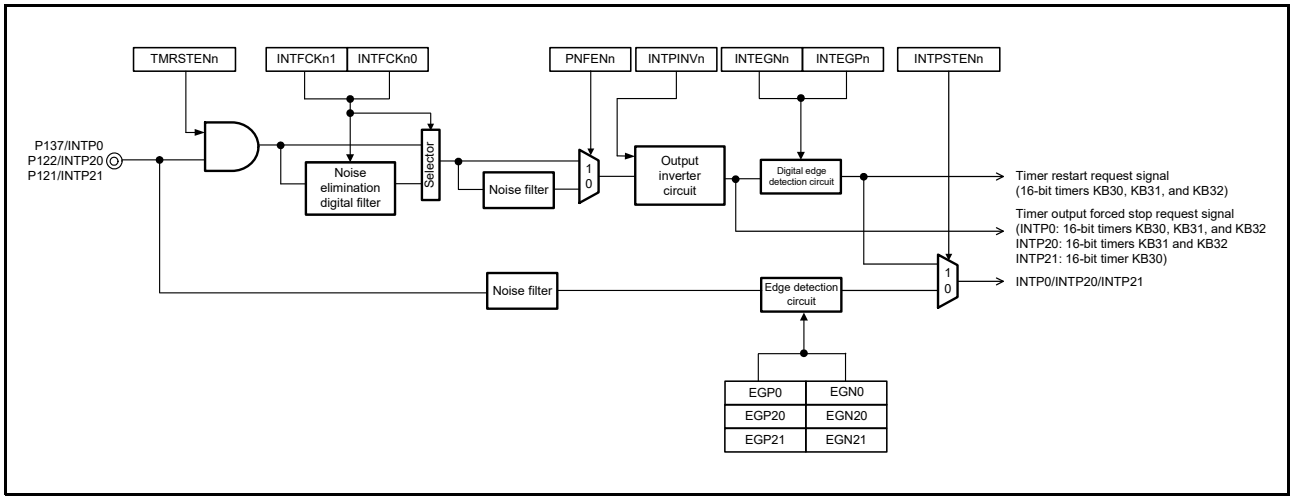
The restart signals for 16-bit timers KB30, KB31, and KB32 as well as the input signal edges detected on the INTPm pins can be selected as triggers for the INTP0, INTP20, and INTP21 interrupts. Thus, the INTPm interrupts can be generated in synchronization with generation of the restart signals for 16-bit timers KB30, KB31, and KB32.

According to the function to be used, set the following registers.

- External interrupt control register n (INTPCTLn)
- External interrupt rising edge enable registers 0 and 1 (EGP0, EGP1)
- External interrupt falling edge enable registers 0 and 1 (EGN0, EGN1)

Note that the width of the active signal required to make a given function operate differs between functions. Set the registers with reference to **Table 15 - 18** and **Figure 15 - 121** and configure external circuits so that the required active signal width is ensured.

Figure 15 - 120 Block Diagram of Hardware for External Interrupts INTP0, INTP20, and INTP21



Remark 1. EGP0 bit: Bit in the external interrupt rising edge enable register (EGP0)
EGP20 and EGP21 bits: Bits in the external interrupt rising edge enable register (EGP1)
EGN0 bit: Bit in the external interrupt falling edge enable register (EGN0)
EGN20 and EGN21 bits: Bits in the external interrupt falling edge enable register (EGN1)

Remark 2. n = 0 to 2

Table 15 - 18 Relationship between INTPm Functions, Register Settings, and Active Signal Width

	INTPm Function	Interlocked Function of Timer KB	Setting of Bits in the External Interrupt Control Register (INTPCTLn)	Setting of Bits in the Registers for Setting the Edge	Active Signal Width Required for Operation of the Function		
					Interrupt	Forced Output Stop of 16-bit Timers KB30, KB31, and KB32	Restart of 16-bit Timers KB30, KB31, and KB32
Interlocking is not in use	Interrupt due to input through the INTPx pin (can release the LSI from STOP mode)	—	INTPSTENn = 0	EGPn, EGNn	Up to 1 μs	—	—
Interlocking is in use	Interrupt due to generation of the timer restart request signal (cannot release the LSI from STOP mode)	Timer output forced stop	TMRSTENn = 1 INTPSTENn = 1	— **Note 1**	55 to 245 ns **Note 2** 2 to 3 clock cycles **Note 3**	55 to 245 ns **Notes 2, 4**	—
		Timer restart		INTEGPn, INTEGNn	55 to 245 ns **Note 2** 2 to 3 clock cycles **Note 3**	—	55 to 245 ns **Note 2** 2 to 3 clock cycles **Notes 3, 5**

Note 1. Forced output stop signals 1 and 2 are active high. The edge selection is valid only for an interrupt.

Note 2. The active signal width is 5 to 45 ns when the noise filter is disabled (PNFENn = 1).

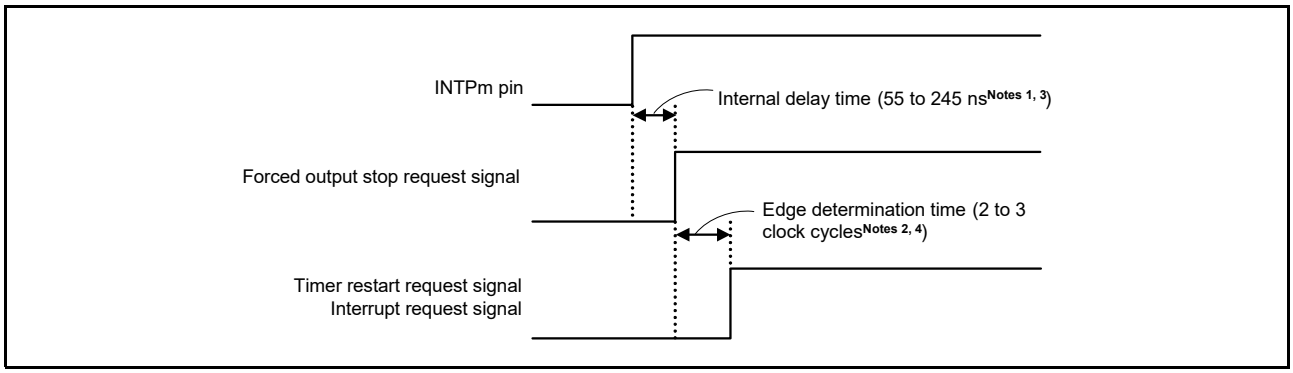
Note 3. The clock is f_{CLK} or f_{PLL} when bit 0 (DSCON) of the PLL control register (DSCCTL) is 1.

Note 4. After forced output stop functions 1 and 2 have been triggered, it takes a delay time of 10 to 40 ns until the state of the output pin changes.

Note 5. After the timer restart request signal has been received, it takes one clock cycle until the timer restart function operates and it takes an additional delay time of 10 to 40 ns until the state of the output pin changes.

Remark m = 0, 20, 21; n = 0 to 2

Figure 15 - 121 Timing of the Forced Output Stop Request Signal and Timer Restart Request Signal Generated from INTP0, INTP20, and INTP21



Note 1. The internal delay time is 5 to 45 ns when the noise filter is disabled (PNFENn = 1).

Note 2. The clock is f_{CLK} or f_{PLL} when bit 0 (DSCON) of the PLL control register (DSCCTL) is 1.

Note 3. After forced output stop functions 1 and 2 have been triggered, it takes a delay time of 10 to 40 ns until the state of the output pin changes.

Note 4. After the timer restart request signal has been received, it takes one clock cycle until the timer restart function operates and it takes an additional delay time of 10 to 40 ns until the state of the output pin changes.

Remark m = 0, 20, 21; n = 0 to 2

For interlocking of the comparator with 16-bit timers KB30, KB31, and KB32, refer to **22.4.6 Notes on using interlocking of the comparators with the 16-bit timers KB30, KB31, and KB32**.

Section 16 Realtime Clock (RTC)

16.1 Functions of Realtime Clock

The realtime clock has the following features.

- Capable of counting years, months, days of the week, dates, hours, minutes, and seconds, for up to 99 years
- Fixed-cycle interrupt (with period selectable from among 0.5 of a second, 1 second, 1 minute, 1 hour, 1 day, or 1 month)
- Alarm interrupt (alarm set by day of week, hour, and minute)
- Pin output function of 1 Hz (This is applicable to the 30-, 32-, 40-, 44-, 48-, 52-, and 64-pin products.)

The realtime clock interrupt signal (INTRTC) can be used to wake up the MCU from the STOP mode, or to trigger transitions of the A/D converter to the SNOOZE mode.

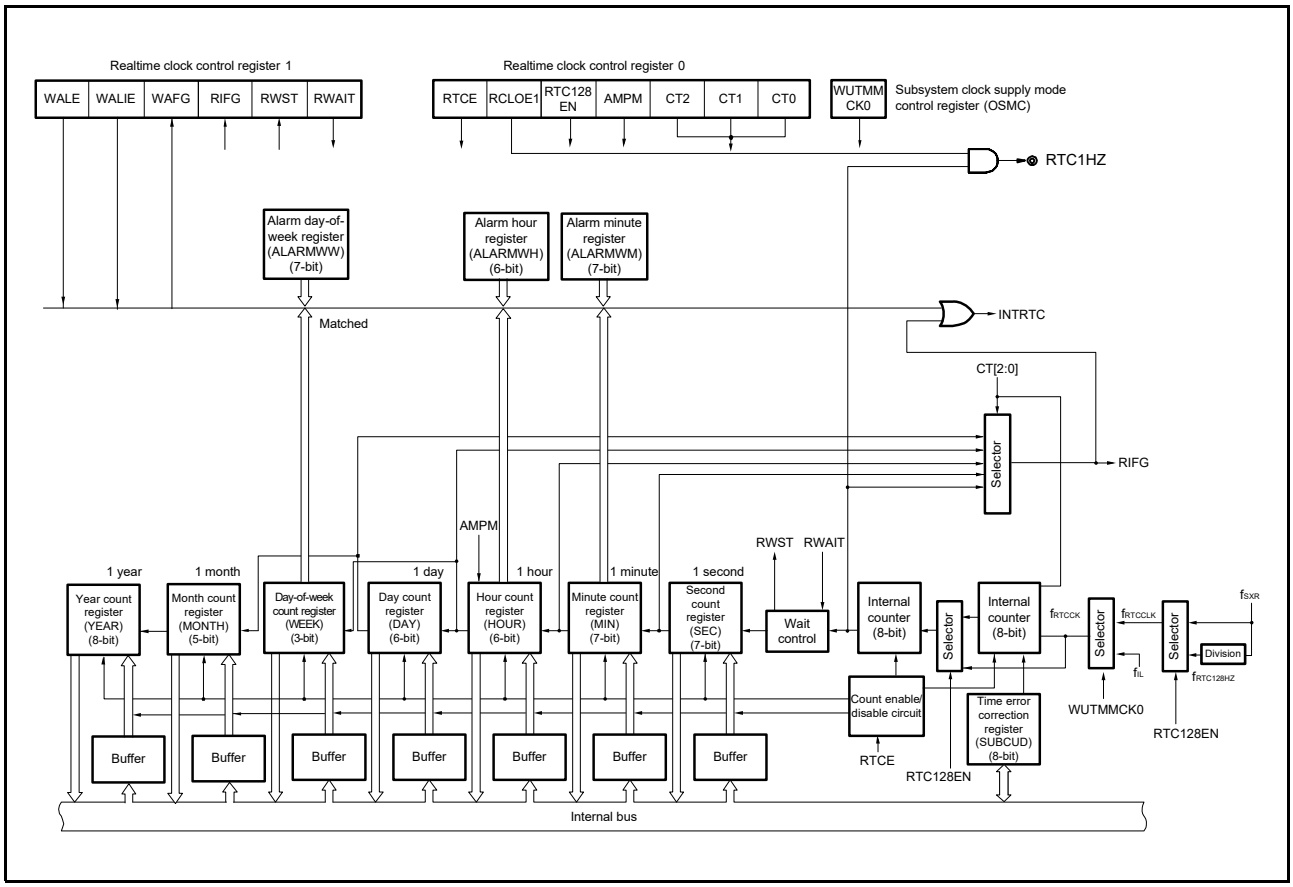
16.2 Configuration of the Realtime Clock

The realtime clock includes the following hardware blocks.

Table 16 - 1 Configuration of the Realtime Clock

Item	Configuration
Counter	Internal counter (16 bits)
Control registers	• Peripheral enable register 0 (PER0) • Subsystem clock supply mode control register (OSMC) • Realtime clock control register 0 (RTCC0) • Realtime clock control register 1 (RTCC1) • Second count register (SEC) • Minute count register (MIN) • Hour count register (HOUR) • Day count register (DAY) • Day-of-week count register (WEEK) • Month count register (MONTH) • Year count register (YEAR) • Time error correction register (SUBCUD) • Alarm minute register (ALARMWM) • Alarm hour register (ALARMWH) • Alarm day-of-week register (ALARMWW)

Figure 16 - 1 Block Diagram of the Realtime Clock



Caution The count of years, months, weeks, days, hours, minutes, and seconds can only proceed when a subsystem clock (f_{SXR} = 32.768 kHz) is selected as the operating clock of the realtime clock. When the low-speed on-chip oscillator clock (f_{IL} = 32.768 kHz) is selected, only the fixed-cycle interrupt is available.

16.3 Registers to Control the Realtime Clock

The following registers are used to control the realtime clock.

- Peripheral enable register 0 (PER0)
- Subsystem clock supply mode control register (OSMC)
- Realtime clock control register 0 (RTCC0)
- Realtime clock control register 1 (RTCC1)
- Second count register (SEC)
- Minute count register (MIN)
- Hour count register (HOUR)
- Day count register (DAY)
- Day-of-week count register (WEEK)
- Month count register (MONTH)
- Year count register (YEAR)
- Time error correction register (SUBCUD)
- Alarm minute register (ALARMWM)
- Alarm hour register (ALARMWH)
- Alarm day-of-week register (ALARMWW)
- Port mode registers xx (PMxx) (xx = 3)
- Port registers xx (Pxx) (xx = 3)

The following shows the register states depending on reset sources.

Reset Source	System-related Registers[Note 1]	Calendar-related Registers[Note 2]
POR	Reset	Not reset
External reset	Retained	Retained
WDT	Retained	Retained
TRAP	Retained	Retained
LVD	Retained	Retained
Other internal reset sources	Retained	Retained

Note 1. RTCC0, RTCC1, and SUBCUD

Note 2. SEC, MIN, HOUR, DAY, WEEK, MONTH, YEAR, ALARMWM, ALARMWH, and ALARMWW

Assertion of the reset signal does not reset the SEC, MIN, HOUR, DAY, WEEK, MONTH, YEAR, ALARMWM, ALARMWH, or ALARMWW register. Initialize all the registers after power on.

16.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. If the realtime clock is to be used, be sure to set bit 7 (RTCWEN) of this register to 1. The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 16 - 2 Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

RTCWEN	Control of access to the realtime clock (RTC)
0	• The SFRs used by the realtime clock (RTC) cannot be written. • The realtime clock (RTC) can operate.
1	• The SFRs used by the realtime clock (RTC) can be read and written. • The realtime clock (RTC) can operate.

Caution 1. When the realtime clock is to be used, start by setting the RTCWEN bit to 1 and then set the following registers once oscillation of the counter clock (f_{RTCCK}) has become stable. If RTCWEN is 0, attempted writing to the control registers of the realtime clock is ignored, and 00H is read (except for the subsystem clock supply mode control register (OSMC), port mode register 3 (PM3), and port register 3 (P3)).

- **Realtime clock control register 0 (RTCC0)**
- **Realtime clock control register 1 (RTCC1)**
- **Second count register (SEC)**
- **Minute count register (MIN)**
- **Hour count register (HOUR)**
- **Day count register (DAY)**
- **Day-of-week count register (WEEK)**
- **Month count register (MONTH)**
- **Year count register (YEAR)**
- **Time error correction register (SUBCUD)**
- **Alarm minute register (ALARMWM)**
- **Alarm hour register (ALARMWH)**
- **Alarm day-of-week register (ALARMWW)**

Caution 2. The subsystem clock supply to peripheral functions other than the realtime clock can be stopped in STOP mode or HALT mode when the subsystem clock is in use, by setting the RTCLPC bit of the subsystem clock supply mode control register (OSMC) to 1.

Caution 3. Be sure to set bits 6 and 1 to 0.

16.3.2 Subsystem clock supply mode control register (OSMC)

The control clock (f_{RTCCK}) for the realtime clock can be selected by the WUTMMCK0 bit of this register. The RTCLPC bit of this register can be used to reduce the power consumption by disabling supply of the clock signal to peripheral functions that are not in use. For details about setting the RTCLPC bit, see **Section 9 Clock Generator**. The OSMC register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is undefined.

Figure 16 - 3 Format of Subsystem Clock Supply Mode Control Register (OSMC)

Address: F00F3H
After reset: Undefined**Note 1**
R/W: R/W**Note 2**

Symbol	<7>	6	5	<4>	3	2	1	<0>
OSMC	RTCLPC	0	0	WUTMMCK0	×	×	0	HIPREC

WUTMMCK0	Selection of the operating clock (f_{RTCCK}) for the control block of the realtime clock
0	Subsystem clock XR (f_{SXR}) or $f_{RTC128HZ}$ ($f_{RTC128HZ}$ is selected when the setting of the RTC128EN bit is 1.)
1	Low-speed on-chip oscillator clock (f_{IL})**Notes 3, 4**

Note 1. The RTCLPC and WUTMMCK bits have the value 0 and the HIPREC bit has the value 1 following a reset.

Note 2. Bits 3, 2, and 0 are read-only. Writing to these bits is ignored.

Note 3. Setting the WUTMMCK0 bit to 1 is prohibited when the subsystem clock X is selected as the operating clock for the control block of the realtime clock.

Note 4. Switching between the subsystem clock and the low-speed on-chip oscillator clock by using the WUTMMCK0 bit is only possible while operations of the realtime clock, 32-bit interval timer, clock output/buzzer output controller, and timer RJ are all stopped.

Caution 1. Counting of years, months, weeks, days, hours, minutes, and seconds can only proceed when the subsystem clock XR (f_{SXR} = 32.768 kHz) or $f_{RTC128HZ}$ is selected as the operating clock for the control block of the realtime clock. When the low-speed on-chip oscillator clock (f_{IL} = 32.768 kHz) is selected, only the fixed-cycle interrupt is available.

Caution 2. Be sure to set bits 6, 5, and 1 to 0.

Remark ×: Don't care

16.3.3 Realtime clock control register 0 (RTCC0)

This is an 8-bit register that is used to start or stop the realtime clock operation, control the RTC1HZ pin, and set a 12- or 24-hour system and the fixed-cycle interrupt. The RTCC0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register is 00H following an internal reset by the power-on reset circuit.

Figure 16 - 4 Format of Realtime Clock Control Register 0 (RTCC0) (1/2)

Address: F022BH
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	3	2	1	0
RTCC0	RTCE	0	RCLOE1	RTC128EN	AMPM	CT2	CT1	CT0

RTCE**Note 1**	Realtime clock operation control
0	Stops counter operation.
1	Starts counter operation.

RCLOE1 **Note 2**	RTC1HZ pin output control
0	Disables output of the RTC1HZ pin (1 Hz).
1	Enables output of the RTC1HZ pin (1 Hz).

RTC128EN	Selection of the operating clock for the realtime clock (f_{RTCCLK})
0	32.768 kHz
1	128 Hz
• Setting this bit to 1 enables the realtime clock to operate with the 128-Hz clock for lower-power operation. • Time error correction cannot be used when the setting of this bit is 1. • The WUTMMCK bit in the OSMC register should be set to 0 when setting this bit to 1.	

AMPM	Selection of 12-/24-hour system
0	12-hour system (a.m. and p.m. are displayed.)
1	24-hour system
• Rewrite the AMPM bit value after setting the RWAIT bit (bit 0 of the realtime clock control register 1 (RTCC1)) to 1. If the AMPM bit value is changed, the values of the hour count register (HOUR) change according to the specified time system. • **Table 16 - 2** shows the time (hour) digits indicated according to the setting of this bit.	

Figure 16 - 4 Format of Realtime Clock Control Register 0 (RTCC0) (2/2)

CT2	CT1	CT0	Fixed-cycle interrupt (INTRTC) selection
0	0	0	Does not use fixed-cycle interrupt.
0	0	1	Once per 0.5 s (synchronized with second count up)
0	1	0	Once per 1 s (same time as second count up)
0	1	1	Once per 1 m (second 00 of every minute)
1	0	0	Once per 1 hour (minute 00 and second 00 of every hour)
1	0	1	Once per 1 day (hour 00, minute 00, and second 00 of every day)
1	1	×	Once per 1 month (Day 1, hour 00 a.m., minute 00, and second 00 of every month)
To change the values of the CT[2:0] bits while counting is in progress (RTCE = 1), rewrite the values of the CT[2:0] bits after disabling interrupt processing of INTRTC by using the interrupt mask flag register. Furthermore, after rewriting the values of the CT[2:0] bits, enable interrupt processing after clearing the RIFG and RTCIF flags.			

Note 1. To shift to the STOP mode immediately after setting the RTCE bit to 1, follow the procedure in **Figure 16 - 18 Procedure for Shifting to HALT or STOP Mode after Setting RTCE Bit to 1**.

Note 2. In the 20-, 24-, and 25-pin products, be sure to set this bit to 0.

Caution 1. Do not change the value of the RCLOE1 bit when RTCE is 1.

Caution 2. 1 Hz is not output even if RCLOE1 is set to 1 when RTCE is 0.

Caution 3. Be sure to set bit 6 to 0.

Remark ×: Don't care

16.3.4 Realtime clock control register 1 (RTCC1)

This is an 8-bit register that is used to control the alarm interrupt and the wait time of the counter. The RTCC1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register is 00H following an internal reset by the power-on reset circuit.

Figure 16 - 5 Format of Realtime Clock Control Register 1 (RTCC1) (1/2)

Address: F022CH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	5	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	0	WAFG	RIFG	0	RWST	RWAIT

WALE	Alarm operation control
0	Match operation is invalid.
1	Match operation is valid.**Note 1**
When setting a value to the WALE bit while counting is in progress (RTCE = 1) and WALIE is 1, rewrite the WALE bit after disabling interrupt processing of INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG and RTCIF flags after rewriting the WALE bit. When setting any of the alarm-related registers (WALIE flag of realtime clock control register 1 (RTCC1), the alarm minute register (ALARMWM), the alarm hour register (ALARMWH), and the alarm day-of-week register (ALARMWW)), set the WALE bit to 0.	

WALIE	Control of alarm interrupt (INTRTC)
0	Does not generate interrupt on matching of alarm.
1	Generates interrupt on matching of alarm.**Note 1**

WAFG	Alarm detection status flag
0	Alarm mismatch
1	Detection of matching of alarm
This is a status flag that indicates detection of matching with the alarm. It is only valid when WALE is 1 and is set to 1 one cycle of f_{RTCCK} after matching of the alarm is detected. This flag is cleared when 0 is written to it. Writing 1 to it is invalid.	

RIFG	Fixed-cycle interrupt status flag
0	Fixed-cycle interrupt is not generated.
1	Fixed-cycle interrupt is generated.
This flag indicates the state of generation of the fixed-cycle interrupt. When the fixed-cycle interrupt is generated, it is set to 1. This flag is cleared when 0 is written to it. Writing 1 to it is invalid.	

Figure 16 - 5 Format of Realtime Clock Control Register 1 (RTCC1) (2/2)

RWST	Wait status flag of realtime clock[Note 2]
0	Counting is in progress.
1	Counter values are readable and writable.
This status flag indicates whether the setting of the RWAIT bit is valid. Before reading or writing the counter value, confirm that the value of this flag is 1.	

RWAIT	Wait control of realtime clock[Note 1]
0	Counting proceeds.
1	Stops the SEC to YEAR counters. Counter values are readable and writable.
This bit controls the operation of the counter. Be sure to write 1 to this bit to read or write the counter value. So that the 16-bit internal counter continues to run, return the value of this bit to 0 on completion of reading or writing within one second. After setting this bit to 1, it takes up to one cycle of f_{RTCCK} until the counter value can be actually read or written (RWST = 1).[Notes 3, 4] When the internal counter (16 bits) overflows while the setting of this bit is 1, an indicator of the counter having overflowed is retained after RWAIT has become 0, after which counting up continues. Note that, when the second count register has been written to, the overflow is not retained.	

Note 1. When the detection of matching for an alarm or the alarm interrupt is to be used, set the fixed-cycle interrupt to “once per second”, and, within 1 second of the generation of the INTRTC interrupt, set the RWAIT bit to 1 and read or write counter values. If the RWAIT bit is set to 1 and counter values are read or written with any given timing, matching for an alarm may not occur and the interrupt request may also not be generated. For details on the procedures for reading and writing counter values, see **16.4.3 Reading from and writing to the counters of the realtime clock**.

Note 2. Bit 1 is read-only.

Note 3. When the RWAIT bit is set to 1 within one cycle of f_{RTCCK} clock after setting the RTCE bit to 1, the setting of the RWST bit actually becoming 1 may take up to two cycles of the operating clock (f_{RTCCK}).

Note 4. When the RWAIT bit is set to 1 within one cycle of f_{RTCCK} clock after release from the standby mode (HALT mode, STOP mode, or SNOOZE mode), the setting of the RWST bit actually becoming 1 may take up to two cycles of the operating clock (f_{RTCCK}).

Caution **Note that using a bit manipulation instruction for writing to the RTCC1 register may lead to clearing of the RIFG and WAFG flags. Therefore, when writing to the RTCC1 register, be sure to use an 8-bit manipulation instruction. To prevent the RIFG and WAFG flags from being cleared during writing, disable writing by setting 1 to the corresponding bit. However, if the RIFG and WAFG flags are not in use and a change to the value does not matter, using a bit manipulation instruction for writing to the RTCC1 register does not create a problem.**

Remark 1. Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

Remark 2. The internal counter (16 bits) is cleared when the second count register (SEC) is written.

16.3.5 Second count register (SEC)

This is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of seconds. This counter is incremented each time the 16-bit internal counter overflows. When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTCCK} later. Set a decimal value of 00 to 59 to this register in BCD code. The SEC register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 16 - 6 Format of Second Count Register (SEC)

Address: F0220H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

Caution **When reading from or writing to this register while the counter is in operation (RTCE = 1), follow the procedures described in 16.4.3 Reading from and writing to the counters of the realtime clock.**

Remark The internal counter (16 bits) is cleared when the second count register (SEC) is written.

16.3.6 Minute count register (MIN)

This is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of minutes. This counter is incremented each time the second counter overflows. When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTCCK} later. Even if the second count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 59 to this register in BCD code. The MIN register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 16 - 7 Format of Minute Count Register (MIN)

Address: F0221H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

Caution **When reading from or writing to this register while the counter is in operation (RTCE = 1), follow the procedures described in 16.4.3 Reading from and writing to the counters of the realtime clock.**

16.3.7 Hour count register (HOUR)

This is an 8-bit register that takes a value of 00 to 23 or 01 to 12 and 21 to 32 (decimal) and indicates the count value of hours. This counter is incremented each time the minute counter overflows. When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTCCK} later. Even if the minute count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Specify a decimal value of 00 to 23, 01 to 12, or 21 to 32 by using BCD code according to the time system specified using bit 3 (AMPM) of the realtime clock control register 0 (RTCC0). If the AMPM bit value is changed, the values of the HOUR register change according to the specified time system. The HOUR register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 16 - 8 Format of Hour Count Register (HOUR)

Address: F0222H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

Caution 1. Bit 5 (HOUR20) of the HOUR register indicates AM (0) or PM (1) when AMPM = 0 (that is, when the 12-hour system is selected).

Caution 2. When reading from or writing to this register while the counter is in operation (RTCE = 1), follow the procedures described in 16.4.3 Reading from and writing to the counters of the realtime clock.

Table 16 - 2 shows the relationship between the setting value of the AMPM bit, the hour count register (HOUR) value, and time.

Table 16 - 2 Displayed Time Digits

24-Hour Display (RTCC0.AMPM = 1)		12-Hour Display (RTCC0.AMPM = 0)	
Time	HOUR Register	Time	HOUR Register
0	00H	12 a.m.	12H
1	01H	1 a.m.	01H
2	02H	2 a.m.	02H
3	03H	3 a.m.	03H
4	04H	4 a.m.	04H
5	05H	5 a.m.	05H
6	06H	6 a.m.	06H
7	07H	7 a.m.	07H
8	08H	8 a.m.	08H
9	09H	9 a.m.	09H
10	10H	10 a.m.	10H
11	11H	11 a.m.	11H
12	12H	12 p.m.	32H
13	13H	1 p.m.	21H
14	14H	2 p.m.	22H
15	15H	3 p.m.	23H
16	16H	4 p.m.	24H
17	17H	5 p.m.	25H
18	18H	6 p.m.	26H
19	19H	7 p.m.	27H
20	20H	8 p.m.	28H
21	21H	9 p.m.	29H
22	22H	10 p.m.	30H
23	23H	11 p.m.	31H

The HOUR register value is set to 12-hour display when the AMPM bit is 0 and to 24-hour display when the AMPM bit is 1. In 12-hour display, the fifth bit of the HOUR register displays 0 for AM and 1 for PM.

16.3.8 Day count register (DAY)

This is an 8-bit register that takes a value of 1 to 31 (decimal) and indicates the count value of days. This counter is incremented each time the hour counter overflows. Counting by the date counter proceeds as shown below.

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February, leap year)
- 01 to 28 (February, normal year)

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTCCK} later. Even if the hour count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 31 to this register in BCD code. The DAY register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 16 - 9 Format of Day Count Register (DAY)

Address: F0224H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

Caution When reading from or writing to this register while the counter is in operation (RTCE = 1), follow the procedures described in 16.4.3 Reading from and writing to the counters of the realtime clock.

16.3.9 Day-of-week count register (WEEK)

This is an 8-bit register that takes a value of 0 to 6 (decimal) and indicates the count value of days of the week. This counter is incremented in synchronization with the date counter. When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTCCK} later. Set a decimal value of 00 to 06 to this register in BCD code. The WEEK register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 16 - 10 Format of Day-of-week Count Register (WEEK)

Address: F0223H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

Caution 1. The value corresponding to the month count register (MONTH) or the day count register (DAY) is not stored in the day-of-week count register (WEEK) automatically. After reset release, set the day-of-week count register as follow.

Day	WEEK
Sunday	00H
Monday	01H
Tuesday	02H
Wednesday	03H
Thursday	04H
Friday	05H
Saturday	06H

Caution 2. When reading from or writing to this register while the counter is in operation (RTCE = 1), follow the procedures described in 16.4.3 Reading from and writing to the counters of the realtime clock.

16.3.10 Month count register (MONTH)

This is an 8-bit register that takes a value of 1 to 12 (decimal) and indicates the count value of months. This counter is incremented each time the day counter overflows. When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTCCK} later. Even if the day count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 12 to this register in BCD code. The MONTH register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 16 - 11 Format of Month Count Register (MONTH)

Address: F0225H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

Caution When reading from or writing to this register while the counter is in operation (RTCE = 1), follow the procedures described in 16.4.3 Reading from and writing to the counters of the realtime clock.

16.3.11 Year count register (YEAR)

This is an 8-bit register that takes a value of 0 to 99 (decimal) and indicates the value of the counter of years. This counter is incremented each time the month count register (MONTH) overflows. Values 00, 04, 08, …, 92, and 96 indicate a leap year. When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTCCK} later. Even if the MONTH register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 99 to this register in BCD code. The YEAR register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 16 - 12 Format of Year Count Register (YEAR)

Address: F0226H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

Caution When reading from or writing to this register while the counter is in operation (RTCE = 1), follow the procedures described in 16.4.3 Reading from and writing to the counters of the realtime clock.

16.3.12 Time error correction register (SUBCUD)

This register is used to correct the time with high accuracy when it is running slow or fast by adjusting the value that is considered an overflow from the internal counter (16 bits) to the second count register (SEC) (reference value: 7FFFH). The SUBCUD register can be set by an 8-bit memory manipulation instruction. The value of this register is 00H following an internal reset by the power-on reset circuit.

Figure 16 - 13 Format of Time Error Correction Register (SUBCUD)

Address: F0227H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
SUBCUD	DEV	F6	F5	F4	F3	F2	F1	F0

DEV	Setting of time error correction timing
0	Corrects time error when the second digits are at 00, 20, or 40 (every 20 seconds).
1	Corrects time error only when the second digits are at 00 (every 60 seconds).
Writing to the SUBCUD register at the following timing is prohibited. • When DEV = 0 is set: For a period of SEC = 00H, 20H, 40H • When DEV = 1 is set: For a period of SEC = 00H	

F6	Setting of time error correction value
0	Increases by {(F5, F4, F3, F2, F1, F0) – 1} × 2.
1	Decreases by {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2.
When (F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *), the time error is not corrected. * is 0 or 1. /F5 to /F0 are the inverted values of the corresponding bits (000011B when 111100B). Range of correction value: (when F6 = 0) 2, 4, 6, 8, … , 120, 122, 124 (when F6 = 1) –2, –4, –6, –8, … , –120, –122, –124	

The range of value that can be corrected by using the time error correction register (SUBCUD) is shown below.

	DEV = 0 (Correction Every 20 Seconds)	DEV = 1 (Correction Every 60 Seconds)
Correctable range	–189.2 ppm to 189.2 ppm	–63.1 ppm to 63.1 ppm
Maximum excludes quantization error	±1.53 ppm	±0.51 ppm
Minimum resolution	±3.05 ppm	±1.02 ppm

Caution **Time error correction cannot be used in the 128-Hz operating mode (RTC128EN = 1); it can only proceed if the setting of RTC128EN is 0.**

Remark If a correctable range is –63.1 ppm or lower and 63.1 ppm or higher, set DEV to 0.

16.3.13 Alarm minute register (ALARMWM)

This register is used to set minutes of alarm. The ALARMWM register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Caution Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 16 - 14 Format of Alarm Minute Register (ALARMWM)

Address: F0228H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

16.3.14 Alarm hour register (ALARMWH)

This register is used to set hours of alarm. The ALARMWH register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Caution Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 16 - 15 Format of Alarm Hour Register (ALARMWH)

Address: F0229H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

Caution Bit 5 (WH20) of the ALARMWH register indicates AM (0) or PM (1) when AMPM = 0 (that is, when the 12-hour system is selected).

16.3.15 Alarm day-of-week register (ALARMWW)

This register is used to set days of the week of alarm. The ALARMWW register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 16 - 16 Format of Alarm Day-of-Week Register (ALARMWW)

Address: F022AH
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

Here is an example of setting the alarm.

Time of Alarm	Day of Week							12-hour Display				24-hour Display			
	Sunday WW0	Monday WW1	Tuesday WW2	Wednesday WW3	Thursday WW4	Friday WW5	Saturday WW6	Hour 10	Hour 1	Minute 10	Minute 1	Hour 10	Hour 1	Minute 10	Minute 1
Every day, 0:00 a.m.	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
Every day, 1:30 a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
Every day, 11:59 a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
Monday through Friday, 0:00 p.m.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
Sunday, 1:30 p.m.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
Monday, Wednesday, Friday, 11:59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

16.3.16 Registers for controlling the port functions multiplexed with the realtime clock output

Set the following registers to control the port functions multiplexed with the realtime clock output.

- Port mode registers xx (PMxx)
- Port registers xx (Pxx)

For details, see **7.3.1 Port mode registers xx (PMxx)** and **7.3.2 Port registers xx (Pxx)**.

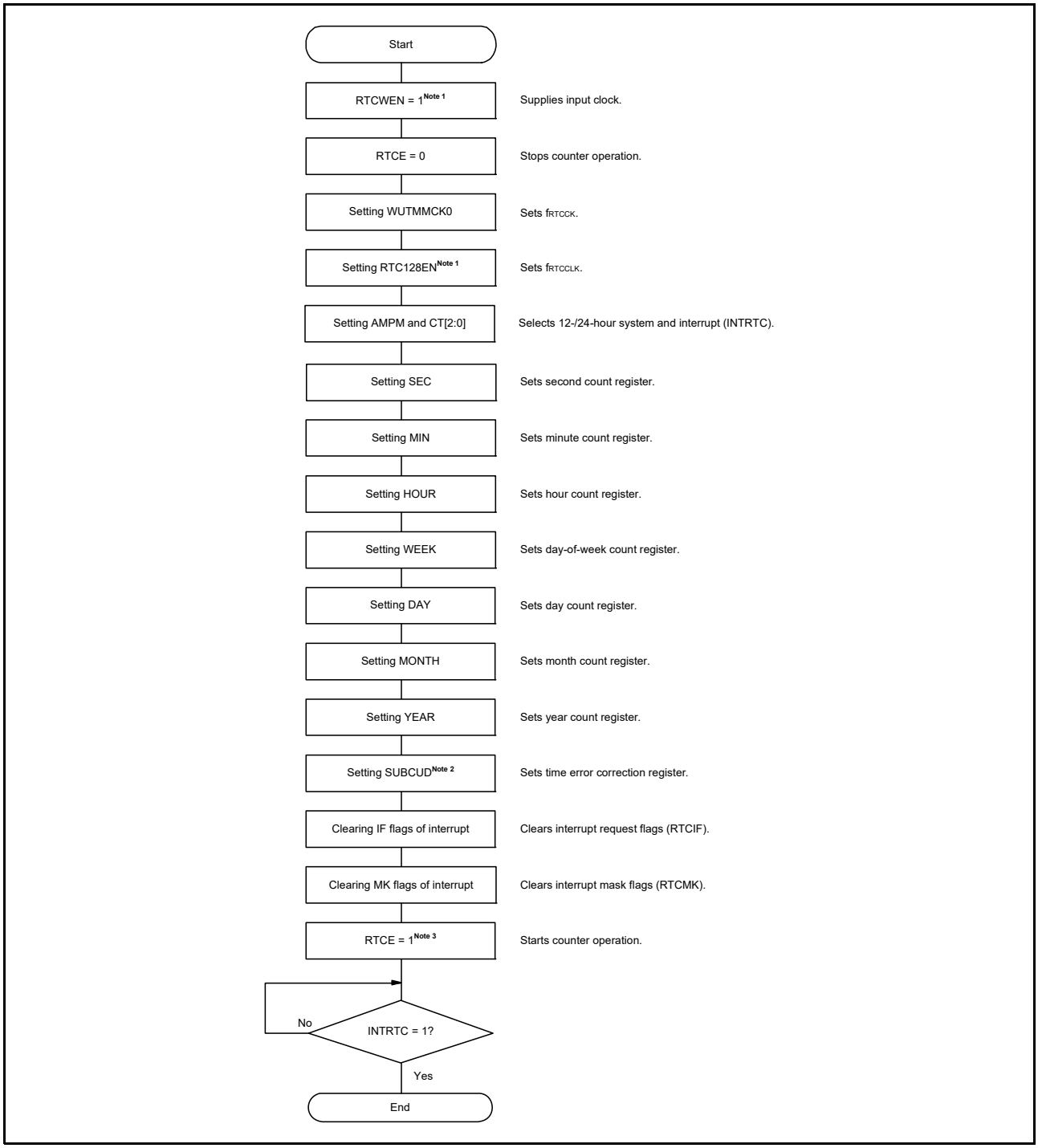
When the pin multiplexed with RTC1HZ is to be used for 1-Hz output, set the PM30 and P30 bits to 0.

Remark xx = 3

16.4 Operations of the Realtime Clock

16.4.1 Starting the realtime clock operation

Figure 16 - 17 Procedure for Starting the Realtime Clock Operation



Note 1. First set the RTCWEN bit to 1 and set the RTC128EN bit as desired, while oscillation of the count clock (f_{RTCCK}) is stable.

Note 2. Set up the SUBCUD register only if the time error must be corrected. For details about how to calculate the correction value, see **16.4.6 Example of time error correction by the realtime clock**.
Time error correction cannot be used while the setting of the RTC128EN bit is 1.

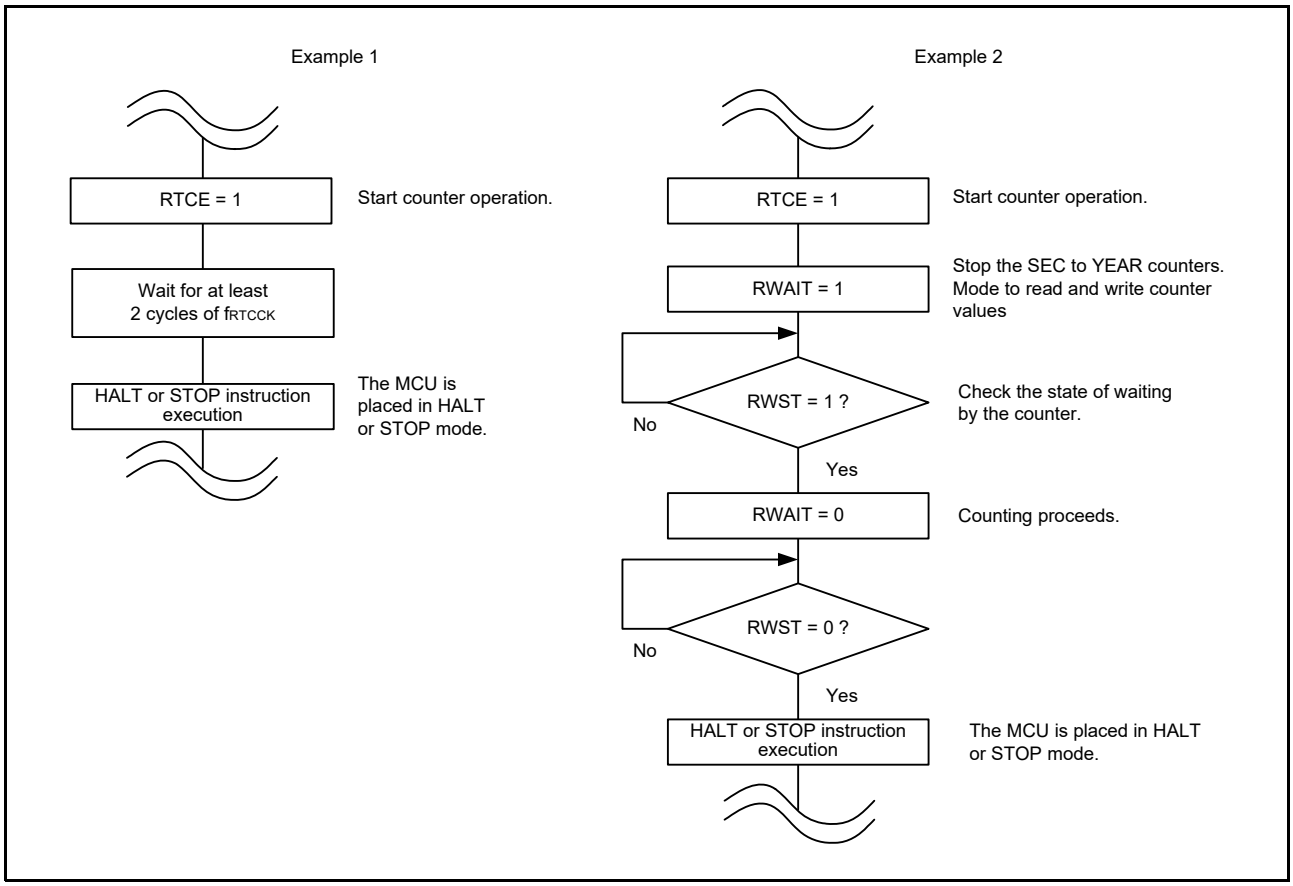
Note 3. Confirm the procedure described in **16.4.2 Shifting to HALT or STOP mode after starting operation** when shifting to HALT or STOP mode without waiting for INTRTC = 1 after RTCE = 1.

16.4.2 Shifting to HALT or STOP mode after starting operation

Take either of the steps listed below when shifting to HALT or STOP mode immediately after setting the RTCE bit to 1. Note that any of these steps is not required when shifting to the HALT or STOP mode after the INTRTC interrupt has occurred.

- Make a transition to HALT or STOP mode when at least two counter clock cycles (f_{RTCCK}) have elapsed after setting the RTCE bit to 1 (see **Example 1** in **Figure 16 - 18 Procedure for Shifting to HALT or STOP Mode after Setting RTCE Bit to 1**).
- After setting the RTCE bit to 1 and then setting the RWAIT bit to 1, poll the RWST bit to check if it has become 1 yet. After setting the RWAIT bit to 0 and polling the RWST bit to check if it has become 0 yet, a transition to HALT/STOP mode will proceed (see **Example 2** in **Figure 16 - 18 Procedure for Shifting to HALT or STOP Mode after Setting RTCE Bit to 1**).

Figure 16 - 18 Procedure for Shifting to HALT or STOP Mode after Setting RTCE Bit to 1



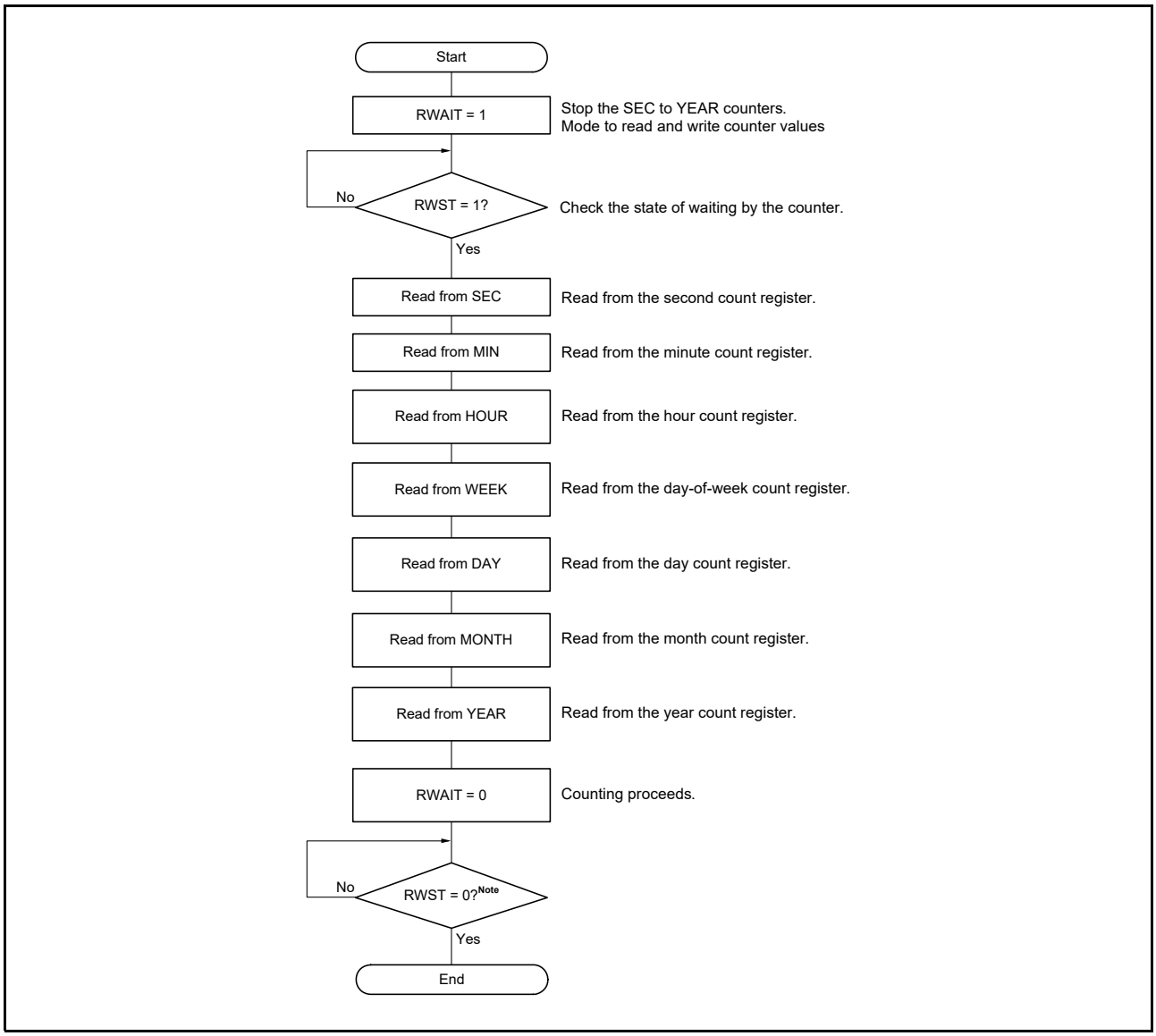
16.4.3 Reading from and writing to the counters of the realtime clock

Read from or write to the counter after setting RWAIT to 1 first.

Set RWAIT to 0 after completion of reading from or writing to the counter.

When the alarm interrupt is in use, read from or write to the counters according to the procedures shown in **Figures 16 - 20** and **16 - 22**.

Figure 16 - 19 Procedure for Reading from the Realtime Clock

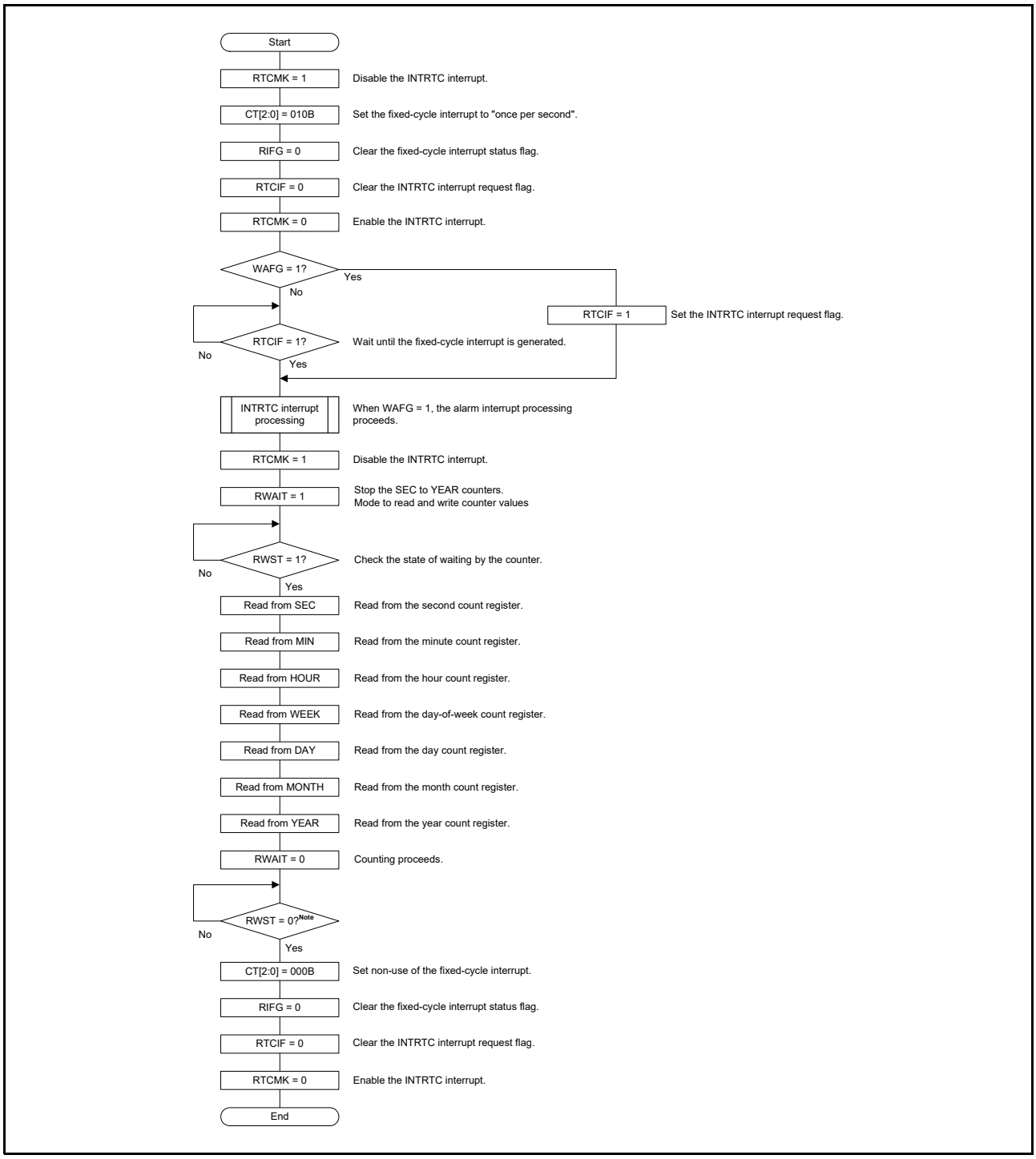


Note Be sure to confirm that RWST = 0 before setting STOP mode.

Caution **Complete the series of process of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.**

Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), day-of-week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be read in any sequence. All the registers do not have to be read and only some registers may be read.

Figure 16 - 20 Procedure for Reading from the Realtime Clock (When the Alarm Interrupt Is in Use)

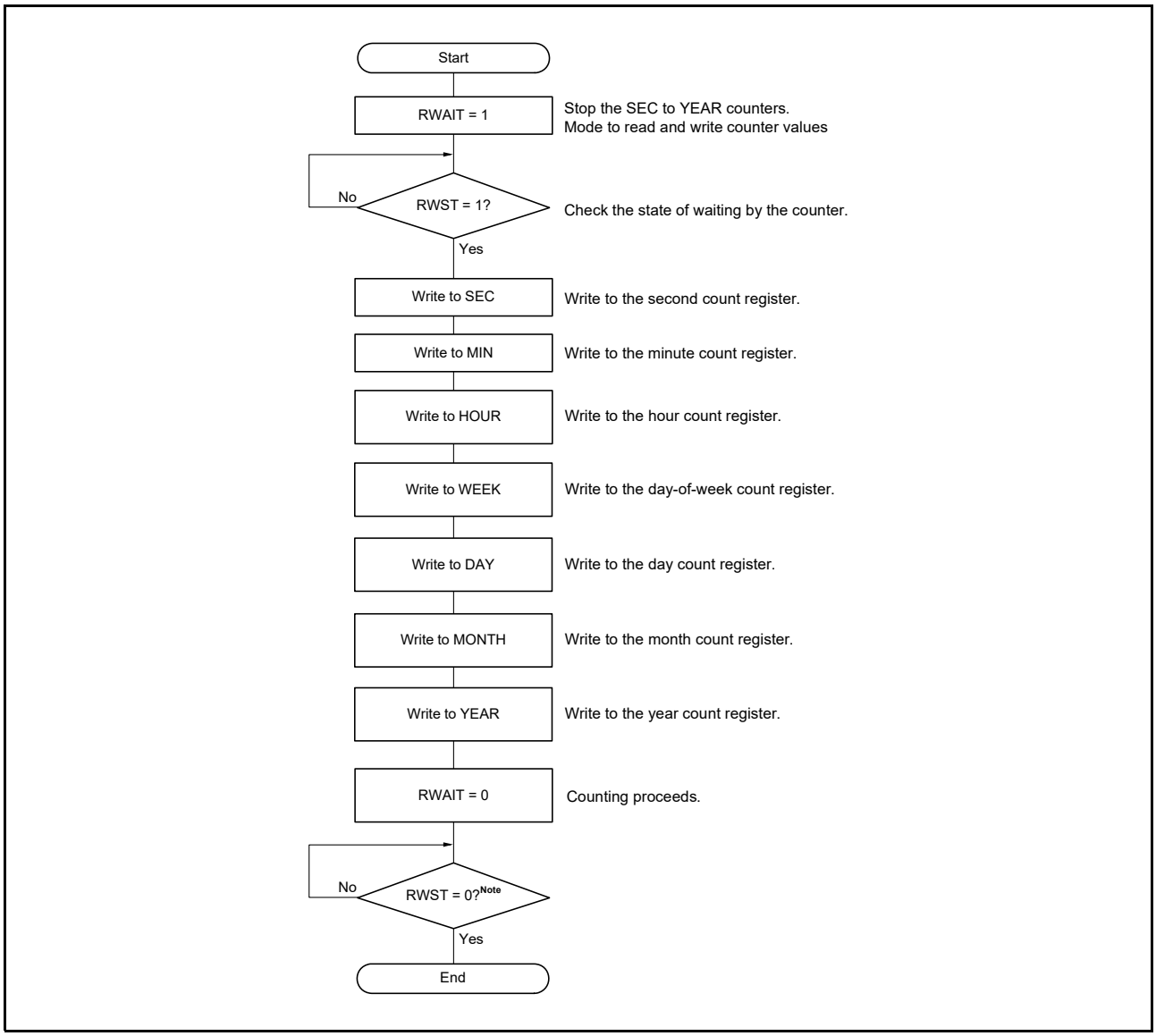


Note Be sure to confirm that RWST = 0 before setting STOP mode.

Caution **Complete the parts of the process from the start of INTRTC interrupt processing to clearing the RWAIT bit to 0 within 1 second.**

Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), day-of-week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be read in any sequence. All the registers do not have to be read and only some registers may be read.

Figure 16 - 21 Procedure for Writing to the Realtime Clock



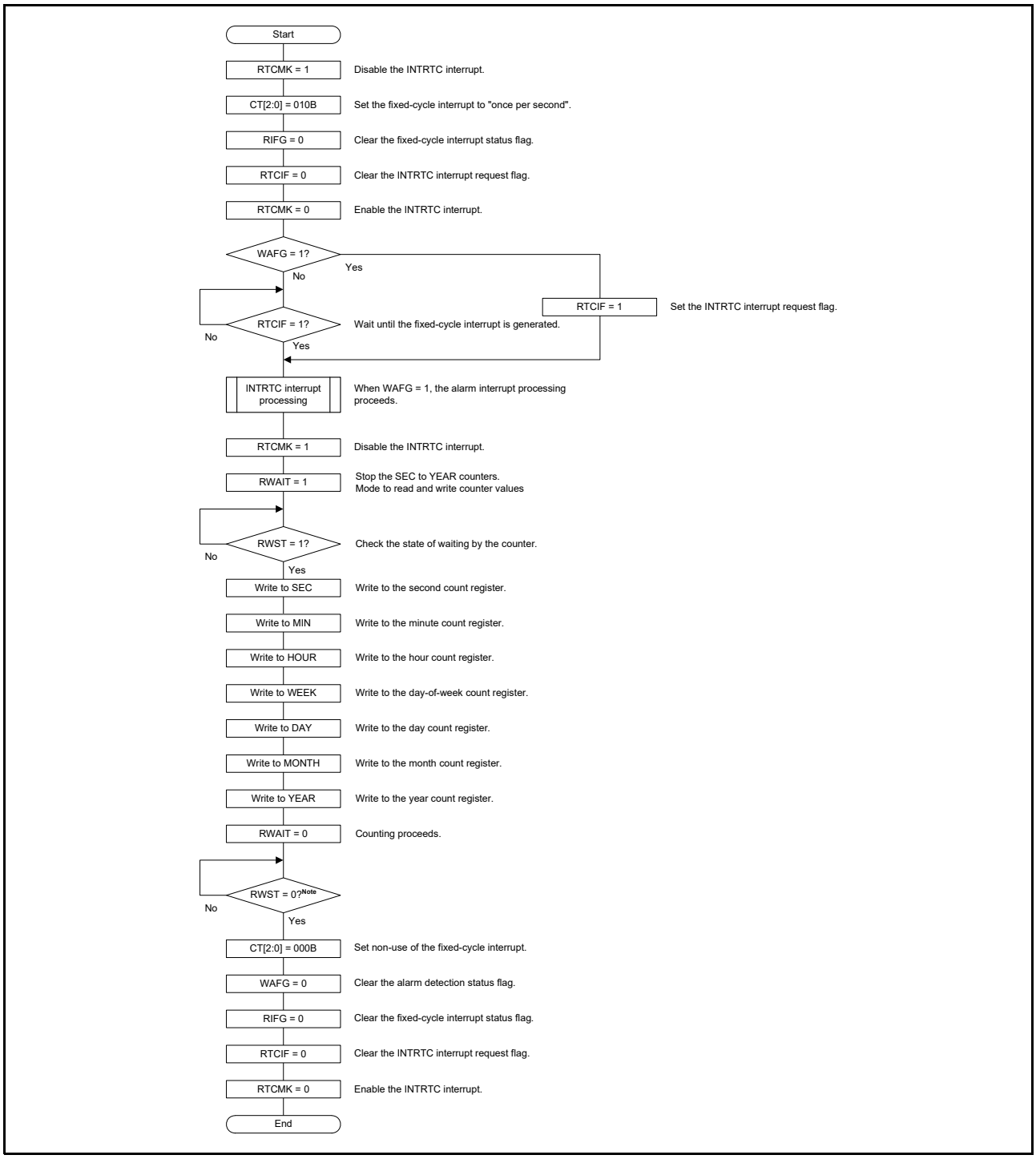
Note Be sure to confirm that RWST = 0 before setting STOP mode.

Caution 1. Complete the series of operations of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.

Caution 2. When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR registers while counting is in progress (RTCE = 1), rewrite the registers after disabling interrupt processing of INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags after rewriting the registers.

Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), day-of-week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be written in any sequence. All the registers do not have to be set and only some registers may be written.

Figure 16 - 22 Procedure for Writing to the Realtime Clock (When the Alarm Interrupt Is in Use)



Note Be sure to confirm that RWST = 0 before setting STOP mode.

Caution 1. Complete the parts of the process from the start of INTRTC interrupt processing to clearing the RWAIT bit to 0 within 1 second.

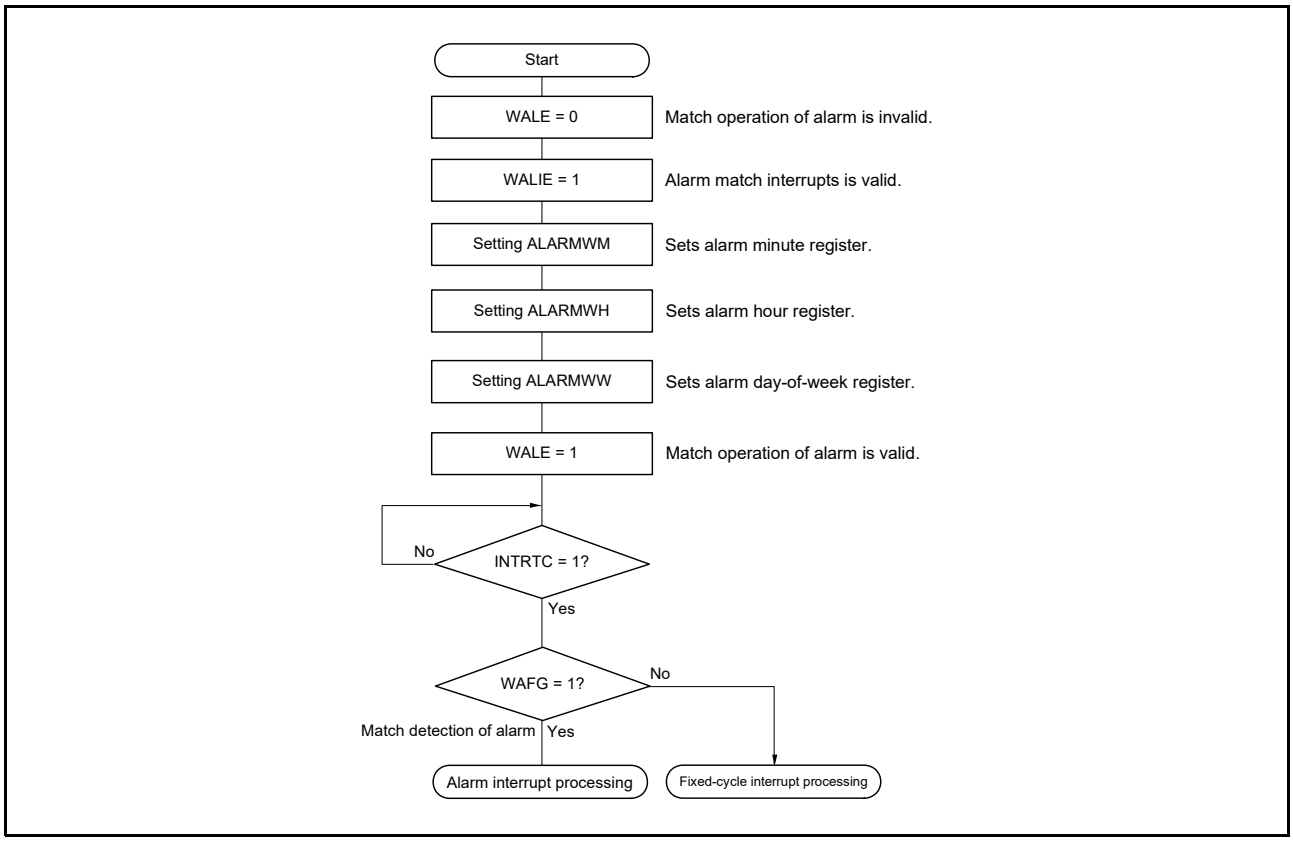
Caution 2. When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR registers while counting is in progress (RTCE = 1), rewrite the registers after disabling interrupt processing of INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags after rewriting the registers.

Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), day-of-week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be written in any sequence. All the registers do not have to be set and only some registers may be written.

16.4.4 Setting alarm by the realtime clock

Set time of alarm after setting WALE to 0 (alarm operation invalid) first.

Figure 16 - 23 Alarm Processing Procedure

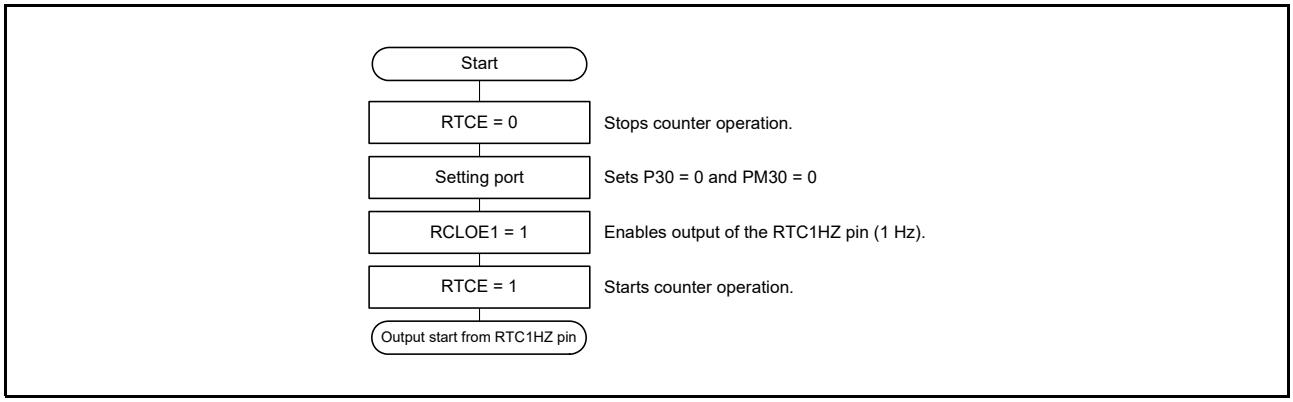


Remark 1. The alarm minute register (ALARMWM), alarm hour register (ALARMWH), and alarm day-of-week register (ALARMWW) may be written in any sequence.

Remark 2. Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). To use these two types of interrupts at the same time, the source of the interrupt can be identified by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) when an INTRTC is generated.

16.4.5 1 Hz output by the realtime clock

Figure 16 - 24 1 Hz Output Setting Procedure



Caution First set the RTCWEN bit to 1, while oscillation of the count clock (f_{RTCCK}) is stable.

16.4.6 Example of time error correction by the realtime clock

Time can be corrected with high accuracy when it is slow or fast, by setting a value to the time error correction register.

Example of calculating the correction value

The correction value used when correcting the count value of the internal counter (16 bits) is calculated by using the following expression.

Set the DEV bit to 0 when the correction range is –63.1 ppm or less, or 63.1 ppm or more.

(When DEV = 0)

Correction value[Note] = Number of correction counts in 1 minute ÷ 3 = (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60 ÷ 3

(When DEV = 1)

Correction value[Note] = Number of correction counts in 1 minute = (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60

Note The correction value is the time error correction value calculated by using bits 6 to 0 of the time error correction register (SUBCUD).
(When F6 = 0) Correction value = {(F5, F4, F3, F2, F1, F0) – 1} × 2
(When F6 = 1) Correction value = – {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2
When (F6, F5, F4, F3, F2, F1, F0) is (*, 0, 0, 0, 0, 0, *), time error correction is not performed. “*” is 0 or 1.
/F5 to /F0 are bit-inverted values (000011B when 111100B).

Remark 1. The correction value is 2, 4, 6, 8, … 120, 122, 124 or –2, –4, –6, –8, … –120, –122, –124.

Remark 2. The oscillation frequency is a value of the count clock (f_{RTCCK}).
It can be calculated from the output frequency of the RTC1HZ pin × 32768 when the time error correction register is set to its initial value (00H).

Remark 3. The target frequency is the frequency resulting after correction performed by using the time error correction register.

Correction example 1

Example of correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz –131.2 ppm)

[Measuring the oscillation frequency]

To measure the oscillation frequency[Note] of each product, a signal at about 32.768 kHz can be output from the PCLBUZ0 pin when the clock error correction register (SUBCUD) is set to its initial value (00H).

Note See **16.4.5 1 Hz output by the realtime clock** for the setting procedure of the RTC1Hz output, and see **18.4 Operations of the Clock Output/Buzzer Output Controller** for the setting procedure for output of about 32 kHz from the PCLBUZ0 pin.

[Calculating the correction value]

When the output frequency from the PCLBUZ0 pin is 32772.3 Hz:

Given that the target frequency is 32768 Hz (32772.3 Hz –131.2 ppm) and the extent of correction is –131.2 ppm (not in the range below –63.1 ppm), set DEV to 0. Accordingly, the expression for calculating the correction value when DEV is 0 is applicable.

Correction value = Error for correction of counting of 1 minute ÷ 3
= (Oscillation frequency ÷ target frequency – 1) × 32768 × 60 ÷ 3
= (32772.3 ÷ 32768 – 1) × 32768 × 60 ÷ 3
= 86

[Calculating the values to be set to (F6 to F0)]

When the correction value is 86:

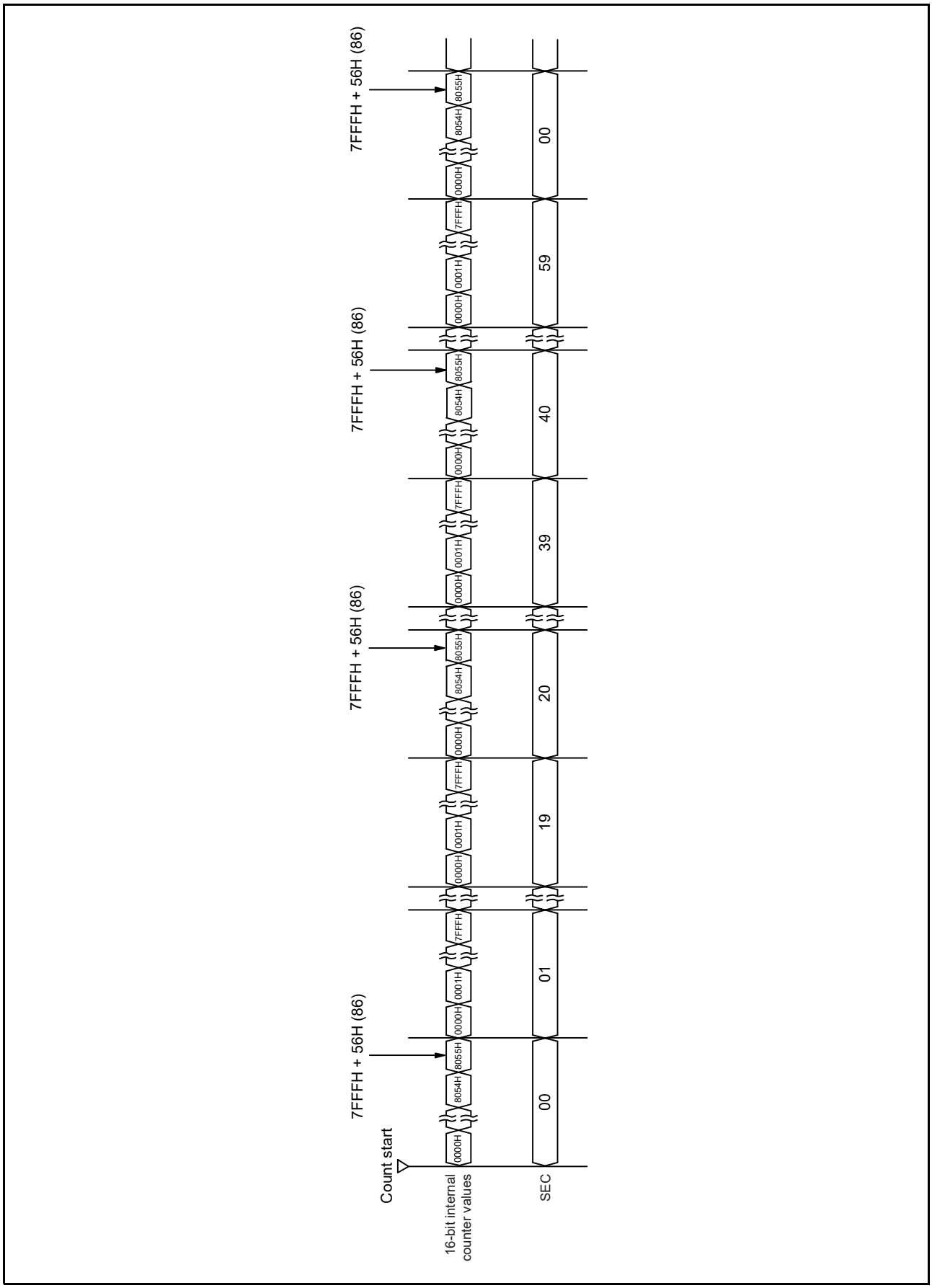
If the correction value is 0 or larger (the clock is running slow), set F6 to 0. Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

{(F5, F4, F3, F2, F1, F0) – 1} × 2 = 86
(F5, F4, F3, F2, F1, F0) = 44
(F5, F4, F3, F2, F1, F0) = (1, 0, 1, 1, 0, 0)

Consequently, when correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz –131.2 ppm), setting the correction register such that DEV is 0 and the correction value is 86 (bits 6 to 0 of the SUBCUD register: 0101100B) results in the desired frequency of 32768 Hz (error of 0 ppm).

Figure 16 - 25 shows the operation for correction when the value of (DEV, F6, F5, F4, F3, F2, F1, F0) is (0, 0, 1, 0, 1, 1, 0, 0).

Figure 16 - 25 Operation for Correction When the Value of (DEV, F6, F5, F4, F3, F2, F1, F0) = (0, 0, 1, 0, 1, 1, 0, 0)



Correction example 2

Example of correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm)

[Measuring the oscillation frequency]

To measure the oscillation frequency[Note] of each product, a signal at about 1 Hz can be output from the RTC1HZ pin when the clock error correction register (SUBCUD) is set to its initial value (00H).

Note See **16.4.5 1 Hz output by the realtime clock** for the setting procedure for output of about 1 Hz from the RTC1HZ pin.

[Calculating the correction value]

When the output frequency from the RTC1HZ pin is 0.9999817 Hz:

Oscillation frequency = 32768 × 0.9999817 ≈ 32767.4 Hz

Given that the target frequency is 32768 Hz (32767.4 Hz + 18.3 ppm), set DEV to 1. Accordingly, the expression for calculating the correction value when DEV is 1 is applicable.

Correction value = Error for correction of counting of 1 minute
= (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60
= (32767.4 ÷ 32768 – 1) × 32768 × 60
= –36

[Calculating the values to be set to (F6 to F0)]

When the correction value is –36:

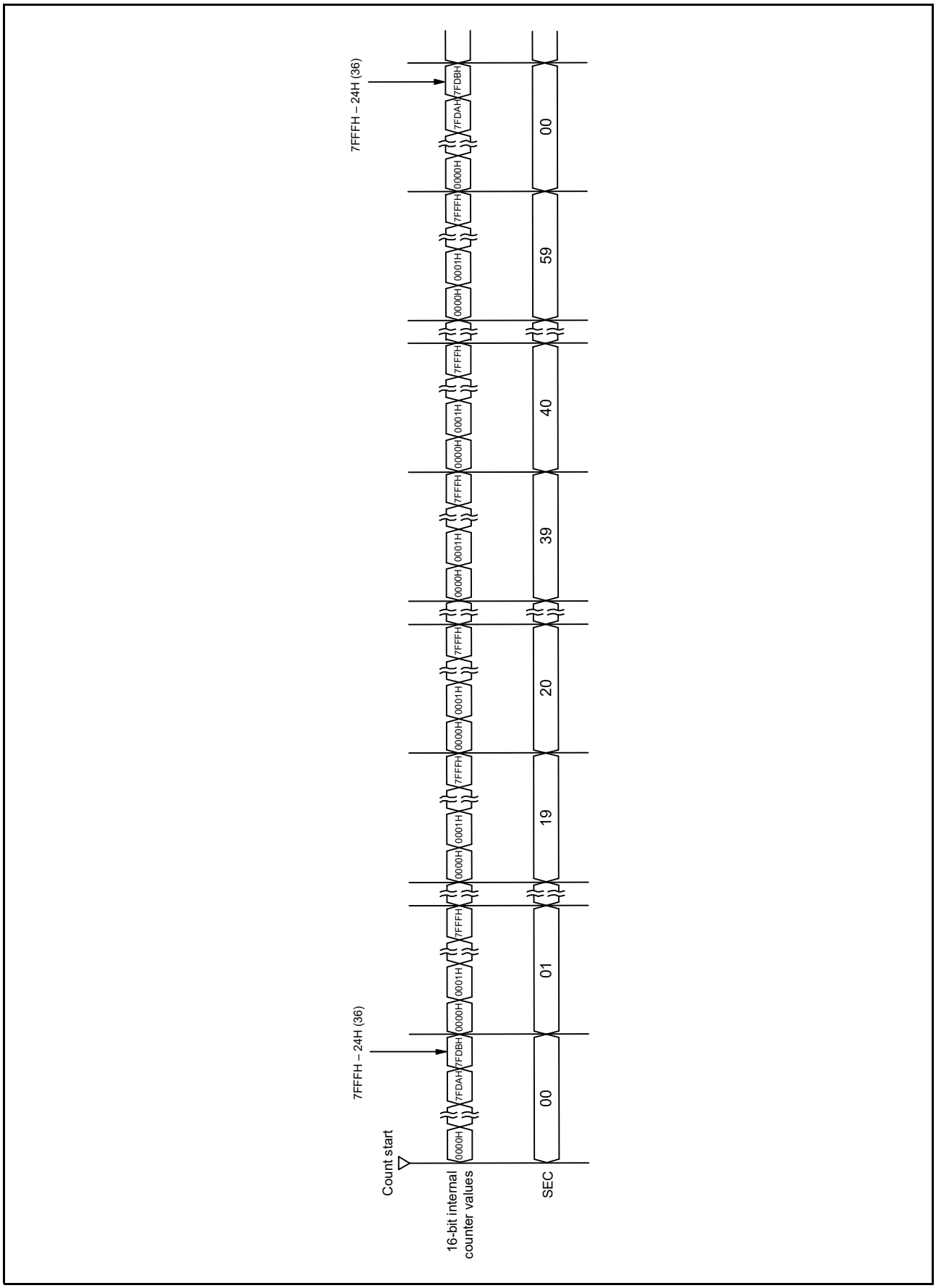
If the correction value is 0 or less (the clock is running fast), set F6 to 1. Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

– {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2 = –36
(/F5, /F4, /F3, /F2, /F1, /F0) = 17
(/F5, /F4, /F3, /F2, /F1, /F0) = (0, 1, 0, 0, 0, 1)
(F5, F4, F3, F2, F1, F0) = (1, 0, 1, 1, 1, 0)

Consequently, when correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm), setting the correction register such that DEV is 1 and the correction value is –36 (bits 6 to 0 of the SUBCUD register: 1101110B) results in the desired frequency of 32768 Hz (error of 0 ppm).

Figure 16 - 26 shows the operation for correction when the value of (DEV, F6, F5, F4, F3, F2, F1, F0) is (1, 1, 1, 0, 1, 1, 1, 0).

Figure 16 - 26 Operation for Correction When the Value of (DEV, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 0, 1, 1, 1, 0)



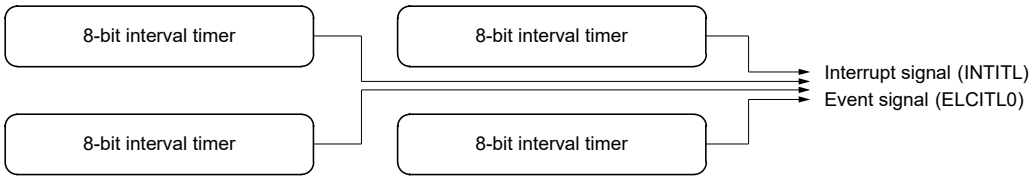
Section 17 32-bit Interval Timer (TML32)

The 32-bit interval timer is made up of four 8-bit interval timers (referred to as channels 0 to 3). Each is capable of operating independently and in that case they all have the same functions. Two 8-bit interval timer channels can be connected to operate as a 16-bit interval timer. Four 8-bit interval timer channels can be connected to operate as a 32-bit interval timer.

17.1 Overview

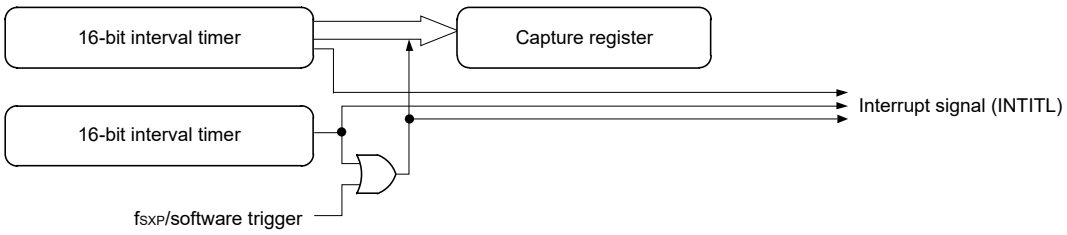
The 32-bit interval timer operates with the f_{MXP}, f_{SXP}, f_{IHP}, or f_{IMP} clock, which is asynchronous with the CPU operation. The 32-bit interval timer has the following functions.

1. 8-bit counter mode
   The 8-bit timers are usable as four 8-bit interval timers that generate interrupts (INTITL) at fixed intervals.



2. 16-bit counter mode
   The 8-bit timers are usable as two 16-bit interval timers that generate interrupts (INTITL) at fixed intervals.

3. 16-bit capture mode
   The 8-bit timers are usable as two 16-bit interval timers that generate interrupts (INTITL) at fixed intervals.
   The value of one of the 16-bit interval timers can also be stored in the capture register in response to a selected capture trigger.



4. 32-bit counter mode
   The 8-bit timers are usable as a 32-bit interval timer that generates interrupts (INTITL) at fixed intervals.

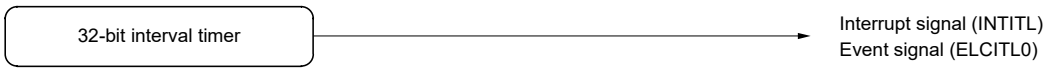


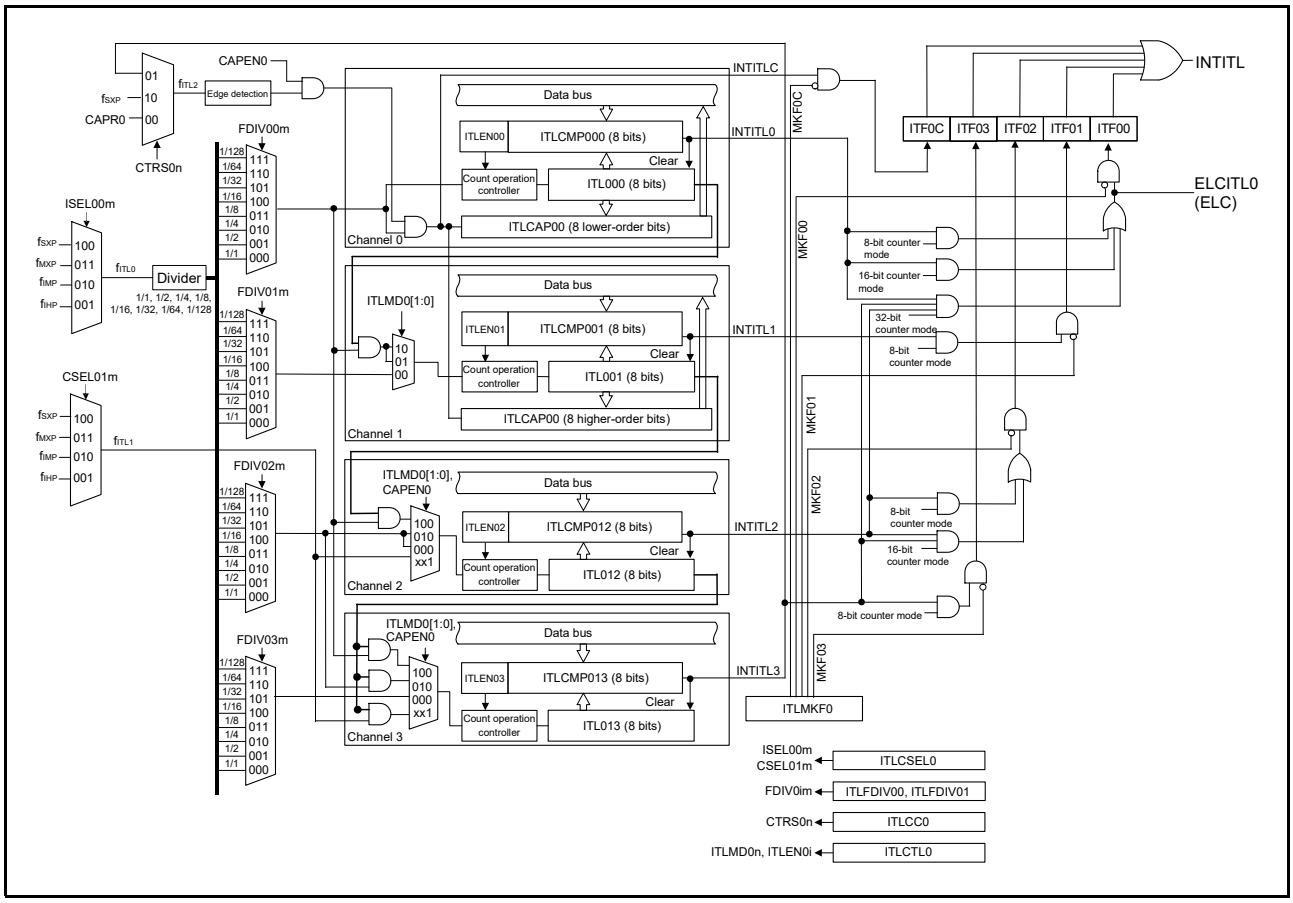
Table 17 - 1 lists the specifications of the 32-bit interval timer operations and **Figure 17 - 1** shows a block diagram of the 32-bit interval timer.

Table 17 - 1 Specifications of 32-bit Interval Timer Operations

Item	Description
Count source (operating clock)	• f_{MXP} • f_{SXP} • f_{IHP} • f_{IMP}
Capture clock (Selectable sources for counting by the timer which can generate a capture trigger)	• f_{MXP} • f_{SXP} • f_{IHP} • f_{IMP}
Frequency division ratio	• 1/1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128
Operating mode	• 8-bit counter mode Channels 0 to 3 independently operate as 8-bit counters. • 16-bit counter mode The combinations of channels 0 and 1 and channels 2 and 3 are cascade-connectable to operate as two 16-bit counters. • 32-bit counter mode Channels 0 to 3 are connected to operate as a 32-bit counter. • 16-bit capture mode Channels 0 and 1 are connected to operate as a 16-bit counter using the count source, channels 2 and 3 are connected to operate as a 16-bit counter using the capture clock, and the connected counters are used for capture operation.
Interrupt	• Five interrupt sources are integrated and output as the single interrupt signal (INTITL). - Output when the counter value in any of channels 0 to 3 matches the compare value. - Output when the capturing of the counter value is completed in capture mode.
ELC	• Output ELCITL0 to trigger the ELC operation when the counter value in channel 0 matches the compare value.

Remark f_{MXP}: High-speed peripheral clock frequency
f_{SXP}: Low-speed peripheral clock frequency
f_{IHP}: High-speed on-chip oscillator peripheral clock frequency
f_{IMP}: Middle-speed on-chip oscillator peripheral clock frequency (4 MHz)

Figure 17 - 1 Block Diagram of 32-bit Interval Timer



- ITL000, ITL001, ITL012, ITL013: These are the 8-bit counters. In 16-bit counter mode, the counters in channels 0 and 1 are connected (ITL000 + ITL001) and the counters in channels 2 and 3 are connected (ITL012 + ITL013). In 32-bit counter mode, the counters in channels 0 to 3 are connected (ITL000 + ITL001 + ITL012+ ITL013).
- ISEL00m, CSEL01m: Bits in the ITLCSEL0 register
- FDIV0im: Bits in the ITLFDIV0n register
- CTRS0n: Bits in the ITLCC0 register
- ITLMD0n, ITLEN0i: Bits in the ITLCTL0 register

Caution **f_{IHP} cannot be set to 64 MHz in 8-, 16-, or 32-bit counter mode. f_{IHP} also cannot be set to 64 MHz in 16-bit capture mode. Use the given interval timer frequency division register n (ITLFDIV0n) to set a frequency no greater than $f_{ITL0}/2$.**

Remark n = 0, 1; m = 0 to 2; i = 0 to 3

17.2 Registers to Control the 32-bit Interval Timer

The following registers are used to control the 32-bit interval timer.

- Peripheral enable register 1 (PER1)
- Peripheral reset control register 1 (PRR1)
- Interval timer compare registers 0mn (ITLCMP0mn) (mn = 00, 01, 12, 13)
- Interval timer compare registers 0n (ITLCMP0n) (n = 0, 1)
- Interval timer capture register 00 (ITLCAP00)
- Interval timer control register (ITLCTL0)
- Interval timer clock select register 0 (ITLCSEL0)
- Interval timer frequency division register 0 (ITLFDIV00)
- Interval timer frequency division register 1 (ITLFDIV01)
- Interval timer capture control register 0 (ITLCC0)
- Interval timer status register (ITLS0)
- Interval timer match detection mask register (ITLMKF0)

17.2.1 Peripheral enable register 1 (PER1)

This register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. If the 32-bit interval timer is to be used, be sure to set bit 4 (TML32EN) of this register to 1. The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 17 - 2 Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	<3>	2	1	<0>
PER1	DACEN	0	PGACMPEN	TML32EN	DTCEN	0	0	DALIEN

TML32EN	Control of supply of an input clock to the 32-bit interval timer
0	Stops supply of an input clock. • The SFRs used by the 32-bit interval timer cannot be written. • When an SFR used by the 32-bit interval timer is read, the value returned is 00H or 0000H.
1	Enables supply of an input clock. • The SFRs used by the 32-bit interval timer can be read and written.

17.2.2 Peripheral reset control register 1 (PRR1)

The PRR1 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. Set bit 4 (TML32RES) of this register to 1 to reset the 32-bit interval timer. The PRR1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 17 - 3 Format of Peripheral Reset Control Register 1 (PRR1)

Address: F00FBH
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	3	2	1	<0>
PRR1	DACRES	0	PGACMPRES	TML32RES	0	0	0	DALIRES

TML32RES	Control resetting of the 32-bit interval timer
0	The 32-bit interval timer is released from the reset state.
1	The 32-bit interval timer is in the reset state. • The SFRs for use with the 32-bit interval timer are initialized.

17.2.3 Interval timer compare registers 0mn (ITLCMP0mn) (mn = 00, 01, 12, 13)

These are compare value registers used in 8-bit counter mode. These registers can be set by an 8-bit memory manipulation instruction. The value of each ITLCMP0mn register following a reset is FFH. A value from 01H to FFH can be specified. Setting of 00H is prohibited. These registers hold values to be compared with the ITL000 to ITL013 counter values.

Figure 17 - 4 Format of Interval Timer Compare Registers 0mn (ITLCMP0mn)

Address: F0360H (ITLCMP000), F0361H (ITLCMP001), F0362H (ITLCMP012), F0363H (ITLCMP013)
After reset: FFH
R/W: R/W**Note**

Symbol	7	6	5	4	3	2	1	0
ITLCMP0mn								

Note Write to the ITLCMP000 to ITLCMP013 registers while the settings of the ITLEN00 to ITLEN03 bits in the ITLCTL0 register are 0, respectively.

17.2.4 Interval timer compare registers 0n (ITLCMP0n) (n = 0, 1)

These are compare value registers used in 16-bit or 32-bit counter mode. These registers can be set by a 16-bit memory manipulation instruction. The value of each ITLCMP0n register following a reset is FFFFH. A value from 0001H to FFFFH can be specified. Setting these registers to 0000H is prohibited. These registers hold values to be compared with the ITL0n counter values. When the settings of the ITLMD01 and ITLMD00 bits in the ITLCTL0 register are 1 and 0, respectively, these registers are used as compare registers in 32-bit counter mode; specify the upper 16-bit compare value in the ITLCMP01 register and the lower 16-bit compare value in the ITLCMP00 register.

Figure 17 - 5 Format of Interval Timer Compare Registers 0n (ITLCMP0n)

Address: F0360H (ITLCMP00), F0362H (ITLCMP01)
After reset: FFFFH
R/W: R/W**Note**

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ITLCMP0n																

Note Write to the ITLCMP00 register while the ITLEN00 bit in the ITLCTL0 register is 0.
Write to the ITLCMP01 register while the ITLEN02 bit in the ITLCTL0 register is 0 in 16-bit counter mode or while the ITLEN00 bit in the ITLCTL0 register is 0 in 32-bit counter mode.

17.2.5 Interval timer capture register 00 (ITLCAP00)

This register holds 16-bit captured values when the interval timers are operating in 16-bit counter mode. The ITLCAP00 register can be read by a 16-bit manipulation instruction. The value of this register following a reset is 0000H. The values of the 16-bit counters (ITL000 + ITL001) are stored in the ITLCAP00 register in response to the capture trigger selected in the ITLCC0 register when the CAPEN0 bit in the ITLCC0 register is 1. When an interrupt on compare match with the ITLCMP01 register is to be used, select the counter clock in the ITLCSEL0 register and set the comparison value in the TLCMP01 register.

Figure 17 - 6 Format of Interval Timer Capture Register 00 (ITLCAP00)

Address: F0364H
After reset: 0000H
R/W: R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ITLCAP00																

17.2.6 Interval timer control register (ITLCTL0)

This register is used to start or stop counting by the interval timer and to select 8-bit, 16-bit, or 32-bit counter mode. This register can be set by a 1-bit or 8-bit manipulation instruction. The value of this register following a reset is 00H.

Figure 17 - 7 Format of Interval Timer Control Register (ITLCTL0) (1/2)

Address: F0366H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	<3>	<2>	<1>	<0>
ITLCTL0	ITLMD01	ITLMD00	0	0	ITLEN03	ITLEN02	ITLEN01	ITLEN00

ITLMD01	ITLMD00	Selection of 8-bit, 16-bit, or 32-bit counter mode[Note 1]
0	0	The interval timer operates in 8-bit counter mode.
0	1	The interval timer operates in 16-bit counter mode (channel 0 is connected with channel 1 and channel 2 is connected with channel 3).
1	0	The interval timer operates in 32-bit counter mode (channels 0 to 3 are connected).
1	1	Setting prohibited.

ITLEN03	8-bit counter mode: ITL013 count enable[Note 2]
0	Counting stops.
1	Counting begins.
In 8-bit counter mode, writing 1 to this bit starts up-counting in the ITL013 counter and writing 0 stops it. In 16-bit counter mode, set this bit to 0. In 32-bit counter mode, set this bit to 0.	

ITLEN02	8-bit counter mode: ITL012 count enable[Note 2] 16-bit counter mode: ITL012 + ITL013 count enable[Note 2]
0	Counting stops.
1	Counting begins.
In 8-bit counter mode, writing 1 to this bit starts up-counting in the ITL012 counter and writing 0 stops it. In 16-bit counter mode, writing 1 to this bit starts up-counting in the ITL012 + ITL013 counter and writing 0 stops it. In 32-bit counter mode, set this bit to 0.	

ITLEN01	8-bit counter mode: ITL001 count enable[Note 2]
0	Counting stops.
1	Counting begins.
In 8-bit counter mode, writing 1 to this bit starts up-counting in the ITL001 counter and writing 0 stops it. In 16-bit counter mode, set this bit to 0. In 32-bit counter mode, set this bit to 0.	

Figure 17 - 7 Format of Interval Timer Control Register (ITLCTL0) (2/2)

ITLEN00	8-bit counter mode: ITL000 count enable[Note 2] 16-bit counter mode: ITL000 + ITL001 count enable[Note 2] 32-bit counter mode: ITL000 + ITL001 + ITL012 + ITL013 count enable[Note 2]
0	Counting stops.
1	Counting begins.
In 8-bit counter mode, writing 1 to this bit starts up-counting in the ITL000 counter and writing 0 stops it. In 16-bit counter mode, writing 1 to this bit starts up-counting in the ITL000 + ITL001 counter and writing 0 stops it. In 32-bit counter mode, writing 1 to this bit starts up-counting in the ITL000 + ITL001 + ITL012 + ITL013 counter and writing 0 stops it.	

Note 1. To change the mode specified in the ITLMD0[1:0] bits, be sure to write to the ITLMD0[1:0] bits only while the ITLEN00, ITLEN01, ITLEN02, and ITLEN03 bits are all 0.

Note 2. When any of the ITLEN03 to ITLEN00 bits is cleared to 0, the corresponding counter is cleared to 0 without synchronization with the counter clock.

Mode	ITLMD01	ITLMD00	ITLEN03	ITLEN02	ITLEN01	ITLEN00	Target Counter
8-bit mode	0	0				✓	ITL000
					✓		ITL001
				✓			ITL012
			✓				ITL013
16-bit mode	0	1	Always set to 0.		Always set to 0.	✓	ITL000 + ITL001
			Always set to 0.	✓	Always set to 0.		ITL012 + ITL013
32-bit mode	1	0	Always set to 0.	Always set to 0.	Always set to 0.	✓	ITL000 + ITL001 + ITL012 + ITL013

Remark ✓: Enables counting in the target counter.

In 8-bit counter mode, two or more bits of ITLEN03 to ITLEN00 can be set to 1 or 0 at the same time.
In 16-bit counter mode, the ITLEN02 and ITLEN00 bits can be set to 1 or 0 at the same time.

17.2.7 Interval timer clock select register 0 (ITLCSEL0)

This register is used to select the count source for the interval timer. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 17 - 8 Format of Interval Timer Clock Select Register 0 (ITLCSEL0)

Address: F0367H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ITLCSEL0	0	CSEL012	CSEL011	CSEL010	0	ISEL002	ISEL001	ISEL000

CSEL012	CSEL011	CSEL010	Selection of interval timer count clock for capturing (f_{ITL1})**Note 1**
0	0	0	Counting stops.
0	0	1	f_{IHP}**Note 2**
0	1	0	f_{IMP}
0	1	1	f_{MXP}
1	0	0	f_{SXP}
Others			Setting prohibited.

ISEL002	ISEL001	ISEL000	Selection of interval timer count clock (f_{ITL0})**Note 1**
0	0	0	Counting stops.
0	0	1	f_{IHP}**Note 2**
0	1	0	f_{IMP}
0	1	1	f_{MXP}
1	0	0	f_{SXP}
Others			Setting prohibited.

Note 1. Be sure to write to the CSEL01[2:0] bits and ISEL00[2:0] bits only while the ITLEN03 to ITLEN00 bits are all 0.

Note 2. f_{IHP} cannot be set to 64 MHz in 8-, 16-, or 32-bit counter mode. f_{IHP} also cannot be set to 64 MHz in 16-bit capture mode. Use the given interval timer frequency division register n (ITLFDIV0n) to set a frequency no greater than $f_{ITL0}/2$.

Remark 1. f_{MXP}: High-speed peripheral clock frequency
f_{SXP}: Low-speed peripheral clock frequency
f_{IHP}: High-speed on-chip oscillator peripheral clock frequency
f_{IMP}: Middle-speed on-chip oscillator peripheral clock frequency

Remark 2. n = 0, 1

17.2.8 Interval timer frequency division register 0 (ITLFDIV00)

This register is used to select the counter clock for the interval timer. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 17 - 9 Format of Interval Timer Frequency Division Register 0 (ITLFDIV00)

Address: F0368H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ITLFDIV00	0	FDIV012	FDIV011	FDIV010	0	FDIV002	FDIV001	FDIV000

FDIV012	FDIV011	FDIV010	8-bit counter mode: Counter clock for ITL001Note 1
0	0	0	f_{ITL0}
0	0	1	$f_{ITL0}/2$
0	1	0	$f_{ITL0}/4$
0	1	1	$f_{ITL0}/8$
1	0	0	$f_{ITL0}/16$
1	0	1	$f_{ITL0}/32$
1	1	0	$f_{ITL0}/64$
1	1	1	$f_{ITL0}/128$
In 8-bit counter mode, ITL001 counts cycles of the counter clock specified in the FDIV012 to FDIV010 bits. In 16-bit counter mode, set these bits to 000B. In 32-bit counter mode, set these bits to 000B.			

FDIV002	FDIV001	FDIV000	8-bit counter mode: Counter clock for ITL000Note 2 16-bit counter mode: Counter clock for ITL000 + ITL001Note 2 32-bit counter mode: Counter clock for ITL000 + ITL001 + ITL012 + ITL013Note 2
0	0	0	f_{ITL0}
0	0	1	$f_{ITL0}/2$
0	1	0	$f_{ITL0}/4$
0	1	1	$f_{ITL0}/8$
1	0	0	$f_{ITL0}/16$
1	0	1	$f_{ITL0}/32$
1	1	0	$f_{ITL0}/64$
1	1	1	$f_{ITL0}/128$
In 8-bit counter mode, ITL000 counts cycles of the counter clock specified in the FDIV002 to FDIV000 bits. In 16-bit counter mode, ITL000 + ITL001 counts cycles of the counter clock specified in the FDIV002 to FDIV000 bits. In 32-bit counter mode, ITL000 +ITL001 + ITL012 + ITL013 counts cycles of the counter clock specified in the FDIV002 to FDIV000 bits.			

Note 1. In 8-bit counter mode, be sure to write to the FDIV012 to FDIV010 bits only while the ITLEN01 bit in the ITLCTL0 register is 0.

Note 2. Be sure to write to the FDIV002 to FDIV000 bits only while the ITLEN00 bit in the ITLCTL0 register is 0.

17.2.9 Interval timer frequency division register 1 (ITLFDIV01)

This register is used to select the counter clock for the interval timer. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 17 - 10 Format of Interval Timer Frequency Division Register 1 (ITLFDIV01)

Address: F0369H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ITLFDIV01	0	FDIV032	FDIV031	FDIV030	0	FDIV022	FDIV021	FDIV020

FDIV032	FDIV031	FDIV030	8-bit counter mode: Counter clock for ITL013Note 1
0	0	0	f_{ITL0}
0	0	1	$f_{ITL0}/2$
0	1	0	$f_{ITL0}/4$
0	1	1	$f_{ITL0}/8$
1	0	0	$f_{ITL0}/16$
1	0	1	$f_{ITL0}/32$
1	1	0	$f_{ITL0}/64$
1	1	1	$f_{ITL0}/128$
In 8-bit counter mode, ITL013 counts cycles of the counter clock specified in the FDIV032 to FDIV030 bits. In 16-bit counter mode, set these bits to 000B. In 32-bit counter mode, set these bits to 000B.			

FDIV022	FDIV021	FDIV020	8-bit counter mode: Counter clock for ITL012Note 2 16-bit counter mode: Counter clock for ITL012 and ITL013Note 2
0	0	0	f_{ITL0}
0	0	1	$f_{ITL0}/2$
0	1	0	$f_{ITL0}/4$
0	1	1	$f_{ITL0}/8$
1	0	0	$f_{ITL0}/16$
1	0	1	$f_{ITL0}/32$
1	1	0	$f_{ITL0}/64$
1	1	1	$f_{ITL0}/128$
In 8-bit counter mode, ITL012 counts cycles of the counter clock specified in the FDIV022 to FDIV020 bits. In 16-bit counter mode, ITL012 + ITL013 counts cycles of the counter clock specified in the FDIV022 to FDIV020 bits. In 32-bit counter mode, these bits are not used; write 000B to them.			

Note 1. In 8-bit counter mode, be sure to write to the FDIV032 to FDIV030 bits only while the ITLEN03 bit in the ITLCTL0 register is 0.

Note 2. In 8-bit or 16-bit counter mode, be sure to write to the FDIV022 to FDIV020 bits only while the ITLEN02 bit in the ITLCTL0 register is 0.

17.2.10 Interval timer capture control register 0 (ITLCC0)

This register is used to enable or disable the capture function of the interval timer, specify whether to hold or clear the capture completion flag, set up the software trigger, and select the capture trigger. This register can be set by a 1-bit or 8-bit manipulation instruction. The value of this register following a reset is 00H.

Figure 17 - 11 Format of Interval Timer Capture Control Register 0 (ITLCC0)

Address: F036AH
After reset: 00H
R/W: R/W

Symbol	7	<6>	5	<4>	3	2	1	0
ITLCC0	CAPEN0	CAPF0CR	CAPF0	CAPR0	CAPC0CR	0	CTRS01	CTRS00

CAPEN0	Capture enable[Note 1]
0	Capturing is disabled.
1	Capturing is enabled.

CAPF0CR	Capture completion flag clear[Note 2]
0	The value of the capture completion flag CAPF0 is held.
1	The value of the capture completion flag CAPF0 is cleared.

CAPF0	Capture completion flag[Note 3]
0	Capturing has not been completed.
1	Capturing has been completed. This flag is set to 1 after a capture trigger selected in the CTRS01 and CTRS00 bits is generated and the captured data is stored in the ITLCAP00 register. Writing 1 to the CAPF0CR bit clears this flag to 0.

CAPR0	Software capture trigger[Notes 4, 7]
0	Trigger operation does not proceed.
1	A software trigger for capturing is generated.

CAPC0CR	Selection of the 16-bit counter (ITL000 + ITL001) clearing after capturing[Note 5]
0	The 16-bit counter (ITL000 + ITL001) is retained after the completion of capturing.
1	The 16-bit counter (ITL000 + ITL001) is cleared after the completion of capturing.

CTRS01	CTRS00	Selection of capture trigger[Notes 6, 7]
0	0	Software trigger
0	1	Interrupt on compare match with ITLCMP01[Note 8]
1	0	fSXP (rising edge)

Note 1. Be sure to write to the CAPEN0 bit only while the ITLEN0[3:0] bits in the ITLCTL0 register are all 0.
Note 2. The CAPF0CR bit is always read as 0.
Note 3. Bit 5 is read-only.
Note 4. The CAPR0 bit is always read as 0.
Note 5. Be sure to write to the CAPC0CR bit only while the ITLEN0[3:0] bits in the ITLCTL0 register are all 0.
Note 6. Be sure to write to the CTRS01 and CTRS00 bits only while the ITLEN0[3:0] bits in the ITLCTL0 register are all 0.
Note 7. In the capture operation, the interval at which the capture trigger is generated should be two or more cycles of the counter clock.

Note 8. When the interrupt on compare match with ITLCMP01 is selected as a capture trigger, the compare match detection flag for channel 2 (ITF02) and capture detection flag (ITF0C) are set on capture of the counter value. When only using the capture detection flag, set the ITLMKF0 register to mask the compare match detection flag for channel 2.

17.2.11 Interval timer status register (ITLS0)

This is a status register for the interval timer. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H. When the value of the ITL0mn register (mn = 00, 01, 12, 13) matches the value specified in the ITLCMP0mn, ITLCMP00, and ITLCMP01 registers, the compare-match flag for the corresponding channel is set. When a capture trigger is generated while the CAPEN0 bit in the ITLCC0 register is 1, the capture detection flag is set after the value of the ITL0n counter is stored in the ITLCAP00 register. The values of the ITF0C and ITF03 to ITF00 flags in this register are ORed and output as the interrupt request signal (INTITL). **Table 17 - 2** shows the conditions for setting the status flags in each timer mode selected by the ITLMD0[1:0] bits.

Figure 17 - 12 Format of Interval Timer Status Register (ITLS0)

Address: F036BH
After reset: 00H
R/W: R/W**Note**

Symbol	7	6	5	4	3	2	1	0
ITLS0	0	0	0	ITF0C	ITF03	ITF02	ITF01	ITF00

ITF0C	Capture detection flag
0	Completion of capturing has not been detected.
1	Completion of capturing has been detected.

ITF03	Compare match detection flag for channel 3
0	A compare match signal has not been detected in channel 3.
1	A compare match signal has been detected in channel 3.

ITF02	Compare match detection flag for channel 2
0	A compare match signal has not been detected in channel 2.
1	A compare match signal has been detected in channel 2.

ITF01	Compare match detection flag for channel 1
0	A compare match signal has not been detected in channel 1.
1	A compare match signal has been detected in channel 1.

ITF00	Compare match detection flag for channel 0
0	A compare match signal has not been detected in channel 0.
1	A compare match signal has been detected in channel 0.

Note Writing 1 to each bit is ignored. To clear the ITF0C or ITF0i flag (i = 0, 1, 2, 3), write 0 to the desired bit and 1 to the other bits by using an 8-bit memory manipulation instruction.

Caution 1. If clearing any of the ITF0C, ITF03, ITF02, ITF01, ITF00 flags to 0 does not lead to the value of the ITLS0 register becoming 00H, an interrupt request (INTITL) is generated and the interrupt request flag (ITLIF) is set to 1.

Caution 2. To clear a flag in the ITLS0 register to 0, only write 0 to a flag that has the setting 1. This is because writing 0 to a flag that has the setting 0 may make detecting a compare match signal or capture detection signal generated at the same time as the writing of 0 impossible. For example, when the ITF01 flag is set to 1, write 00011101B to the ITLS0 register to clear the ITF01 flag.

Table 17 - 2 Conditions for Setting the Status Flags in Each Timer Mode

Mode	ITLMD01	ITLMD00	CAPEN0	Status Flag	Conditions for Setting Status Flag
8-bit mode	0	0	×	ITF00	The next rising edge of the counter clock following a match between the ITLCMP000 and ITL000 values
			×	ITF01	The next rising edge of the counter clock following a match between the ITLCMP001 and ITL001 values
			×	ITF02	The next rising edge of the counter clock following a match between the ITLCMP012 and ITL012 values
			×	ITF03	The next rising edge of the counter clock following a match between the ITLCMP013 and ITL013 values
16-bit mode	0	1	×	ITF00	The next rising edge of the counter clock following a match between the ITLCMP00 and ITL000 + ITL001 values
			×	ITF02	The next rising edge of the counter clock following a match between the ITLCMP01 and ITL012 + ITL013 values
			1	ITF0C	The ITL000 + ITL001 value is stored in ITLCAP00 after a capture trigger is generated.
32-bit mode	1	0	—	ITF00	The next rising edge of the counter clock following a match between the ITLCMP00 + ITLCMP01 and ITL000 + ITL001 + ITL012 + ITL013 values

Remark ×: Don’t care

17.2.12 Interval timer match detection mask register (ITLMKF0)

This register is used to enable or disable setting of each effective bit in the interval timer status register (ITLS0) to 1. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H. Setting an MKF0C or MKF03 to MKF00 bit to 1 masks the corresponding status flag (ITF0C, ITF03 to ITF00), after which the given flag is not set to 1 even if a compare match with a compare register or capture completion is detected. Since the status flag will not be set to1, masking also prevents generation of the interval detection interrupt (INTITL).

Figure 17 - 13 Format of Interval Timer Match Detection Mask Register (ITLMKF0)

Address: F036CH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ITLMKF0	0	0	0	MKF0C	MKF03	MKF02	MKF01	MKF00

MKF0C	Mask for capture detection status flag[Note]
0	The ITF0C flag is not masked.
1	The ITF0C flag is masked.

MKF03	Mask for compare match status flag for channel 3[Note]
0	The ITF03 flag is not masked.
1	The ITF03 flag is masked.

MKF02	Mask for compare match status flag for channel 2[Note]
0	The ITF02 flag is not masked.
1	The ITF02 flag is masked.

MKF01	Mask for compare match status flag for channel 1[Note]
0	The ITF01 flag is not masked.
1	The ITF01 flag is masked.

MKF00	Mask for compare match status flag for channel 0[Note]
0	The ITF00 flag is not masked.
1	The ITF00 flag is masked.

Note Setting all functional bits to 1 for masking prevents setting of the corresponding bits in the ITLS0 register. This in turn prevents software detection of compare matches and completion of capture. When compare match for any of channels 0 to 3 is to be used, be sure to set the bit corresponding to the given status flag to 0 so that the flag is not masked. For the state of completion of capture, on the other hand, the CAPF0 flag in the interval timer capture control register 0 (ITLCC0) can be used to detect this even when the MKF0C bit is set to 1 to mask the ITF0C flag.

17.3 Operation

17.3.1 Counter mode settings

The 32-bit interval timer has three counter modes: 8-bit counter mode, 16-bit counter mode, and 32-bit counter mode. **Tables 17 - 3** to **17 - 5** show the registers and settings for use in 8-bit counter mode, 16-bit counter mode, and 32-bit counter mode, respectively.

Table 17 - 3 Registers and Settings Used in 8-bit Counter Mode

Register	Bit	Setting
Interval timer compare registers 0mn (ITLCMP0mn)	Bits 7 to 0	Specify 8-bit compare values for channels 0 to 3.
Interval timer control register 0 (ITLCTL0)	ITLEN00	Specify whether to start or stop counting in channel 0.
	ITLEN01	Specify whether to start or stop counting in channel 1.
	ITLEN02	Specify whether to start or stop counting in channel 2.
	ITLEN03	Specify whether to start or stop counting in channel 3.
	ITLMD00	Set to 0.
	ITLMD01	Set to 0.
Interval timer frequency division registers n (ITLFDIV0n)	FDIV00[2:0]	Select the count clock for channel 0.
	FDIV01[2:0]	Select the count clock for channel 1.
	FDIV02[2:0]	Select the count clock for channel 2.
	FDIV03[2:0]	Select the count clock for channel 3.
Interval timer clock select register 0 (ITLCSEL0)	ISEL00[2:0]	Select the count clock for the interval timer.
	CSEL01[2:0]	Set to 000B.
Interval timer capture control register 0 (ITLCC0)	Bits 7 to 0	Set to 0.

Remark mn = 00, 01, 12, 13

Table 17 - 4 Registers and Settings Used in 16-bit Counter Mode

Register	Bit	Setting
Interval timer compare registers 0n (ITLCMP0n)	Bits 15 to 0	Specify 16-bit compare values for channels 0 and 1, and channels 2 and 3.
Interval timer control register 0 (ITLCTL0)	ITLEN00	Specify whether to start or stop counting in channels 0 and 1.
	ITLEN01	Set to 0.
	ITLEN02	Specify whether to start or stop counting in channels 2 and 3.
	ITLEN03	Set to 0.
	ITLMD00	Set to 1.
	ITLMD01	Set to 0.
Interval timer frequency division registers n (ITLFDIV0n)	FDIV00[2:0]	Select the count clock for channels 0 and 1.
	FDIV01[2:0]	Set to 000B.
	FDIV02[2:0]	Select the count clock for channels 2 and 3.
	FDIV03[2:0]	Set to 000B.
Interval timer clock select register 0 (ITLCSEL0)	ISEL00[2:0]	Select the count clock for the interval timer.
	CSEL01[2:0]	Set to 000B.
Interval timer capture control register 0 (ITLCC0)	Bits 7 to 0	Set to 0.

Remark n = 0, 1

Table 17 - 5 Registers and Settings Used in 32-bit Counter Mode

Register	Bit	Setting
Interval timer compare registers 0n (ITLCMP0n)	Bits 15 to 0	Specify a compare value in 32-bit counter mode. Specify the lower 16 bits of the compare value in channels 0 and 1 (ITLCMP00) and the upper 16 bits of the compare value in channels 2 and 3 (ITLCMP01).
Interval timer control register 0 (ITLCTL0)	ITLEN00	Specify whether to start or stop counting in channels 0 to 3.
	ITLEN01	Set to 0.
	ITLEN02	Set to 0.
	ITLEN03	Set to 0.
	ITLMD00	Set to 0.
	ITLMD01	Set to 1.
Interval timer frequency division registers n (ITLFDIV0n)	FDIV00[2:0]	Select the count clock for channels 0 to 3.
	FDIV01[2:0]	Set to 000B.
	FDIV02[2:0]	Set to 000B.
	FDIV03[2:0]	Set to 000B.
Interval timer clock select register 0 (ITLCSEL0)	ISEL00[2:0]	Select the count clock for the interval timer.
	CSEL01[2:0]	Set to 000B.
Interval timer capture control register 0 (ITLCC0)	Bits 7 to 0	Set to 0.

Remark n = 0, 1

17.3.2 Capture mode settings

When the 16-bit capture mode is to be used for channels 0 and 1, the counter value is stored in interval timer capture register 00 (ITLCAP00) in response to a selected capture trigger.

Table 17 - 6 Registers and Settings Used in 16-bit Capture Mode

Register	Bit	Setting
Interval timer compare register 00 (ITLCMP00)	Bits 15 to 0	Specify 16-bit compare values for channels 0 and 1.
Interval timer compare register 01 (ITLCMP01)Note	Bits 15 to 0	Specify 16-bit compare values for channels 2 and 3.
Interval timer control register 0 (ITLCTL0)	ITLEN00	Specify whether to start or stop counting in channels 0 and 1.
	ITLEN01	Set to 0.
	ITLEN02	Specify whether to start or stop counting in channels 2 and 3.
	ITLEN03	Set to 0.
	ITLMD00	Set to 1.
	ITLMD01	Set to 0.
Interval timer frequency division registers n (ITLFDIV0n)	FDIV00[2:0]	Select the count clock for channel 0.
	FDIV01[2:0]	Set to 000B.
	FDIV02[2:0]	Set to 000B.
	FDIV03[2:0]	Set to 000B.
Interval timer clock select register 0 (ITLCSEL0)	ISEL00[2:0]	Select the count clock for the interval timer in channels 0 and 1.
	CSEL01[2:0]	Select the count clock for the interval timer for capturing in channels 2 and 3.
Interval timer capture control register 0 (ITLCC0)	CAPEN0	Set to 1.
	CAPC0CR	Specify whether to clear or hold the counter value in channels 0 and 1 after the completion of capturing.
	CTRS0[1:0]	Select a capture trigger.

Note Channels 2 and 3 can only be used in 16-bit counter mode when an interrupt on compare match with ITLCMP01 is not to be used as a capture trigger.

Remark n = 0, 1

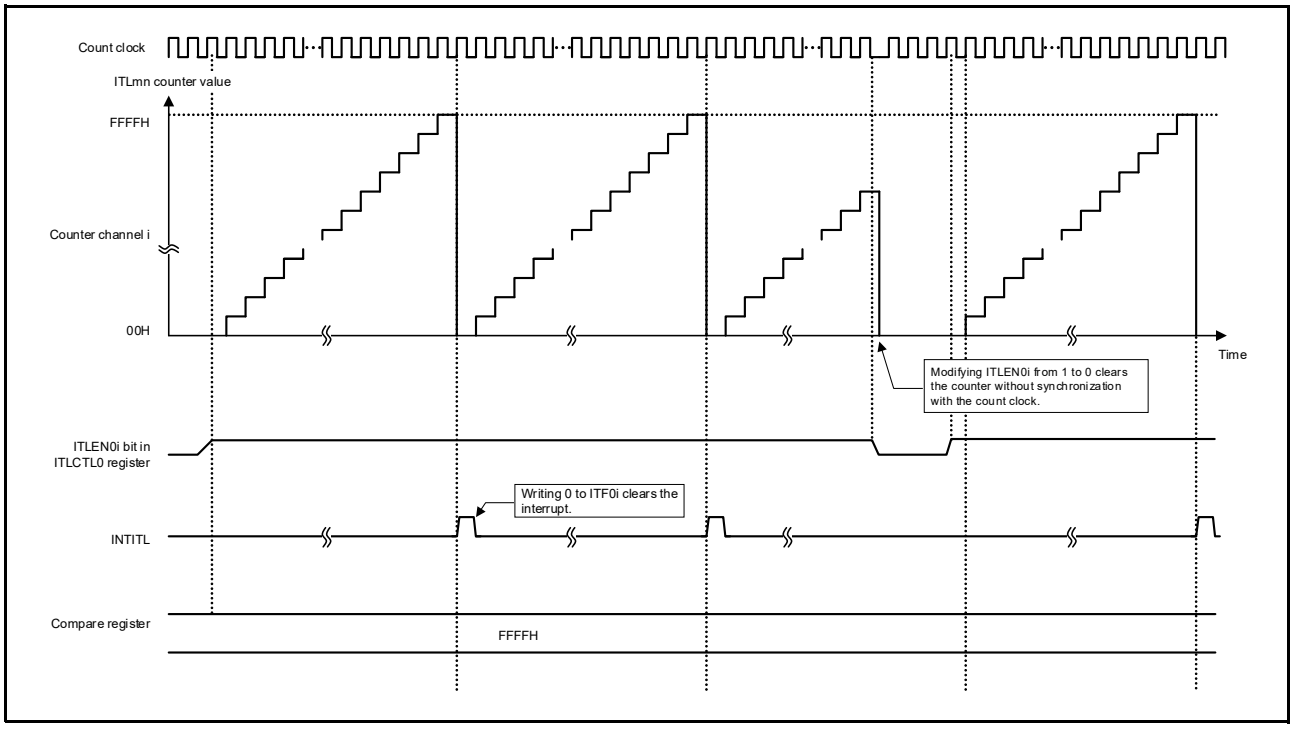
17.3.3 Timer operation

The ITL0mn counter counts up cycles of the counting clock specified in the interval timer frequency division registers 0 and 1 (ITLFDIV00 and ITLFDIV01). An interrupt request signal (INTITL) is generated on the counting of the next clock cycle after the value of the counter matches the comparison value. The interrupt request signal (INTITL) remains high until the value of the ITLS0 register becomes 00H.

While the interrupt request signal (INTITL) is high, neither the generation of a further interrupt request (INTITL) nor setting of the interrupt request flag (ITLIF) proceeds even if a compare match or capture completion is detected for an operating channel.

Clearing the ITLEN03 to ITLEN00 bits to 0 clears the counter value.

Figure 17 - 14 Example of Timer Operation



Remark mn = 00, 01, 12, 13; i = 0 to 3

17.3.4 Capture operation

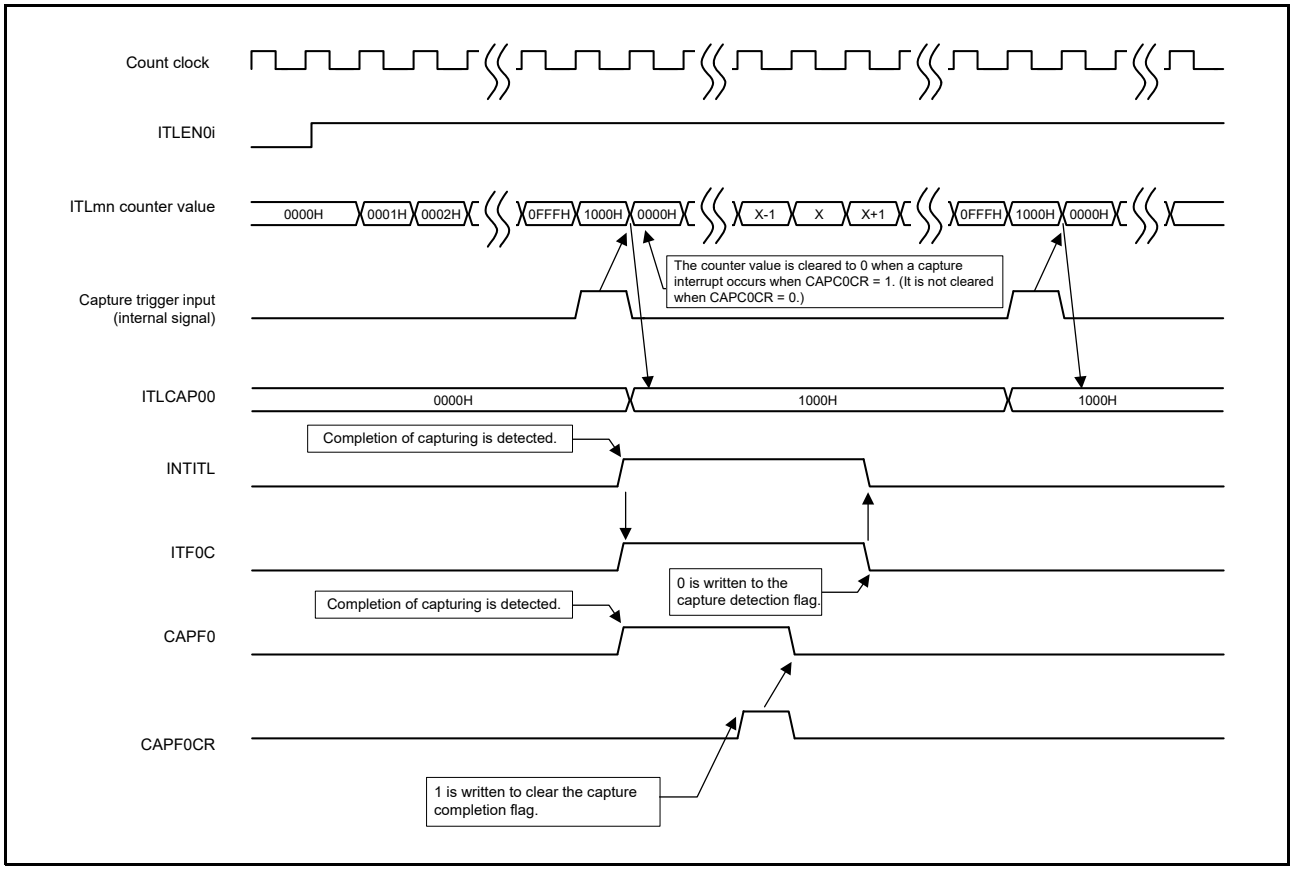
When the setting of the CAPEN0 bit in the interval timer capture control register 0 (ITLCC0) is 1, the values in the 16-bit counters (ITL000 and ITL001) are stored in interval timer capture register 00 (ITLCAP00) in response to the capture trigger specified in the ITLCC0 register.

The capture trigger is selectable from among the interrupt on compare match with the ITLCMP01 register, f_{SXP}, and a software trigger (setting the CAPR0 bit to 1). To use the interrupt on compare match with the ITLCMP01 register as the capture trigger, set interval timer clock select register 0 (ITLCSEL0) to select the clock for counting, and set interval timer compare register 01 (ITLCMP01) to specify the comparison value. When using f_{SXP} or a software trigger (setting the CAPR0 bit to 1) as a capture trigger, channels 2 and 3 can be used in 16-bit counter mode.

After a capture trigger is input and the counter value is stored in the interval timer capture register 00, the interrupt request signal (INTITL) is output, the capture completion flag (CAPF0) and capture detection flag (ITF0C) are set to 1, and the flag values are retained until they are explicitly cleared**Note**. The CAPF0 flag can be cleared by setting the CAPF0CR bit to 1. The ITF0C flag in the ITLS0 register can be cleared by writing 0 to it. Since capture operations operate with the counter clock, the interval at which the capture trigger is generated should be at least five cycles of the counter clock. If a capture trigger is generated again within two cycles of the counter clock after an earlier capture trigger was generated, the CAPF0 bit may not be set.

Note If the value of the ITLS0 register is other than 00H, neither interrupt operation nor setting of the interrupt request flag (ITLIF) will proceed even when the ITF0C flag is set to 1 because the interrupt request signal (INTITL) is kept at the high level.

Figure 17 - 15 Example of Capture Operation



Remark mn = 00, 01, 12, 13; i = 0 to 3

When the counter value matches the comparison value while the CAPC0CR bit in the ITLCC0 register is set to 1 (mode where the 16-bit counter (ITL000 + ITL001) is cleared following the completion of capture), counting of the next clock cycle clears the counter value. Note that the ITF00 flag is set when the 16-bit counter (ITL000 + ITL001) matches the comparison value before a capture trigger is input. The counter value is not cleared in this way if the CAPC0CR bit is set to 0 (mode where the 16-bit counter (ITL000 + ITL001) retains its value after the completion of capture). The ITF00 flag is set when the 16-bit counter (ITL000 + ITL001) matches the comparison value.

17.3.5 Interrupt

Table 17 - 7 shows the interrupt sources in 8-bit, 16-bit, and 32-bit counter modes.
The ITF0C, and ITF03 to ITF00 flags are interrupt status flags in the ITLS0 register. When any of the interrupt status flag is set, an interrupt request signal (INTITL) is output.

Table 17 - 7 Interrupt Sources in 8-bit, 16-bit, and 32-bit Counter Modes

Interrupt Source	Interrupt Condition in 8-bit Counter Mode	Interrupt Condition in 16-bit Counter Mode	Interrupt Condition in 32-bit Counter Mode
ITF00	Next rising edge of the counter clock after a compare match in channel 0	Next rising edge of the counter clock after a compare match in channels 0 and 1	Next rising edge of the counter clock after a compare match
ITF01	Next rising edge of the counter clock after a compare match in channel 1	Not generated	Not generated
ITF02	Next rising edge of the counter clock after a compare match in channel 2	Next rising edge of the counter clock after a compare match in channels 2 and 3	Not generated
ITF03	Next rising edge of the counter clock after a compare match in channel 3	Not generated	Not generated
ITF0C	Not generated; this is the case when the setting of ITLCC0 is 00H.	Timing of storing the counter value in the capture register after a capture trigger is input	Not generated; this is the case when the setting of ITLCC0 is 00H.

If the value of the ITLS0 register is other than 00H, the interrupt request signal (INTITL) is kept at the high level. Accordingly, neither the generation of a further interrupt request (INTITL) nor setting of the interrupt request flag (ITLIF) will proceed, even when a compare match or completion of capture is detected for an operating channel.
However, if the value of the ITLS0 register is not 00H after any bit in the ITLS0 register is set to 0 by an 8-bit memory manipulation instruction, a low-level pulse signal is output on the INTITL pin and the interrupt request flag (ITLIF) is set to 1. Accordingly, clearing a status flag in the ITLS0 register to 0 while ITLIF = 0 during vector interrupt processing or other processing enables the detection of an interrupt in response to another status bit having the setting 1. **Figure 17 - 16** shows the relationship between clearing of the detection flags and the interval detection interrupt signal.

The following describes the operation shown in **Figure 17 - 16**.
When a compare match in channel 1 is detected while the value of the ITLS0 register is 00H, the ITF01 flag is set to 1 and the interval detection interrupt signal (INTITL) is driven high. While the interval detection interrupt signal (INTITL) is kept at the high level, neither the generation of a further interrupt request (INTITL) nor setting of the interrupt request flag (ITLIF) will proceed, even when a compare match or completion of capture is detected for an operating channel.
Note that if another detection flag is set to 1 immediately before clearing the ITF0x (x = 0 to 3, C) flag to 0, the output of the INTITL pin temporarily goes to the low level after clearing of the given ITF0x flag, leading to setting of the interrupt request flag (ITLIF) to 1.

<1> The ITF01 flag is set to 1 in response to a compare match in channel 1 and the interval detection interrupt signal (INTITL) and interrupt request flag (ITLIF) are driven high. The interval detection interrupt processing is executed after clearing of the interrupt request flag (ITLIF) to 0.
<2> Check which detection flag in the ITLS0 register is set to 1 from within the interval detection interrupt processing. In the case shown in **Figure 17 - 16**, the ITF01 and ITF00 flags being set to 1 can be confirmed.
<3> Clear the ITF01 and ITF00 flags detected in step <2> by using an 8-bit memory manipulation instruction to write 00011100B to the ITLS0 register so that its value becomes 00H.**Note**
<4> The respective processing sequences in response to the ITF01 and ITF00 flags being set to 1 are then executed.**Note**

<5> The ITF01 flag is set to 1 in response to a further compare match in channel 1 and the interval detection interrupt signal (INTITL) and interrupt request flag (ITLIF) are driven high. The interval detection interrupt processing is executed after clearing of the interrupt request flag (ITLIF) to 0.

<6> Check which detection flag in the ITLS0 register is set to 1 from within the interval detection interrupt processing. In the case shown in **Figure 17 - 16**, the ITF01 flag being set to 1 can be confirmed.

<7> Clear the ITF01 flag detected in step <6> by using an 8-bit memory manipulation instruction to write 00011101B to the ITLS0 register so that its value becomes 00H. Though the ITF00 flag is also set to 1 in response to the compare match in channel 0 at this time, the ITF00 flag is not cleared because the processing for the flag does not proceed.

<8> As the ITF00 flag is set to 1 at the time the ITF01 flag is cleared to 0 in step <7>, the INTITL signal is temporarily driven low and the interrupt request flag (ITLIF) is set to 1. If the interrupt enable flag (IE) is not cleared to 0 at this time, this interrupt request is held pending.

<9> The processing in response to the ITF01 flag being set to 1 is then executed.

<10> As the interrupt request flag (ITLIF) is still set to 1 after return from the processing in response to the ITF01 flag being set to 1, clear the interrupt request flag (ITLIF) to 0, after which the interval detection interrupt processing that was held pending proceeds.

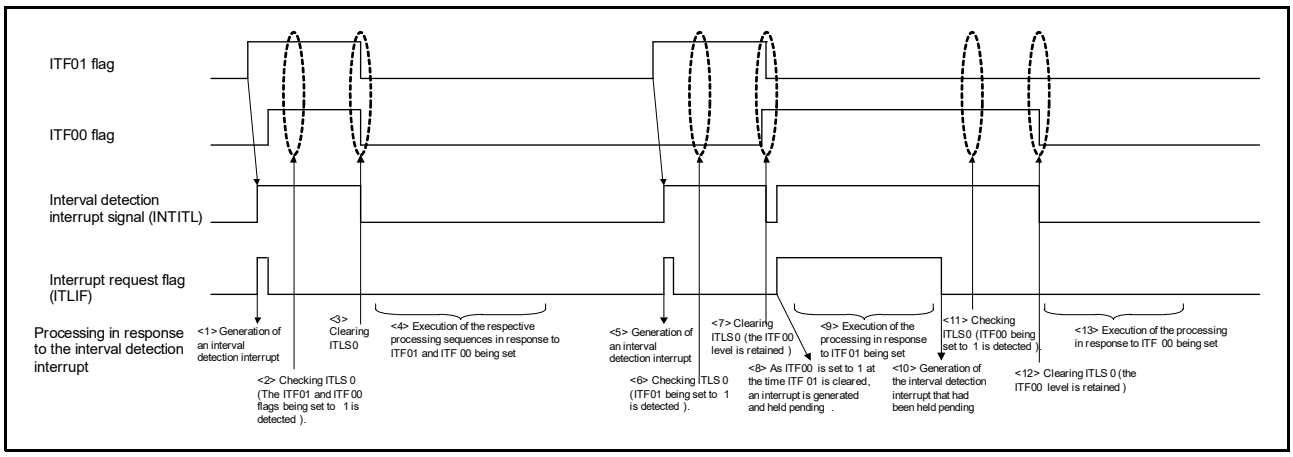
<11> Check which detection flag in the ITLS0 register is set to 1 from within the interval detection interrupt processing. In the case shown in **Figure 17 - 16**, the ITF00 flag being set to 1 can be confirmed.

<12> Clear the ITF00 flag detected in step <11> by using an 8-bit memory manipulation instruction to write 00011101B to the ITLS0 register so that its value becomes 00H.

<13> The processing in response to the ITF00 flag being set to 1 is then executed.

Note Missing an interrupt source can also be prevented by repeating the processing for clearing an interrupt source per flag.

Figure 17 - 16 Example of Clearing the Detected Flags



17.3.6 Interval timer setting procedures

The following shows the procedures for setting up the 32-bit interval timer.

Figure 17 - 17 Procedure for Starting the 32-bit Interval Timer

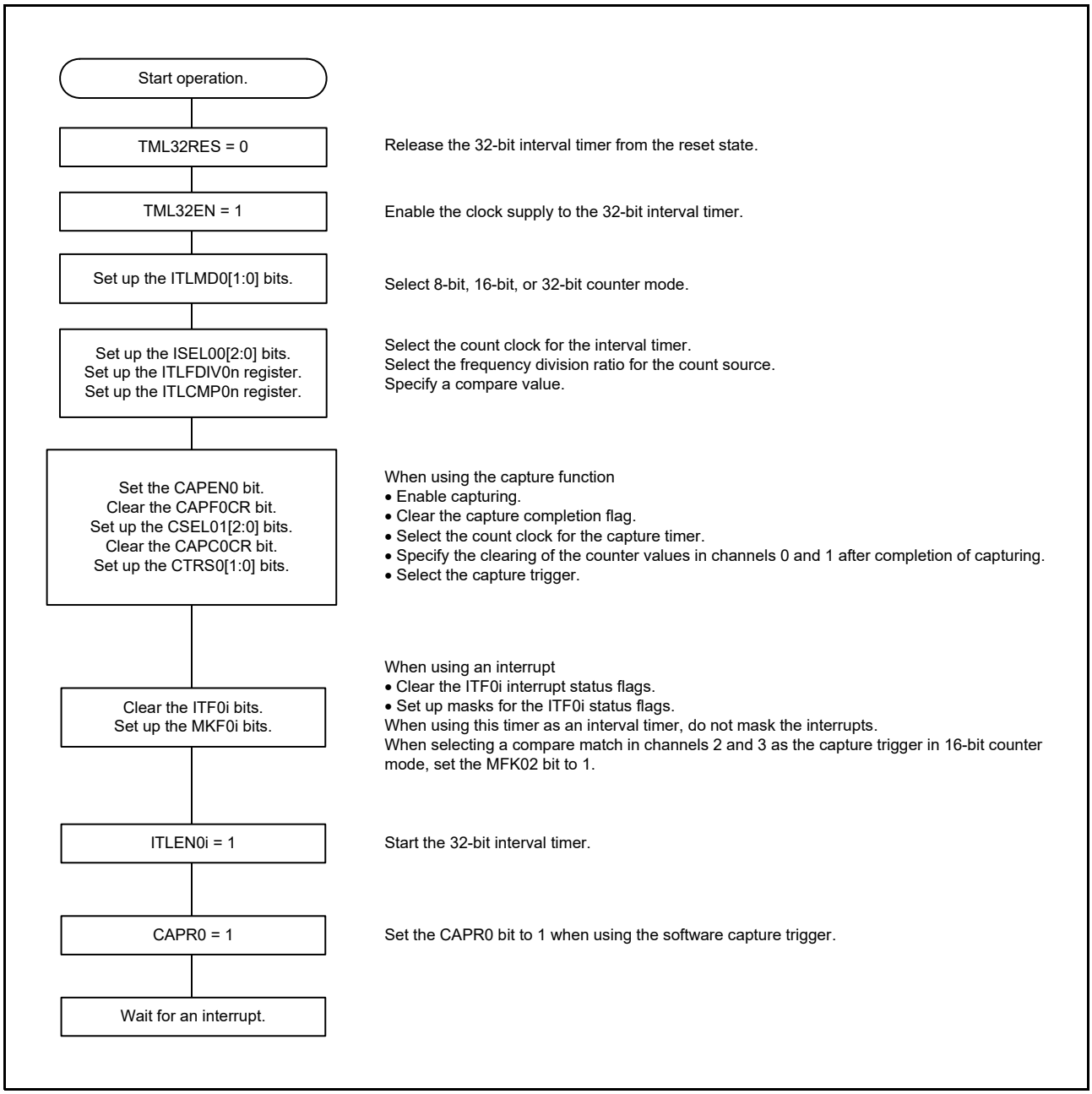


Figure 17 - 18 Procedure for Stopping the 32-bit Interval Timer

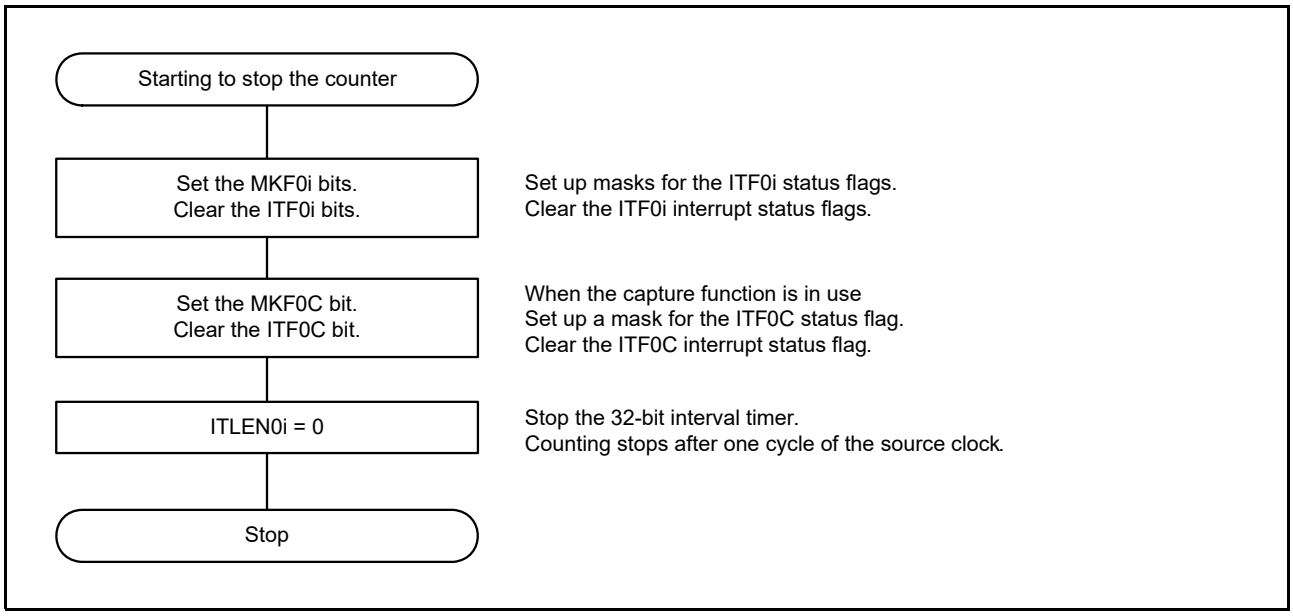


Figure 17 - 19 Procedure for Changing the Operating Mode of the 32-bit Interval Timer

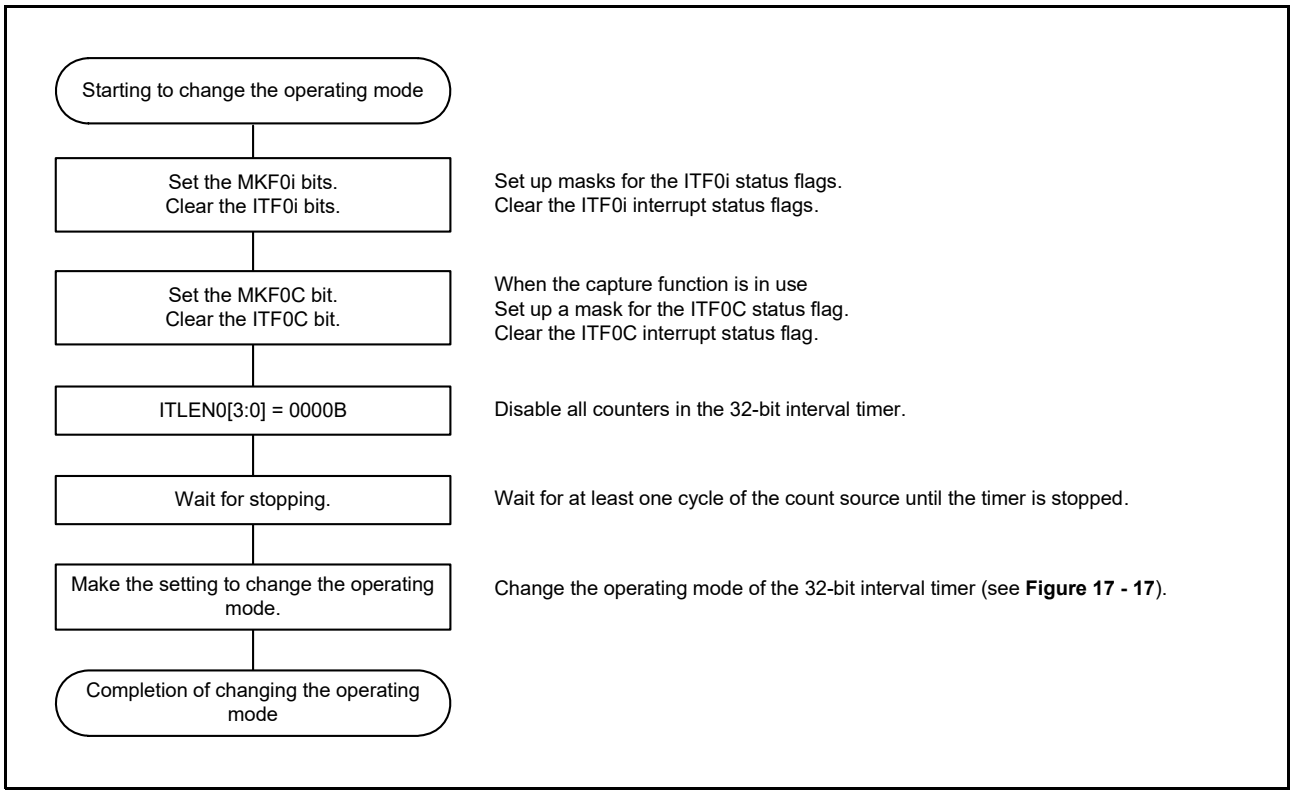
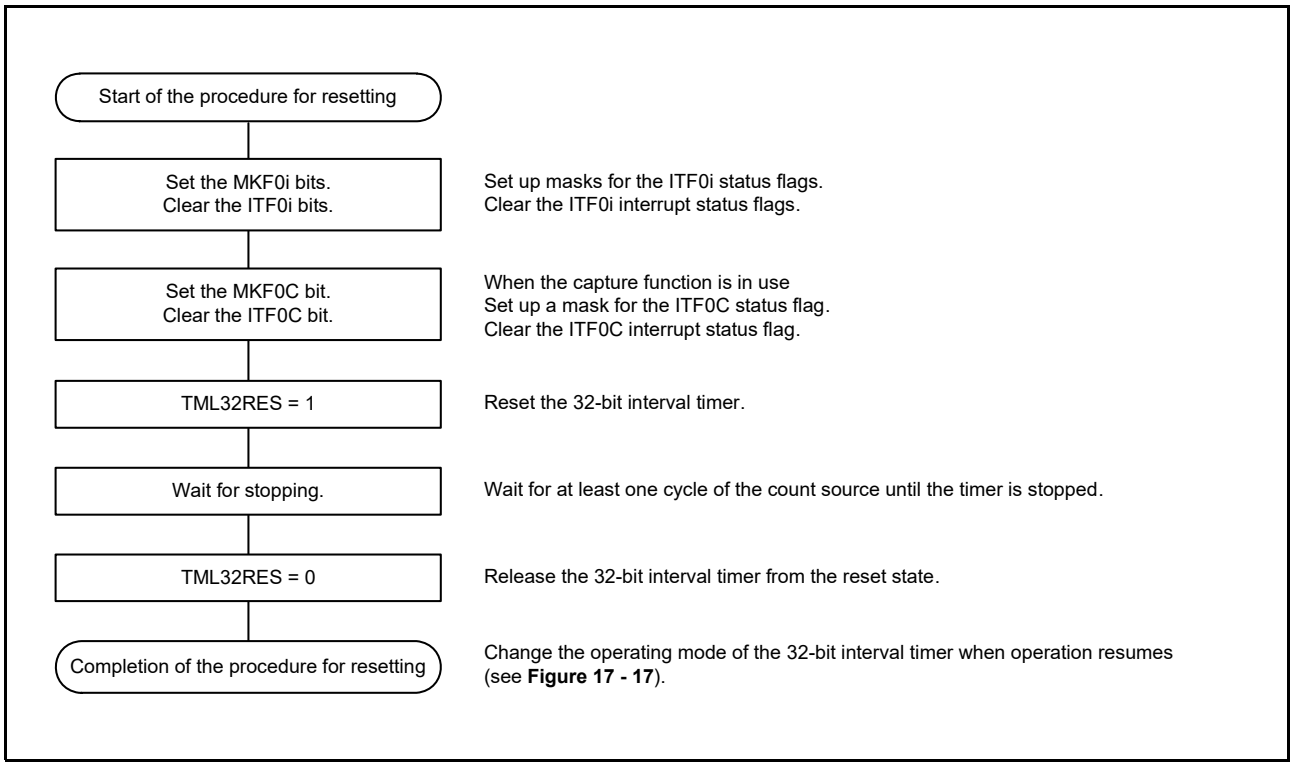


Figure 17 - 20 Procedure for Resetting the 32-bit Interval Timer



Section 18 Clock Output/Buzzer Output Controller (PCLBUZ)

The number of output pins for the clock output/buzzer output controller depends on the product.

Output Pins	20-, 24-, and 25-pin Products	30-, 32-, 40-, 44-, 48-, 52-, and 64-pin Products
PCLBUZ0	—	✓
PCLBUZ1	✓	✓

Caution Most of the following descriptions in this section use the 64-pin products as an example.

18.1 Functions of Clock Output/Buzzer Output Controller

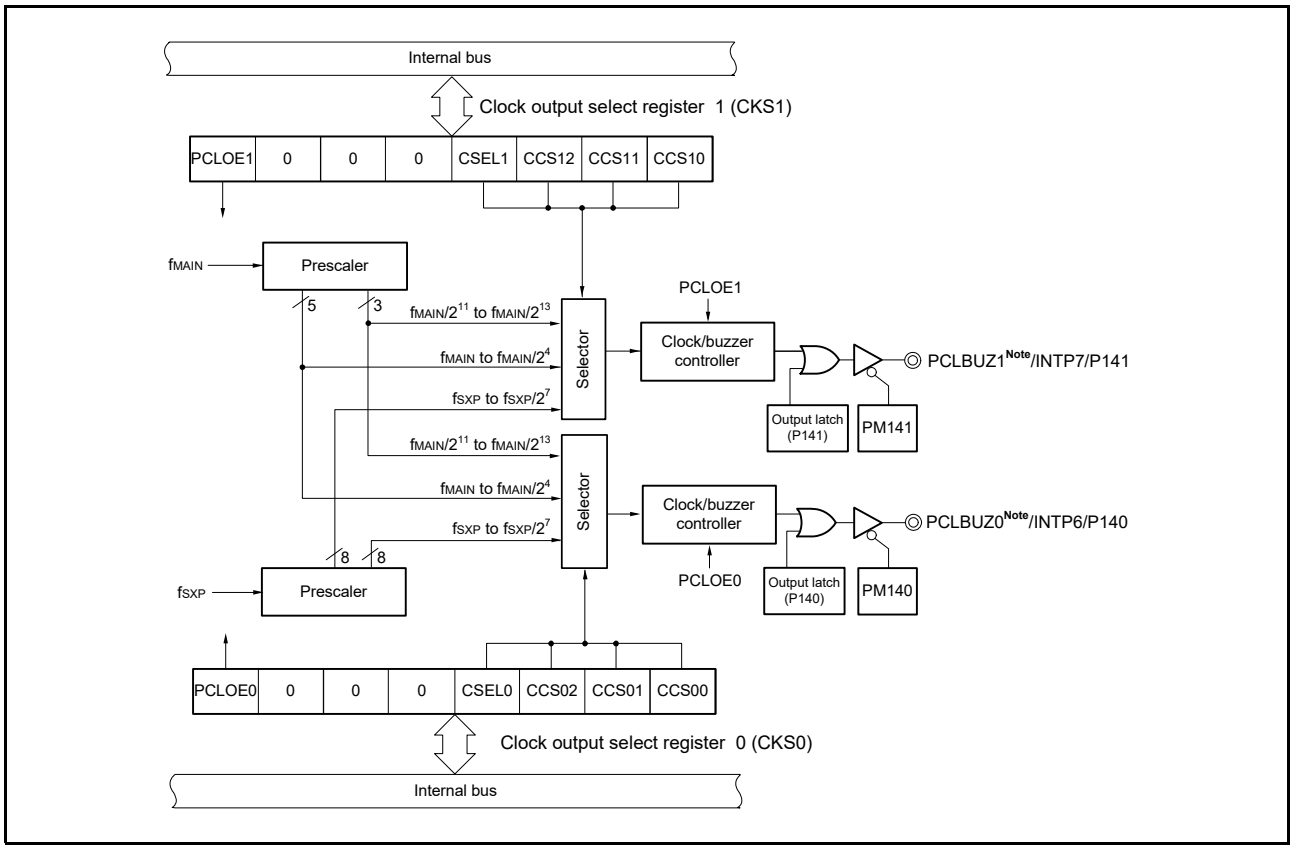
In clock output, the controller outputs a clock signal for supply to peripheral ICs. In buzzer output, the controller outputs a square wave at the buzzer frequency. This module has two output channels (PCLBUZn) and each of them can be specified to output a clock or buzzer signal. Switching of the output is handled by clock output select registers n (CKSn). **Figure 18 - 1** shows the block diagram of clock output/buzzer output controller.

Caution Output of the low-speed peripheral clock (f_{SXP}) from the PCLBUZn pin is not possible when the following conditions are both satisfied:

- **The setting of the RTCLPC bit in the subsystem clock supply mode control register (OSMC) is 1.**
- **Operation is in the HALT mode with the subsystem clock (f_{SUB}) selected as the CPU clock.**

Remark n = 0, 1

Figure 18 - 1 Block Diagram of Clock Output/Buzzer Output Controller



Note For output frequencies available from PCLBUZ0 and PCLBUZ1, refer to **43.4 AC Characteristics** or **44.4 AC Characteristics**.

Remark The clock output/buzzer output pins in above diagram show the information of 64-pin products with PIOR03 = 0 and PIOR04 = 0.

18.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 18 - 1 Configuration of Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	• Clock output select registers n (CKSn) (n = 0, 1) • Port mode registers xx (PMxx) (xx = 1, 3, 5, 14) • Port registers xx (Pxx) (xx = 1, 3, 5, 14) • Port output mode registers xx (POMxx) (xx = 1, 5) • Port mode control A registers xx (PMCAxx) (xx = 1)

18.3 Registers to Control the Clock Output/Buzzer Output Controller

The following registers are used to control the clock output/buzzer output controller.

- Clock output select registers n (CKSn) (n = 0, 1)
- Port mode registers xx (PMxx) (xx = 1, 3, 5, 14)
- Port registers xx (Pxx) (xx = 1, 3, 5, 14)
- Port output mode registers xx (POMxx) (xx = 1, 5)
- Port mode control A registers xx (PMCAxx) (xx = 1)

18.3.1 Clock output select registers n (CKSn) (n = 0, 1)

The CKSn registers enable or disable the output from the clock or buzzer frequency output pin (PCLBUZn), and set the output clock. Use the CKSn register to select the clock to be output from the PCLBUZn pin. The CKSn registers are set by a 1-bit or 8-bit memory manipulation instruction. The value of each CKSn register following a reset is 00H.

Figure 18 - 2 Format of Clock Output Select Register n (CKSn)

Address: FFFA5H (CKS0), FFFA6H (CKS1)
After reset: 00H
R/W: R/W

Symbol	<7>	6	5	4	3	2	1	0
CKSn	PCLOEn	0	0	0	CSELn	CCSn2	CCSn1	CCSn0

PCLOEn	PCLBUZn pin output enable/disable specification
0	Output disabled
1	Output enabled

CSELn	CCSn2	CCSn1	CCSn0		PCLBUZn pin output clock selection				
					f_{MAIN} = 5 MHz	f_{MAIN} = 10 MHz	f_{MAIN} = 20 MHz	f_{MAIN} = 32 MHz	f_{MAIN} = 48 MHz
0	0	0	0	f_{MAIN}	5 MHz**Note**	10 MHz**Note**	Setting prohibited	Setting prohibited	Setting prohibited
0	0	0	1	$f_{MAIN}/2$	2.5 MHz	5 MHz**Note**	10 MHz**Note**	16 MHz**Note**	Setting prohibited
0	0	1	0	$f_{MAIN}/2^2$	1.25 MHz	2.5 MHz	5 MHz**Note**	8 MHz**Note**	12 MHz**Note**
0	0	1	1	$f_{MAIN}/2^3$	625 kHz	1.25 MHz	2.5 MHz	4 MHz	6 MHz
0	1	0	0	$f_{MAIN}/2^4$	312.5 kHz	625 kHz	1.25 MHz	2 MHz	3 MHz
0	1	0	1	$f_{MAIN}/2^{11}$	2.44 kHz	4.88 kHz	9.77 kHz	15.63 kHz	15.63 kHz
0	1	1	0	$f_{MAIN}/2^{12}$	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz	11.72 kHz
0	1	1	1	$f_{MAIN}/2^{13}$	610 Hz	1.22 kHz	2.44 kHz	3.91 kHz	5.86 kHz
1	0	0	0	f_{SXP}	32.768 kHz				
1	0	0	1	$f_{SXP}/2$	16.384 kHz				
1	0	1	0	$f_{SXP}/2^2$	8.192 kHz				
1	0	1	1	$f_{SXP}/2^3$	4.096 kHz				
1	1	0	0	$f_{SXP}/2^4$	2.048 kHz				
1	1	0	1	$f_{SXP}/2^5$	1.024 kHz				
1	1	1	0	$f_{SXP}/2^6$	512 Hz				
1	1	1	1	$f_{SXP}/2^7$	256 Hz				

Note The selectable output clock frequency depends on the power supply voltage (V_{DD}). See **43.4 AC Characteristics** or **44.4 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 (CSELn = 0), set PCLOEn = 0 before executing the STOP instruction. When the subsystem clock is selected (CSELn = 1), PCLOEn = 1 can be set because the clock can be output while the RTCLPC bit of the subsystem clock supply mode control (OSMC) register is set to 0 and moreover while STOP mode is set.

(Caution and Remarks are listed on the next page.)

Caution 3. It is not possible to output the low-speed peripheral clock (f_{SXP}) from the PCLBUZn pin while the RTCLPC bit of the subsystem clock supply mode control register (OSMC) is set to 1 and moreover while HALT mode is set with the subsystem clock (f_{SUB}) selected as CPU clock.

Remark 1. n = 0, 1

Remark 2. f_{MAIN}: Main system clock frequency
f_{SUB}: Subsystem clock frequency
f_{SXP}: Low-speed peripheral clock frequency

18.3.2 Registers for controlling the port functions multiplexed with the clock or buzzer outputs

Set the following registers to control the port functions multiplexed with the clock or buzzer outputs.

- Port mode registers xx (PMxx)
- Port registers xx (Pxx)
- Port output mode registers xx (POMxx)
- Port mode control A registers xx (PMCAxx)

For details, see the following sections.

- **7.3.1 Port mode registers xx (PMxx)**
- **7.3.2 Port registers xx (Pxx)**
- **7.3.5 Port output mode registers xx (POMxx)**
- **7.3.7 Port mode control A registers xx (PMCAxx)**

When the pins multiplexed with PCLBUZ0 and PCLBUZ1 are to be used for clock or buzzer outputs, set the bits of the following registers corresponding to the given multiplexed port pins to 0.

- Port mode register xx (PMxx)
- Port register xx (Pxx)
- Port output mode register xx (POMxx)
- Port mode control A register xx (PMCAxx)

Example: When P140/PCLBUZ0/INTP6 is to be used for clock or buzzer output
Set the PM140 bit of port mode register 14 to 0.
Set the P140 bit of port register 14 to 0.

Remark xx = 1, 3, 5, 14
Note that POM14, PMCA3, and PMCA5 are not present in the RL78/G24 products.

18.4 Operations of the Clock Output/Buzzer Output Controller

This module has two output channels (PCLBUZn) and each of them can be specified to output a clock or buzzer signal. The PCLBUZ0 and PCLBUZ1 pins respectively output the clock specified by clock output select registers 0 and 1 (CKS0 and CKS1).

18.4.1 Operation of output pins

Follow the steps below to enable output from a PCLBUZn pin.

<1> Set the corresponding bits of the following registers to 0 to select the port pin for use as a PCLBUZn pin.
- Port mode register xx (PMxx)
- Port register xx (Pxx)

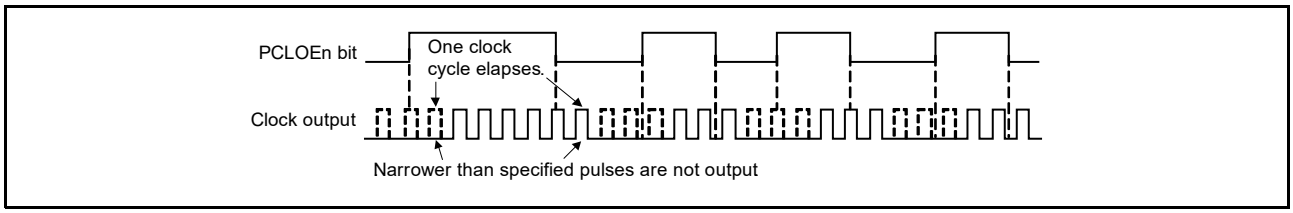
<2> Select the output frequency with bits 3 to 0 (CSELn, CCSn[2:0]) of the clock output select register (CKSn) for a PCLBUZn pin (output is still disabled).

<3> Set bit 7 (PCLOEn) of the CKSn register to 1 to enable clock/buzzer output.

Remark 1. The controller used for the clock output starts or stops output one clock cycle after that in which the PCLOEn bit was set to enable or disable the output, respectively. Pulses that are narrower than the specified width are not output at those times. **Figure 18 - 3** shows the relationship between the setting of the PCLOEn bit and the timing of clock output.

Remark 2. n = 0, 1

Figure 18 - 3 Timing of the Clock Output from a PCLBUZn Pin



18.5 Point for Caution When the Clock Output/Buzzer Output Controller Is to Be Used

When the main system clock is selected for the PCLBUZn output (CSELn = 0), if STOP mode is entered within 1.5 clock cycles of output from a PCLBUZn pin having been disabled (PCLOEn = 0), the width of a PCLBUZn pulse being output at that time becomes shorter.

Section 19 Watchdog Timer (WDT)

19.1 Functions of Watchdog Timer

The counting operation of the watchdog timer is set by the user option byte (000C0H).
The watchdog timer operates with the low-speed on-chip oscillator clock (f_{IL}) divided by 2 (1/2 f_{IL}).
The watchdog timer is used to detect program malfunctions. If a malfunction is detected, an internal reset signal is generated. Any among the following cases is considered a program malfunction.

- The watchdog timer counter overflows
- A 1-bit manipulation instruction is used to write to the watchdog timer enable register (WDTE).
- A value other than ACH is written to the WDTE register.
- Writing to the WDTE register proceeds while the window is closed.

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 **Section 32 Reset Function**.
When 75% of the overflow time + 1/4 f_{IL} is reached, an interval interrupt can be generated.

19.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 19 - 1 Configuration of Watchdog Timer

Item	Configuration
Counter	Internal counter (17 bits)
Control register	Watchdog timer enable register (WDTE)

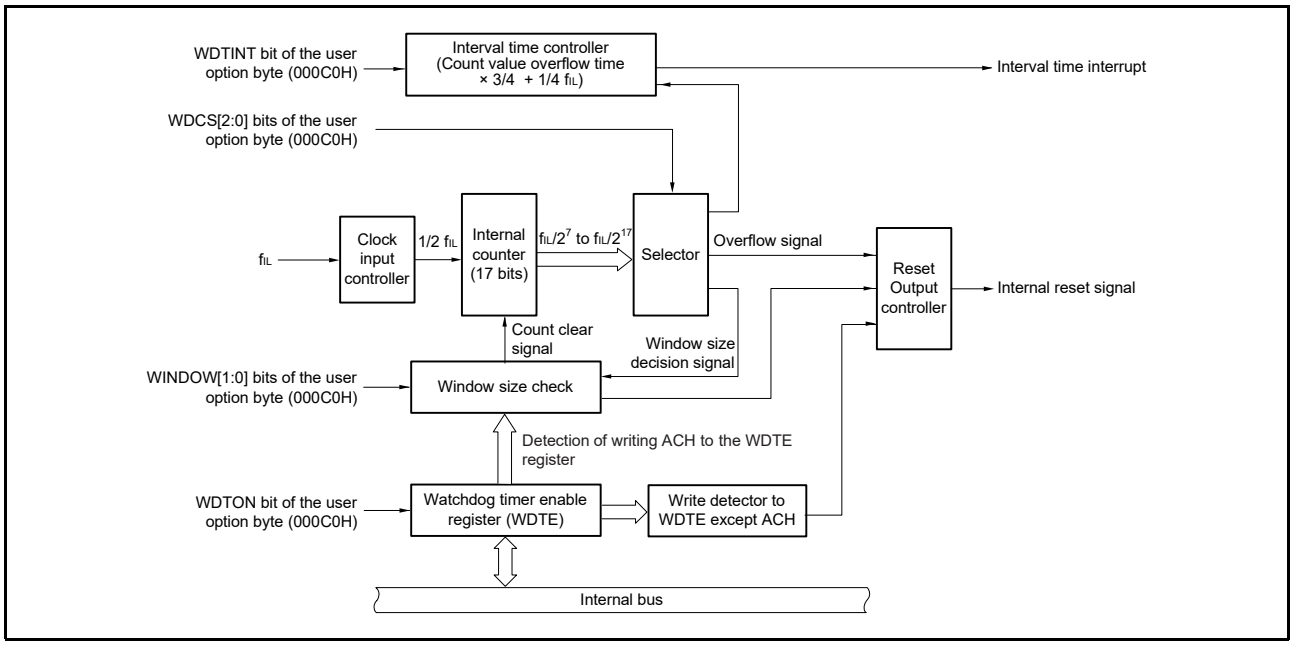
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the user option byte.

Table 19 - 2 Setting of the User Option Byte and Watchdog Timer

Setting of Watchdog Timer	User Option Byte (000C0H)
Watchdog timer interval interrupt	Bit 7 (WDTINT)
Window open period	Bits 6 and 5 (WINDOW[1:0])
Controlling counter operation of watchdog timer	Bit 4 (WDTON)
Overflow time of watchdog timer	Bits 3 to 1 (WDCS[2:0])
Controlling counter operation of watchdog timer (in HALT/STOP mode)	Bit 0 (WDSTBYON)

Remark For the user option byte, see **Section 38 Option Bytes**.

Figure 19 - 1 Block Diagram of Watchdog Timer



Remark fIL: Low-speed on-chip oscillator clock

19.3 Register to Control the Watchdog Timer

The following register is used to control the watchdog timer.

- Watchdog timer enable register (WDTE)

19.3.1 Watchdog timer enable register (WDTE)

Writing ACH to the WDTE register clears and then restarts counting by the watchdog timer counter. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 9AH or 1AH**Note**.

Figure 19 - 2 Format of Watchdog Timer Enable Register (WDTE)

Address: FFFABH
After reset: 9AH/1AH**Note**
R/W: 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 user option byte (000C0H). To operate watchdog timer, set the WDTON bit to 1.

WDTON bit setting value	WDTE register reset value
0 (Watchdog timer counting disabled)	1AH
1 (Watchdog timer counting 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)).

19.4 Operation of Watchdog Timer

19.4.1 Controlling operation of watchdog timer

1. When the watchdog timer is used, its operation is specified by the user option byte (000C0H).
   - Enable counting operation of the watchdog timer by setting bit 4 (WDTON) of the user option byte (000C0H) to 1 (the counter starts operating after release from the reset state) (for details, see **Section 38 Option Bytes**).

WDTON	Watchdog Timer Counter
0	Counter operation disabled (counting stopped after release from the reset state)
1	Counter operation enabled (counting started after release from the reset state)

   - Set the time at which the counter is to overflow by using bits 3 to 1 (WDCS[2:0]) of the user option byte (000C0H). For details, see **19.4.2 Setting overflow time of watchdog timer** and **Section 38 Option Bytes**.
   - Set a window open period by using bits 6 and 5 (WINDOW[1:0]) of the user option byte (000C0H) (for details, see **19.4.3 Setting window open period of watchdog timer** and **Section 38 Option Bytes**).
2. After release from the reset state, 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 user option byte, the watchdog timer is cleared and starts counting again.
4. After that, writing to the WDTE register the second and subsequent times must proceed while the window is open. If the WDTE register is written during a window close period, an internal reset signal is generated.
5. If the time at which an overflow is to occur elapses without ACH having been written to the WDTE register, an internal reset signal is generated.
   An internal reset signal is also 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 4 cycles of the clock at f_{IL} may occur before the watchdog timer is cleared.

Caution 3. The watchdog timer can be cleared immediately before the counter value overflows.

Caution 4. The operation of the watchdog timer in the HALT, STOP, and SNOOZE modes differs as follows depending on the set value of bit 0 (WDSTBYON) of the user 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.

19.4.2 Setting overflow time of watchdog timer

Set the overflow time of the watchdog timer by using bits 3 to 1 (WDCS[2:0]) of the user 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 19 - 3 Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer (f_{IL} = 37.683 kHz (max.))
0	0	0	$2^7/f_{IL}$ (3.39 ms)
0	0	1	$2^8/f_{IL}$ (6.79 ms)
0	1	0	$2^9/f_{IL}$ (13.58 ms)
0	1	1	$2^{10}/f_{IL}$ (27.17 ms)
1	0	0	$2^{12}/f_{IL}$ (108.69 ms)
1	0	1	$2^{14}/f_{IL}$ (434.78 ms)**Note**
1	1	0	$2^{15}/f_{IL}$ (869.56 ms)**Note**
1	1	1	$2^{17}/f_{IL}$ (3478.26 ms)**Note**

Note Using the watchdog timer under the following conditions may lead to the generation of an interval interrupt (INTWDTI) after one cycle of the watchdog timer clock once the watchdog timer counter has been cleared.

- the watchdog timer interval interrupt is in use, and
- ACH is written to the WDTE register (FFFABH) when the watchdog timer counter has reached or exceeded 75% of the overflow time.

This interrupt can be masked by clearing the watchdog timer counter through steps 1 to 5 below.

<1> Set the WDTIMK bit of the interrupt mask flag register 0 (MK0L) to 1 before clearing the watchdog timer counter.
<2> Clear the watchdog timer counter.
<3> Wait for at least 80 μs.
<4> Clear the WDTIIF bit of the interrupt request flag register 0 (IF0L) to 0.
<5> Clear the WDTIMK bit of the interrupt mask flag register 0 (MK0L) to 0.

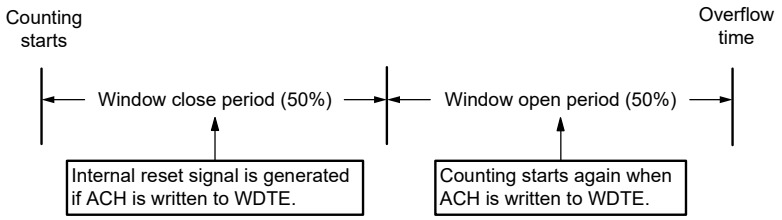
Remark f_{IL}: Low-speed on-chip oscillator clock frequency

19.4.3 Setting window open period of watchdog timer

Set the window open period of the watchdog timer by using bits 6 and 5 (WINDOW[1:0]) of the user 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 19 - 4 Setting Window Open Period of Watchdog Timer

WINDOW1	WINDOW0	Window Open Period of Watchdog Timer
0	1	50%
1	1	100%
Others		Setting prohibited

Remark If the overflow time is set to $2^{10}/f_{IL}$, the times over which the window is open and closed are as follows.

	Setting of Window Open Period	
	50%	100%
Window close time	0 to 18.38 ms	None
Window open time	18.38 to 27.17 ms	0 to 27.17 ms

<When window open period is 50%>

- Overflow time:
  $2^{10}/f_{IL}$ (max.) = 2^{10}/37.683 kHz = 27.17 ms
- Window close time:
  0 to $2^{10}/f_{IL}$ (min.) × (1 – 0.5) = 0 to 2^{10}/27.852 kHz × 0.5 = 0 to 18.38 ms
- Window open time:
  $2^{10}/f_{IL}$ (min.) × (1 – 0.5) to $2^{10}/f_{IL}$ (max.) = 2^{10}/27.853 kHz × 0.5 to 2^{10}/37.683 kHz = 18.38 to 27.17 ms

19.4.4 Setting watchdog timer interval interrupt

Depending on the setting of bit 7 (WDTINT) of the user option byte (000C0H), an interval interrupt (INTWDTI) can be generated when 75% of the overflow time + 1/4 f_{IL} is reached.

Table 19 - 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% of the overflow time + 1/4 f_{IL} 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.

Section 20 A/D Converter (ADC)

The number of analog input channels of the A/D converter differs, depending on the product.

	20-pin	24-pin	25- to 32-pin	40-pin	44- to 48-pin	52- to 64-pin
Number of the analog input channels	12 (ANI0 to ANI3, ANI19 to ANI24, ANI29, ANI30)	13 (ANI0 to ANI3, ANI18 to ANI23, ANI26, ANI29, ANI30)	16 (ANI0 to ANI3, ANI18 to ANI27, ANI29, ANI30)	19 (ANI0 to ANI6, ANI18 to ANI27, ANI29, ANI30)	21 (ANI0 to ANI7, ANI18 to ANI30)	23 (ANI0 to ANI7, ANI16 to ANI30)

20.1 Function of A/D Converter

The A/D converter is used to convert analog input signals into digital values. 12-bit, 10-bit, or 8-bit resolution can be selected by the ADTYP[1:0] bits of A/D converter mode register 2 (ADM2). The A/D converter has the following functions.

- 12-bit/10-bit/8-bit resolution A/D conversion
  12-bit, 10-bit, or 8-bit resolution A/D conversion for up to four analog input channels selected from among ANI0 to ANI7 and ANI16 to ANI30 repeatedly proceeds. Interrupt request signals (INTAD0 to INTAD3)**Note** are generated on completion of each A/D conversion operation.
- Advanced mode
  Setting the ADVMOD bit in the ADM3 register to 1 enables the advanced mode. In this mode, the ADC can operate with the 48-MHz operating clock.
  In advanced mode, conversion of signals on four channels is possible, so conversion of up to four analog input signals is specifiable. Sequential conversion of up to four signals or simultaneous sampling of the signals on up to three input channels is also possible. Moreover, whether or not to generate an interrupt request (INTAD0 to INTAD3) after conversion is also specifiable for each conversion channel. If an interrupt request is generated, the conversion interrupt status register (ADINTST) can be used to confirm which conversion channel was responsible for its generation. In this document, unless “advanced” appears in a mode name, the advanced mode is disabled in the operation being described.
  Example: “Software trigger mode” indicates the software trigger mode with the advanced mode disabled. On the other hand, “Advanced software trigger mode” indicates the software trigger mode with the advanced mode enabled.

Note When the advanced mode is disabled, INTAD0 is only generated. On the other hand, when the advanced mode is enabled, an interrupt request corresponding to a setting for A/D conversion from among INTAD0 to INTAD3 is generated. Notation for the advanced mode described above applies hereinafter.

Remark The ADC cannot operate with the 48-MHz clock when the advanced mode is disabled. Set the supply clock to a frequency of no greater than 32 MHz.

Table 20 - 1 shows a comparison of functions with the advanced mode is enabled and disabled.

Table 20 - 1 Comparison of Functions with the Advanced Mode Is Enabled and Disabled (1/2)

	Advanced Mode Enabled	Advanced Mode Disabled (Conventional Mode)
Maximum operating clock frequency (f_{CLK})	48 MHz	32 MHz
Resolution	8 bits, 10 bits, or 12 bits	8 bits, 10 bits, or 12 bits
Number of channels for simultaneous sampling	1 to 3	1
Input channel	ANI0 to ANI7, ANI16 to ANI30, PGA output, temperature sensor output voltage, internal reference voltage	ANI0 to ANI7, ANI16 to ANI30, temperature sensor output voltage, internal reference voltage
Trigger mode	Software trigger (no-wait mode) A/D conversion is started by setting ADM3.ADTRSWT to 1.	Software trigger no-wait mode A/D conversion is started by setting the ADCE bit to 1 by software, and then setting ADCS to 1 after the A/D power supply stabilization wait time has passed.
		Hardware trigger wait mode The power to the A/D converter is turned on by detecting a hardware trigger while the power is off and the A/D converter is in the conversion standby state, and conversion is then started automatically after the A/D power supply stabilization wait time has passed.
	Hardware trigger (no-wait mode) A/D conversion is started by detecting a hardware trigger specified for each conversion channel.	Hardware trigger no-wait mode Conversion is started by detecting a hardware trigger.
		Hardware trigger wait mode The power to the A/D converter is turned on by detecting a hardware trigger while the power is off and the A/D converter is in the conversion standby state, and conversion is then started automatically after the A/D power supply stabilization wait time has passed. When using the SNOOZE mode function, specify the hardware trigger wait mode.
Conversion trigger	• Channel 01 timer array unit counting or capture end interrupt signal (INTTM01) • Realtime clock interrupt signal (INTRTC) • 32-bit interval timer channel 0 interrupt signal (ELCITL0) • Event input from the ELC • 16-bit timer KB30 A/D trigger signal • 16-bit timer KB31 A/D trigger signal • 16-bit timer KB32 A/D trigger signal • Timer RD2 A/D conversion trigger 0 • Timer RD2 A/D conversion trigger 1 • Software trigger	• Channel 01 timer array unit counting or capture end interrupt signal (INTTM01) • Realtime clock interrupt signal (INTRTC) • 32-bit interval timer channel 0 interrupt signal (ELCITL0) • Event input from the ELC • Software trigger
Channel selection mode	Up to four analog input channels can be assigned for the corresponding conversion triggers. When a trigger is generated, A/D conversion proceeds on the corresponding analog input channel.	Select mode A/D conversion proceeds on the analog input of one selected channel.
		Scan mode A/D conversion proceeds on the analog inputs to four channels in order. Four consecutive channels can be selected from among ANI0 to ANI7 as analog input channels.

Table 20 - 1 Comparison of Functions with the Advanced Mode Is Enabled and Disabled (2/2)

	Advanced Mode Enabled	Advanced Mode Disabled (Conventional Mode)
Conversion	A/D conversion proceeds once for each analog input channel selected as a conversion channel.	One-shot conversion mode A/D conversion proceeds once for each input channel selected as a conversion channel.
		Sequential conversion mode A/D conversion sequentially proceeds for each channel selected as a conversion channel until the conversion is stopped by software.

Operating Voltage Mode**Note 1**	Number of Sampling Clock		Note
	Advanced Mode Enabled	Advanced Mode Disabled	
Normal 1	20 f_{AD} (input channels: ANI0 to ANI7)**Note 2**	43 f_{AD}	Set the number of sampling clock cycles so that the sampling capacitor is sufficiently charged according to the output impedance of the analog input source.
	27 f_{AD} (input channels: ANI0 to ANI7, ANI16 to ANI30)**Note 3**		
	32 f_{AD} (PGA gain: ×4 to ×16)		
	63 f_{AD} (PGA gain: ×32)		
Normal 2	240 f_{AD}	160 f_{AD}	
Low voltage 1	53 f_{AD}	53 f_{AD}	
Low voltage 2	80 f_{AD}	80 f_{AD}	

Note 1. The operation mode that can be selected differs depending on the analog input channel, V_{DD} voltage, AV_{REFP} voltage, trigger mode, and f_{CLK}. See **Table 20 - 6 Selection of A/D Conversion Time** for details.

Note 2. This applies when ADSPMOD = 01H.

Note 3. This applies when ADSPMOD = 00H.

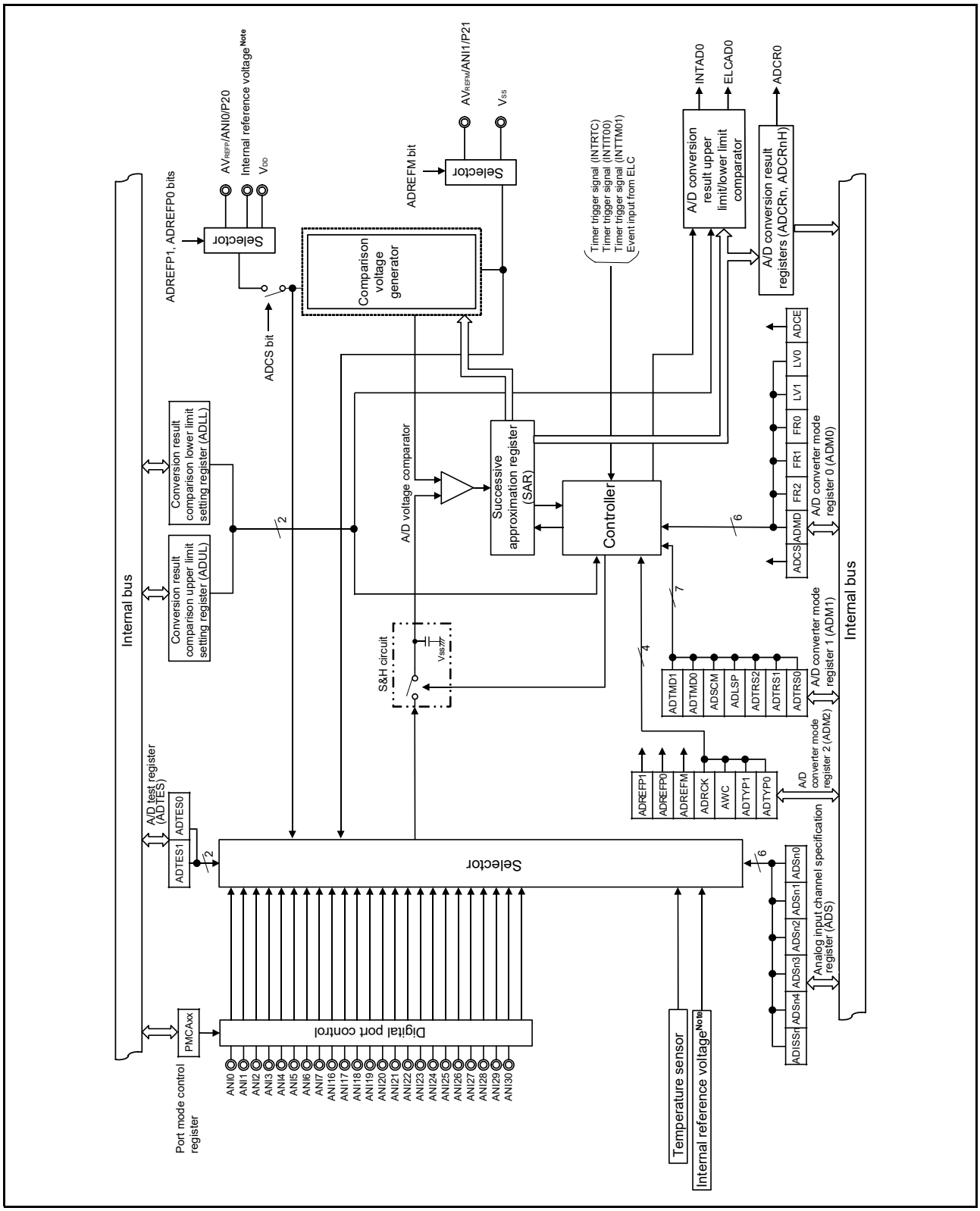
Table 20 - 2 shows the selectable conversion targets in each operating voltage mode in the advanced mode.

Table 20 - 2 Selectable Conversion Targets in Each Operating Voltage Mode in the Advanced Mode

Operating Voltage Mode	Conversion Targets				
	ANI0 to ANI7 (ADSPMOD = 01H)	ANI0 to ANI30 (ADSPMOD = 00H)	PGA Output	Temperature Sensor/Internal Reference Voltage	Simultaneous Sampling**Note**
Normal 1	✓	✓	✓	—	✓
Normal 2	—	✓	—	✓	—
Low voltage 1	—	✓	✓	—	—
Low voltage 2	—	✓	—	✓	—

Note When the simultaneous sampling is to proceed, a single round of A/D conversion should take no longer than 3.3 µs.

Figure 20 - 1 Block Diagram of the A/D Converter with the Advanced Mode Disabled

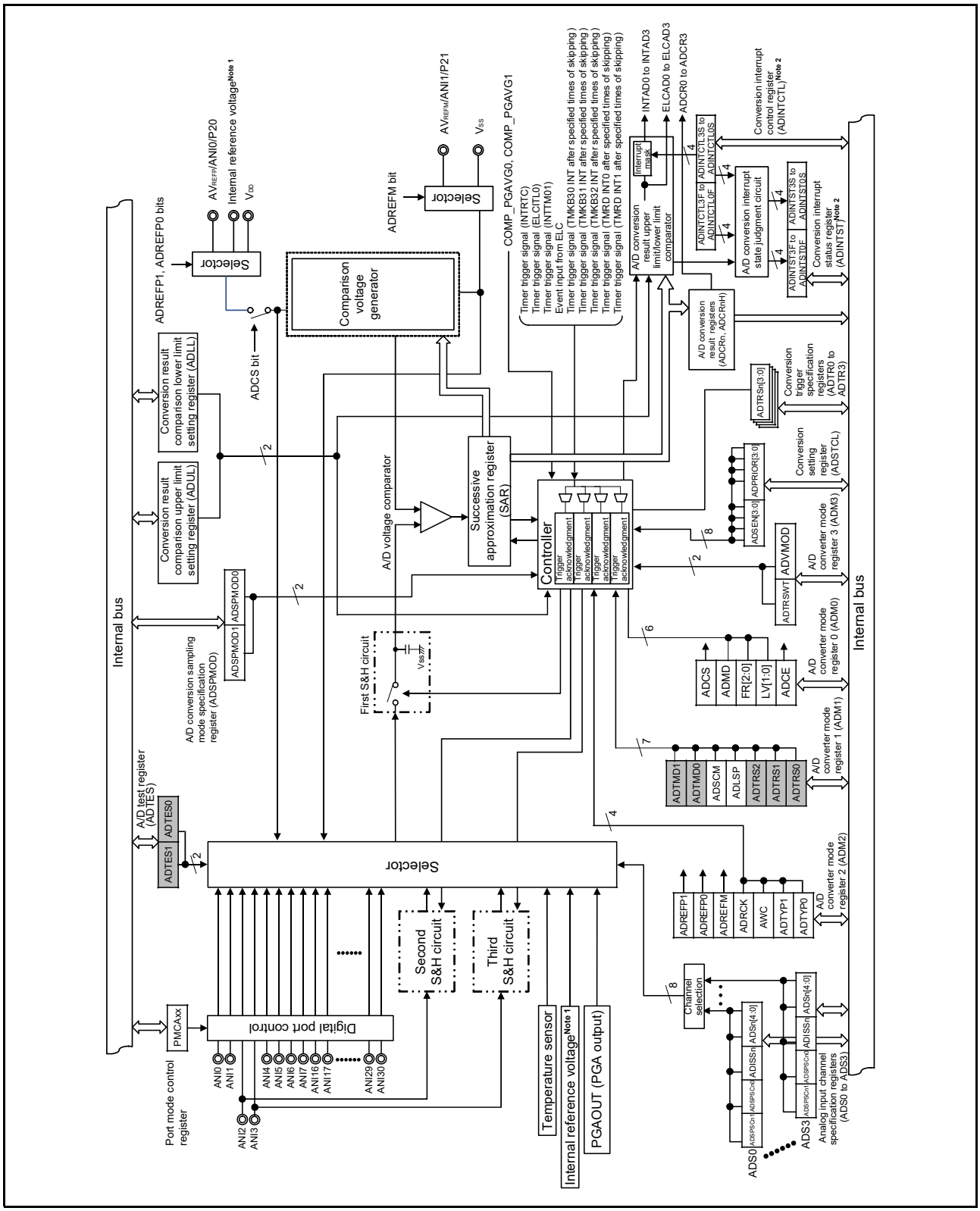


Note For details on the internal reference voltage, see **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** and **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.

Remark 1. Analog input pins in this figure are for a 64-pin product.

Remark 2. n = 0 to 3

Figure 20 - 1 Block Diagram of the A/D Converter with the Advanced Mode Enabled



20.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

1. ANI0 to ANI7 and ANI16 to ANI30 pins
   These are the analog input pins of the 23 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. Sample & hold circuit (S&H 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.
3. 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 AV_{REF}) 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 AV_{REF}), the MSB of the SAR register is reset.
   After that, bit 10 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 11, to which the result has already been set.

   - Bit 11 = 0: (1/4 AV_{REF})
   - Bit 11 = 1: (3/4 AV_{REF})

   The voltage tap of the comparison voltage generator and the analog input voltage are compared and bit 10 of the SAR register is manipulated according to the result of the comparison.

   - Analog input voltage ≥ Voltage tap of comparison voltage generator: Bit 10 = 1
   - Analog input voltage ≤ Voltage tap of comparison voltage generator: Bit 10 = 0

   Comparison is continued like this to bit 0 of the SAR register.
4. Comparison voltage generator
   The comparison voltage generator generates the voltage to be compared with the input from an analog input pin.

   Remark AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage**Note**, and V_{DD}.

   Note For details about the internal reference voltage, see **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** and **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.
5. Successive approximation register (SAR)
   The SAR register is used to set voltage tap data whose values from the comparison voltage generator match the voltage values of the analog input pins, one 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 12-bit/10-bit A/D conversion result register (ADCRn). When all the specified A/D conversion operations have ended, a corresponding A/D conversion end interrupt request signal (INTAD0 to INTAD3) is generated.

6. 12-bit/10-bit A/D conversion result register (ADCRn)
   Each time A/D conversion ends, the conversion result is loaded from the successive approximation register, the ADCRn register holds the result as follows:
   When conversion is with 12-bit resolution, it holds the A/D conversion result in its lower 12 bits (the higher 4 bits are fixed to 0).
   When conversion is with 10-bit resolution, it holds the A/D conversion result in its higher 10 bits (the lower 6 bits are fixed to 0).
7. 8-bit A/D conversion result register (ADCRnH)
   The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCRnH register holds the higher 8 bits of the A/D conversion result.
8. Controller
   This circuit controls the times for the conversion of analog input signals to digital signals, starting and stopping of the conversion operations, and the order of handling the conversion channels according to their priority levels. When A/D conversion has been completed, this controller generates INTAD0 to INTAD3 through the A/D conversion result upper limit/lower limit comparator.
9. 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 and ADREFP0 bits of A/D converter mode register 2 (ADM2) to 0 and 1, respectively.
   The analog signals input to ANI2 to ANI7 and ANI16 to ANI30 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**Note** as the + side reference voltage of the A/D converter.

   Note For details about the internal reference voltage, see **Section 43 Electrical Characteristics (TA = –40 to +105°C)** and **Section 44 Electrical Characteristics (TA = –40 to +125°C)**.

10. AVREFM pin
    This pin inputs an external reference voltage (AVREFM). To use 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.
11. PGA circuit
    The output from the PGA circuit is selectable as an analog input to the A/D converter.
12. Dedicated sample & hold (S&H) circuits for channels 2 and 3 (two channels)
    The dedicated sample & hold circuits for channels 2 and 3 are for use in simultaneous sampling. These circuits are controlled by the controller described in **8. Controller**.

20.3 Registers to Control the A/D Converter

The following registers are used to control the A/D converter.

- Peripheral enable register 0 (PER0)
- Peripheral reset control register 0 (PRR0)
- A/D converter mode register 0 (ADM0)
- A/D converter mode register 1 (ADM1)
- A/D converter mode register 2 (ADM2)
- A/D converter mode register 3 (ADM3)**Note 1**
- 12-bit/10-bit A/D conversion result register and registers (ADCR, ADCRn) (n = 0 to 3)
- 8-bit A/D conversion result registers H, nH (ADCRH, ADCRnH) (n = 0 to 3)
- Analog input channel specification register (ADS)**Note 2**
- Analog input channel specification registers n for advanced mode (ADSn) (n = 0 to 3)**Note 1**
- Conversion setting register (ADSCTL)**Note 1**
- Conversion trigger specification registers n (ADTRn) (n = 0 to 3)**Note 1**
- Conversion interrupt control register (ADINTCTL)**Note 1**
- Conversion interrupt status register (ADINTST)**Note 1**
- A/D conversion sampling mode specification register (ADSPMOD)**Note 1**
- Conversion result comparison upper limit setting register (ADUL)
- Conversion result comparison lower limit setting register (ADLL)
- A/D test register (ADTES)**Note 2**
- Port mode registers xx (PMxx) (xx = 0 to 2, 12, 14)
- Port mode control A registers xx (PMCAxx) (xx = 0 to 2, 12, 14)

Note 1. This register is exclusively for use in advanced mode, so it should not be changed from the initial value when the advanced mode is disabled.

Note 2. This register is only usable when the advanced mode is disabled, so it should not be changed from the initial value when the advanced mode is enabled.

20.3.1 Peripheral enable register 0 (PER0)

The PER0 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. If the A/D converter is to be 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. The value of this register following a reset is 00H.

Figure 20 - 2 Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

ADCEN	Control of supply of an input clock to the A/D converter
0	Stops supply of an input clock. • The SFRs used by the A/D converter cannot be written. • When an SFR used by the A/D converter is read, the value returned is 00H or 0000H.
1	Enables supply of an input clock. • The SFRs used by the A/D converter can be read and written.

Caution 1. When setting the A/D converter, make sure that the setting of the ADCEN bit is 1 before setting the following registers.
If ADCEN = 0, the value of each register which controls the A/D converter is its initial value and writing to any of those registers is ignored (except for port mode registers 0 to 2, 12, and 14 (PM0 to PM2, PM12, and PM14) and port mode control A registers 0 to 2, 12, and 14 (PMCA0 to PMCA2, PMCA12, and PMCA14)).

- **A/D converter mode register 0 (ADM0)**
- **A/D converter mode register 1 (ADM1)**
- **A/D converter mode register 2 (ADM2)**
- **A/D converter mode register 3 (ADM3)**
- **12-bit/10-bit A/D conversion result register (ADCRn)**
- **8-bit A/D conversion result register (ADCRnH)**
- **Analog input channel specification register (ADS)**
- **Analog input channel specification register (ADSn)**
- **Conversion setting register (ADSCTL)**
- **Conversion trigger specification register n (ADTRn)**
- **Conversion interrupt control register (ADINTCTL)**
- **Conversion interrupt status register (ADINTST)**
- **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 bits 6 and 1 to 0.

Remark n = 0 to 3

20.3.2 Peripheral reset control register 0 (PRR0)

The PRR0 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. To place the A/D converter in the reset state, set bit 5 (ADCRES) of this register to 1. The PRR0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 20 - 3 Format of Peripheral Reset Control Register 0 (PRR0)

Address: F00F1H
After reset: 00H
R/W: R/W

Symbol	7	6	<5>	<4>	<3>	<2>	1	<0>
PRR0	0	0	ADCRES	IICA0RES	SAU1RES	SAU0RES	0	TAU0RES

ADCRES	Control resetting of the A/D converter
0	The A/D converter is released from the reset state.
1	The A/D converter is in the reset state. • The SFRs for use with the A/D converter are initialized.

Caution Be sure to clear bits 7, 6 and 1 to 0.

20.3.3 A/D converter mode register 0 (ADM0)

The ADM0 register sets the time for converting analog input to digital data, and starts and stops conversion. The ADM0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 20 - 4 Format of A/D Converter Mode Register 0 (ADM0)

Address: FFF30H
After reset: 00H
R/W: 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 is stopped or in standby.
1	Enables conversion operation [When read] While in the software trigger no-wait mode: Conversion is enabled. While in the software trigger wait mode: A/D power supply stabilization wait state + conversion in progress While in the hardware trigger no-wait mode: Conversion is enabled. While in the hardware trigger wait mode: A/D power supply stabilization wait state + conversion in progress While in the advanced mode: Conversion is enabled.

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 FR[2:0] and LV[1:0] bits and A/D conversion, see **Table 20 - 6 Selection of A/D Conversion Time**.

Note 2. While in the software trigger no-wait 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 + 2 cycles of the conversion clock (f_{AD}) from the start of operation for the operation to stabilize. Therefore, immediately after the ADCS bit is set to 1 after at least 1 μs + 2 cycles of the conversion clock (f_{AD}) have elapsed from the time ADCE bit is set to 1, the conversion result becomes valid. When ADCS is set to 1 while ADCE = 0, A/D conversion starts after the stabilization wait time has passed. If ADCS is set to 1 before at least 1 μs + 2 cycles of the conversion clock (f_{AD}) have elapsed, ignore data of the first conversion.

Caution 1. Change the ADMD, FR[2:0], and LV[1:0] bits while conversion is stopped (ADCS = 0, ADCE = 0).
Caution 2. Setting change from ADCS = 1 and ADCE = 1 to ADCS = 1 and ADCE = 0 is prohibited.
Caution 3. Do not change the ADCS and ADCE bits from 0 to 1 at the same time by using an 8-bit manipulation instruction. Be sure to follow the procedure described in 20.7 A/D Converter Setup Flowchart.
Caution 4. Do not set ADMS to 1 when the advanced mode is enabled.
Caution 5. Do not overwrite ADCS with 1 when the setting of ADCS is 1 in the advanced mode.
Caution 6. Do not overwrite ADCE with 1 when the setting of ADCE is 1 in the advanced mode.

Table 20 - 3 Settings of the ADCS and ADCE Bits

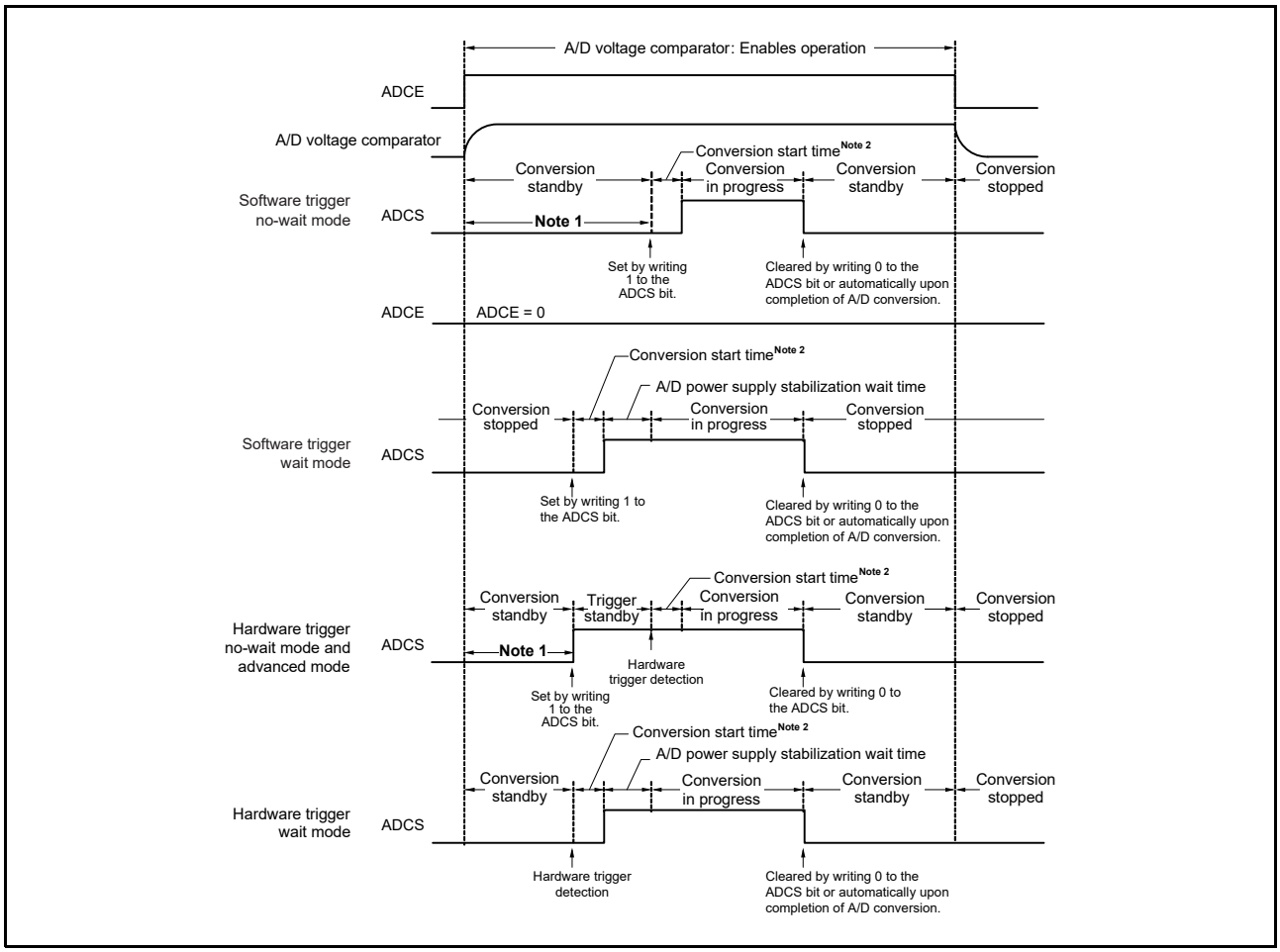
ADVMOD	ADCS	ADCE	State of the A/D Converter
0/1	0	0	Conversion is stopped.
0/1	0	1	A/D conversion is disabled but the comparator is enabled.
0	1	0	Conversion in progress or standby in software trigger wait mode or setting prohibited in modes other than software trigger mode
0	1	1	Conversion in progress in modes other than software trigger wait mode
1	1	0	Setting prohibited. Do not set ADCS to 1 before setting ADCE to 1. First set ADCE to 1, and then set ADCS 1.
1	1	1	Waiting for the trigger or conversion in progress (after having received the trigger)

Table 20 - 4 Conditions for Setting and Clearing the ADCS Bit

A/D Conversion Mode			Conditions for Setting	Conditions for Clearing
Software trigger no-wait mode	Select mode	Sequential conversion mode	1 is written to ADCS.	0 is written to ADCS.
		One-shot conversion mode		• 0 is written to ADCS. • A/D conversion ends.
	Scan mode	Sequential conversion mode		0 is written to ADCS.
		One-shot conversion mode		• 0 is written to ADCS. • Conversion ends on the specified four channels.
Software trigger wait mode	Select mode	Sequential conversion mode		0 is written to ADCS.
		One-shot conversion mode		• 0 is written to ADCS. • A/D conversion ends.
	Scan mode	Sequential conversion mode		0 is written to ADCS.
		One-shot conversion mode		• 0 is written to ADCS. • Conversion ends on the specified four channels.
Hardware trigger no-wait mode	Select mode	Sequential conversion mode		0 is written to ADCS.
		One-shot conversion mode		0 is written to ADCS.
	Scan mode	Sequential conversion mode		0 is written to ADCS.
		One-shot conversion mode		0 is written to ADCS.
Advanced mode**Note**	Select mode	One-shot conversion mode		0 is written to ADCS.
Hardware trigger wait mode	Select mode	Sequential conversion mode	A hardware trigger is input.	0 is written to ADCS.
		One-shot conversion mode		• 0 is written to ADCS. • A/D conversion ends.
	Scan mode	Sequential conversion mode		0 is written to ADCS.
		One-shot conversion mode		• 0 is written to ADCS. • Conversion ends on the specified four channels.

Note The scan mode and sequential conversion mode are not supported in advanced mode.

Figure 20 - 5 Timing Chart When the A/D Voltage Comparator Is Used



Note 1. While in the software trigger no-wait mode, hardware trigger no-wait mode, or advanced mode, the time from the rising of the ADCE bit to the rising of the ADCS bit must be 1 μs + 2 cycles of the conversion clock (f_{AD}) or longer to stabilize the internal circuit.

Note 2. The following shows the maximum time to start conversion.

ADM1	ADM0			Conversion Clock (f_{AD})	Conversion Start Time (Number of f_{CLK} Clock Cycles) (Common to Software and Hardware Trigger Modes)		
ADLSP	FR2	FR1	FR0		No-wait Mode	Wait Mode	Advanced Mode
0	0	0	0	$f_{CLK}/32$	31	1	36
0	0	0	1	$f_{CLK}/16$	15	1	20
0	0	1	0	$f_{CLK}/8$	7	1	12
0	0	1	1	$f_{CLK}/4$	3	1	8
0	1	0	0	$f_{CLK}/2$	1	1	6
0	1	0	1	f_{CLK}	1	1	5
1	0	1	1	$f_{CLK}/4$	3	1	8
1	1	0	0	$f_{CLK}/2$	1	1	6
1	1	0	1	f_{CLK}	1	1	5

However, for the second and subsequent conversion in sequential conversion mode and for conversion of the channels specified for 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 state.

Caution 2. While in the one-shot conversion mode of the hardware trigger no-wait mode or advanced mode, the ADCS bit 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 state).

Caution 4. In advanced mode, three cycles of the f_{CLK} clock are required from the occurrence of a trigger source until detection of the trigger. Table 20 - 5 lists the required numbers of clock cycles from the occurrence of a trigger or completion of the most recently executed conversion until A/D conversion starts in advanced mode.

Caution 5. To complete A/D conversion, specify at least the following time as the hardware trigger interval:
Hardware trigger no-wait mode: 2 cycles of the f_{CLK} clock + conversion start time + A/D conversion time
Hardware trigger wait mode: 2 cycles of the f_{CLK} clock + conversion start time + A/D power supply stabilization wait time + A/D conversion time
Advanced mode: 3 cycles of the f_{CLK} clock + conversion start time + A/D conversion time

Remark f_{CLK}: CPU/peripheral hardware clock frequency
Conversion start delay: Delay that is incurred after the conversion start time before the start of the A/D conversion time; one cycle of the f_{AD} clock
Interrupt output delay: Delay that is incurred from the completion of A/D conversion until INTAD0 to INTAD3 is generated; one cycle of the f_{AD} clock

Table 20 - 5 Required Numbers of Clock Cycles from the Occurrence of a Trigger or Completion of the Most Recently Executed Conversion Until A/D Conversion Starts in Advanced Mode

Name of the Preparation Time	Situation	Required Number of Clock Cycles
<1> Preparation time for a new conversion	Time from the occurrence of a trigger source until the start of A/D conversion when the trigger is generated while conversion is not proceeding and the converter has been placed in the trigger standby state	Conversion start time (4 cycles of f_{CLK} + 1 cycle of f_{AD})
<2> Preparation time for a conversion triggered by a high-priority source	Time from the occurrence of the trigger source until the start of A/D conversion when a trigger source with high priority is generated while a conversion triggered by a source with low priority is in progress	7 cycles of f_{CLK} + 8 cycles of f_{AD}
<3> Preparation time for a conversion held pending	Time from the completion of the conversion currently being executed until the start of the A/D conversion that was held pending	4 cycles of f_{CLK} + 1 cycle of f_{AD}

Table 20 - 6 Selection of A/D Conversion Time (1/11)

1. Normal modes 1 and 2 with no A/D power supply stabilization wait time
(software trigger no-wait select mode and hardware trigger no-wait select mode)

A/D Converter Mode Register 0 A/D Converter Mode Register 1						Mode	Conversion Clock (f_{AD})	Number of Clock Cycles for Conversion Start Delay	Number of Clock Cycles for Conversion	Number of Clock Cycles for Interrupt Output Delay	A/D Conversion Time (Conversion Start Delay Time + Conversion Time + Interrupt Output Delay Time)					
(ADM1)	(ADM0)											2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V				
ADLSP	FR2	FR1	FR0	LV1	LV0							f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	0	0	Normal 1	f_{CLK}/32	1 f_{AD}	64 f_{AD}	1 f_{AD}	2112/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	66 μs
0	0	0	1				f_{CLK}/16	1 f_{AD}	64 f_{AD}	1 f_{AD}	1056/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	66 μs	33 μs
0	0	1	0				f_{CLK}/8	1 f_{AD}	64 f_{AD}	1 f_{AD}	528/f_{CLK}	Setting prohibited	Setting prohibited	66 μs	33 μs	16.5 μs
0	0	1	1				f_{CLK}/4	1 f_{AD}	64 f_{AD}	1 f_{AD}	264/f_{CLK}	Setting prohibited	Setting prohibited	33 μs	16.5 μs	8.25 μs
0	1	0	0				f_{CLK}/2	1 f_{AD}	64 f_{AD}	1 f_{AD}	132/f_{CLK}	Setting prohibited	Setting prohibited	16.5 μs	8.25 μs	4.125 μs
0	1	0	1				f_{CLK}	1 f_{AD}	64 f_{AD}	1 f_{AD}	66/f_{CLK}	Setting prohibited	Setting prohibited	8.25 μs	4.125 μs	2.0625 μs
1	0	1	1				f_{CLK}/4	1 f_{AD}	64 f_{AD}	1 f_{AD}	264/f_{CLK}	Setting prohibited	66 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	1 f_{AD}	64 f_{AD}	1 f_{AD}	132/f_{CLK}	Setting prohibited	33 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	1 f_{AD}	64 f_{AD}	1 f_{AD}	66/f_{CLK}	66 μs	16.5 μs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									
0	0	0	0	0	1	Normal 2	f_{CLK}/32	1 f_{AD}	181 f_{AD}	1 f_{AD}	5856/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	183 μs
0	0	0	1				f_{CLK}/16	1 f_{AD}	181 f_{AD}	1 f_{AD}	2928/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	183 μs	91.5 μs
0	0	1	0				f_{CLK}/8	1 f_{AD}	181 f_{AD}	1 f_{AD}	1464/f_{CLK}	Setting prohibited	Setting prohibited	183 μs	91.5 μs	45.75 μs
0	0	1	1				f_{CLK}/4	1 f_{AD}	181 f_{AD}	1 f_{AD}	732/f_{CLK}	Setting prohibited	Setting prohibited	91.5 μs	45.75 μs	22.875 μs
0	1	0	0				f_{CLK}/2	1 f_{AD}	181 f_{AD}	1 f_{AD}	366/f_{CLK}	Setting prohibited	Setting prohibited	45.75 μs	22.875 μs	11.4375 μs
0	1	0	1				f_{CLK}	1 f_{AD}	181 f_{AD}	1 f_{AD}	183/f_{CLK}	Setting prohibited	Setting prohibited	22.875 μs	11.4375 μs	5.71875 μs
1	0	1	1				f_{CLK}/4	1 f_{AD}	181 f_{AD}	1 f_{AD}	732/f_{CLK}	Setting prohibited	183 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	1 f_{AD}	181 f_{AD}	1 f_{AD}	366/f_{CLK}	Setting prohibited	91.5 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	1 f_{AD}	181 f_{AD}	1 f_{AD}	183/f_{CLK}	183 μs	45.75 μs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									

Caution 1. The A/D conversion time must be within the relevant range of conversion clock (f_{AD}) and conversion times (t_{CONV}) described in 43.6.1 A/D converter characteristics or 44.6.1 A/D converter characteristics.

Caution 2. Rewrite the FR[2:0] and LV[1:0] bits to different values while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion times do not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. Additionally, the conversion times do not include clock frequency errors. Consider clock frequency errors when selecting the conversion time.

Caution 4. When the internal reference voltage or the temperature sensor output voltage is selected as the conversion target, use normal mode 2.

Caution 5. When the internal reference voltage is selected for the + side reference voltage, normal modes 1 and 2 cannot be used. In such cases, use low voltage mode 1 or 2.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 20 - 6 Selection of A/D Conversion Time (2/11)

2. Low voltage modes 1 and 2 with no A/D power supply stabilization wait time
(software trigger no-wait select mode and hardware trigger no-wait select mode)

A/D Converter Mode Register 0 A/D Converter Mode Register 1						Mode	Conversion Clock (f_{AD})	Number of Clock Cycles for Conversion Start Delay	Number of Clock Cycles for Conversion	Number of Clock Cycles for Interrupt Output Delay	A/D Conversion Time (Conversion Start Delay Time + Conversion Time + Interrupt Output Delay Time)					
(ADM1)	(ADM0)											1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	1.8 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	2.7 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V
ADLSP	FR2	FR1	FR0	LV1	LV0							f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	1	0	Low voltage 1	f_{CLK}/32	1 f_{AD}	80 f_{AD}	1 f_{AD}	2624/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	82 µs
0	0	0	1				f_{CLK}/16	1 f_{AD}	80 f_{AD}	1 f_{AD}	1312/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	82 µs	41 µs
0	0	1	0				f_{CLK}/8	1 f_{AD}	80 f_{AD}	1 f_{AD}	656/f_{CLK}	Setting prohibited	Setting prohibited	82 µs	41 µs	20.5 µs
0	0	1	1				f_{CLK}/4	1 f_{AD}	80 f_{AD}	1 f_{AD}	328/f_{CLK}	Setting prohibited	Setting prohibited	41 µs	20.5 µs	10.25 µs
0	1	0	0				f_{CLK}/2	1 f_{AD}	80 f_{AD}	1 f_{AD}	164/f_{CLK}	Setting prohibited	Setting prohibited	20.5 µs	10.25 µs	5.125 µs
0	1	0	1				f_{CLK}	1 f_{AD}	80 f_{AD}	1 f_{AD}	82/f_{CLK}	Setting prohibited	Setting prohibited	10.25 µs	5.125 µs	Setting prohibited
1	0	1	1				f_{CLK}/4	1 f_{AD}	80 f_{AD}	1 f_{AD}	328/f_{CLK}	Setting prohibited	82 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	1 f_{AD}	80 f_{AD}	1 f_{AD}	164/f_{CLK}	Setting prohibited	41 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	1 f_{AD}	80 f_{AD}	1 f_{AD}	82/f_{CLK}	82 µs	20.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									
0	0	0	0	1	1	Low voltage 2	f_{CLK}/32	1 f_{AD}	107 f_{AD}	1 f_{AD}	3488/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	109 µs
0	0	0	1				f_{CLK}/16	1 f_{AD}	107 f_{AD}	1 f_{AD}	1744/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	109 µs	54.5 µs
0	0	1	0				f_{CLK}/8	1 f_{AD}	107 f_{AD}	1 f_{AD}	872/f_{CLK}	Setting prohibited	Setting prohibited	109 µs	54.5 µs	27.25 µs
0	0	1	1				f_{CLK}/4	1 f_{AD}	107 f_{AD}	1 f_{AD}	436/f_{CLK}	Setting prohibited	Setting prohibited	54.5 µs	27.25 µs	13.625 µs
0	1	0	0				f_{CLK}/2	1 f_{AD}	107 f_{AD}	1 f_{AD}	218/f_{CLK}	Setting prohibited	Setting prohibited	27.25 µs	13.625 µs	6.8125 µs
0	1	0	1				f_{CLK}	1 f_{AD}	107 f_{AD}	1 f_{AD}	109/f_{CLK}	Setting prohibited	Setting prohibited	13.625 µs	6.8125 µs	Setting prohibited
1	0	1	1				f_{CLK}/4	1 f_{AD}	107 f_{AD}	1 f_{AD}	436/f_{CLK}	Setting prohibited	109 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	1 f_{AD}	107 f_{AD}	1 f_{AD}	218/f_{CLK}	Setting prohibited	54.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	1 f_{AD}	107 f_{AD}	1 f_{AD}	109/f_{CLK}	109 µs	27.25 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									

Caution 1. The A/D conversion time must be within the relevant range of conversion clock (f_{AD}) and conversion times (t_{CONV}) described in 43.6.1 A/D converter characteristics or 44.6.1 A/D converter characteristics.

Caution 2. Rewrite the FR[2:0] and LV[1:0] bits to different values while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion times do not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. Additionally, the conversion times do not include clock frequency errors. Consider clock frequency errors when selecting the conversion time.

Caution 4. When the internal reference voltage or the temperature sensor output voltage is selected as the conversion target, use low voltage mode 2 with the conversion clock (f_{AD}) with a frequency of no more than 16 MHz.

(Caution and Remark are listed on the next page.)

Caution 5. When the internal reference voltage is selected for the + side reference voltage, the conversion clock (f_{AD}) becomes 1 to 2 MHz.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 20 - 6 Selection of A/D Conversion Time (3/11)

3. Normal modes 1 and 2 with A/D power supply stabilization wait time
(software trigger wait select mode and hardware trigger wait select mode[Note 1])

A/D Converter Mode Register 0 A/D Converter Mode Register 1						Mode	Conversion Clock (f_{AD})	Number of Clock Cycles for A/D Power Supply Stabilization Wait	Number of Clock Cycles for Conversion	Number of Clock Cycles for Interrupt Output Delay[Note 2]	A/D Conversion Time (A/D Power Supply Stabilization Wait Time + Conversion Time + Interrupt Output Delay Time)				
(ADM1)	(ADM0)											2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V				
ADLSP	FR2	FR1	FR0	LV1	LV0							f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	0	0	Normal 1	f_{CLK}/32	4 f_{AD}	64 f_{AD}	4 f_{AD}	2304/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	72 μs
0	0	0	1				f_{CLK}/16	4 f_{AD}	64 f_{AD}	4 f_{AD}	1152/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	72 μs	36 μs
0	0	1	0				f_{CLK}/8	6 f_{AD}	64 f_{AD}	4 f_{AD}	592/f_{CLK}	Setting prohibited	Setting prohibited	74 μs	37 μs	18.5 μs
0	0	1	1				f_{CLK}/4	10 f_{AD}	64 f_{AD}	4 f_{AD}	312/f_{CLK}	Setting prohibited	Setting prohibited	39 μs	19.5 μs	9.75 μs
0	1	0	0				f_{CLK}/2	18 f_{AD}	64 f_{AD}	4 f_{AD}	172/f_{CLK}	Setting prohibited	Setting prohibited	21.5 μs	10.75 μs	5.375 μs
0	1	0	1				f_{CLK}	34 f_{AD}	64 f_{AD}	4 f_{AD}	102/f_{CLK}	Setting prohibited	Setting prohibited	12.75 μs	6.375 μs	3.1875 μs
1	0	1	1				f_{CLK}/4	4 f_{AD}	64 f_{AD}	4 f_{AD}	288/f_{CLK}	Setting prohibited	72 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	4 f_{AD}	64 f_{AD}	4 f_{AD}	144/f_{CLK}	Setting prohibited	36 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	6 f_{AD}	64 f_{AD}	4 f_{AD}	74/f_{CLK}	74 μs	18.5 μs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									
0	0	0	0	0	1	Normal 2	f_{CLK}/32	4 f_{AD}	181 f_{AD}	4 f_{AD}	6048/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	189 μs
0	0	0	1				f_{CLK}/16	4 f_{AD}	181 f_{AD}	4 f_{AD}	3024/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	189 μs	94.5 μs
0	0	1	0				f_{CLK}/8	6 f_{AD}	181 f_{AD}	4 f_{AD}	1528/f_{CLK}	Setting prohibited	Setting prohibited	191 μs	95.5 μs	47.75 μs
0	0	1	1				f_{CLK}/4	10 f_{AD}	181 f_{AD}	4 f_{AD}	780/f_{CLK}	Setting prohibited	Setting prohibited	97.5 μs	48.75 μs	24.375 μs
0	1	0	0				f_{CLK}/2	18 f_{AD}	181 f_{AD}	4 f_{AD}	406/f_{CLK}	Setting prohibited	Setting prohibited	50.75 μs	25.375 μs	12.6875 μs
0	1	0	1				f_{CLK}	34 f_{AD}	181 f_{AD}	4 f_{AD}	219/f_{CLK}	Setting prohibited	Setting prohibited	27.375 μs	13.6875 μs	6.84375 μs
1	0	1	1				f_{CLK}/4	4 f_{AD}	181 f_{AD}	4 f_{AD}	756/f_{CLK}	Setting prohibited	189 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	4 f_{AD}	181 f_{AD}	4 f_{AD}	378/f_{CLK}	Setting prohibited	94.5 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	6 f_{AD}	181 f_{AD}	4 f_{AD}	191/f_{CLK}	191 μs	47.75 μs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									

Note 1. For the second and subsequent conversion in sequential conversion mode and for conversion of the channels specified for scan 1, 2, and 3 in scan mode, the conversion start time and A/D power supply stabilization wait time do not occur after a hardware trigger is detected. For details, see **Table 20 - 6 Selection of A/D Conversion Time (1/11)**.

Note 2. This number denotes the number of clock cycles for interrupt output delay in the one-shot conversion mode. When the sequential conversion mode is selected, the number of conversion clock (f_{AD}) cycles becomes shorter by three cycles.

Caution 1. The A/D conversion time must be within the relevant range of conversion clock (f_{AD}) and conversion times (t_{CONV}) described in 43.6.1 A/D converter characteristics or 44.6.1 A/D converter characteristics. Note that the conversion time (t_{CONV}) does not include A/D power supply stabilization wait time.

Caution 2. Rewrite the FR[2:0] and LV[1:0] bits to different values while conversion is stopped (ADCS = 0, ADCE = 0).

(Cautions and Remark are listed on the next page.)

Caution 3. The above conversion times do not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. Additionally, the conversion times do not include clock frequency errors. Consider clock frequency errors when selecting the conversion time.

Caution 4. The conversion time for the hardware trigger wait mode includes the time until the A/D power supply is stabilized from the hardware trigger is detected. The conversion time for the software trigger wait mode includes the time until the A/D power supply is stabilized from setting ADCS to 1.

Caution 5. When the internal reference voltage or the temperature sensor output voltage is selected as the conversion target, use normal mode 2.

Caution 6. When the internal reference voltage is selected for the + side reference voltage, normal modes 1 and 2 cannot be used. In such cases, use low voltage mode 1 or 2.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 20 - 6 Selection of A/D Conversion Time (4/11)

4. Low voltage modes 1 and 2 with A/D power supply stabilization wait time
(software trigger wait select mode and hardware trigger wait select mode[Note 1])

A/D Converter Mode Register 0 A/D Converter Mode Register 1						Mode	Conversion Clock (f_{AD})	Number of Clock Cycles for A/D Power Supply Stabilization Wait	Number of Clock Cycles for Conversion	Number of Clock Cycles for Interrupt Output Delay[Note 2]	A/D Conversion Time (A/D Power Supply Stabilization Wait Time + Conversion Time + Interrupt Output Delay Time)				
(ADM1)	(ADM0)											1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	1.8 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	2.7 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V
ADLSP	FR2	FR1	FR0	LV1	LV0							f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	1	0	Low voltage 1	f_{CLK}/32	4 f_{AD}	80 f_{AD}	4 f_{AD}	2816/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	88 µs
0	0	0	1				f_{CLK}/16	4 f_{AD}	80 f_{AD}	4 f_{AD}	1408/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	88 µs	44 µs
0	0	1	0				f_{CLK}/8	6 f_{AD}	80 f_{AD}	4 f_{AD}	720/f_{CLK}	Setting prohibited	Setting prohibited	90 µs	45 µs	22.5 µs
0	0	1	1				f_{CLK}/4	10 f_{AD}	80 f_{AD}	4 f_{AD}	376/f_{CLK}	Setting prohibited	Setting prohibited	47 µs	23.5 µs	11.75 µs
0	1	0	0				f_{CLK}/2	18 f_{AD}	80 f_{AD}	4 f_{AD}	204/f_{CLK}	Setting prohibited	Setting prohibited	25.5 µs	12.75 µs	6.375 µs
0	1	0	1				f_{CLK}	34 f_{AD}	80 f_{AD}	4 f_{AD}	118/f_{CLK}	Setting prohibited	Setting prohibited	14.75 µs	7.375 µs	Setting prohibited
1	0	1	1				f_{CLK}/4	4 f_{AD}	80 f_{AD}	4 f_{AD}	352/f_{CLK}	Setting prohibited	88 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	4 f_{AD}	80 f_{AD}	4 f_{AD}	176/f_{CLK}	Setting prohibited	44 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	6 f_{AD}	80 f_{AD}	4 f_{AD}	90/f_{CLK}	90 µs	22.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									
0	0	0	0	1	1	Low voltage 2	f_{CLK}/32	4 f_{AD}	107 f_{AD}	4 f_{AD}	3680/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	115 µs
0	0	0	1				f_{CLK}/16	4 f_{AD}	107 f_{AD}	4 f_{AD}	1840/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	115 µs	57.5 µs
0	0	1	0				f_{CLK}/8	6 f_{AD}	107 f_{AD}	4 f_{AD}	936/f_{CLK}	Setting prohibited	Setting prohibited	117 µs	58.5 µs	29.25 µs
0	0	1	1				f_{CLK}/4	10 f_{AD}	107 f_{AD}	4 f_{AD}	484/f_{CLK}	Setting prohibited	Setting prohibited	60.5 µs	30.25 µs	15.125 µs
0	1	0	0				f_{CLK}/2	18 f_{AD}	107 f_{AD}	4 f_{AD}	258/f_{CLK}	Setting prohibited	Setting prohibited	32.25 µs	16.125 µs	8.0625 µs
0	1	0	1				f_{CLK}	34 f_{AD}	107 f_{AD}	4 f_{AD}	145/f_{CLK}	Setting prohibited	Setting prohibited	18.125 µs	9.0625 µs	Setting prohibited
1	0	1	1				f_{CLK}/4	4 f_{AD}	107 f_{AD}	4 f_{AD}	460/f_{CLK}	Setting prohibited	115 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	4 f_{AD}	107 f_{AD}	4 f_{AD}	230/f_{CLK}	Setting prohibited	57.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	6 f_{AD}	107 f_{AD}	4 f_{AD}	117/f_{CLK}	117 µs	29.25 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									

Note 1. For the second and subsequent conversion in sequential conversion mode and for conversion of the channels specified for scan 1, 2, and 3 in scan mode, the conversion start time and A/D power supply stabilization wait time do not occur after a hardware trigger is detected. For details, see **Table 20 - 6 Selection of A/D Conversion Time (2/11)**.

Note 2. This number denotes the number of clock cycles for interrupt output delay in the one-shot conversion mode. When the sequential conversion mode is selected, the number of conversion clock (f_{AD}) cycles becomes shorter by three cycles.

Caution 1. The A/D conversion time must be within the relevant range of conversion clock (f_{AD}) and conversion times (t_{CONV}) described in 43.6.1 A/D converter characteristics or 44.6.1 A/D converter characteristics. Note that the conversion time (t_{CONV}) does not include A/D power supply stabilization wait time.

Caution 2. Rewrite the FR[2:0] and LV[1:0] bits to different values while conversion is stopped (ADCS = 0, ADCE = 0).

(Cautions and Remark are listed on the next page.)

Caution 3. The above conversion times do not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. Additionally, the conversion times do not include clock frequency errors. Consider clock frequency errors when selecting the conversion time.

Caution 4. The conversion time for the hardware trigger wait mode includes the time until the A/D power supply is stabilized from the hardware trigger is detected. The conversion time for the software trigger wait mode includes the time until the A/D power supply is stabilized from setting ADCS to 1.

Caution 5. When the internal reference voltage or the temperature sensor output voltage is selected as the conversion target, use low voltage mode 2 with the conversion clock (f_{AD}) with a frequency of no more than 16 MHz.

Caution 6. When the internal reference voltage is selected for the + side reference voltage, the conversion clock (f_{AD}) becomes 1 to 2 MHz.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 20 - 6 Selection of A/D Conversion Time (5/11)

5. Normal modes 1 and 2 with no A/D power supply stabilization wait time
(software trigger no-wait scan mode and hardware trigger no-wait scan mode)

A/D Converter Mode Register 0 A/D Converter Mode Register 1						Mode	Conversion Clock (f_{AD})	Number of Clock Cycles for Conversion Start Delay	Number of Clock Cycles for Conversion	Number of Clock Cycles for Interrupt Output Delay	A/D Conversion Time (Conversion Start Delay Time + Conversion Time × 4 + Interrupt Output Delay Time)				
(ADM1)	(ADM0)											2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V			
ADLSP	FR2	FR1	FR0	LV1	LV0							f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	0	0	Normal 1	f_{CLK}/32	1 f_{AD}	64 f_{AD}	1 f_{AD}	8256/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	258 µs
0	0	0	1				f_{CLK}/16	1 f_{AD}	64 f_{AD}	1 f_{AD}	4128/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	258 µs	129 µs
0	0	1	0				f_{CLK}/8	1 f_{AD}	64 f_{AD}	1 f_{AD}	2064/f_{CLK}	Setting prohibited	Setting prohibited	258 µs	129 µs	64.5 µs
0	0	1	1				f_{CLK}/4	1 f_{AD}	64 f_{AD}	1 f_{AD}	1032/f_{CLK}	Setting prohibited	Setting prohibited	129 µs	64.5 µs	32.25 µs
0	1	0	0				f_{CLK}/2	1 f_{AD}	64 f_{AD}	1 f_{AD}	516/f_{CLK}	Setting prohibited	Setting prohibited	64.5 µs	32.25 µs	16.125 µs
0	1	0	1				f_{CLK}	1 f_{AD}	64 f_{AD}	1 f_{AD}	258/f_{CLK}	Setting prohibited	Setting prohibited	32.25 µs	16.125 µs	8.0625 µs
1	0	1	1				f_{CLK}/4	1 f_{AD}	64 f_{AD}	1 f_{AD}	1032/f_{CLK}	Setting prohibited	258 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	1 f_{AD}	64 f_{AD}	1 f_{AD}	516/f_{CLK}	Setting prohibited	129 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	1 f_{AD}	64 f_{AD}	1 f_{AD}	258/f_{CLK}	258 µs	64.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									
0	0	0	0	0	1	Normal 2	f_{CLK}/32	1 f_{AD}	181 f_{AD}	1 f_{AD}	23232/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	726 µs
0	0	0	1				f_{CLK}/16	1 f_{AD}	181 f_{AD}	1 f_{AD}	11616/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	726 µs	363 µs
0	0	1	0				f_{CLK}/8	1 f_{AD}	181 f_{AD}	1 f_{AD}	5808/f_{CLK}	Setting prohibited	Setting prohibited	726 µs	363 µs	181.5 µs
0	0	1	1				f_{CLK}/4	1 f_{AD}	181 f_{AD}	1 f_{AD}	2904/f_{CLK}	Setting prohibited	Setting prohibited	363 µs	181.5 µs	90.75 µs
0	1	0	0				f_{CLK}/2	1 f_{AD}	181 f_{AD}	1 f_{AD}	1452/f_{CLK}	Setting prohibited	Setting prohibited	181.5 µs	90.75 µs	45.375 µs
0	1	0	1				f_{CLK}	1 f_{AD}	181 f_{AD}	1 f_{AD}	726/f_{CLK}	Setting prohibited	Setting prohibited	90.75 µs	45.375 µs	22.6875 µs
1	0	1	1				f_{CLK}/4	1 f_{AD}	181 f_{AD}	1 f_{AD}	2904/f_{CLK}	Setting prohibited	726 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	1 f_{AD}	181 f_{AD}	1 f_{AD}	1452/f_{CLK}	Setting prohibited	363 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	1 f_{AD}	181 f_{AD}	1 f_{AD}	726/f_{CLK}	726 µs	181.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									

Caution 1. The A/D conversion time must be within the relevant range of conversion clock (f_{AD}) and conversion times (t_{CONV}) described in 43.6.1 A/D converter characteristics or 44.6.1 A/D converter characteristics.

Caution 2. Rewrite the FR[2:0] and LV[1:0] bits to different values while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion times do not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. Additionally, the conversion times do not include clock frequency errors. Consider clock frequency errors when selecting the conversion time.

Caution 4. When the internal reference voltage or the temperature sensor output voltage is selected as the conversion target, use normal mode 2.

Caution 5. When the internal reference voltage is selected for the + side reference voltage, normal modes 1 and 2 cannot be used. In such cases, use low voltage mode 1 or 2.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 20 - 6 Selection of A/D Conversion Time (6/11)

6. Low voltage modes 1 and 2 with no A/D power supply stabilization wait time
(software trigger no-wait scan mode and hardware trigger no-wait scan mode)

A/D Converter Mode Register 0 A/D Converter Mode Register 1						Mode	Conversion Clock (f_{AD})	Number of Clock Cycles for Conversion Start Delay	Number of Clock Cycles for Conversion	Number of Clock Cycles for Interrupt Output Delay	A/D Conversion Time (Conversion Start Delay Time + Conversion Time × 4 + Interrupt Output Delay Time)					
(ADM1)	(ADM0)											1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	1.8 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	2.7 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V
ADLSP	FR2	FR1	FR0	LV1	LV0							f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	1	0	Low voltage 1	f_{CLK}/32	1 f_{AD}	80 f_{AD}	1 f_{AD}	10304/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	322 µs
0	0	0	1				f_{CLK}/16	1 f_{AD}	80 f_{AD}	1 f_{AD}	5152/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	322 µs	161 µs
0	0	1	0				f_{CLK}/8	1 f_{AD}	80 f_{AD}	1 f_{AD}	2576/f_{CLK}	Setting prohibited	Setting prohibited	322 µs	161 µs	80.5 µs
0	0	1	1				f_{CLK}/4	1 f_{AD}	80 f_{AD}	1 f_{AD}	1288/f_{CLK}	Setting prohibited	Setting prohibited	161 µs	80.5 µs	40.25 µs
0	1	0	0				f_{CLK}/2	1 f_{AD}	80 f_{AD}	1 f_{AD}	644/f_{CLK}	Setting prohibited	Setting prohibited	80.5 µs	40.25 µs	20.125 µs
0	1	0	1				f_{CLK}	1 f_{AD}	80 f_{AD}	1 f_{AD}	322/f_{CLK}	Setting prohibited	Setting prohibited	40.25 µs	20.125 µs	Setting prohibited
1	0	1	1				f_{CLK}/4	1 f_{AD}	80 f_{AD}	1 f_{AD}	1288/f_{CLK}	Setting prohibited	322 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	1 f_{AD}	80 f_{AD}	1 f_{AD}	644/f_{CLK}	Setting prohibited	161 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	1 f_{AD}	80 f_{AD}	1 f_{AD}	322/f_{CLK}	322 µs	80.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									
0	0	0	0	1	1	Low voltage 2	f_{CLK}/32	1 f_{AD}	107 f_{AD}	1 f_{AD}	13760/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	430 µs
0	0	0	1				f_{CLK}/16	1 f_{AD}	107 f_{AD}	1 f_{AD}	6880/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	430 µs	215 µs
0	0	1	0				f_{CLK}/8	1 f_{AD}	107 f_{AD}	1 f_{AD}	3440/f_{CLK}	Setting prohibited	Setting prohibited	430 µs	215 µs	107.5 µs
0	0	1	1				f_{CLK}/4	1 f_{AD}	107 f_{AD}	1 f_{AD}	1720/f_{CLK}	Setting prohibited	Setting prohibited	215 µs	107.5 µs	53.75 µs
0	1	0	0				f_{CLK}/2	1 f_{AD}	107 f_{AD}	1 f_{AD}	860/f_{CLK}	Setting prohibited	Setting prohibited	107.5 µs	53.75 µs	26.875 µs
0	1	0	1				f_{CLK}	1 f_{AD}	107 f_{AD}	1 f_{AD}	430/f_{CLK}	Setting prohibited	Setting prohibited	53.75 µs	26.875 µs	Setting prohibited
1	0	1	1				f_{CLK}/4	1 f_{AD}	107 f_{AD}	1 f_{AD}	1720/f_{CLK}	Setting prohibited	430 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	1 f_{AD}	107 f_{AD}	1 f_{AD}	860/f_{CLK}	Setting prohibited	215 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	1 f_{AD}	107 f_{AD}	1 f_{AD}	430/f_{CLK}	430 µs	107.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									

Caution 1. The A/D conversion time must be within the relevant range of conversion clock (f_{AD}) and conversion times (t_{CONV}) described in 43.6.1 A/D converter characteristics or 44.6.1 A/D converter characteristics.

Caution 2. Rewrite the FR[2:0] and LV[1:0] bits to different values while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion times do not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. Additionally, the conversion times do not include clock frequency errors. Consider clock frequency errors when selecting the conversion time.

Caution 4. When the internal reference voltage or the temperature sensor output voltage is selected as the conversion target, use low voltage mode 2 with the conversion clock (f_{AD}) with a frequency of no more than 16 MHz.

(Caution and Remark are listed on the next page.)

Caution 5. When the internal reference voltage is selected for the + side reference voltage, the conversion clock (f_{AD}) becomes 1 to 2 MHz.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 20 - 6 Selection of A/D Conversion Time (7/11)

7. Normal modes 1 and 2 with A/D power supply stabilization wait time
(software trigger wait scan mode and hardware trigger wait scan mode[Note 1])

A/D Converter Mode Register 0 A/D Converter Mode Register 1						Mode	Conversion Clock (f_{AD})	Number of Clock Cycles for A/D Power Supply Stabilization Wait	Number of Clock Cycles for Conversion	Number of Clock Cycles for Interrupt Output Delay[Note 2]	A/D Conversion Time (A/D Power Supply Stabilization Wait Time + Conversion Time × 4 + Interrupt Output Delay Time)				
(ADM1)	(ADM0)											2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V				
ADLSP	FR2	FR1	FR0	LV1	LV0							f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	0	0	Normal 1	f_{CLK}/32	4 f_{AD}	64 f_{AD}	4 f_{AD}	8448/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	264 µs
0	0	0	1				f_{CLK}/16	4 f_{AD}	64 f_{AD}	4 f_{AD}	4224/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	264 µs	132 µs
0	0	1	0				f_{CLK}/8	6 f_{AD}	64 f_{AD}	4 f_{AD}	2128/f_{CLK}	Setting prohibited	Setting prohibited	266 µs	133 µs	66.5 µs
0	0	1	1				f_{CLK}/4	10 f_{AD}	64 f_{AD}	4 f_{AD}	1080/f_{CLK}	Setting prohibited	Setting prohibited	135 µs	67.5 µs	33.75 µs
0	1	0	0				f_{CLK}/2	18 f_{AD}	64 f_{AD}	4 f_{AD}	556/f_{CLK}	Setting prohibited	Setting prohibited	69.5 µs	34.75 µs	17.375 µs
0	1	0	1				f_{CLK}	34 f_{AD}	64 f_{AD}	4 f_{AD}	294/f_{CLK}	Setting prohibited	Setting prohibited	36.75 µs	18.375 µs	9.1875 µs
1	0	1	1				f_{CLK}/4	4 f_{AD}	64 f_{AD}	4 f_{AD}	1056/f_{CLK}	1056 µs	264 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	4 f_{AD}	64 f_{AD}	4 f_{AD}	528/f_{CLK}	528 µs	132 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	6 f_{AD}	64 f_{AD}	4 f_{AD}	266/f_{CLK}	266 µs	66.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									
0	0	0	0	0	1	Normal 2	f_{CLK}/32	4 f_{AD}	181 f_{AD}	4 f_{AD}	23424/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	732 µs
0	0	0	1				f_{CLK}/16	4 f_{AD}	181 f_{AD}	4 f_{AD}	11712/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	732 µs	366 µs
0	0	1	0				f_{CLK}/8	6 f_{AD}	181 f_{AD}	4 f_{AD}	5872/f_{CLK}	Setting prohibited	Setting prohibited	734 µs	367 µs	183.5 µs
0	0	1	1				f_{CLK}/4	10 f_{AD}	181 f_{AD}	4 f_{AD}	2952/f_{CLK}	Setting prohibited	Setting prohibited	369 µs	184.5 µs	92.25 µs
0	1	0	0				f_{CLK}/2	18 f_{AD}	181 f_{AD}	4 f_{AD}	1492/f_{CLK}	Setting prohibited	Setting prohibited	186.5 µs	93.25 µs	46.625 µs
0	1	0	1				f_{CLK}	34 f_{AD}	181 f_{AD}	4 f_{AD}	762/f_{CLK}	Setting prohibited	Setting prohibited	95.25 µs	47.625 µs	23.8125 µs
1	0	1	1				f_{CLK}/4	4 f_{AD}	181 f_{AD}	4 f_{AD}	2928/f_{CLK}	Setting prohibited	732 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	4 f_{AD}	181 f_{AD}	4 f_{AD}	1464/f_{CLK}	Setting prohibited	366 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	6 f_{AD}	181 f_{AD}	4 f_{AD}	734/f_{CLK}	734 µs	183.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									

Note 1. For the second and subsequent conversion in sequential conversion mode and for conversion of the channels specified for scan 1, 2, and 3 in scan mode, the conversion start time and A/D power supply stabilization wait time do not occur after a hardware trigger is detected. For details, see **Table 20 - 6 Selection of A/D Conversion Time (1/11)**.

Note 2. This number denotes the number of clock cycles for interrupt output delay in the one-shot conversion mode. When the sequential conversion mode is selected, the number of conversion clock (f_{AD}) cycles becomes shorter by three cycles.

Caution 1. The A/D conversion time must be within the relevant range of conversion clock (f_{AD}) and conversion times (t_{CONV}) described in 43.6.1 A/D converter characteristics or 44.6.1 A/D converter characteristics. Note that the conversion time (t_{CONV}) does not include A/D power supply stabilization wait time.

Caution 2. Rewrite the FR[2:0] and LV[1:0] bits to different values while conversion is stopped (ADCS = 0, ADCE = 0).

(Cautions and Remark are listed on the next page.)

Caution 3. **The above conversion times do not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. Additionally, the conversion times do not include clock frequency errors. Consider clock frequency errors when selecting the conversion time.**

Caution 4. **The conversion time for the hardware trigger wait mode includes the time until the A/D power supply is stabilized from the hardware trigger is detected. The conversion time for the software trigger wait mode includes the time until the A/D power supply is stabilized from setting ADCS to 1.**

Caution 5. **When the internal reference voltage or the temperature sensor output voltage is selected as the conversion target, use normal mode 2.**

Caution 6. **When the internal reference voltage is selected for the + side reference voltage, normal modes 1 and 2 cannot be used. In such cases, use low voltage mode 1 or 2.**

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 20 - 6 Selection of A/D Conversion Time (8/11)

8. Low voltage modes 1 and 2 with A/D power supply stabilization wait time
(software trigger wait scan mode and hardware trigger wait scan mode[Note 1])

A/D Converter Mode Register 0 A/D Converter Mode Register 1						Mode	Conversion Clock (f_{AD})	Number of Clock Cycles for A/D Power Supply Stabilization Wait	Number of Clock Cycles for Conversion	Number of Clock Cycles for Interrupt Output Delay[Note 2]	A/D Conversion Time (A/D Power Supply Stabilization Wait Time + Conversion Time × 4 + Interrupt Output Delay Time)				
(ADM1)	(ADM0)											1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	1.8 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	2.7 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V
ADLSP	FR2	FR1	FR0	LV1	LV0							f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	1	0	Low voltage 1	f_{CLK}/32	4 f_{AD}	80 f_{AD}	4 f_{AD}	10496/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	328 µs
0	0	0	1				f_{CLK}/16	4 f_{AD}	80 f_{AD}	4 f_{AD}	5248/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	328 µs	164 µs
0	0	1	0				f_{CLK}/8	6 f_{AD}	80 f_{AD}	4 f_{AD}	2640/f_{CLK}	Setting prohibited	Setting prohibited	330 µs	165 µs	82.5 µs
0	0	1	1				f_{CLK}/4	10 f_{AD}	80 f_{AD}	4 f_{AD}	1336/f_{CLK}	Setting prohibited	Setting prohibited	167 µs	83.5 µs	41.75 µs
0	1	0	0				f_{CLK}/2	18 f_{AD}	80 f_{AD}	4 f_{AD}	684/f_{CLK}	Setting prohibited	Setting prohibited	85.5 µs	42.75 µs	21.375 µs
0	1	0	1				f_{CLK}	34 f_{AD}	80 f_{AD}	4 f_{AD}	358/f_{CLK}	Setting prohibited	Setting prohibited	44.75 µs	22.375 µs	Setting prohibited
1	0	1	1				f_{CLK}/4	4 f_{AD}	80 f_{AD}	4 f_{AD}	1312/f_{CLK}	Setting prohibited	328 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	4 f_{AD}	80 f_{AD}	4 f_{AD}	656/f_{CLK}	Setting prohibited	164 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	6 f_{AD}	80 f_{AD}	4 f_{AD}	330/f_{CLK}	330 µs	82.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									
0	0	0	0	1	1	Low voltage 2	f_{CLK}/32	4 f_{AD}	107 f_{AD}	4 f_{AD}	13952/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	436 µs
0	0	0	1				f_{CLK}/16	4 f_{AD}	107 f_{AD}	4 f_{AD}	6976/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	436 µs	218 µs
0	0	1	0				f_{CLK}/8	6 f_{AD}	107 f_{AD}	4 f_{AD}	3504/f_{CLK}	Setting prohibited	Setting prohibited	438 µs	219 µs	109.5 µs
0	0	1	1				f_{CLK}/4	10 f_{AD}	107 f_{AD}	4 f_{AD}	1768/f_{CLK}	Setting prohibited	Setting prohibited	221 µs	110.5 µs	55.25 µs
0	1	0	0				f_{CLK}/2	18 f_{AD}	107 f_{AD}	4 f_{AD}	900/f_{CLK}	Setting prohibited	Setting prohibited	112.5 µs	56.25 µs	28.125 µs
0	1	0	1				f_{CLK}	34 f_{AD}	107 f_{AD}	4 f_{AD}	466/f_{CLK}	Setting prohibited	Setting prohibited	58.25 µs	29.125 µs	Setting prohibited
1	0	1	1				f_{CLK}/4	4 f_{AD}	107 f_{AD}	4 f_{AD}	1744/f_{CLK}	Setting prohibited	436 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	4 f_{AD}	107 f_{AD}	4 f_{AD}	872/f_{CLK}	Setting prohibited	218 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	6 f_{AD}	107 f_{AD}	4 f_{AD}	438/f_{CLK}	438 µs	109.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									

Note 1. For the second and subsequent conversion in sequential conversion mode and for conversion of the channels specified for scan 1, 2, and 3 in scan mode, the conversion start time and A/D power supply stabilization wait time do not occur after a hardware trigger is detected. For details, see **Table 20 - 6 Selection of A/D Conversion Time (2/11)**.

Note 2. This number denotes the number of clock cycles for interrupt output delay in the one-shot conversion mode. When the sequential conversion mode is selected, the number of conversion clock (f_{AD}) cycles becomes shorter by three cycles.

Caution 1. The A/D conversion time must be within the relevant range of conversion clock (f_{AD}) and conversion times (t_{CONV}) described in 43.6.1 A/D converter characteristics or 44.6.1 A/D converter characteristics. Note that the conversion time (t_{CONV}) does not include A/D power supply stabilization wait time.

Caution 2. Rewrite the FR[2:0] and LV[1:0] bits to different values while conversion is stopped (ADCS = 0, ADCE = 0).

(Cautions and Remark are listed on the next page.)

Caution 3. The above conversion times do not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. Additionally, the conversion times do not include clock frequency errors. Consider clock frequency errors when selecting the conversion time.

Caution 4. The conversion time for the hardware trigger wait mode includes the time until the A/D power supply is stabilized from the hardware trigger is detected. The conversion time for the software trigger wait mode includes the time until the A/D power supply is stabilized from setting ADCS to 1.

Caution 5. When the internal reference voltage or the temperature sensor output voltage is selected as the conversion target, use low voltage mode 2 with the conversion clock (f_{AD}) with a frequency of no more than 16 MHz.

Caution 6. When the internal reference voltage is selected for the + side reference voltage, the conversion clock (f_{AD}) becomes 1 to 2 MHz.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 20 - 6 Selection of A/D Conversion Time (9/11)

9. Normal mode 1 with no A/D power supply stabilization wait time (in advanced mode) when ANI0 to ANI7 are to be A/D converted

A/D Converter Mode Register 0 A/D Converter Mode Register 1						Mode	Conversion Clock (f_{AD})	Number of Clock Cycles for Conversion Start Delay	Number of Clock Cycles for Conversion Note	Number of Clock Cycles for Interrupt Output Delay	A/D Conversion Time (Conversion Start Delay Time + Conversion Time + Interrupt Output Delay Time)						
(ADM1)	(ADM0)											2.4 V ≤ V_{DD} ≤ 5.5 V					2.7 V ≤ V_{DD} ≤ 5.5 V
ADLSP	FR2	FR1	FR0	LV1	LV0							f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz	f_{CLK} = 48 MHz
0	0	0	0	0	0	Normal 1	f_{CLK}/32	1 f_{AD}	41 f_{AD}	1 f_{AD}	1376/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	43 μs	28.667 μs
0	0	0	1				f_{CLK}/16	1 f_{AD}	41 f_{AD}	1 f_{AD}	688/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	43 μs	21.5 μs	14.333 μs
0	0	1	0				f_{CLK}/8	1 f_{AD}	41 f_{AD}	1 f_{AD}	344/f_{CLK}	Setting prohibited	Setting prohibited	43 μs	21.5 μs	10.75 μs	7.1667 μs
0	0	1	1				f_{CLK}/4	1 f_{AD}	41 f_{AD}	1 f_{AD}	172/f_{CLK}	Setting prohibited	Setting prohibited	21.5 μs	10.75 μs	5.375 μs	3.5833 μs
0	1	0	0				f_{CLK}/2	1 f_{AD}	41 f_{AD}	1 f_{AD}	86/f_{CLK}	Setting prohibited	Setting prohibited	10.75 μs	5.375 μs	2.6875 μs	1.7917 μs
0	1	0	1				f_{CLK}	1 f_{AD}	41 f_{AD}	1 f_{AD}	43/f_{CLK}	Setting prohibited	Setting prohibited	5.375 μs	2.6875 μs	1.3438 μs	0.8958 μs
1	0	1	1				f_{CLK}/4	1 f_{AD}	41 f_{AD}	1 f_{AD}	172/f_{CLK}	Setting prohibited	43 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	1 f_{AD}	41 f_{AD}	1 f_{AD}	86/f_{CLK}	Setting prohibited	21.5 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	1 f_{AD}	41 f_{AD}	1 f_{AD}	43/f_{CLK}	43 μs	10.75 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited										

Note The listed value denotes the number of clock cycles for conversion when the setting of ADSPMOD[1:0] in the ADSPMOD register is 01B.

Caution 1. The A/D conversion time must be within the relevant range of conversion clock (f_{AD}) and conversion times (t_{CONV}) described in 43.6.1 A/D converter characteristics or 44.6.1 A/D converter characteristics.

Caution 2. Rewrite the FR[2:0] and LV[1:0] bits to different values while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion times do not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. The conversion start time applies when no contention is present. For details on the conversion start time when contention is present, see Note 2 for Figure 20 - 5. Additionally, the conversion times do not include clock frequency errors. Consider clock frequency errors when selecting the conversion time.

Caution 4. When the simultaneous sampling is to proceed, the following conditions must be met.
ADLSP = 0, FR[2:0] = 100, LV[1:0] = 00, f_{CLK} ≥ 32 MHz, V_{DD} ≥ 2.7 V
ADLSP = 0, FR[2:0] = 101, LV[1:0] = 00, f_{CLK} ≥ 16 MHz, V_{DD} ≥ 2.7 V

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 20 - 6 Selection of A/D Conversion Time (10/11)

10. Normal modes 1 and 2 with no A/D power supply stabilization wait time (in advanced mode) when ANI0 to ANI7, and ANI16 to ANI30 are to be A/D converted

A/D Converter Mode Register 0 A/D Converter Mode Register 1						Mode	Conversion Clock (f_{AD})	Number of Clock Cycles for Conversion Start Delay	Number of Clock Cycles for Conversion Note	Number of Clock Cycles for Interrupt Output Delay	A/D Conversion Time (Conversion Start Delay Time + Conversion Time + Interrupt Output Delay Time)						
(ADM1)	(ADM0)											2.4 V ≤ V_{DD} ≤ 5.5 V					2.7 V ≤ V_{DD} ≤ 5.5 V
ADLSP	FR2	FR1	FR0	LV1	LV0							f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz	f_{CLK} = 48 MHz
0	0	0	0	0	0	Normal 1	f_{CLK}/32	1 f_{AD}	48 f_{AD}	1 f_{AD}	1600/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	50 µs	33.333 µs
0	0	0	1				f_{CLK}/16	1 f_{AD}	48 f_{AD}	1 f_{AD}	800/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	50 µs	25 µs	16.667 µs
0	0	1	0				f_{CLK}/8	1 f_{AD}	48 f_{AD}	1 f_{AD}	400/f_{CLK}	Setting prohibited	Setting prohibited	50 µs	25 µs	12.5 µs	8.3333 µs
0	0	1	1				f_{CLK}/4	1 f_{AD}	48 f_{AD}	1 f_{AD}	200/f_{CLK}	Setting prohibited	Setting prohibited	25 µs	12.5 µs	6.25 µs	4.1667 µs
0	1	0	0				f_{CLK}/2	1 f_{AD}	48 f_{AD}	1 f_{AD}	100/f_{CLK}	Setting prohibited	Setting prohibited	12.5 µs	6.25 µs	3.125 µs	2.0833 µs
0	1	0	1				f_{CLK}	1 f_{AD}	48 f_{AD}	1 f_{AD}	50/f_{CLK}	Setting prohibited	Setting prohibited	6.25 µs	3.125 µs	1.5625 µs	1.0417 µs
1	0	1	1				f_{CLK}/4	1 f_{AD}	48 f_{AD}	1 f_{AD}	200/f_{CLK}	Setting prohibited	50 µs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	1 f_{AD}	48 f_{AD}	1 f_{AD}	100/f_{CLK}	Setting prohibited	25 µs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	1 f_{AD}	48 f_{AD}	1 f_{AD}	50/f_{CLK}	50 µs	12.5 µs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited										
0	0	0	0	0	1	Normal 2	f_{CLK}/32	1 f_{AD}	261 f_{AD}	1 f_{AD}	8416/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	263 µs	175.33 µs
0	0	0	1				f_{CLK}/16	1 f_{AD}	261 f_{AD}	1 f_{AD}	4208/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	263 µs	131.5 µs	87.667 µs
0	0	1	0				f_{CLK}/8	1 f_{AD}	261 f_{AD}	1 f_{AD}	2104/f_{CLK}	Setting prohibited	Setting prohibited	263 µs	131.5 µs	65.75 µs	43.833 µs
0	0	1	1				f_{CLK}/4	1 f_{AD}	261 f_{AD}	1 f_{AD}	1052/f_{CLK}	Setting prohibited	Setting prohibited	131.5 µs	65.75 µs	32.875 µs	21.917 µs
0	1	0	0				f_{CLK}/2	1 f_{AD}	261 f_{AD}	1 f_{AD}	526/f_{CLK}	Setting prohibited	Setting prohibited	65.75 µs	32.875 µs	16.438 µs	10.958 µs
0	1	0	1				f_{CLK}	1 f_{AD}	261 f_{AD}	1 f_{AD}	263/f_{CLK}	Setting prohibited	Setting prohibited	32.875 µs	16.438 µs	8.2188 µs	5.4792 µs
1	0	1	1				f_{CLK}/4	1 f_{AD}	261 f_{AD}	1 f_{AD}	1052/f_{CLK}	Setting prohibited	263 µs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	1 f_{AD}	261 f_{AD}	1 f_{AD}	526/f_{CLK}	Setting prohibited	131.5 µs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	1 f_{AD}	261 f_{AD}	1 f_{AD}	263/f_{CLK}	263 µs	65.75 µs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited										

Note The listed value denotes the number of clock cycles for conversion when the setting of ADSPMOD[1:0] in the ADSPMOD register is 00B.

Caution 1. The A/D conversion time must be within the relevant range of conversion clock (f_{AD}) and conversion times (t_{CONV}) described in 43.6.1 A/D converter characteristics or 44.6.1 A/D converter characteristics.

Caution 2. Rewrite the FR[2:0] and LV[1:0] bits to different values while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion times do not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. The conversion start time applies when no contention is present. For details on the conversion start time when contention is present, see Note 2 for Figure 20 - 5. Additionally, the conversion times do not include clock frequency errors. Consider clock frequency errors when selecting the conversion time.

Caution 4. When the internal reference voltage or the temperature sensor output voltage is selected as the conversion target, use normal mode 2.

(Cautions and Remark are listed on the next page.)

Caution 5. When the internal reference voltage is selected for the + side reference voltage, the conversion clock (f_{AD}) becomes 1 to 2 MHz.

Caution 6. When the PGA output is selected as the conversion target, use normal mode 1.

Caution 7. When the PGA is selected as the conversion target, the PGA stabilization wait time is automatically added. The stabilization wait time varies depending on the gain setting of the PGA.
× 4 to × 16: 5 f_{AD}
× 32: 36 f_{AD}

Caution 8. When the simultaneous sampling is to proceed, the following conditions must be met.
ADLSP = 0, FR[2:0] = 100, LV[1:0] = 00, f_{CLK} ≥ 32 MHz, V_{DD} ≥ 2.7 V
ADLSP = 0, FR[2:0] = 101, LV[1:0] = 00, f_{CLK} ≥ 16 MHz, V_{DD} ≥ 2.7 V

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 20 - 6 Selection of A/D Conversion Time (11/11)

11. Low voltage modes 1 and 2 with no A/D power supply stabilization wait time (advanced mode)

A/D Converter Mode Register 0 A/D Converter Mode Register 1						Mode	Conversion Clock (f_{AD})	Number of Clock Cycles for Conversion Start Delay	Number of Clock Cycles for Conversion	Number of Clock Cycles for Interrupt Output Delay	A/D Conversion Time (Conversion Start Delay Time + Conversion Time + Interrupt Output Delay Time)						
(ADM1)	(ADM0)											1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	1.8 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	2.7 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	2.7 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V
ADLSP	FR2	FR1	FR0	LV1	LV0							f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz	f_{CLK} = 48 MHz
0	0	0	0	1	0	Low voltage 1	f_{CLK}/32	1 f_{AD}	80 f_{AD}	1 f_{AD}	2624/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	82 µs	54.667 µs
0	0	0	1				f_{CLK}/16	1 f_{AD}	80 f_{AD}	1 f_{AD}	1312/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	82 µs	41 µs	27.333 µs
0	0	1	0				f_{CLK}/8	1 f_{AD}	80 f_{AD}	1 f_{AD}	656/f_{CLK}	Setting prohibited	Setting prohibited	82 µs	41 µs	20.5 µs	13.667 µs
0	0	1	1				f_{CLK}/4	1 f_{AD}	80 f_{AD}	1 f_{AD}	328/f_{CLK}	Setting prohibited	Setting prohibited	41 µs	20.5 µs	10.25 µs	6.8333 µs
0	1	0	0				f_{CLK}/2	1 f_{AD}	80 f_{AD}	1 f_{AD}	164/f_{CLK}	Setting prohibited	Setting prohibited	20.5 µs	10.25 µs	5.125 µs	3.4167 µs
0	1	0	1				f_{CLK}	1 f_{AD}	80 f_{AD}	1 f_{AD}	82/f_{CLK}	Setting prohibited	Setting prohibited	10.25 µs	5.125 µs	Setting prohibited	Setting prohibited
1	0	1	1				f_{CLK}/4	1 f_{AD}	80 f_{AD}	1 f_{AD}	328/f_{CLK}	Setting prohibited	82 µs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	1 f_{AD}	80 f_{AD}	1 f_{AD}	164/f_{CLK}	Setting prohibited	41 µs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	1 f_{AD}	80 f_{AD}	1 f_{AD}	82/f_{CLK}	82 µs	20.5 µs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited										
0	0	0	0	1	1	Low voltage 2	f_{CLK}/32	1 f_{AD}	107 f_{AD}	1 f_{AD}	3488/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	109 µs	72.667 µs
0	0	0	1				f_{CLK}/16	1 f_{AD}	107 f_{AD}	1 f_{AD}	1744/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	109 µs	54.5 µs	36.333 µs
0	0	1	0				f_{CLK}/8	1 f_{AD}	107 f_{AD}	1 f_{AD}	872/f_{CLK}	Setting prohibited	Setting prohibited	109 µs	54.5 µs	27.25 µs	18.167 µs
0	0	1	1				f_{CLK}/4	1 f_{AD}	107 f_{AD}	1 f_{AD}	436/f_{CLK}	Setting prohibited	Setting prohibited	54.5 µs	27.25 µs	13.625 µs	9.0833 µs
0	1	0	0				f_{CLK}/2	1 f_{AD}	107 f_{AD}	1 f_{AD}	218/f_{CLK}	Setting prohibited	Setting prohibited	27.25 µs	13.625 µs	6.8125 µs	4.5417 µs
0	1	0	1				f_{CLK}	1 f_{AD}	107 f_{AD}	1 f_{AD}	109/f_{CLK}	Setting prohibited	Setting prohibited	13.625 µs	6.8125 µs	Setting prohibited	Setting prohibited
1	0	1	1				f_{CLK}/4	1 f_{AD}	107 f_{AD}	1 f_{AD}	436/f_{CLK}	Setting prohibited	109 µs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	1 f_{AD}	107 f_{AD}	1 f_{AD}	218/f_{CLK}	Setting prohibited	54.5 µs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	1 f_{AD}	107 f_{AD}	1 f_{AD}	109/f_{CLK}	109 µs	27.25 µs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited										

Caution 1. The A/D conversion time must be within the relevant range of conversion clock (f_{AD}) and conversion times (t_{CONV}) described in 43.6.1 A/D converter characteristics or 44.6.1 A/D converter characteristics.

Caution 2. Rewrite the FR[2:0] and LV[1:0] bits to different values while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion times do not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. The conversion start time applies when no contention is present. For details on the conversion start time when contention is present, see Note 2 for Figure 20 - 5. Additionally, the conversion times do not include clock frequency errors. Consider clock frequency errors when selecting the conversion time.

Caution 4. When the internal reference voltage or the temperature sensor output voltage is selected as the conversion target, use low voltage mode 2 with the conversion clock (f_{AD}) with a frequency of no more than 16 MHz.

(Cautions and Remark are listed on the next page.)

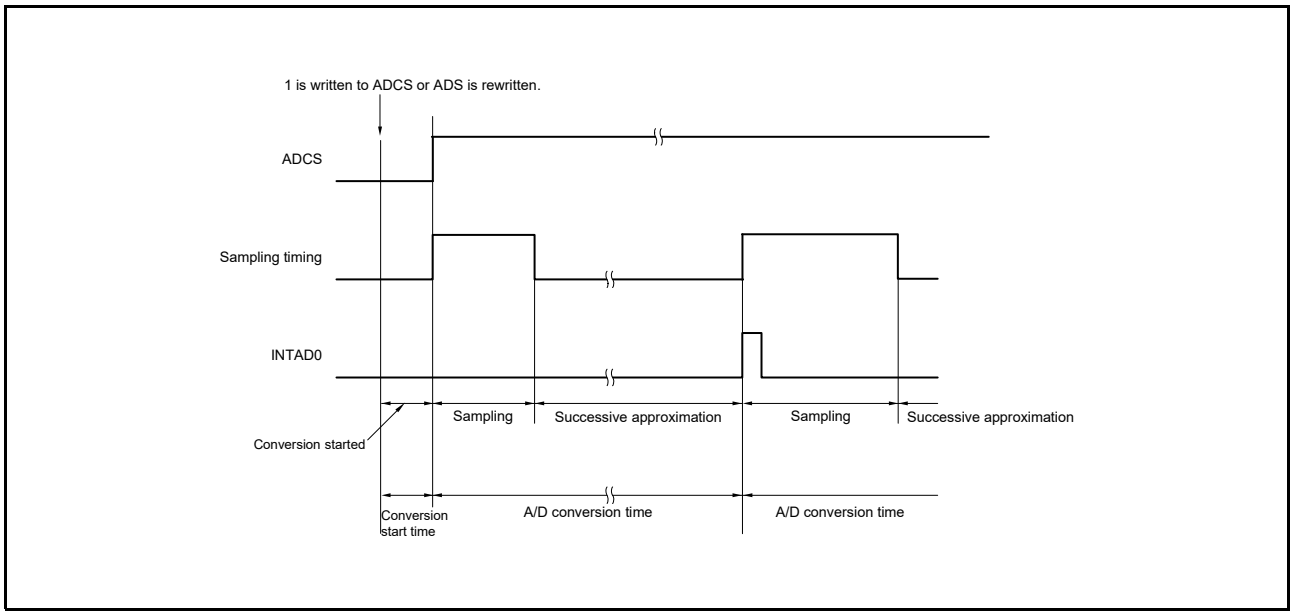
Caution 5. When the internal reference voltage is selected for the + side reference voltage, the conversion clock (f_{AD}) becomes 1 to 2 MHz.

Caution 6. When the PGA output is selected as the conversion target, use low voltage mode 1.

Caution 7. Simultaneous sampling is prohibited in the low voltage modes 1 and 2.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Figure 20 - 6 A/D Converter Sampling and A/D Conversion Timing (Example for Software Trigger No-wait Mode)



20.3.4 A/D converter mode register 1 (ADM1)

The ADM1 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. The value of this register following a reset is 00H.

Figure 20 - 7 Format of A/D Converter Mode Register 1 (ADM1)

Address: FFF32H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ADM1	ADTMD1	ADTMD0	ADSCM	0	ADLSP	ADTRS2	ADTRS1	ADTRS0

ADTMD1	ADTMD0	Selection of the A/D conversion trigger mode
0	×	Software trigger no-wait mode or software trigger wait mode
1	0	Hardware trigger no-wait mode
1	1	Hardware trigger wait mode

ADLSP	fCLK input frequency setting
0	4 MHz < fCLK ≤ 48 MHz
1	1 MHz ≤ fCLK ≤ 4 MHz

ADSCM	Specification of the A/D conversion mode[Note 1]
0	Sequential conversion mode
1	One-shot conversion mode

ADTRS2	ADTRS1	ADTRS0	Selection of the hardware trigger signal
0	0	0	Channel 01 timer array unit counting or capture end interrupt signal (INTTM01)
0	1	0	Realtime clock interrupt signal (INTRTC)
0	1	1	32-bit interval timer channel 0 interrupt signal (ELCITL0)
1	0	0	Event input from the ELC[Note 2]
Setting values other than the above is prohibited.			

<R> **Note 1.** When the advanced mode is enabled, selecting the sequential conversion mode is prohibited. Set the ADSCM bit to 1 (one-shot conversion mode).

Note 2. Event input from the ELC cannot be used in the SNOOZE mode.

Caution 1. Only rewrite the value of the ADM1 register while conversion operation 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 cycles of the fCLK clock + conversion start time + A/D conversion time
Hardware trigger wait mode: 2 cycles of the fCLK 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 INTRTC or ELCITL0 will not be recognized as a valid hardware trigger for up to four cycles of the fCLK clock after the first INTRTC or ELCITL0 is input.

Caution 4. The settings of the ADTMD[1:0] and ADTRS[2:0] bits are only effective when the advanced mode is disabled. Do not change their values from the initial values when the advanced mode is enabled.

(Remarks are listed on the next page.)

Remark 1. ×: Don't care.

Remark 2. f_{CLK}: CPU/peripheral hardware clock frequency

20.3.5 A/D converter mode register 2 (ADM2)

The ADM2 register is used to select the + side and - side reference voltages 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. The value of this register following a reset is 00H.

Figure 20 - 8 Format of A/D Converter Mode Register 2 (ADM2) (1/2)

Address: F0010H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	<3>	<2>	<1>	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	ADTYP1	ADTYP0

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**Note 1**
1	1	Discharged

- Use the following procedure to rewrite the ADREFP[1:0] bits.
  - <1> Set ADCE = 0.
  - <2> Set ADREFP[1:0] to 11B.
    This step is only necessary when the value of the ADREFP[1:0] bits is changed to 10B.
  - <3> Reference voltage discharge time: 1 μs
    This step is only necessary when the value of the ADREFP[1:0] bits is changed to 10B.
  - <4> Change the value of the ADREFP[1:0] bit.
  - <5> Reference voltage stabilization wait time A
  - <6> Set ADCE = 1.
  - <7> Reference voltage stabilization wait time B
    When the ADREFP[1:0] bits are set to 10B, A = 5 μs and B = 1 μs + 2 cycles of the conversion clock (f_{AD}).
    When the ADREFP[1:0] bits are set to 00B or 01B, a wait A is not required and B = 1 μs + 2 cycles of the conversion clock (f_{AD}).

  After <7> stabilization time, start the A/D conversion.
- When the ADREFP[1:0] bits are set to 10B, A/D conversion of the temperature sensor output voltage and internal reference voltage**Note 1** cannot proceed.
  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

Figure 20 - 8 Format of A/D Converter Mode Register 2 (ADM2) (2/2)

ADRCK	Checking the upper limit and lower limit conversion result values
0	The interrupt signals (INTAD0 to INTAD3) are output when the ADLL register ≤ the ADCRn register ≤ the ADUL register (AREA 1).
1	The interrupt signals (INTAD0 to INTAD3) are output when the ADCRn register < the ADLL register (AREA 2) and the ADUL register < the ADCRn register (AREA 3).

Figure 20 - 9 shows the generation range of the interrupt signals (INTAD0 to INTAD3) for AREA 1 to AREA 3.

AWC	Specification of the SNOOZE mode
0	Does not use the SNOOZE mode function.
1	Uses 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 or middle-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.
- When using the SNOOZE mode function, set AWC to 0 in software trigger wait mode, and set AWC to 1 in hardware trigger wait mode.
- Using the SNOOZE mode function in the software trigger no-wait mode, hardware trigger no-wait mode, or advanced mode is prohibited.
- Using the SNOOZE mode function in hardware trigger wait mode in sequential conversion mode is prohibited.
- When using the SNOOZE mode function, specify a hardware trigger interval of at least “transition time to SNOOZE mode[Note 2] + conversion start time + A/D power supply stabilization wait time + A/D conversion time + 2 cycles of the f_{CLK} clock”.
- Even when using the SNOOZE mode, be sure to set the AWC bit to 0 in normal operation and change it to 1 just before shifting to STOP mode.
  Also, be sure to change the AWC bit to 0 after returning from STOP mode to normal operation.
  If the AWC bit is left set to 1, A/D conversion will not start normally in spite of the subsequent SNOOZE mode or normal operation.

ADTYP1	ADTYP0	Selection of the A/D conversion resolution
0	0	10-bit resolution
0	1	8-bit resolution
1	0	12-bit resolution
1	1	Setting prohibited

Note 1. For details about the internal reference voltage, see **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** and **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.

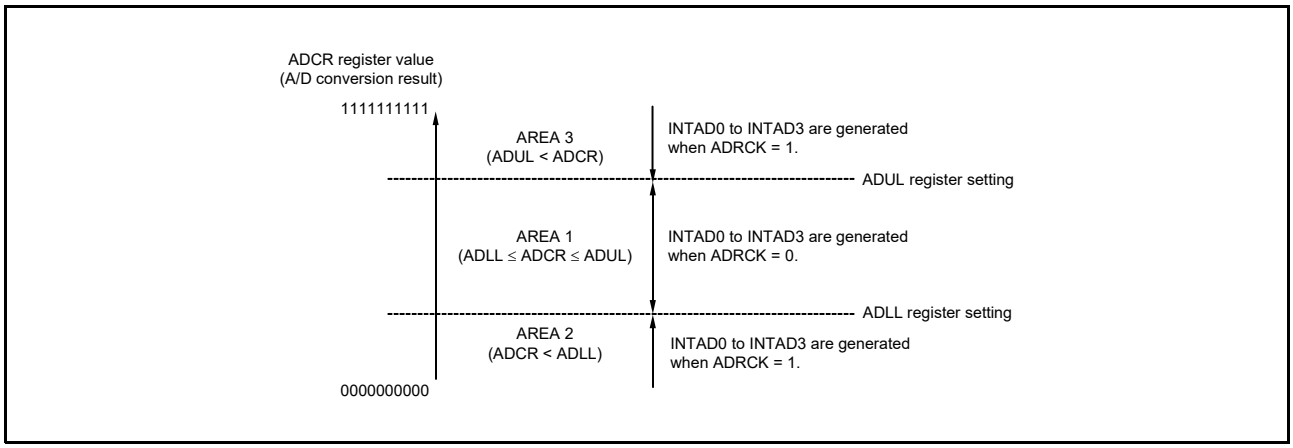
Note 2. Refer to “**Transition time from STOP mode to SNOOZE mode**” in **31.3.3 SNOOZE mode**.

Caution 1. Only rewrite the value of the ADM2 register while conversion operation is stopped (ADCS = 0, ADCE = 0).

Caution 2. Do not set the ADREFP1 bit to 1 when shifting to STOP mode, or to HALT mode while the CPU is operating with the subsystem clock. When the internal reference voltage is selected (ADREFP[1:0] = 10B), the A/D converter reference voltage current (I_{ADREF}) indicated in 43.3.2 Supply current characteristics or 44.3.2 Supply current characteristics will be 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 20 - 9 Interrupt Signal Generation Range According to the Setting of the ADRCK Bit



Remark If INTAD0 to INTAD3 are not generated due to the combination of the settings of the ADUL and ADLL registers and the ADRCK bit, the result of A/D conversion is not stored in the ADCRn and ADCRnH registers.

20.3.6 A/D converter mode register 3 (ADM3)

The ADM3 register is used to generate a software trigger and enable or disable the advanced mode. The ADM3 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 20 - 10 Format of A/D Converter Mode Register 3 (ADM3)

Address: F0014H
After reset: 00H
R/W: R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM3	ADTRSWT	0	0	0	0	0	0	ADVMOD

ADTRSWT	Software trigger
0	A software trigger is not generated.
1	A software trigger is generated. This bit is automatically cleared to 0 after it has been set to 1.Note

ADVMOD	Setting of the advanced mode
0	The advanced mode is disabled.
1	The advanced mode is enabled.

Note The ADTRSWT bit is always read as 0.

Caution **<Timing of ADVMOD setting>**
Change the value of the ADVMOD bit while conversion operations are stopped and before setting the other operation control registers.
Since a change to the ADVMOD setting does not require applying a reset to other registers, clear or set the related registers to suit the state according to whether the advanced mode is enabled or disabled.
<Setting the ADTRSWT and ADVMOD bits at the same time is prohibited>
While ADTRSWT = 0 and ADVMOD = 0, do not change the values of these bits from 0 to 1 at the same time by using an 8-bit memory manipulation instruction. Be sure to set the ADVMOD bit to 1 before setting the ADTRSWT bit.

20.3.7 12-bit/10-bit A/D conversion result register and registers (ADCR, ADCRn) (n = 0 to 3)

The ADCR and ADCRn registers are 16-bit registers that hold the A/D conversion result. When A/D conversion with 12-bit resolution is selected, the higher 4 bits are fixed to 0. When A/D conversion with 10-bit resolution is selected, 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 ADCRn registers can be read by a 16-bit memory manipulation instruction. The value of each of these registers following a reset is 0000H.

In select mode when the advanced mode is disabled, the conversion results are stored in the ADCR and ADCR0 registers**Note**. In scan mode when the advanced mode is disabled, the conversion results of scan 0 are stored in the ADCR and ADCR0 registers, and the conversion results of scan 1 to 3 are stored in the ADCR1 to ADCR3 registers**Note**. When the advanced mode is enabled, the conversion results of the channel selected by the ADSn register are stored in the ADCRn register.

Note If the A/D conversion result is outside the range specified by using the A/D conversion result comparison function (set up by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see **Figure 20 - 9 Interrupt Signal Generation Range According to the Setting of the ADRCK Bit**), the result is not stored.

Figure 20 - 11 Format of 12-bit/10-bit A/D Conversion Result Register and Registers (ADCR, ADCRn)

Address: FFF1FH, FFF1EH (ADCR)**Note**, F0021H, F0020H (ADCR0)**Note**, F0023H, F0022H (ADCR1), F0025H, F0024H (ADCR2), F0027H, F0026H (ADCR3)

After reset: 0000H

R/W: R

When A/D conversion with 12-bit resolution is selected

	FFF1FH (for ADCR)								FFF1EH (for ADCR)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCRn	0	0	0	0												

When A/D conversion with 10-bit resolution is selected

	FFF1FH (for ADCR)								FFF1EH (for ADCR)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCRn											0	0	0	0	0	0

Note The contents of the ADCR register are stored in the ADCR0 register. In advanced mode, refer to the ADCR0 register.

Caution 1. When 8-bit resolution A/D conversion is selected (ADM2.ADTYP[1:0] = 01B) and the ADCRn register is read, 0 is read from the bits other than the higher 8 bits.

Caution 2. When the ADCRn register is accessed in 16-bit units, and A/D conversion with 10-bit resolution is selected, the higher 10 bits of the conversion result are read in order starting at bit 15 of the ADCRn register.
When A/D conversion with 12-bit resolution is selected, the higher 12 bits of the conversion result are read in order starting at bit 11 of the ADCRn register.

(Caution and Remark are listed on the next page.)

Caution 3. Writing to any of the registers related to the ANI signal, that is, the A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), analog input channel specification register for advanced mode n (ADSn), or port mode control A register xx (PMCAxx), may lead to the contents of the ADCRn register becoming undefined. Read conversion results after conversion before writing to any of the ADM0, ADS, ADSn, PMCAxx registers.
Using timing other than the above may cause an incorrect conversion result to be read.

Remark n = 0 to 3

20.3.8 8-bit A/D conversion result registers H, nH (ADCRH, ADCRnH) (n = 0 to 3)

The ADCRH and ADCRnH registers are 8-bit registers that hold the A/D conversion result. The higher 8 bits of 12-bit resolution are stored[Note]. The ADCRnH registers can be read by an 8-bit memory manipulation instruction. The value of each of these registers following a reset is 00H.

Note If the A/D conversion result is outside the range specified by using the A/D conversion result comparison function (set up by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see **Figure 20 - 9 Interrupt Signal Generation Range According to the Setting of the ADRCK Bit**), the result is not stored.

Figure 20 - 12 Format of 8-bit A/D Conversion Result Registers H, nH (ADCRH, ADCRnH)

Address: FFF1FH (ADCRH)[Note], F0021H (ADCR0H)[Note], F0023H (ADCR1H), F0025H (ADCR2H), F0027H (ADCR3H)

After reset: 00H

R/W: R

Symbol	7	6	5	4	3	2	1	0
ADCRnH								

Note The contents of the ADCRH register are stored in the ADCR0H register.

Caution **Writing to any of the registers related to the ANI signal, that is, the A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), analog input channel specification register for advanced mode n (ADSn), or port mode control A register xx (PMCAxx), may lead to the contents of the ADCRnH register becoming undefined. Read conversion results after conversion before writing to any of the ADM0, ADS, ADSn, PMCAxx registers.**
Using timing other than the above may cause an incorrect conversion result to be read.

Remark n = 0 to 3

20.3.9 Analog input channel specification register (ADS)

The ADS 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. The value of this register following a reset is 00H.

Figure 20 - 13 Format of Analog Input Channel Specification Register (ADS) (1/2)

Address: FFF31H
After reset: 00H
R/W: 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	P03/ANI16 pin
0	1	0	0	0	1	ANI17	P02/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	ANI20	P10/ANI20 pin
0	1	0	1	0	1	ANI21	P11/ANI21 pin
0	1	0	1	1	0	ANI22	P12/ANI22 pin
0	1	0	1	1	1	ANI23	P13/ANI23 pin
0	1	1	0	0	0	ANI24	P14/ANI24 pin
0	1	1	0	0	1	ANI25	P115/ANI25 pin
0	1	1	0	1	0	ANI26	P16/ANI26 pin
0	1	1	0	1	1	ANI27	P17/ANI27 pin
0	1	1	1	0	0	ANI28	P146/ANI28 pin
0	1	1	1	0	1	ANI29	P00/ANI29 pin
0	1	1	1	1	0	ANI30	P01/ANI30 pin
1	0	0	0	0	0	—	Temperature sensor output voltage
1	0	0	0	0	1	—	Internal reference voltage Note 1
Other than the above						Setting prohibited Note 2	

Figure 20 - 13 Format of Analog Input Channel Specification Register (ADS) (2/2)

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
Other than the above						Setting prohibited			

Note 1. For details about the internal reference voltage, see **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** and **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.

Note 2. Only the advanced mode supports conversion of the PGA signals.

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

Caution 2. Set the port that is specified as the analog input by the port mode control A register xx (PMCAxx) to the input mode by using the corresponding port mode register xx (PMxx).

Caution 3. When specifying an input channel by the ADS register, do not select the pin that is specified as digital I/O by the port mode control A register xx (PMCAxx).

Caution 4. Rewrite the value of the ADISS bit while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 5. 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 6. 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 7. If the ADISS bit is set to 1, the internal reference voltage 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 detailed setting flow, see 20.7.6 Settings when temperature sensor output voltage or internal reference voltage is selected (example for software trigger no-wait mode and one-shot conversion mode).
For details about the internal reference voltage, see Section 43 Electrical Characteristics (T_A = –40 to +105°C) or Section 44 Electrical Characteristics (T_A = –40 to +125°C).

Caution 8. Do not set the ADISS bit to 1 when shifting to STOP mode, or to HALT mode while the CPU is operating with the subsystem clock. When the ADISS bit is set to 1, the A/D converter reference voltage current (I_{ADREF}) indicated in 43.3.2 Supply current characteristics or 44.3.2 Supply current characteristics will be added.

Caution 9. When the ADISS bit is set to 1, the hardware trigger wait mode and one-shot conversion mode are not available.

Caution 10. Set the ADS register to 00H when the advanced mode is enabled (ADM3.ADVMOD = 1).

20.3.10 Analog input channel specification registers n for advanced mode (ADSn) (n = 0 to 3)

The ADSn registers are used to specify the input channels of the analog voltages to be A/D converted in advanced mode. The ADSn registers can be set by an 8-bit memory manipulation instruction. The value of each of these registers following a reset is 00H.

Figure 20 - 14 Format of Analog Input Channel Specification Registers n for Advanced Mode (ADSn) (1/2)

Address: F0015H (ADS0), F0016H (ADS1), F0017H (ADS2), F0018H (ADS3)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ADSn	ADSPSCn1	ADSPSCn0	ADISSn	ADSn4	ADSn3	ADSn2	ADSn1	ADSn0

ADSPSCn1	ADSPSCn0	Channel specification for simultaneous sampling
0	0	Simultaneous sampling does not proceed.
0	1	The first S&H circuit is used.
1	0	The second S&H circuit is used.
1	1	The third S&H circuit is used.

ADISSn	ADSn4	ADSn3	ADSn2	ADSn1	ADSn0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P20/ANI0/AVREFP pin
	0	0	0	0	1	ANI1	P21/ANI1/AVREFM pin
	0	0	0	1	0	ANI2	P22/ANI2 pin
	0	0	0	1	1	ANI3	P23/ANI3 pin
	0	0	1	0	0	ANI4	P24/ANI4 pin
	0	0	1	0	1	ANI5	P25/ANI5 pin
	0	0	1	1	0	ANI6	P26/ANI6 pin
	0	0	1	1	1	ANI7	P27/ANI7 pin
	1	0	0	0	0	ANI16	P03/ANI16 pin
	1	0	0	0	1	ANI17	P02/ANI17 pin
	1	0	0	1	0	ANI18	P147/ANI18 pin
	1	0	0	1	1	ANI19	P120/ANI19 pin
	1	0	1	0	0	ANI20	P10/ANI20 pin
	1	0	1	0	1	ANI21	P11/ANI21 pin
	1	0	1	1	0	ANI22	P12/ANI22 pin
	1	0	1	1	1	ANI23	P13/ANI23 pin
	1	1	0	0	0	ANI24	P14/ANI24 pin
	1	1	0	0	1	ANI25	P15/ANI25 pin
	1	1	0	1	0	ANI26	P16/ANI26 pin

Figure 20 - 14 Format of Analog Input Channel Specification Registers for Advanced Mode (ADSn) (2/2)

ADISSn	ADSn4	ADSn3	ADSn2	ADSn1	ADSn0	Analog input channel	Input source
0	1	1	0	1	1	ANI27	P17/ANI27 pin
	1	1	1	0	0	ANI28	P146/ANI28 pin
	1	1	1	0	1	ANI29	P00/ANI29 pin
	1	1	1	1	0	ANI30	P01/ANI30 pin
	1	1	1	1	1	—	PGA input
1	0	0	0	0	0	—	Temperature sensor output voltage
	0	0	0	0	1	—	Internal reference voltage
Other than the above						Setting prohibited	

Caution 1. When the setting of the ADSPSCn[1:0] bits is 10B or 11B, do not change the settings of the ADISSn and ADSn[4:0] bits from the initial values.

Caution 2. Do not set the same value to the ADSPSCn[1:0] bits of the ADS0 to ADS3 registers. Note, however, that setting of 00B is always allowed. Set the ADSPSCn[1:0] bits to 00B unless the simultaneous sampling is to be used.

<R> **Caution 3. Before rewriting the value of the ADSn register, take measures such as disabling the corresponding hardware trigger to prevent the triggering of A/D conversion. Alternatively, discard the result of the first conversion after having rewritten the given register and use the second and subsequent results.**

Caution 4. Set the port that is specified as the analog input by the port mode control A register xx (PMCAxx) to the input mode by using the corresponding port mode register xx (PMxx).

Caution 5. When specifying an input channel by the ADSn register, do not select the pin that is specified as digital I/O by the port mode control A register xx (PMCAxx).

Caution 6. 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 7. 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 8. If the ADISSn bit is set to 1, the internal reference voltage cannot be used for the + side reference voltage. After the ADISSn bit is set to 1, the initial conversion result cannot be used. For the detailed setting flow, see 20.7.7 Settings when temperature sensor output voltage or internal reference voltage is selected (example for advanced mode).
For details about the internal reference voltage, see Section 43 Electrical Characteristics (T_A = –40 to +105°C) or Section 44 Electrical Characteristics (T_A = –40 to +125°C).

Caution 9. Using two from among ADS0 to ADS3 to make the setting for conversion of the internal reference voltage (ADISSn = 1, ADSn[4:0] = 00001B) and temperature sensor output voltage (ADISSn = 1, ADSn[4:0] = 00000B) at the same time is prohibited. (Example: ADS0 = 00_1_00000B, ADS1 = 00_1_00001B)

Caution 10. Selecting PGA input as the target for conversion in normal 2 mode and low voltage 2 mode is prohibited.

Caution 11. Setting an ANI signal that is used by the PGA, DAC, CMP or other on-chip peripheral modules as the target for A/D conversion is prohibited.

Caution 12. When the ADSPSCn[1:0] bits are to be set to 10B or 11B, set the ADSn[4:0] bits to 00000B.

Remark n = 0 to 3

20.3.11 Conversion setting register (ADSCTL)

The ADSCTL register is used to enable or disable the ADSn registers and specify the priorities for conversion of the ANI channels selected by the ADSn registers in the advanced mode. The ADSCTL register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 20 - 15 Format of Conversion Setting Register (ADSCTL)

Address: F0019H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
ADSCTL	ADSEN3	ADSEN2	ADSEN1	ADSEN0	ADPRIOR3	ADPRIOR2	ADPRIOR1	ADPRIOR0

ADSENn	Enabling the ADSn register
0	Disables the ADSn register.
1	Enables the ADSn register.

ADPRIORn	Priority of conversion for the channel specified by the ADSn register
0	The channel specified by the ADSn register has low priority.
1	The channel specified by the ADSn register has high priority.

Caution 1. Only rewrite the value of the ADTRn register while conversion operation is stopped (ADCS = 0, ADCE = 0).

Caution 2. In simultaneous sampling, settings of the priority of conversion for the channel specified as the first, second, and third S&H circuits should be 0 (initial value).

Remark 1. If the ADSn registers are assigned the same priority and the respective trigger sources for conversion are also the same, conversion proceeds in ascending order of the register number after the trigger source has been generated. An example is given below.

- When a single trigger source is specified for the ADSn registers while ADPRIOR3 = 0, ADPRIOR2 = 0, and ADPRIOR1 = 0, conversion proceeds in the order ADS1, ADS2, and ADS3 in response to generation of the trigger source. Note, however, that in simultaneous sampling, conversion proceeds in the order the first, second, and third S&H circuits regardless of the register number.

Remark 2. n = 0 to 3

20.3.12 Conversion trigger specification registers n (ADTRn) (n = 0 to 3)

The ADTRn registers are used to specify the trigger sources for the channels specified by each ADSn register in the advanced mode. The ADTRn register can be set by an 8-bit memory manipulation instruction. The value of each of these registers following a reset is 00H.

Figure 20 - 16 Format of Conversion Trigger Specification Registers n (ADTRn)

Address: F001AH (ADTR0), F001BH (ADTR1), F001CH (ADTR2), F001DH (ADTR3)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ADTRn	0	0	0	0	ADTRSn3	ADTRSn2	ADTRSn1	ADTRSn0

ADTRSn3	ADTRSn2	ADTRSn1	ADTRSn0	Trigger Source (n = 0 to 3)
0	0	0	0	Channel 01 timer array unit counting or capture end interrupt signal (INTTM01)
0	0	1	0	Realtime clock interrupt signal (INTRTC)
0	0	1	1	32-bit interval timer channel 0 interrupt signal (ELCITL0)
0	1	0	0	Event input from the ELC
1	0	0	0	16-bit timer KB30 A/D trigger signal
1	0	0	1	16-bit timer KB31 A/D trigger signal
1	0	1	0	16-bit timer KB32 A/D trigger signal
1	0	1	1	Timer RD2 A/D conversion trigger 0
1	1	0	0	Timer RD2 A/D conversion trigger 1
1	1	1	1	Software trigger
Other than the above				Setting prohibited

Caution 1. Only rewrite the value of the ADTRn register while conversion operation is stopped (ADCS = 0, ADCE = 0).

Caution 2. When a single trigger is to be specified for different ADSn registers, specify the same priority for the given ADSn registers. The following shows an example of violations. Do not make such settings as follows.
ADTR0 = ADTR1 = 03H, and ADSCTL.ADPRIOR0 = 0, ADSCTL.ADPRIOR1 = 1

Remark n = 0 to 3

20.3.13 Conversion interrupt control register (ADINTCTL)

The ADINTCTL register is used to control output of the INTAD0 to INTAD3 interrupts and storage of the conversion state information in the ADINTST register in the advanced mode. The setting of this register does not affect the interrupt (ELCADn (n = 0 to 3)) output signals for the ELC generated on successful completion of A/D conversion. The ADINTCTL register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 20 - 17 Format of Conversion Interrupt Control Register (ADINTCTL)

Address: F0028H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
ADINTCTL	ADINT CTL3S	ADINT CTL3F	ADINT CTL2S	ADINT CTL2F	ADINT CTL1S	ADINT CTL1F	ADINT CTL0S	ADINT CTL0F

ADINT CTLnS	Control when conversion was successful
0	Output of the INTAD0 to INTAD3 signals in response to successful completion of conversion for the analog input channel specified by ADSn and storage of the conversion state information in the ADINTST register are disabled.
1	Output of the INTAD0 to INTAD3 signals in response to successful completion of conversion for the analog input channel specified by ADSn and storage of the conversion state information in the ADINTST register are enabled.

ADINT CTLnF	Control when conversion failed
0	Storage of the state information for the analog input channel specified by ADSn in the ADINTST register in response to failure in conversion is disabled.
1	Storage of the state information for the analog input channel specified by ADSn in the ADINTST register in response to failure in conversion is enabled.

Caution **Only rewrite the value of the ADINTCTL register while conversion operation is stopped (ADCS = 0, ADCE = 0).**

Remark n = 0 to 3

20.3.14 Conversion interrupt status register (ADINTST)

The ADINTST register is used to indicate the states of conversion results in the advanced mode. The ADINTST register can be set by an 8-bit memory manipulation instruction, and can be read by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 20 - 18 Format of Conversion Interrupt Status Register (ADINTST)

Address: F0029H
After reset: 00H
R/W: R/W**Note**

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
ADINTST	ADINT ST3S	ADINT ST3F	ADINT ST2S	ADINT ST2F	ADINT ST1S	ADINT ST1F	ADINT ST0S	ADINT ST0F

ADINT STnS	State when conversion was successful
0	Conversion for the analog input channel specified by ADSn has not been completed.
1	Conversion for the analog input channel specified by ADSn has been completed.

ADINT STnF	State when conversion failed
0	Conversion for the analog input channel specified by ADSn did not fail.
1	Conversion for the analog input channel specified by ADSn failed.

Note Writing 1 to any bit of this register is not possible. Writing 0 to a bit of this register clears the bit to 0. When clearing a bit, use an 8-bit memory manipulation instruction to set the bit to be cleared to 0 and dummy-write 1 to the other bits.

Remark 1. n = 0 to 3

Remark 2. Conversion judged failed are given below.

- Low-priority conversion that was interrupted by a high-priority interrupt.
- Conversion that was interrupted by setting the ADCS bit to 0 while the conversion was in progress

20.3.15 A/D conversion sampling mode specification register (ADSPMOD)

The ADSPMOD register is used to change the number of clock cycles for sampling in the advanced mode. When ANI0 to ANI7 are only in use for analog input channels, the number of sampling clock cycles for A/D conversion can be set to 20 f_{AD}. If the number of sampling clock cycles is set to 20 f_{AD}, A/D conversion is completed in 1 μs when the ADC is operating with the sampling clock frequency of 48 MHz. The ADSPMOD register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 20 - 19 Format of A/D Conversion Sampling Mode Specification Register (ADSPMOD)

Address: F001FH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ADSPMOD	0	0	0	0	0	0	ADSPMOD1	ADSPMOD0

ADSPMOD1	ADSPMOD0	Specification of the number of sampling clock cycles for A/D conversion
0	0	27 f_{AD}
0	1	20 f_{AD}**Note**
1	0	Setting prohibited
1	1	Setting prohibited

Note
- This setting is only allowed in normal mode 1.
- Do not set ADSPMOD[1:0] to 01B when any of the following analog input channel is specified in the ADSn register enabled by the ADSCTL register.
  - ANI16 to ANI30
  - PGA output
  - Temperature sensor output voltage
  - Internal reference voltage

<R> **Caution** **Do not rewrite the value of the ADSPMOD register during A/D conversion. Also, before rewriting the value, take measures such as disabling the corresponding hardware trigger to prevent the triggering of A/D conversion while the A/D converter is in the conversion standby state.**

20.3.16 Conversion result comparison upper limit setting register (ADUL)

The ADUL 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 (INTAD0 to INTAD3) generation is controlled in the range specified by the ADRCK bit of A/D converter mode register 2 (ADM2) (see **Figure 20 - 9 Interrupt Signal Generation Range According to the Setting of the ADRCK Bit**). The ADUL register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is FFH.

Figure 20 - 20 Format of Conversion Result Comparison Upper Limit Setting Register (ADUL)

Address: F0011H
After reset: FFH
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ADUL	ADUL7	ADUL6	ADUL5	ADUL4	ADUL3	ADUL2	ADUL1	ADUL0

20.3.17 Conversion result comparison lower limit setting register (ADLL)

The ADLL 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 (INTAD0 to INTAD3) generation is controlled in the range specified by the ADRCK bit of A/D converter mode register 2 (ADM2) (see **Figure 20 - 9 Interrupt Signal Generation Range According to the Setting of the ADRCK Bit**). The ADLL register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 20 - 21 Format of Conversion Result Comparison Lower Limit Setting Register (ADLL)

Address: F0012H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ADLL	ADLL7	ADLL6	ADLL5	ADLL4	ADLL3	ADLL2	ADLL1	ADLL0

Caution 1. When A/D conversion with 10-bit resolution is selected, the ADCRn[15:8] bit value is compared with the values in the ADUL and ADLL registers. When A/D conversion with 12-bit resolution is selected, the ADCRn[11:4] bit value is compared with the values in 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 register must be greater than that of the ADLL register.

20.3.18 A/D test register (ADTES)

The ADTES register is used to select the + side reference voltage or - side reference voltage for the A/D converter, an analog input channel (ANIxx), the temperature sensor output voltage, or the internal reference voltage[Note] as the target for A/D conversion. Using the A/D test register is prohibited in advanced mode. Doing so creates a risk of damage to the circuits. When using this register to test the A/D 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.

The ADTES register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 20 - 22 Format of A/D Test Register (ADTES)

Address: F0013H
After reset: 00H
R/W: 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/internal reference voltage[Note] (set by 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 ADREFP[1:0] bits of the ADM2 register)
Other than the above		Setting prohibited

Note For details about the internal reference voltage, see **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** or **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.

20.3.19 Registers for controlling the port function of analog input pins

Set the following registers to control the port functions multiplexed with the analog inputs of the A/D converter.

- Port mode registers xx (PMxx)
- Port mode control A registers xx (PMCAxx)

For details, see **7.3.1 Port mode registers xx (PMxx)** and **7.3.7 Port mode control A registers xx (PMCAxx)**.

When the pins multiplexed with ANI0 to ANI7 and ANI16 to ANI30 are to be used for analog inputs of the A/D converter, set the port mode register xx (PMxx) bit and port mode control A register xx (PMCAxx) bit corresponding to each port to 1.

Remark xx = 0 to 2, 12, 14

20.4 A/D Converter Operations

The A/D converter conversion operations are described below.

<1> The voltage input to the selected analog input channel is sampled by the S&H circuit.

<2> When sampling has been done for a certain time, the S&H circuit is placed in the hold state and the sampled voltage is held until the A/D conversion operation has ended.

<3> Bit 11 of the successive approximation register (SAR) is set to 1. The series resistor string voltage tap is set to 1/2 AVREF 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 AVREF, the MSB of the SAR register remains set to 1. If the analog input is smaller than 1/2 AVREF, the MSB is reset to 0.

<5> Next, bit 10 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 value of bit 11, to which the result of comparison has already been set, as described below.

- Bit 11 = 1: (3/4) AVREF
- Bit 11 = 0: (1/4) AVREF

The voltage tap and sampled voltage are compared and bit 10 of the SAR register is manipulated according to the result of the comparison as follows.

- Sampled voltage ≥ Voltage tap: Bit 10 = 1
- Sampled voltage < Voltage tap: Bit 10 = 0

<6> Comparison is continued in this way up to bit 0 of the SAR register.

<7> Upon completion of the comparison for 12 bits, an effective digital result value remains in the SAR register, and the result value is transferred to the A/D conversion result registers (ADCRn, ADCRnH) and then latched**Note 1**. At the same time, the A/D conversion end interrupt request signals (INTAD0 to INTAD3) can also be generated**Note 1**.

<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 ADM2.ADRCK bit and the ADUL and ADLL registers (see **Figure 20 - 9 Interrupt Signal Generation Range According to the Setting of the ADRCK Bit**), the A/D conversion end interrupt request signals (INTAD0 to INTAD3) are not generated and no A/D conversion results are stored in the ADCRn and ADCRnH 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 and advanced mode, either. Instead, 1 is retained.

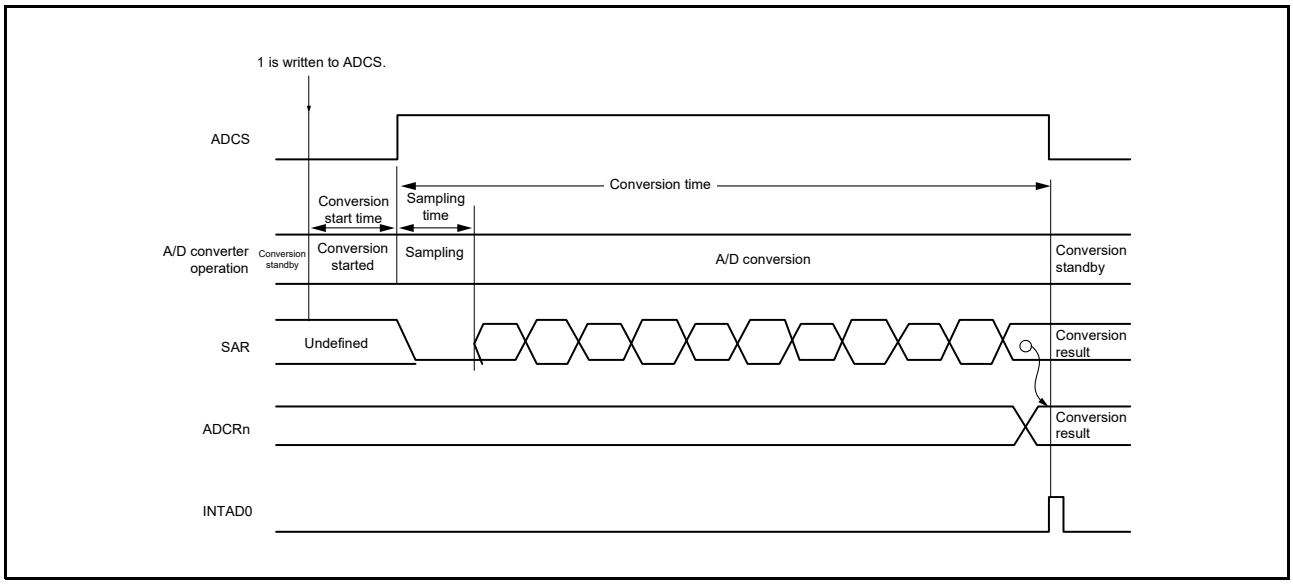
Remark 1. Two types of the A/D conversion result registers are available.

- ADCRn register (16 bits): Stores 12-bit/10-bit A/D conversion value
- ADCRnH register (8 bits): Stores 8-bit A/D conversion value

Remark 2. AVREF: The + side reference voltage of the A/D converter. This can be selected from AVREFP, the internal reference voltage, and VDD.
For details about the internal reference voltage, see **Section 43 Electrical Characteristics (TA = –40 to +105°C)** or **Section 44 Electrical Characteristics (TA = –40 to +125°C)**.

Remark 3. n = 0 to 3

Figure 20 - 23 Conversion Operation of A/D Converter (Software Trigger No-wait Mode)



In one-shot conversion mode with the advanced mode disabled, the ADCS bit is automatically cleared to 0 after completion of A/D conversion.

Writing to the analog input channel specification register (ADS) during A/D conversion while the advanced mode is disabled 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.

The values of the A/D conversion result registers, ADCRn and ADCRnH, following a reset are 0000H and 00H, respectively.

20.5 Input Voltage and Conversion Results

The relationship between the analog voltage input to the analog input pins (ANI0 to ANI7, ANI16 to ANI30) and the theoretical A/D conversion result (stored in the 12-bit/10-bit A/D conversion result register (ADCRn)) is shown by the following expression.

$$ADCRn = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 4096 + 0.5\right)$$

or

$$(ADCRn - 0.5) \times \frac{AV_{REF}}{4096} \leq V_{AIN} < (ADCRn + 0.5) \times \frac{AV_{REF}}{4096}$$

where, INT( ): Function which returns integer part of value in parentheses

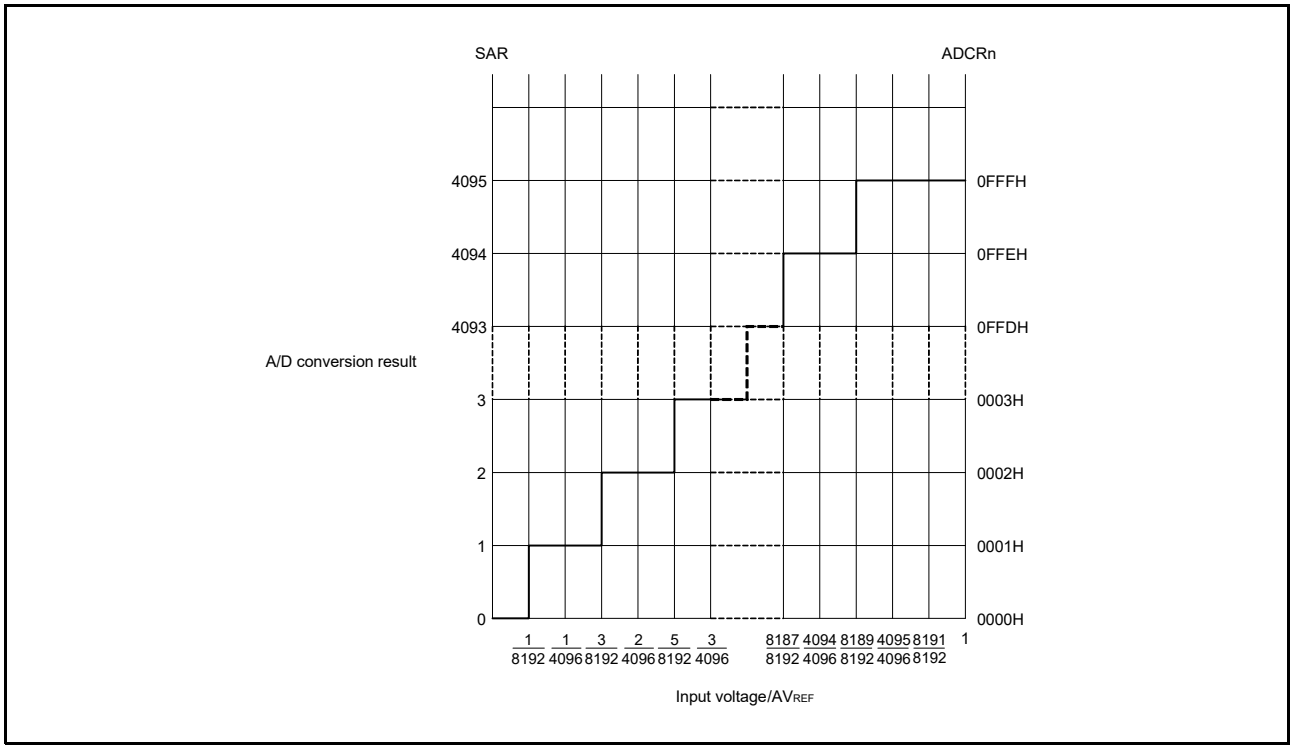
V_{AIN}: Analog input voltage

AV_{REF}: AV_{REF} pin voltage

ADCRn: 12-bit/10-bit A/D conversion result register (ADCRn) value

Figure 20 - 24 shows relationship between analog input voltage and A/D conversion result.

Figure 20 - 24 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[Note], and V_{DD}.

Note For details about the internal reference voltage, see **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** or **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.

20.6 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 **20.7 A/D Converter Setup Flowchart**.

20.6.1 Software trigger no-wait mode (select mode, sequential conversion mode)

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the standby state.

<2> After the software counts up to the stabilization wait time (1 μs + 2 cycles of the conversion clock (f_{AD})), 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 registers (ADCR, ADCRH, ADCR0, and ADCR0H), and the A/D conversion end interrupt request signal (INTAD0) 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.

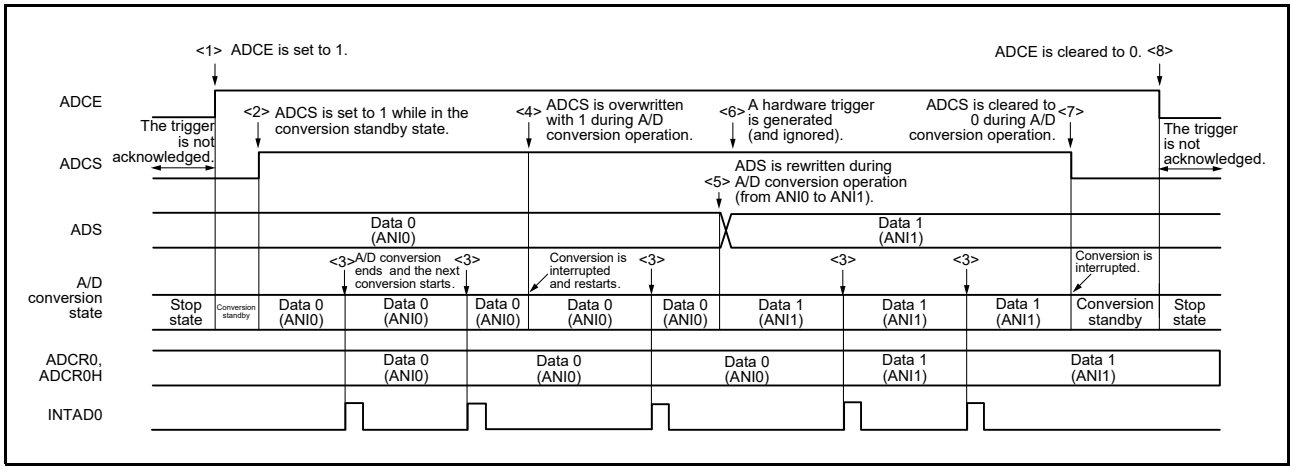
<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<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 A/D converter enters the standby state.

<8> When ADCE is cleared to 0 while in the A/D conversion standby state, the A/D converter enters the stop state.

Figure 20 - 25 Example of Software Trigger No-wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



Caution When <4> or <5> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 20 - 6 Selection of A/D Conversion Time (3/11) and Table 20 - 6 Selection of A/D Conversion Time (4/11).)

20.6.2 Software trigger no-wait mode (select mode, one-shot conversion mode)

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the standby state.

<2> After the software counts up to the stabilization wait time (1 μs + 2 cycles of the conversion clock (f_{AD})), 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 registers (ADCR, ADCRH, ADCR0, and ADCR0H), and the A/D conversion end interrupt request signal (INTAD0) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the standby state.

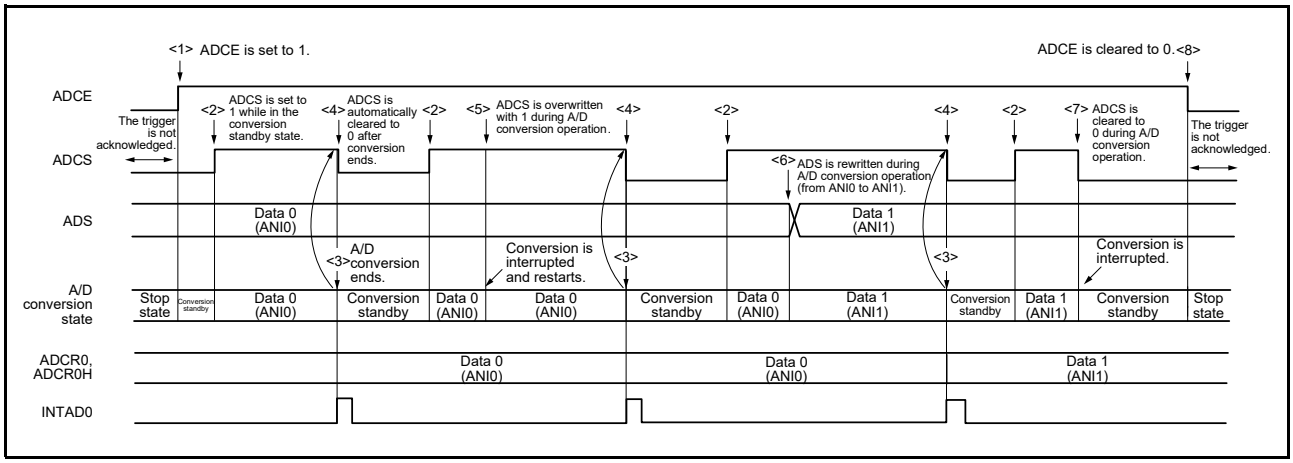
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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 A/D converter enters the standby state.

<8> When ADCE is cleared to 0 while in the A/D conversion standby state, the A/D converter enters the stop state. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby state.

Figure 20 - 26 Example of Software Select No-wait Mode (Select Mode, One-shot Conversion Mode) Operation Timing



Caution When <4>, <5>, or <6> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 20 - 6 Selection of A/D Conversion Time (3/11) and Table 20 - 6 Selection of A/D Conversion Time (4/11).)

20.6.3 Software trigger no-wait mode (scan mode, sequential conversion mode)

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the standby state.

<2> After the software counts up to the stabilization wait time (1 μs + 2 cycles of the conversion clock (f_{AD})), 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 registers (ADCRn, ADCRnH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD0) is generated immediately after A/D conversion of the four channels ends. After A/D conversion of the four channels ends, the next A/D conversion of the specified channels 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.

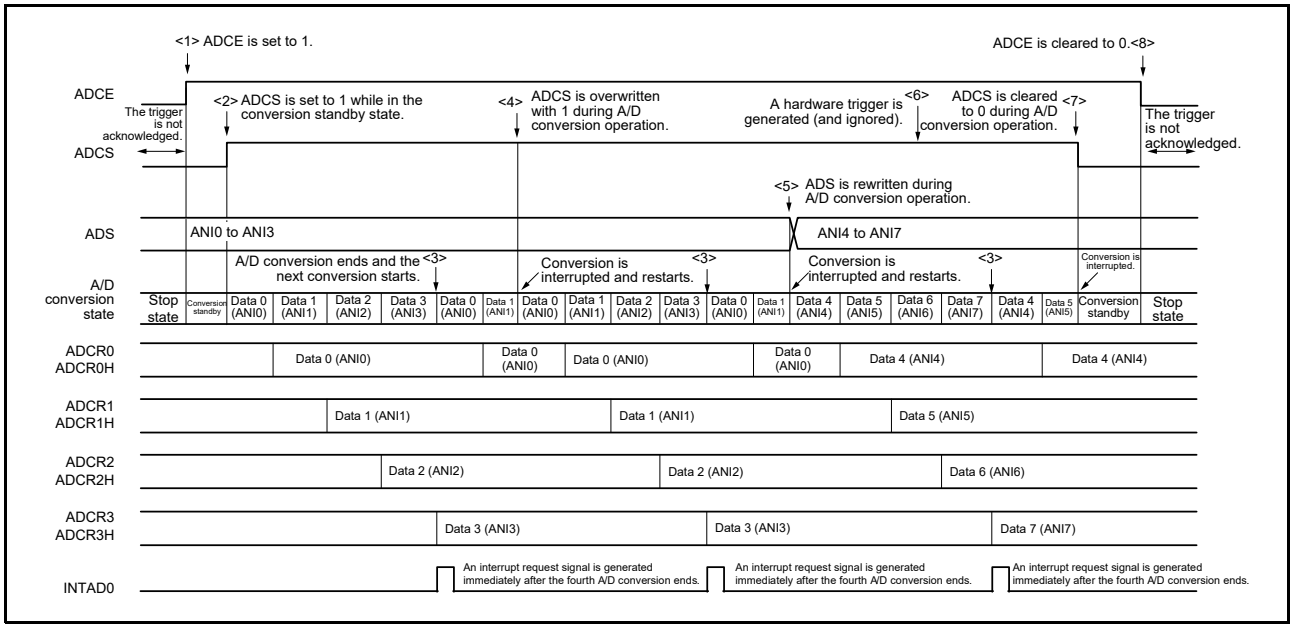
<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<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 A/D converter enters the standby state.

<8> When ADCE is cleared to 0 while in the A/D conversion standby state, the A/D converter enters the stop state.

Figure 20 - 27 Example of Software Trigger No-wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



Caution When <4> or <5> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 20 - 6 Selection of A/D Conversion Time (3/11) and Table 20 - 6 Selection of A/D Conversion Time (4/11).)

20.6.4 Software trigger no-wait mode (scan mode, one-shot conversion mode)

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the standby state.

<2> After the software counts up to the stabilization wait time (1 μs + 2 cycles of the conversion clock (f_{AD})), 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 registers (ADCRn, ADCRnH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD0) is generated immediately after A/D conversion of the four channels ends.

<4> After A/D conversion of the four channels ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the standby state.

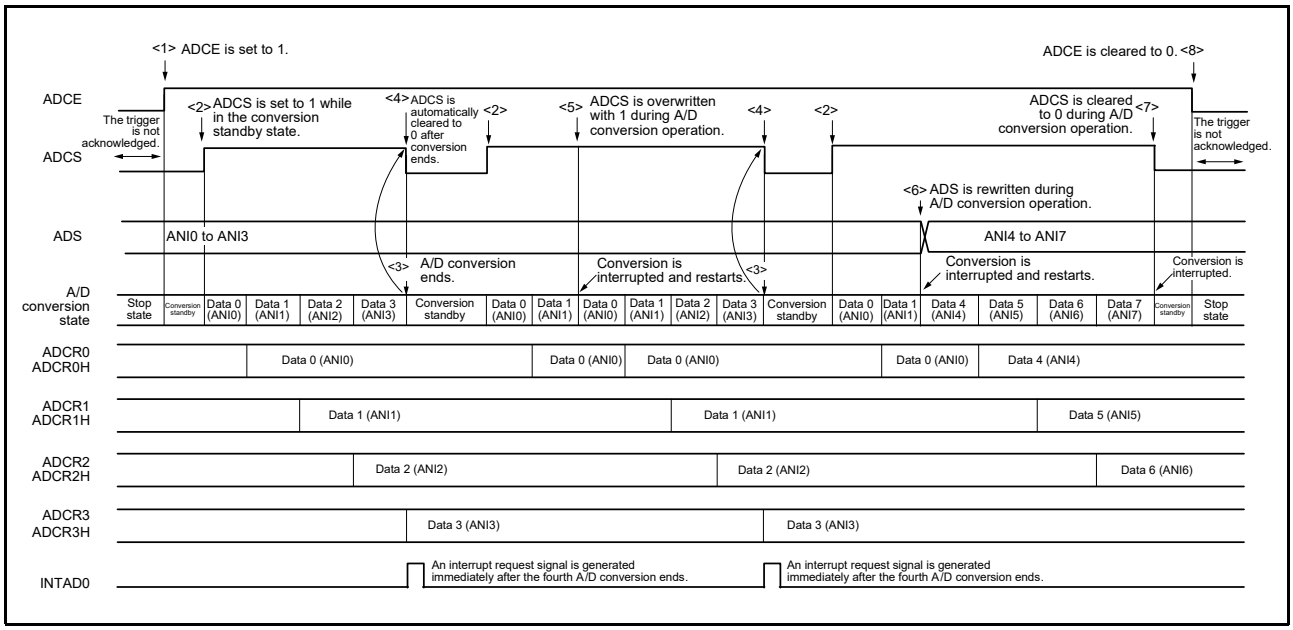
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or 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 A/D converter enters the standby state.

<8> When ADCE is cleared to 0 while in the A/D conversion standby state, the A/D converter enters the stop state. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby state.

Figure 20 - 28 Example of Software Trigger No-wait Mode (Scan Mode, One-shot Conversion Mode) Operation Timing



Caution When <4>, <5>, or <6> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 20 - 6 Selection of A/D Conversion Time (3/11) and Table 20 - 6 Selection of A/D Conversion Time (4/11).)

20.6.5 Software trigger wait mode (select mode, sequential conversion mode)

<1> To shift to software trigger wait mode, the ADCE bit of A/D converter mode register 0 (ADM0) must be set to 0 (stop state).

<2> If ADCS is set to 1 in the stop state, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS) (software trigger wait mode).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result registers (ADCR, ADCRH, ADCR0, and ADCR0H), and the A/D conversion end interrupt request signal (INTAD0) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

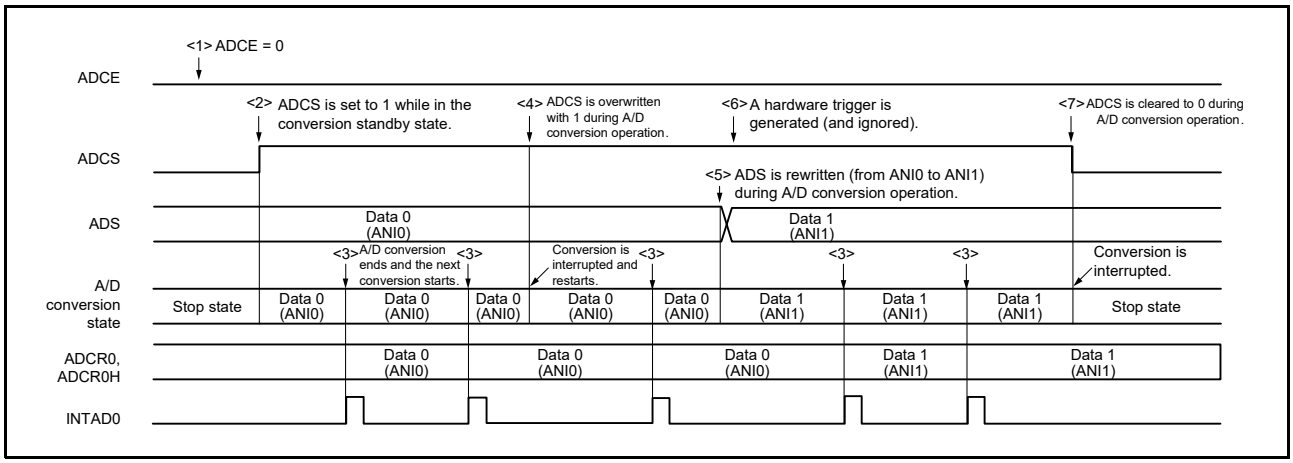
<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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 A/D converter enters the stop state.

Figure 20 - 29 Example of Software Trigger Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



Caution When <4> or <5> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 20 - 6 Selection of A/D Conversion Time (3/11) and Table 20 - 6 Selection of A/D Conversion Time (4/11).)

20.6.6 Software trigger wait mode (select mode, one-shot conversion mode)

<1> To shift to software trigger wait mode, the ADCE bit of A/D converter mode register 0 (ADM0) must be set to 0 (stop state).

<2> If ADCS is set to 1 in the stop state, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS) (software trigger wait mode).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result registers (ADCR, ADCRH, ADCR0, and ADCR0H), and the A/D conversion end interrupt request signal (INTAD0) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the stop state.

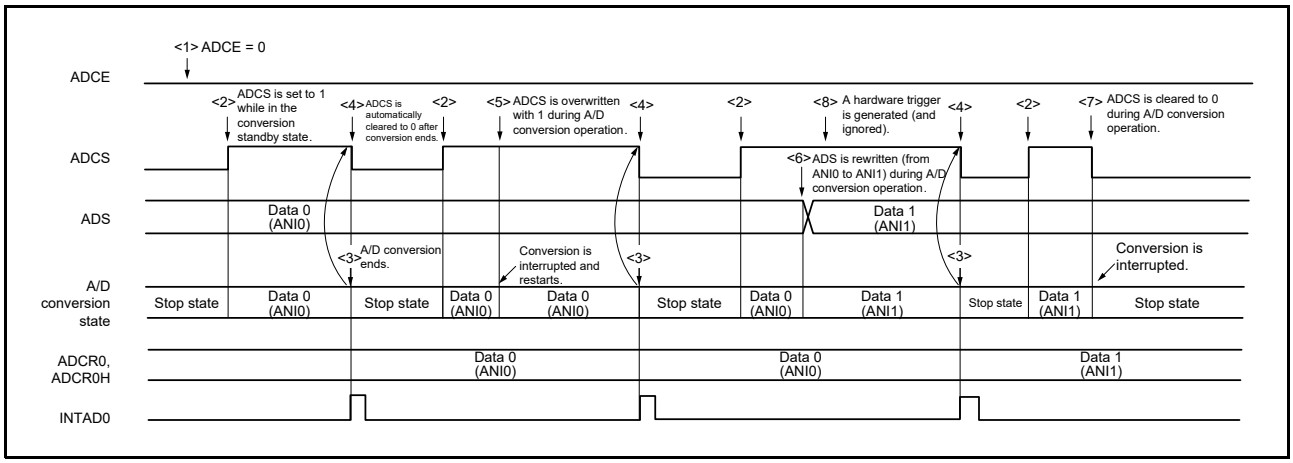
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is initialized.

<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 A/D converter enters the stop state.

<8> When a hardware trigger is input during conversion operation, the trigger is not accepted.

Figure 20 - 30 Example of Software Trigger Wait Mode (Select Mode, One-shot Conversion Mode) Operation Timing



Caution When <4>, <5>, or <6> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 20 - 6 Selection of A/D Conversion Time (3/11) and Table 20 - 6 Selection of A/D Conversion Time (4/11).)

20.6.7 Software trigger wait mode (scan mode, sequential conversion mode)

<1> To shift to software trigger wait mode, the ADCE bit of A/D converter mode register 0 (ADM0) must be set to 0 (stop state).

<2> If ADCS is set to 1 in the stop state, 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) (software trigger wait mode).
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 registers (ADCRn, ADCRnH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD0) is generated immediately after A/D conversion of the four channels ends. After A/D conversion of the four channels ends, the next A/D conversion of the specified channels automatically starts.

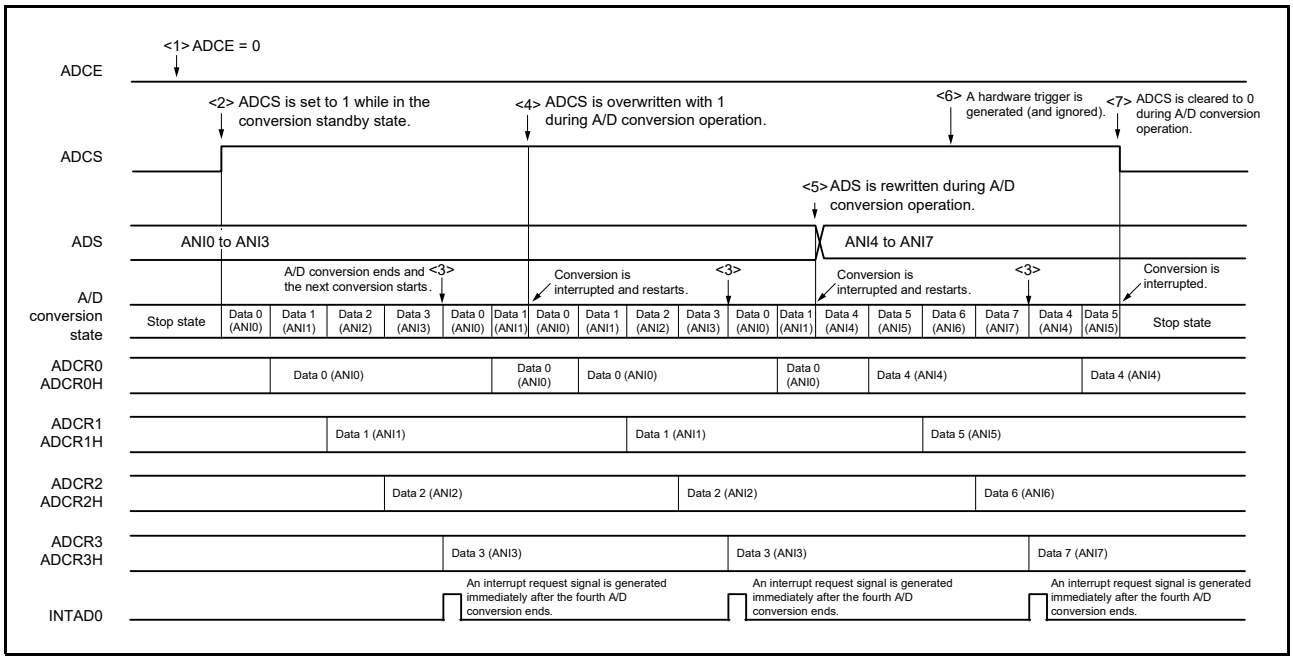
<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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 first channel respecified by the ADS register. The partially converted data is discarded.

<6> When a hardware trigger is input during conversion operation, the trigger is not accepted.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the A/D converter enters the stop state.

Figure 20 - 31 Example of Software Trigger Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



Caution When <4> or <5> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 20 - 6 Selection of A/D Conversion Time (3/11) and Table 20 - 6 Selection of A/D Conversion Time (4/11).)

20.6.8 Software trigger wait mode (scan mode, one-shot conversion mode)

<1> To shift to software trigger wait mode, the ADCE bit of A/D converter mode register 0 (ADM0) must be set to 0 (stop state).

<2> If ADCS is set to 1 in the stop state, 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) (software trigger wait mode).
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 registers (ADCRn, ADCRnH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD0) is generated immediately after A/D conversion of the four channels ends.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the stop state.

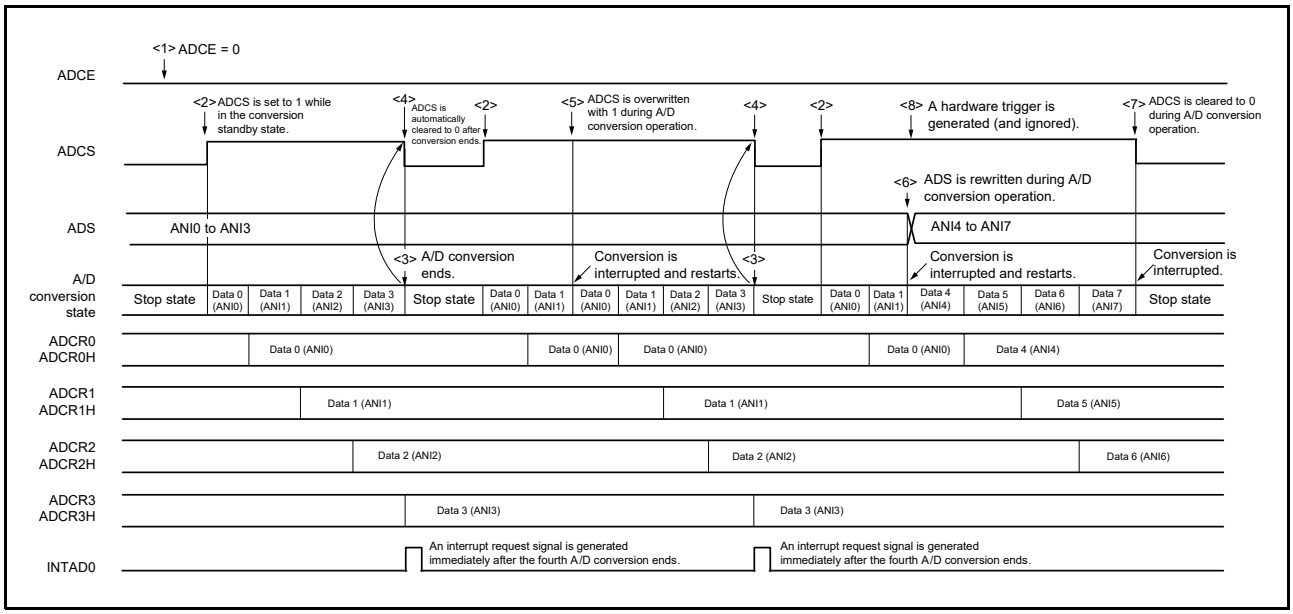
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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 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 A/D converter enters the stop state.

<8> When a hardware trigger is input during conversion operation, the trigger is not accepted.

Figure 20 - 32 Example of Software Trigger Wait Mode (Scan Mode, One-shot Conversion Mode) Operation Timing



Caution When <4>, <5>, or <6> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 20 - 6 Selection of A/D Conversion Time (3/11) and Table 20 - 6 Selection of A/D Conversion Time (4/11).)

20.6.9 Hardware trigger no-wait mode (select mode, sequential conversion mode)

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the standby state.

<2> After the software counts up to the stabilization wait time (1 μs + 2 cycles of the conversion clock (f_{AD})), the ADCS bit of the ADM0 register is set to 1 to place the A/D converter in the hardware trigger standby state (and conversion does not start at this stage). Note that, while in this state, 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 registers (ADCR, ADCRH, ADCR0, and ADCR0H), and the A/D conversion end interrupt request signal (INTAD0) 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.

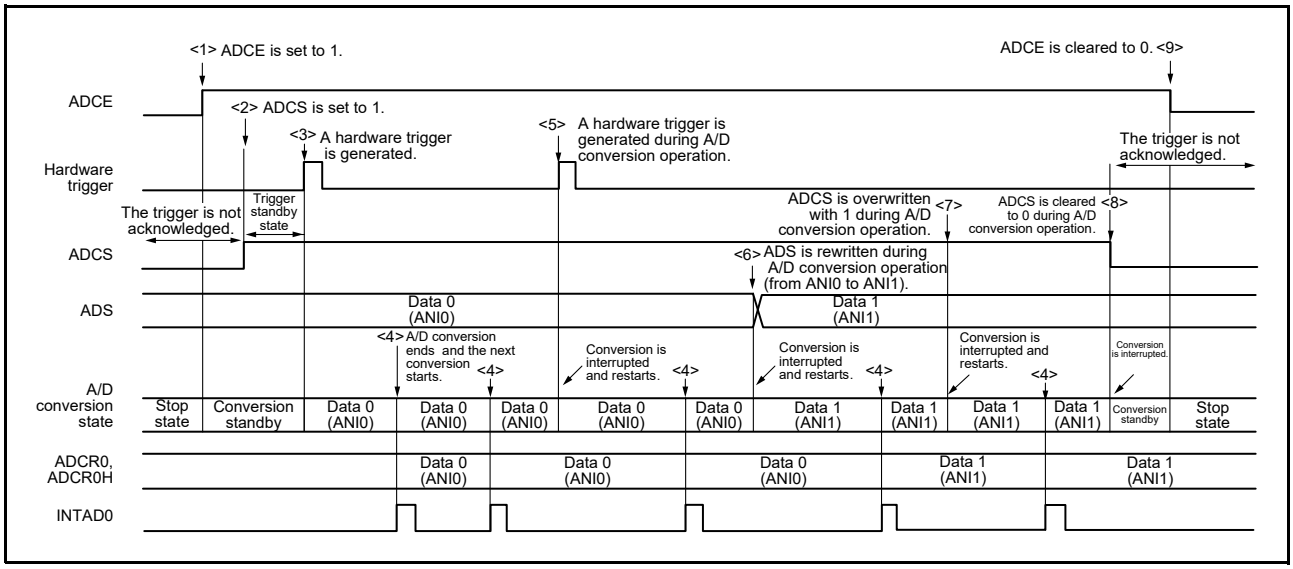
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<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 A/D converter enters the standby state. However, the A/D converter does not stop in this state.

<9> When ADCE is cleared to 0 while in the A/D conversion standby state, the A/D converter enters the stop state. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 20 - 33 Example of Hardware Trigger No-wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



Caution When <4>, <5>, or <6> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 20 - 6 Selection of A/D Conversion Time (3/11) and Table 20 - 6 Selection of A/D Conversion Time (4/11).)

20.6.10 Hardware trigger no-wait mode (select mode, one-shot conversion mode)

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the standby state.

<2> After the software counts up to the stabilization wait time (1 μs + 2 cycles of the conversion clock (f_{AD})), the ADCS bit of the ADM0 register is set to 1 to place the A/D converter in the hardware trigger standby state (and conversion does not start at this stage). Note that, while in this state, 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 registers (ADCR, ADCRH, ADCR0, and ADCR0H), and the A/D conversion end interrupt request signal (INTAD0) is generated.

<5> After A/D conversion ends, the ADCS bit remains set to 1, and the A/D converter enters the standby state.

<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.

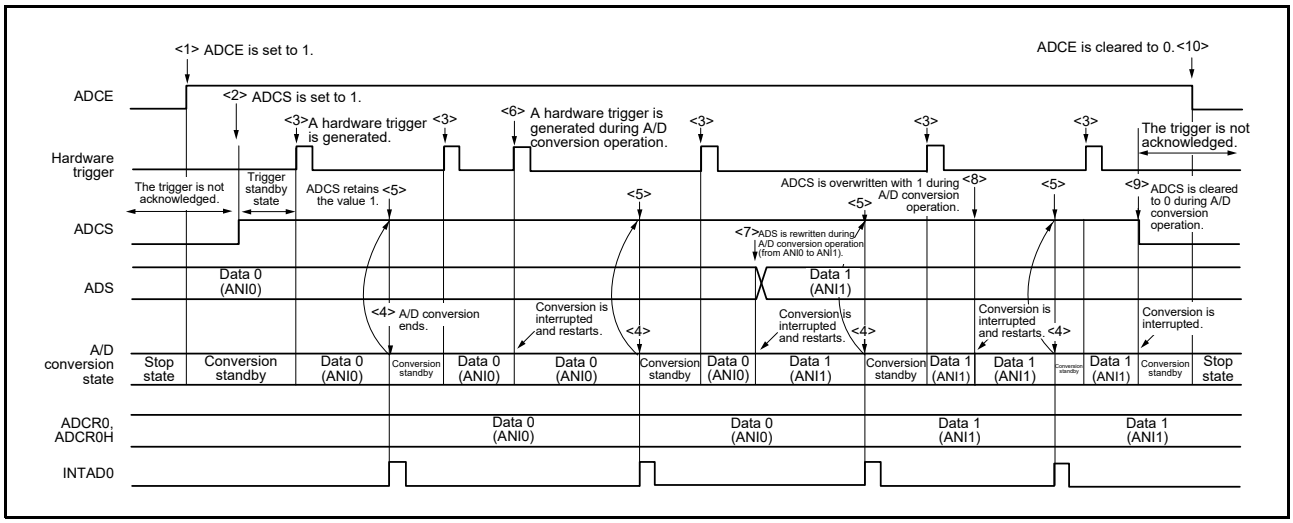
<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<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 A/D converter enters the standby state. However, the A/D converter does not stop in this state.

<10> When ADCE is cleared to 0 while in the A/D conversion standby state, the A/D converter enters the stop state. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 20 - 34 Example of Hardware Trigger No-wait Mode (Select Mode, One-shot Conversion Mode) Operation Timing



Caution When <4>, <5>, or <6> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 20 - 6 Selection of A/D Conversion Time (3/11) and Table 20 - 6 Selection of A/D Conversion Time (4/11).)

20.6.11 Hardware trigger no-wait mode (scan mode, sequential conversion mode)

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the standby state.

<2> After the software counts up to the stabilization wait time (1 μs + 2 cycles of the conversion clock (f_{AD})), the ADCS bit of the ADM0 register is set to 1 to place the A/D converter in the hardware trigger standby state (and conversion does not start at this stage). Note that, while in this state, 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 registers (ADCRn, ADCRnH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD0) is generated immediately after A/D conversion of the four channels ends. After A/D conversion of the four channels ends, the next A/D conversion of the specified channels 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.

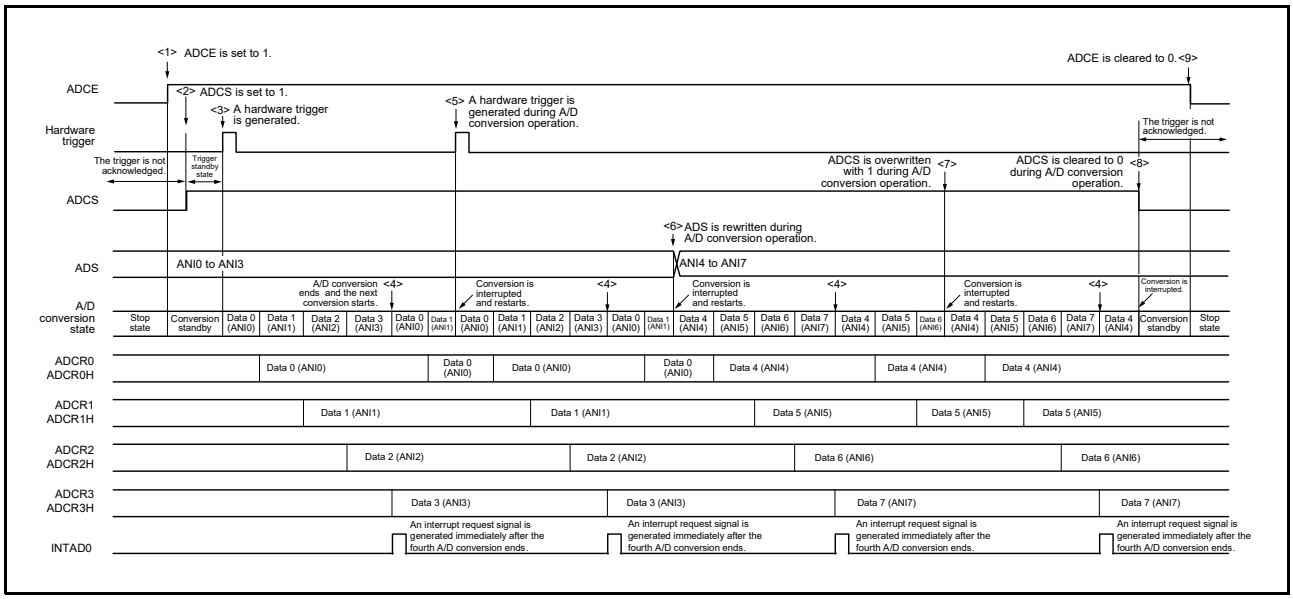
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<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 A/D converter enters the standby state. However, the A/D converter does not stop in this state.

<9> When ADCE is cleared to 0 while in the A/D conversion standby state, the A/D converter enters the stop state. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 20 - 35 Example of Hardware Trigger No-wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



Caution When <4>, <5>, or <6> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 20 - 6 Selection of A/D Conversion Time (3/11) and Table 20

- 6 Selection of A/D Conversion Time (4/11).)

20.6.12 Hardware trigger no-wait mode (scan mode, one-shot conversion mode)

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the standby state.

<2> After the software counts up to the stabilization wait time (1 μs + 2 cycles of the conversion clock (f_{AD})), the ADCS bit of the ADM0 register is set to 1 to place the A/D converter in the hardware trigger standby state (and conversion does not start at this stage). Note that, while in this state, 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 registers (ADCRn, ADCRnH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD0) is generated immediately after A/D conversion of the four channels ends.

<5> After A/D conversion of the four channels ends, the ADCS bit remains set to 1, and the A/D converter enters the standby state.

<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.

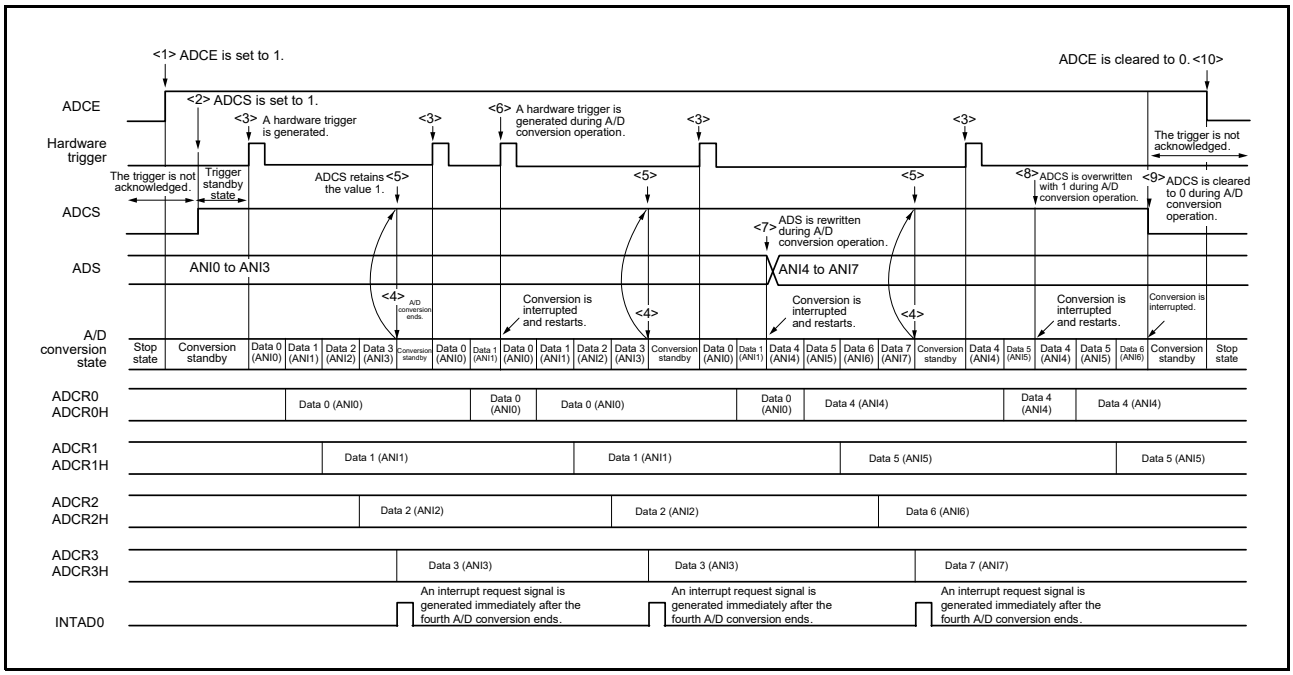
<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<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 A/D converter enters the standby state. However, the A/D converter does not stop in this state.

<10> When ADCE is cleared to 0 while in the A/D conversion standby state, the A/D converter enters the stop state. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 20 - 36 Example of Hardware Trigger No-wait Mode (Scan Mode, One-shot Conversion Mode) Operation Timing



Caution When <4>, <5>, or <6> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 20 - 6 Selection of A/D Conversion Time (3/11) and Table 20 - 6 Selection of A/D Conversion Time (4/11).)

20.6.13 Hardware trigger wait mode (select mode, sequential conversion mode)

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the hardware trigger standby state.

<2> If a hardware trigger is input while in the hardware trigger standby state, 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 registers (ADCR, ADCRH, ADCR0, and ADCR0H), and the A/D conversion end interrupt request signal (INTAD0) is generated. After A/D conversion ends, the next A/D conversion immediately starts. (At this time, no hardware trigger is necessary.)

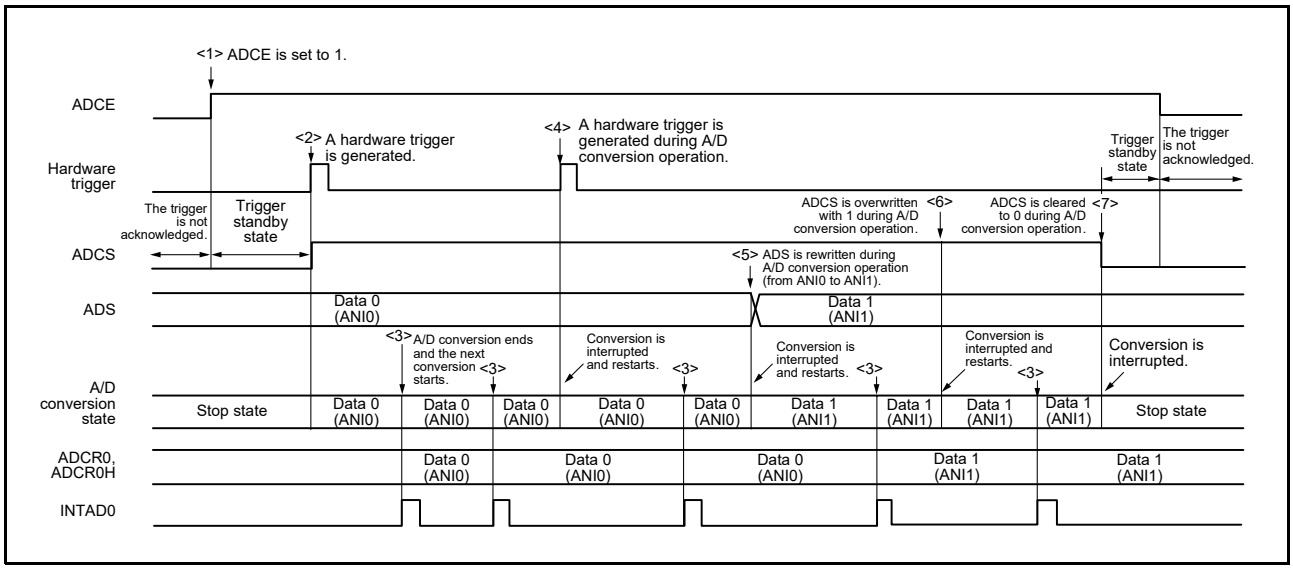
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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 hardware trigger standby state is entered, and the A/D converter is placed in the stop state. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 20 - 37 Example of Hardware Trigger Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



Caution When <4>, <5>, or <6> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 20 - 6 Selection of A/D Conversion Time (3/11) and Table 20 - 6 Selection of A/D Conversion Time (4/11).)

20.6.14 Hardware trigger wait mode (select mode, one-shot conversion mode)

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the hardware trigger standby state.

<2> If a hardware trigger is input while in the hardware trigger standby state, 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 registers (ADCR, ADCRH, ADCR0, and ADCR0H), and the A/D conversion end interrupt request signal (INTAD0) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the stop state.

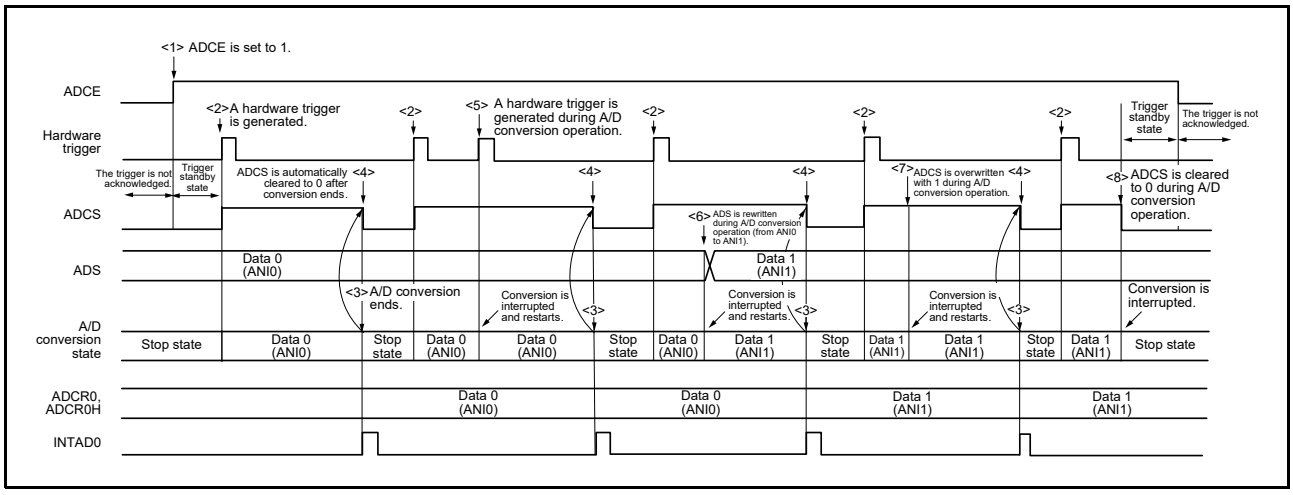
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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 hardware trigger standby state is entered, and the A/D converter is placed in the stop state. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 20 - 38 Example of Hardware Trigger Wait Mode (Select Mode, One-shot Conversion Mode) Operation Timing



Caution 1. When <4>, <5>, or <6> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 20 - 6 Selection of A/D Conversion Time (3/11) and Table 20 - 6 Selection of A/D Conversion Time (4/11).)

Caution 2. In hardware trigger wait mode (select mode, one-shot conversion mode), setting the ADISS bit to 1 (selecting the temperature sensor output voltage and internal reference voltage as input sources) is not possible.

20.6.15 Hardware trigger wait mode (scan mode, sequential conversion mode)

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the standby state.

<2> If a hardware trigger is input while in the hardware trigger standby state, 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 registers (ADCRn, ADCRnH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD0) is generated immediately after A/D conversion of the four channels ends. After A/D conversion of the four channels ends, the next A/D conversion of the specified channels automatically starts.

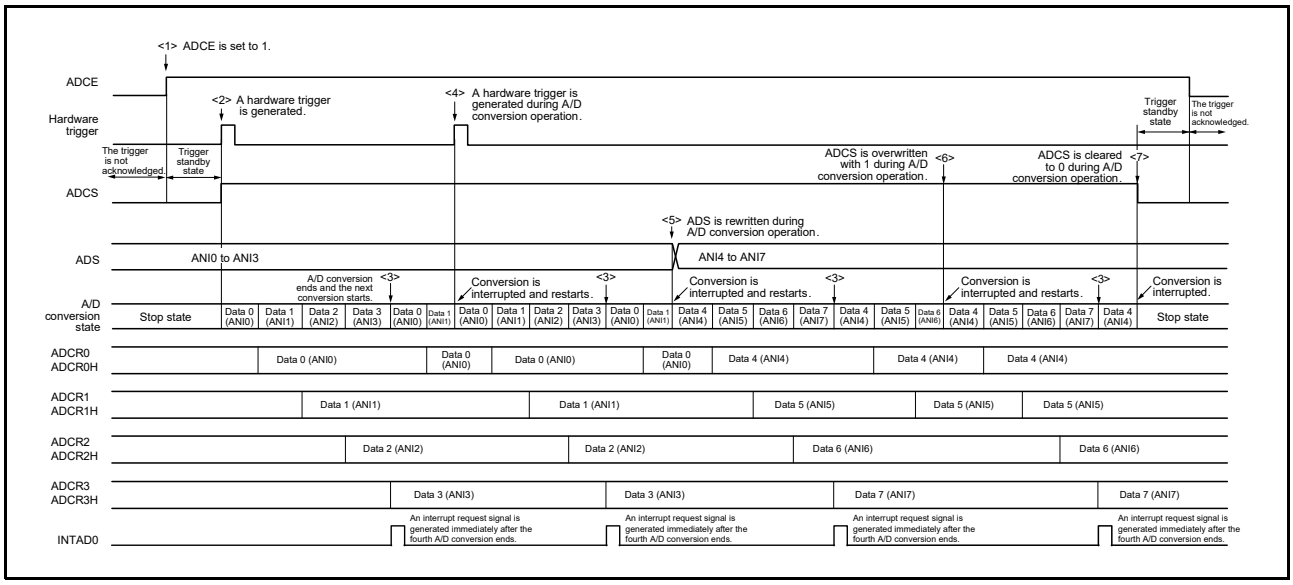
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<5> When the value of the ADS register is rewritten or 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 hardware trigger standby state is entered, and the A/D converter is placed in the stop state. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 20 - 39 Example of Hardware Trigger Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



Caution When <4>, <5>, or <6> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 20 - 6 Selection of A/D Conversion Time (3/11) and Table 20 - 6 Selection of A/D Conversion Time (4/11).)

20.6.16 Hardware trigger wait mode (scan mode, one-shot conversion mode)

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the standby state.

<2> If a hardware trigger is input while in the hardware trigger standby state, 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 registers (ADCRn, ADCRnH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD0) is generated immediately after A/D conversion of the four channels ends.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the stop state.

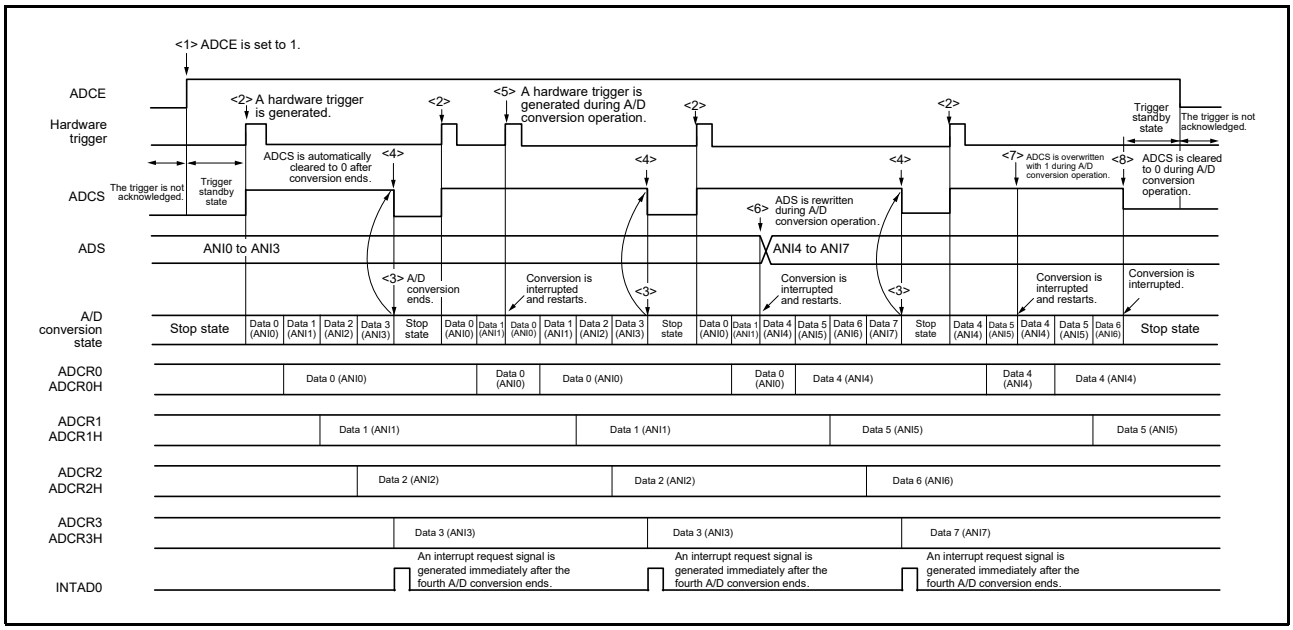
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or 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 hardware trigger standby state is entered, and the A/D converter is placed in the stop state. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 20 - 40 Example of Hardware Trigger Wait Mode (Scan Mode, One-shot Conversion Mode) Operation Timing



Caution When <4>, <5>, or <6> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 20 - 6 Selection of A/D Conversion Time (3/11) and Table 20 - 6 Selection of A/D Conversion Time (4/11).)

20.6.17 Advanced mode with setting of operation triggered by hardware and software

The following shows an example of operation with channels, in each of which conversion starts in response to hardware or software triggers with high or low priority. The setting conditions are as follows.

Mode	Advanced Mode	Wait Mode	Conversion Mode
State	Enabled	No wait	One-shot conversion mode

	Channel Setting Conditions			
	Channel Enabled or Disabled	Trigger	Priority	INTAD0 to INTAD3 Enabled or Disabled
ADS0	Enabled	Timer RD20 (A/D conversion trigger 0)	High	Enabled
ADS1	Enabled	Timer RD21 (A/D conversion trigger 1)	High	Enabled
ADS2	Enabled	Software	Low	Enabled
ADS3	Disabled	Initial value	Initial value	Initial value

<1> Setting the ADCE bit in the A/D converter mode register 0 (ADM0) places the A/D converter in standby when the conversion is stopped. The ADC does not accept a trigger while the setting of the ADCS bit is 0.

<2> After the software counts up to the stabilization wait time (1 μs + 2 cycles of the conversion clock (f_{AD})), the ADCS bit of the ADM0 register is set to 1 to place the A/D converter in the trigger standby state (and conversion does not start at this stage).

<3> Input of the software trigger in the trigger standby state (by setting ADTRSWT to 1) starts A/D conversion on the analog input channel specified by analog input channel specification register 2 (ADS2).

<4> The generation of a trigger with the high priority while A/D conversion for ADS2 with the low priority is in progress forcibly terminates the conversion for ADS2. The failed A/D conversion status bit (ADINTST2F) is set to 1 in response. A/D conversion for ADS0 with the high priority starts at the same time.

<5> Completion of A/D conversion for ADS0 is followed by the following processing. The ADC is placed in standby if no A/D conversion is held pending.

- The conversion result is stored in the A/D conversion result register ADCR0 or ADCR0H.
- The successful A/D conversion status bit (ADINTST0S) is set to 1.
- The A/D conversion end interrupt request signal (INTAD0) for ADS0 is generated.

<6> If a trigger is generated while ADCS = 1, the corresponding A/D conversion starts. In this example, a hardware trigger for ADS1 is generated and A/D conversion starts on the analog input channel specified by ADS1.

<7> Once A/D conversion is completed, confirm generation of INTAD0 to INTAD3, and clear the interrupt request flag register corresponding to the next A/D conversion. Also clear the ADINTST register.

- For details on clearing of the interrupt request flag registers, see **Section 29 Interrupt Functions**.
- Set the ADINTST register to FDH to clear the successful A/D conversion status bit in ADS0.
- Set the ADINTST register to F7H to clear the successful A/D conversion status bit in ADS1.
- Set the ADINTST register to D7H to clear the successful A/D conversion status bit in ADS2.
- Set the ADINTST register to D5H to clear the successful A/D conversion status bits in ADS0 to ADS2 all at once.

<8> If another trigger is generated while A/D conversion with the same or high priority is in progress, A/D conversion in response to the new trigger is held pending. In this example, if a software trigger for AD0 with low priority is generated while A/D conversion in a channel specified by ADS2 with high priority is in progress, A/D conversion for ADS0 is held pending.

- In this example, A/D conversion in response to a software trigger has started, but has been forcibly terminated due to an interrupt by A/D conversion with high priority. The status bit indicating a failure of conversion in a channel specified by ADS2 has been set. Accordingly, clear the INTAD2 interrupt request flag register and the ADINTST2F bit before generating a software trigger again.
  - For details on clearing the interrupt request flag registers, see **Section 29 Interrupt Functions**.
  - Set the ADINTST register to EFH to clear the failed A/D conversion status bit in ADS2.

<9> On completion of the conversion that was currently in progress at the time of the arrival of the trigger for which conversion was held pending, the previously pending A/D conversion starts in order of priority.

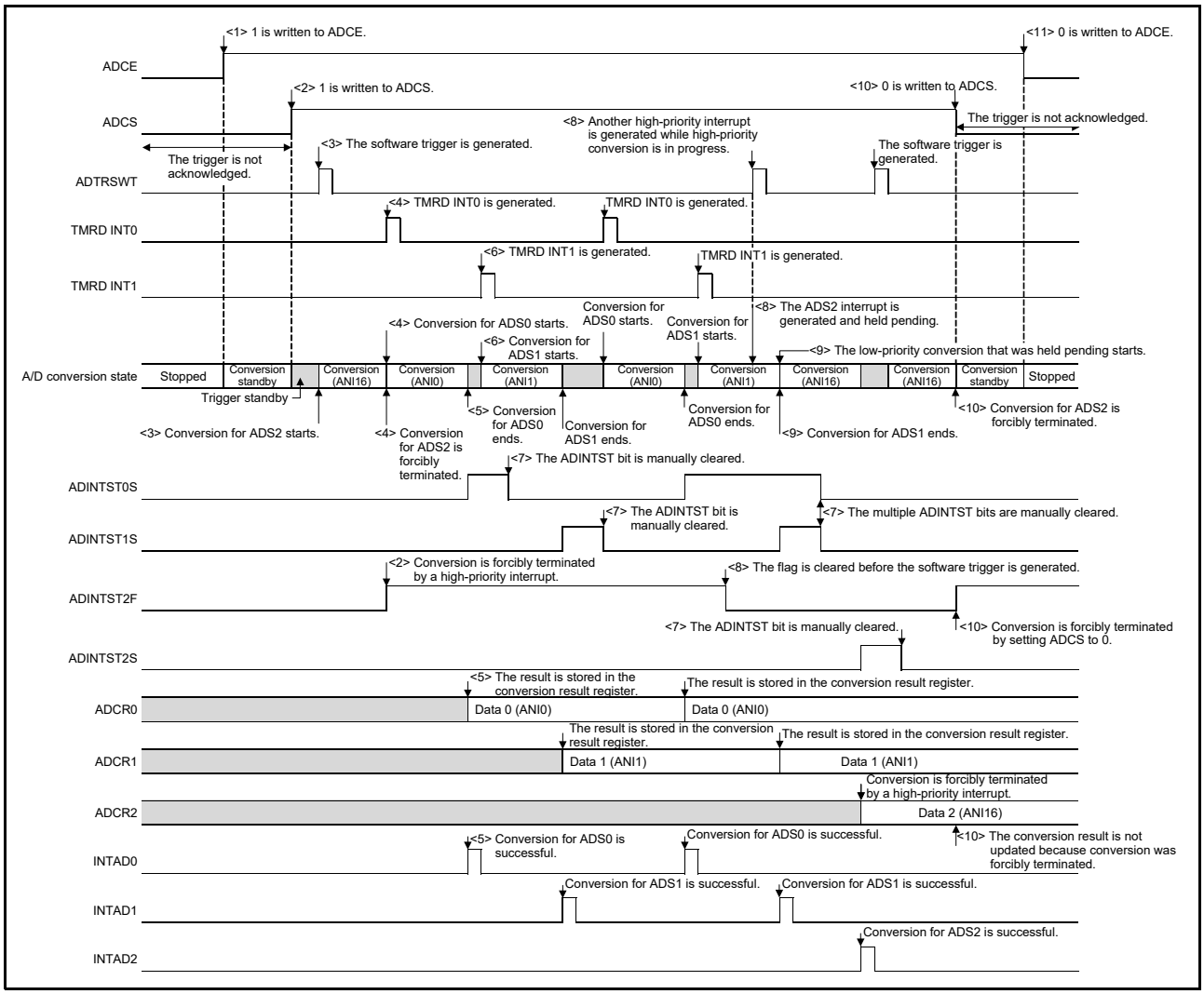
- In this example, A/D conversion in a channel specified by ADS2 starts, which has been held pending.

<10> Setting the ADCS bit to 0 during A/D conversion forcibly terminates the conversion in progress, and places the ADC in standby. The status bit indicating a failure in conversion is set for the forcibly terminated conversion.

- In this example, ADINTST2F is set to 1 because conversion in a channel specified by ADS2 has been forcibly terminated.
- In this example, the A/D conversion result is not updated because conversion in a channel specified by ADS2 has been forcibly terminated.

<11> Clearing the ADCE bit to 0 while the ADC is in standby stops ADC operation.

Figure 20 - 41 Example of Timing of Operations in Advanced Mode with Hardware Triggers and Software Triggers



20.6.18 Advanced mode with simultaneous sampling and setting of operation triggered by hardware and software

The following shows an example of operation with channels, in each of which conversion starts in response to software triggers with low priority, with simultaneous sampling of three channels. The setting conditions are as follows.

Mode	Advanced Mode	Wait Mode	Conversion Mode
State	Enabled	No wait	One-shot conversion mode

	Channel Setting Conditions				
	Channel Enabled or Disabled	Simultaneous Sampling Setting	Trigger	Priority	INTAD Enabled or Disabled
ADS0	Enabled	First S&H (ADS0.ADSPSCn[1:0] = 01B)	INTTM01	Initial value	Enabled
ADS1	Enabled	Second S&H (ADS1.ADSPSCn[1:0] = 10B)	INTTM01	Initial value	Enabled
ADS2	Enabled	Third S&H (ADS2.ADSPSCn[1:0] = 11B)	INTTM01	Initial value	Enabled
ADS3	Enabled	— (ADS2.ADSPSCn[1:0] = 00B)	Software	Low	Enabled

<1> Setting the ADCE bit in the A/D converter mode register 0 (ADM0) places the A/D converter in standby when the conversion is stopped. The ADC does not accept a trigger while the setting of the ADCS bit is 0.

<2> After the software counts up to the stabilization wait time (1 μs + 2 cycles of the conversion clock (f_{AD})), the ADCS bit of the ADM0 register is set to 1 to place the A/D converter in the trigger standby state (and conversion does not start at this stage).

<3> Input of the software trigger in the trigger standby state (by setting ADTRSWT to 1) starts A/D conversion on the analog input channel specified by analog input channel specification register 3 (ADS3).

<4> The generation of the trigger INTTM01 for simultaneous sampling while a conversion on the analog input channel specified by ADS3 with low priority is in progress forcibly terminates the conversion for ADS3. The failed A/D conversion status bit (ADINTST3F) is set to 1 in response. The simultaneous sampling in channels specified by ADS0 to ADS2 starts at the same time.

<5> In simultaneous sampling, the channel specified as the first S&H circuit (by setting ADS0.ADSPSCn[1:0] = 01B) starts A/D conversion in response to a trigger. Each of the channels specified as the second S&H circuit (by setting ADS1.ADSPSCn[1:0] = 10B) and as the third S&H circuit (by setting ADS2.ADSPSCn[1:0] = 11B) retains the sampled value in the S&H circuit.

<6> Completion of A/D conversion in a channel specified by ADS0 is followed by the following processing. A/D conversion in the second S&H circuit specified by ADS1 is automatically starts. The analog input retained in the second S&H circuit is to be A/D converted in a channel specified by ADS1.

- The conversion result is stored in the A/D conversion result register ADCR0 or ADCR0H.
- The successful A/D conversion status bit (ADINTST0S) is set to 1.
- The A/D conversion end interrupt request signal (INTAD0) for ADS0 is generated.

<7> Completion of A/D conversion in a channel specified by ADS1 is followed by the following processing. A/D conversion in the third S&H circuit specified by ADS2 is automatically starts. The analog input retained in the third S&H circuit is to be A/D converted in a channel specified by ADS2.

- The conversion result is stored in the A/D conversion result register ADCR0 or ADCR0H.
- The successful A/D conversion status bit (ADINTST0S) is set to 1.
- The A/D conversion end interrupt request signal (INTAD0) for ADS0 is generated.

<8> Completion of A/D conversion is followed by the following processing. The ADC is placed in standby unless another A/D conversion is held pending.
- The conversion result is stored in the A/D conversion result register ADCR2 or ADCR2H.
- The successful A/D conversion status bit (ADINTST2S) is set to 1.
- The A/D conversion end interrupt request signal (INTAD2) is generated.

<9> Once A/D conversion is completed, confirm generation of INTAD0 to INTAD3, and clear the interrupt request flag register corresponding to the next A/D conversion. Also clear the ADINTST register.
- For details on clearing of the interrupt request flag register, see **Section 29 Interrupt Functions**.
- Set the ADINTST register to FDH to clear the successful A/D conversion status bit in ADS0.
- Set the ADINTST register to F7H to clear the successful A/D conversion status bit in ADS1.
- Set the ADINTST register to DFH to clear the successful A/D conversion status bit in ADS2.
- Set the ADINTST register to D5H to clear the successful A/D conversion status bits in ADS0 to ADS2 all at once.

<10> The software trigger conversion is to proceed when the simultaneous sampling is completed. Accordingly, clear the previous software trigger INTAD3 and the ADINTST register that has been set. In this example, clearing of the status bit indicating a failure in conversion in ADINTST is required because conversion in a channel specified by ADS3 was not successful.
- Set the ADINTST resister to BFH to clear the status bit indicating a failure in conversion in a channel specified by ADS3.

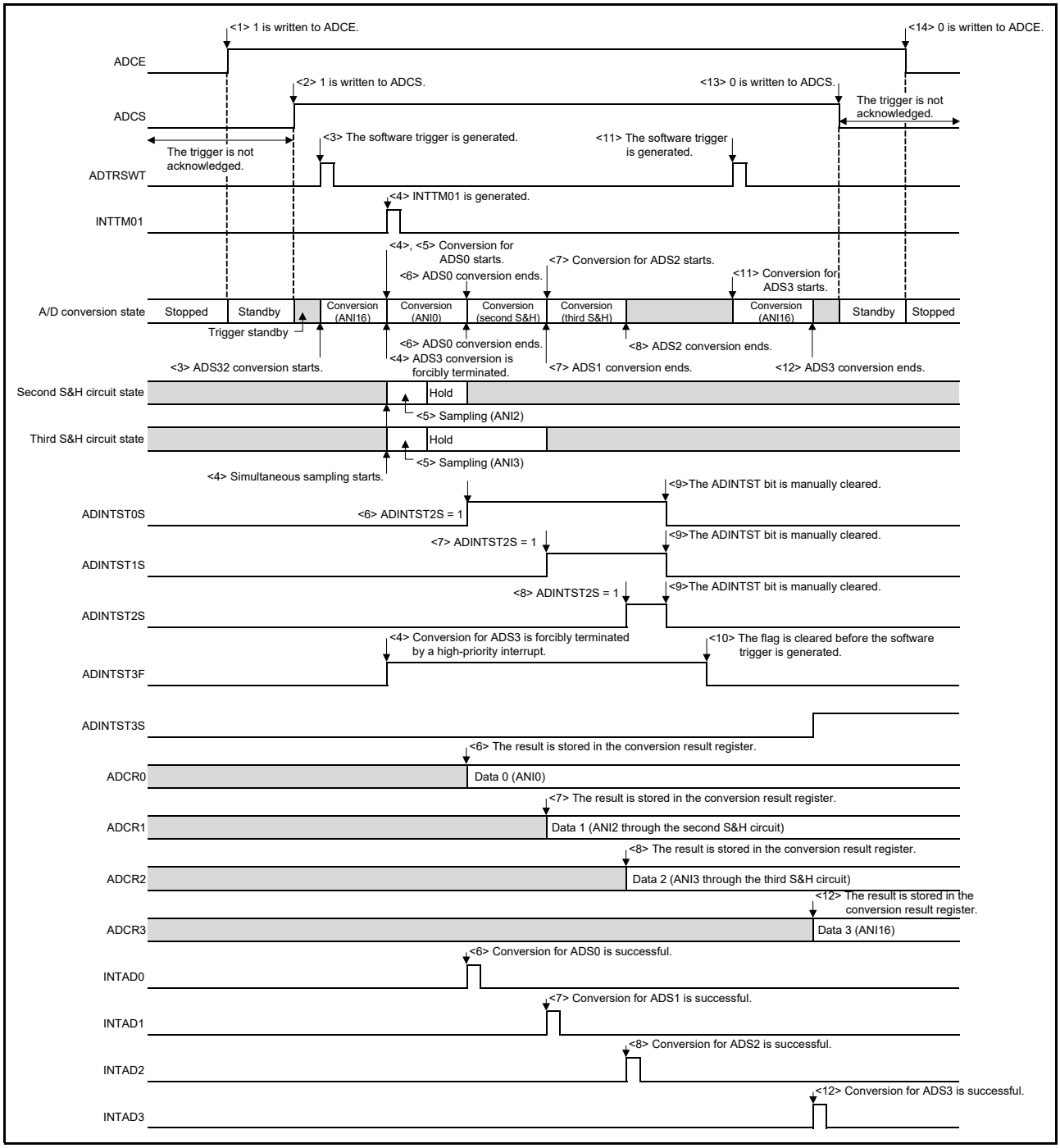
<11> Generate a software trigger to start a conversion in a channel specified by ADS3.

<12> Completion of A/D conversion in a channel specified by ADS3 is followed by the following processing. The ADC is placed in the trigger standby state unless another A/D conversion is held pending.
- The conversion result is stored in the A/D conversion result register ADCR3 or ADCR3H.
- The successful A/D conversion status bit (ADINTST3S) is set to 1.
- The A/D conversion end interrupt request signal (INTAD3) is generated.

<13> Setting the ADCS bit to 0 while in the trigger standby state places the ADC in the conversion standby state.

<14> Clearing the ADCE bit to 0 while the ADC is in standby stops ADC operation.

Figure 20 - 42 Example of Timing of Operations in Advanced Mode with Simultaneous Sampling Enabled and Software Triggers

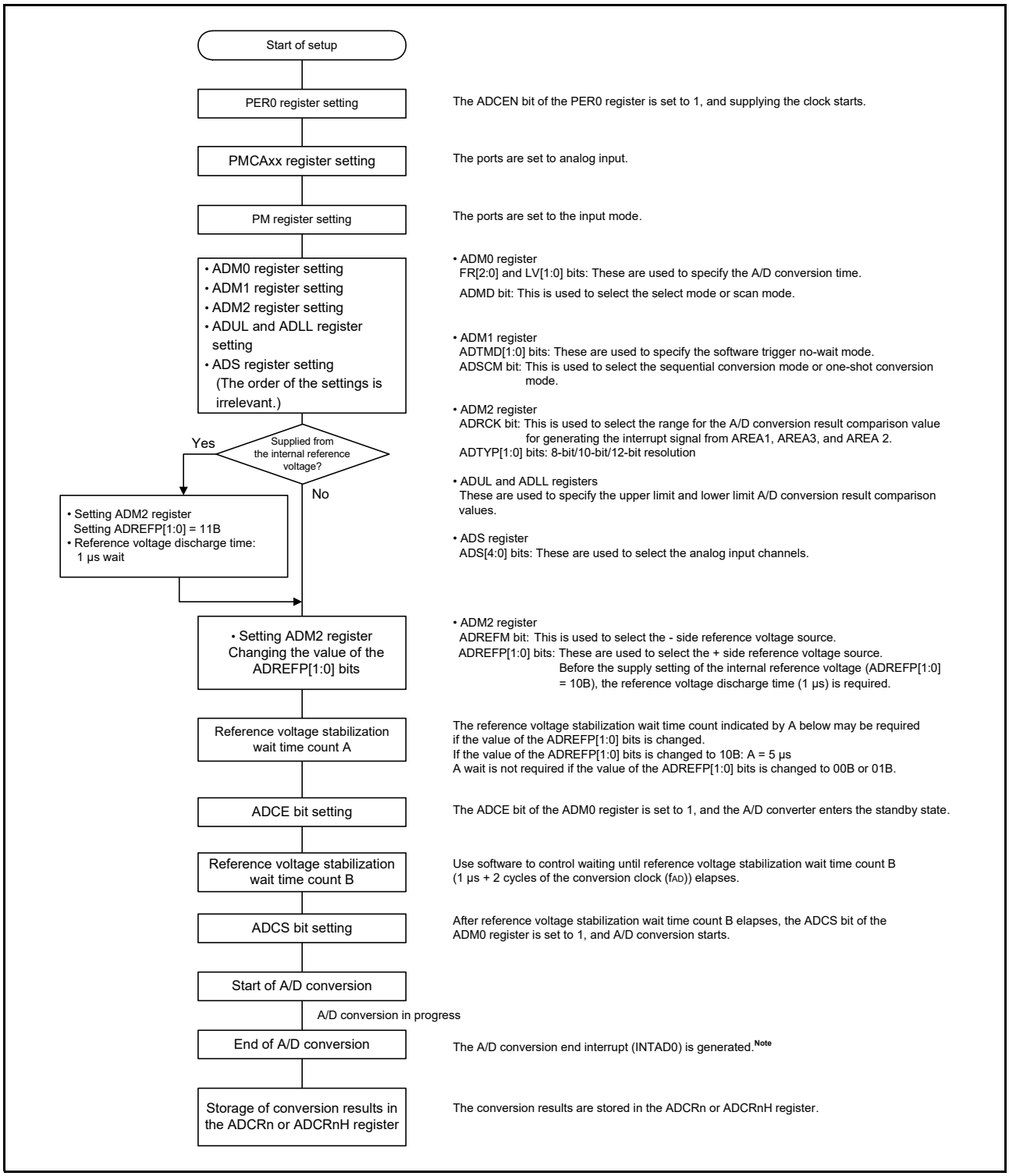


20.7 A/D Converter Setup Flowchart

The A/D converter setup flowchart in each operation mode is described below.

20.7.1 Settings in software trigger no-wait mode

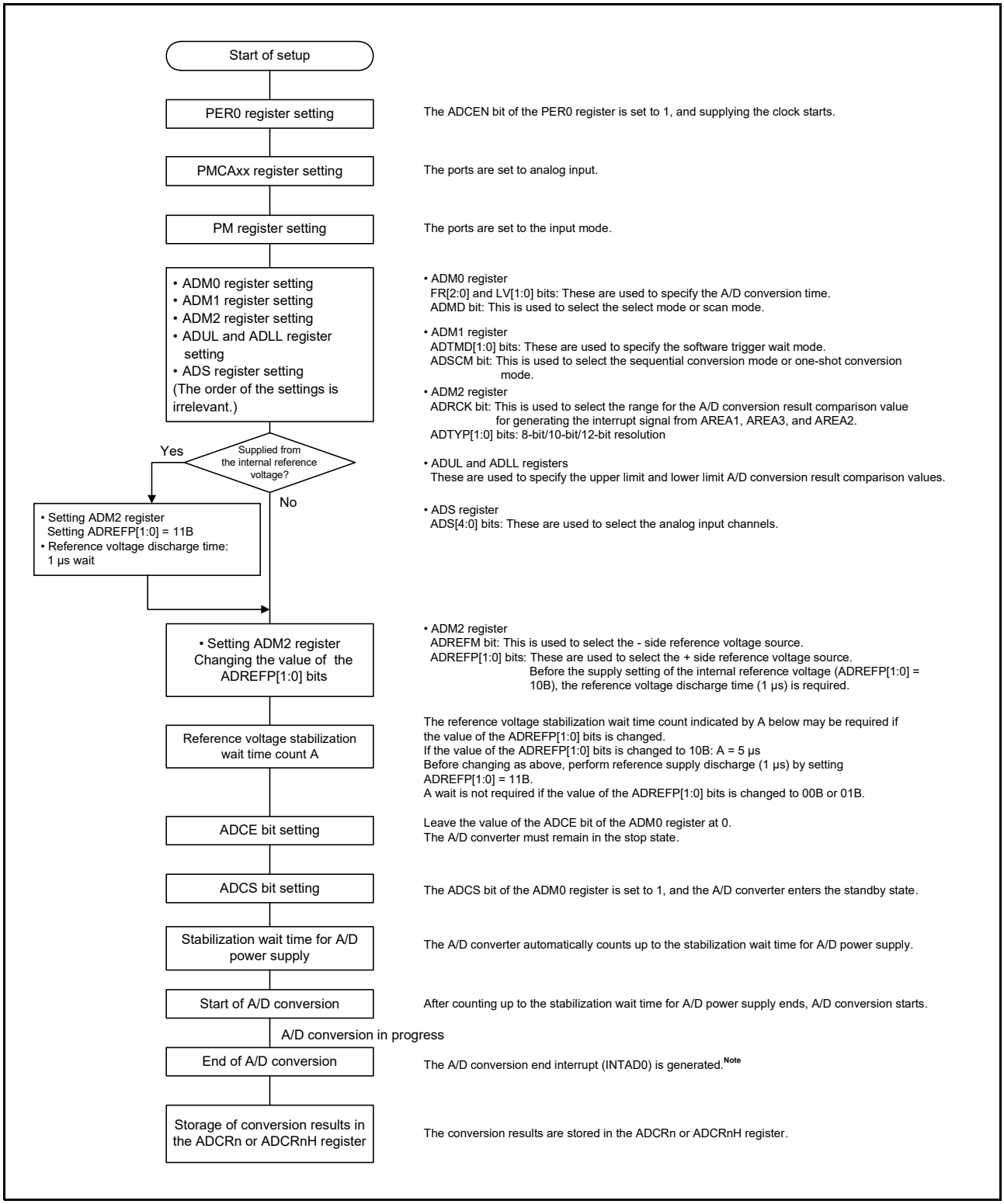
Figure 20 - 43 Settings in Software Trigger No-wait Mode



Note Depending on the settings of the ADRCK bit and ADUL and ADLL registers, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCRn or ADCRnH register.

20.7.2 Settings in software trigger wait mode

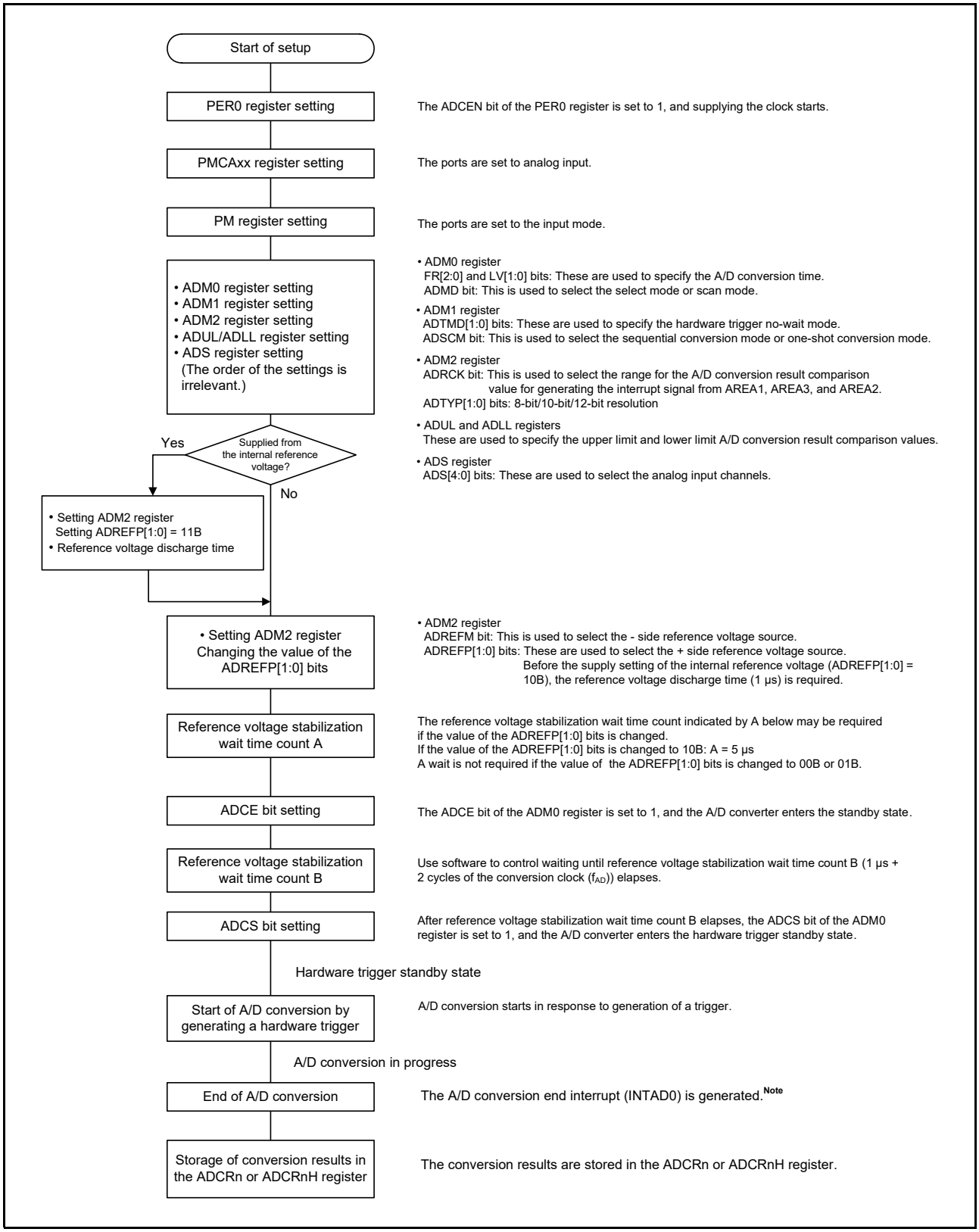
Figure 20 - 44 Settings in Software Trigger Wait Mode



Note Depending on the settings of the ADRCK bit and ADUL and ADLL registers, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCRn or ADCRnH register.

20.7.3 Settings in hardware trigger no-wait mode

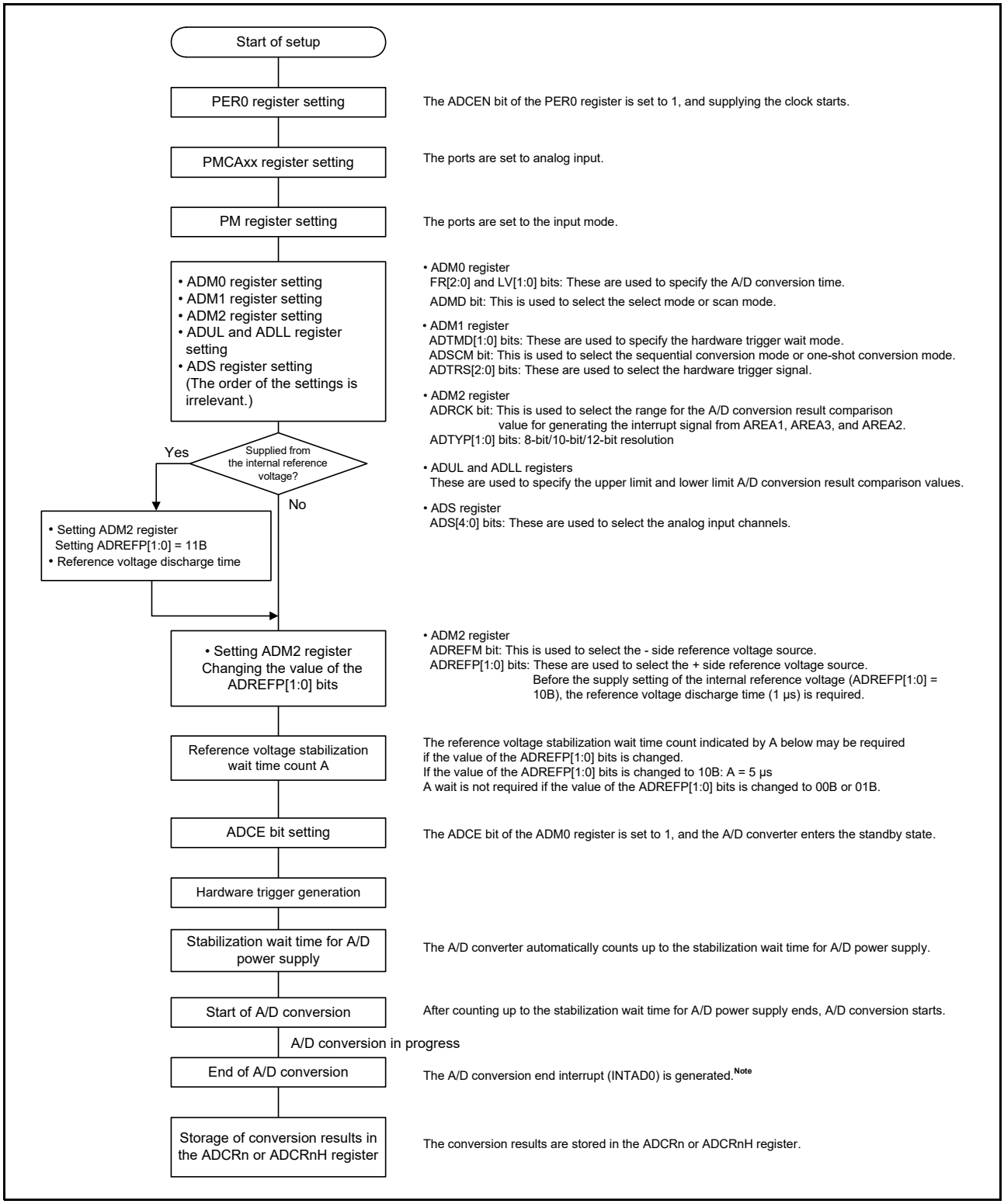
Figure 20 - 45 Settings in Hardware Trigger No-wait Mode



Note Depending on the settings of the ADRCK bit and ADUL and ADLL registers, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCRn or ADCRnH register.

20.7.4 Settings in hardware trigger wait mode

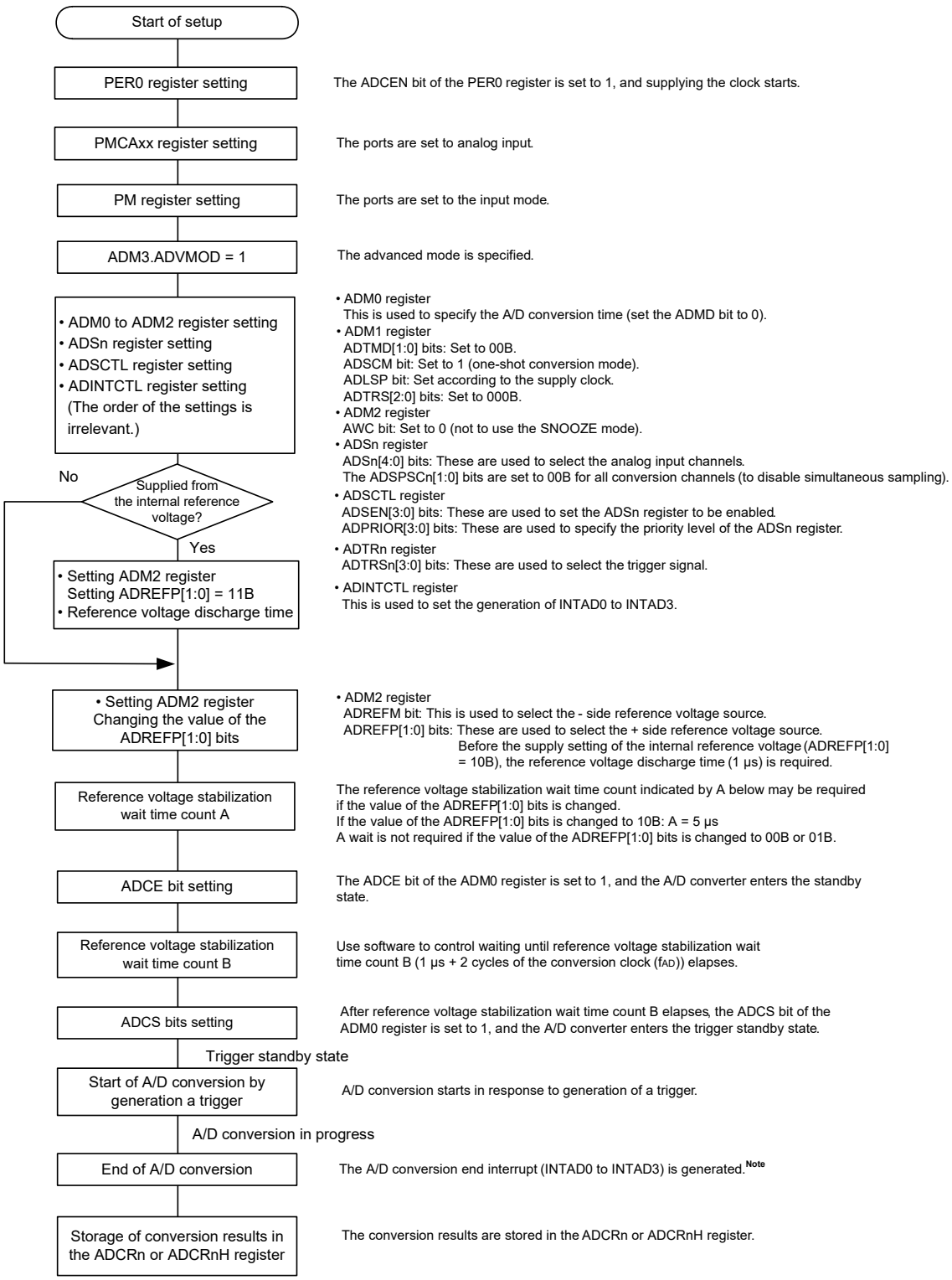
Figure 20 - 46 Settings in Hardware Trigger Wait Mode



Note Depending on the settings of the ADRCK bit and ADUL and ADLL registers, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCRn or ADCRnH register.

20.7.5 Settings in advanced mode

Figure 20 - 47 Settings in Advanced Mode

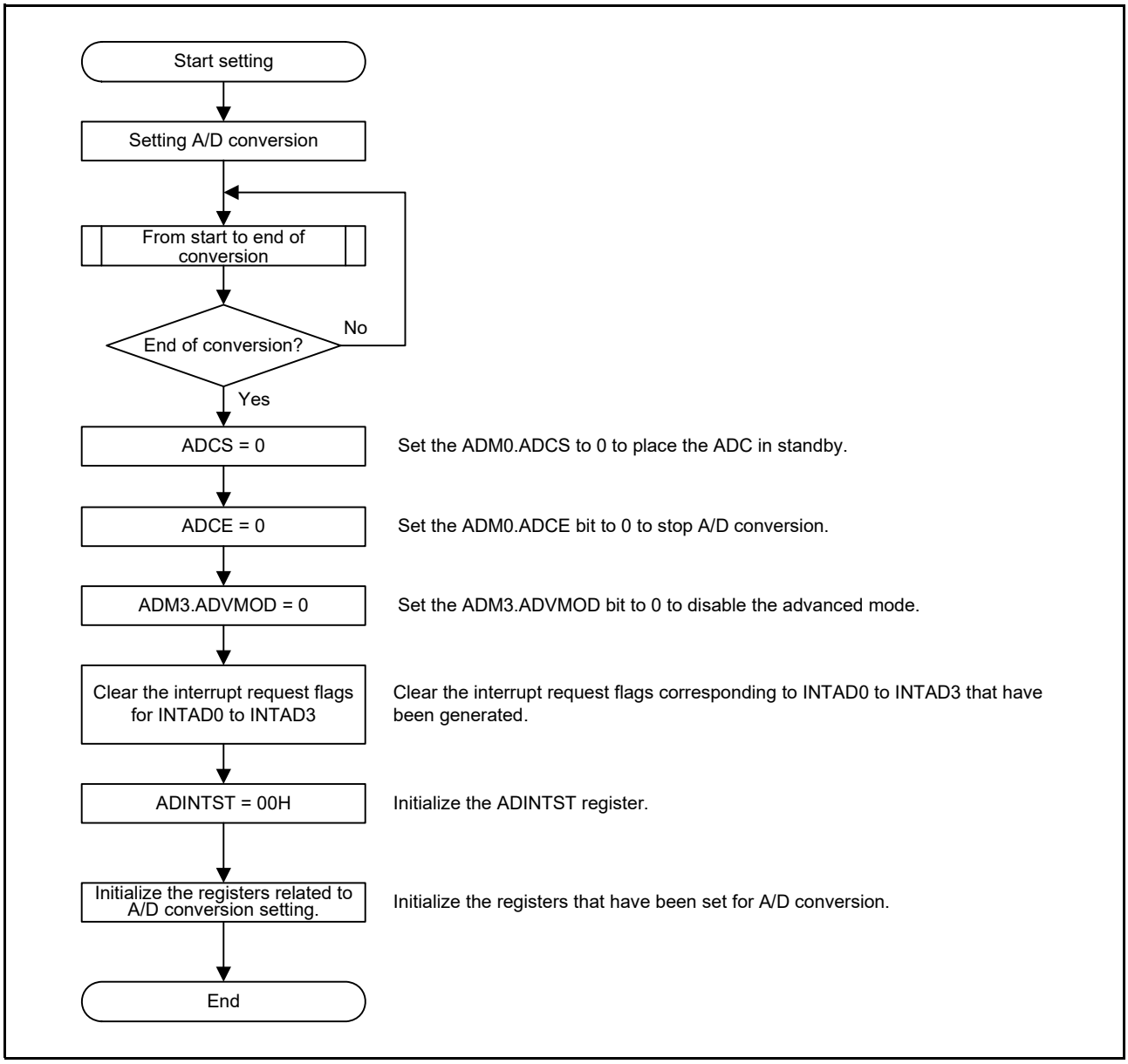


Note If the output of an interrupt signal is disabled by a flag in the ADINTCTL register, the result is still stored in the ADCRn and ADCRnH registers.

20.7.5.1 Procedure of ending the advanced mode

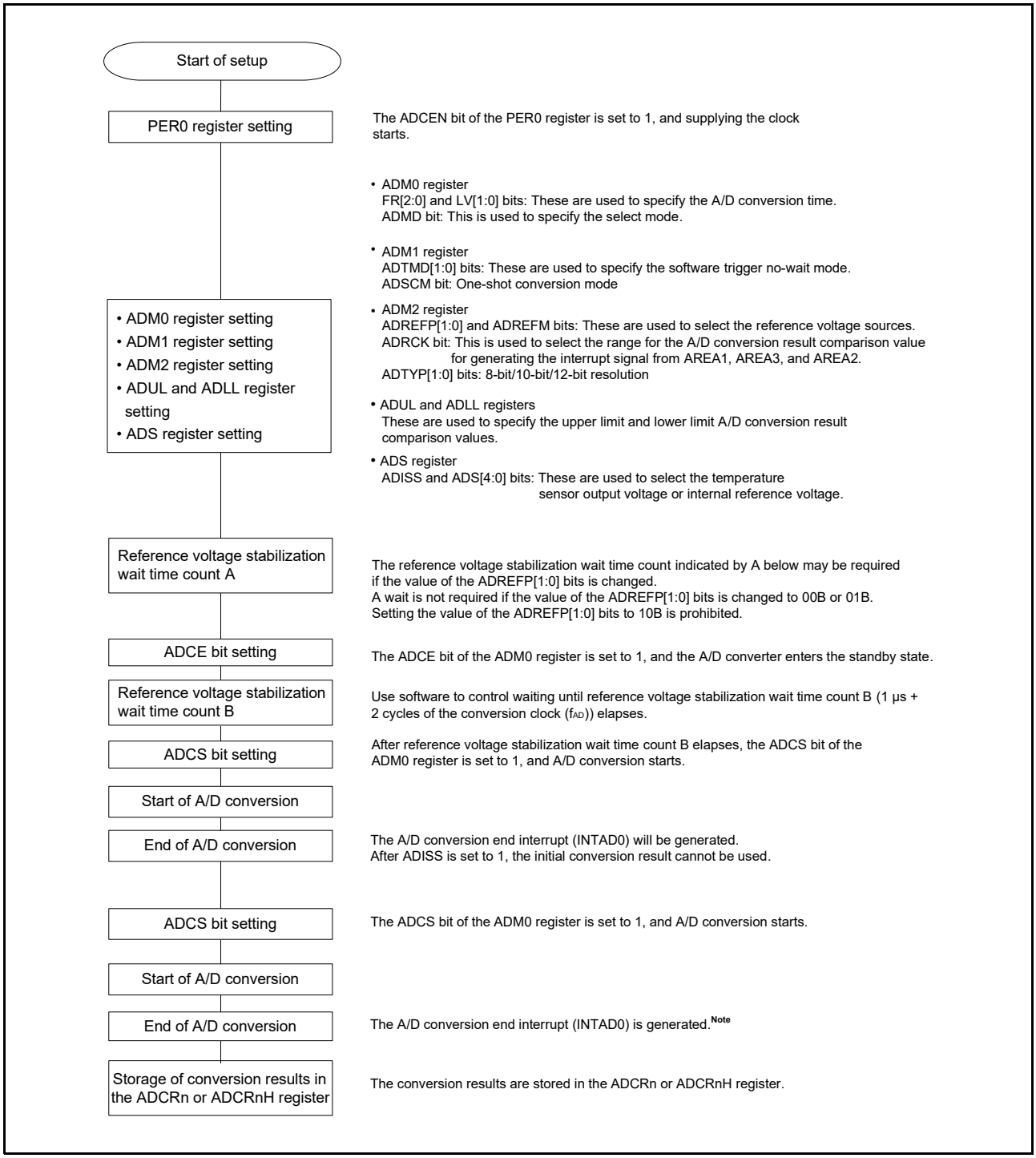
Follow the steps in **Figure 20 - 48** to end the advanced mode after A/D conversion in advanced mode has been completed.

Figure 20 - 48 Procedure of Ending the Advanced Mode



20.7.6 Settings when temperature sensor output voltage or internal reference voltage is selected (example for software trigger no-wait mode and one-shot conversion mode)

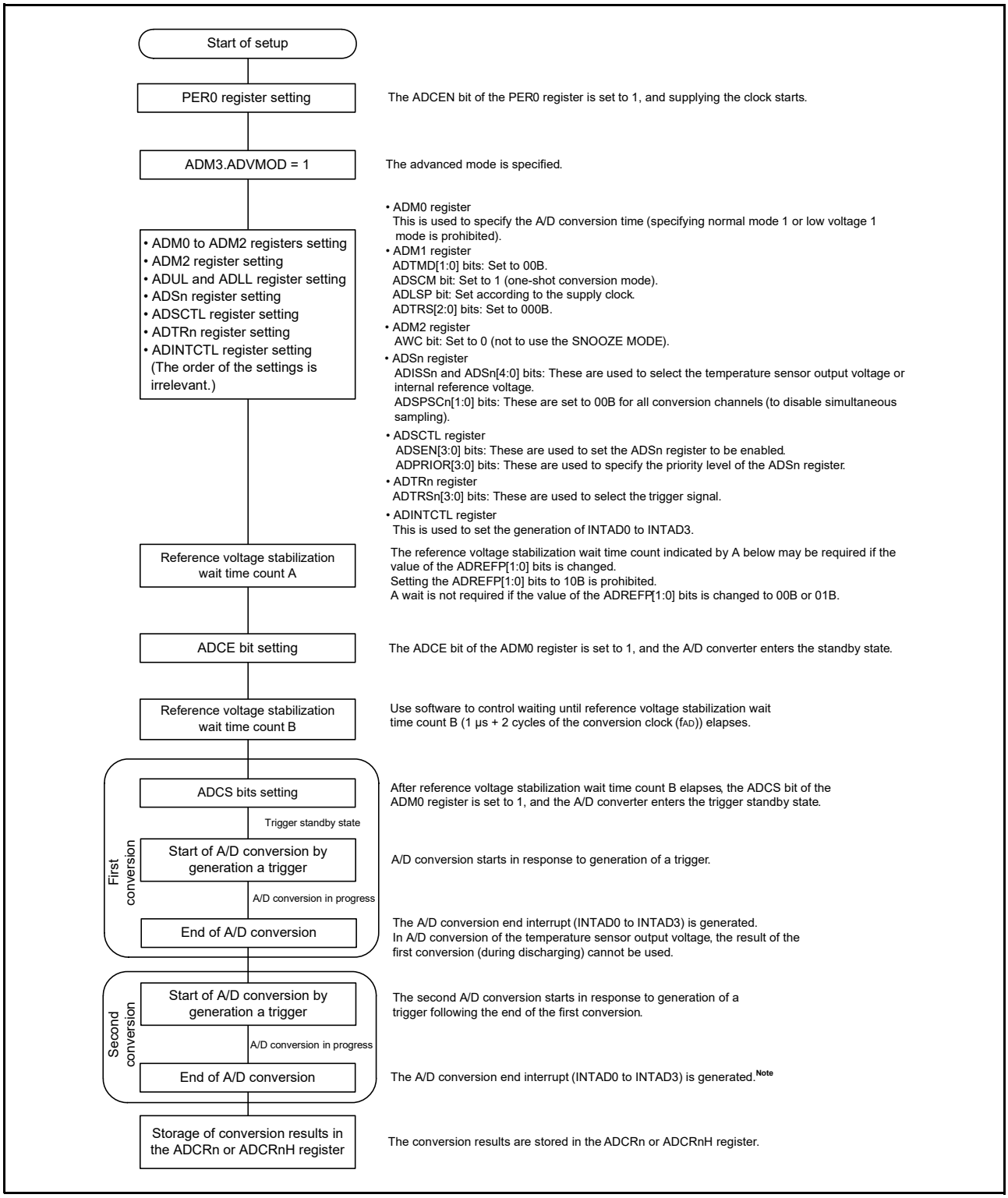
Figure 20 - 49 Settings When Temperature Sensor Output Voltage or Internal Reference Voltage Is Selected (Example for Software Trigger No-wait Mode and One-shot Conversion Mode)



Note Depending on the settings of the ADRCK bit and ADUL and ADLL registers, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCRn or ADCRnH register.

20.7.7 Settings when temperature sensor output voltage or internal reference voltage is selected (example for advanced mode)

Figure 20 - 50 Settings When Temperature Sensor Output Voltage or Internal Reference Voltage Is Selected (in Advanced Mode)

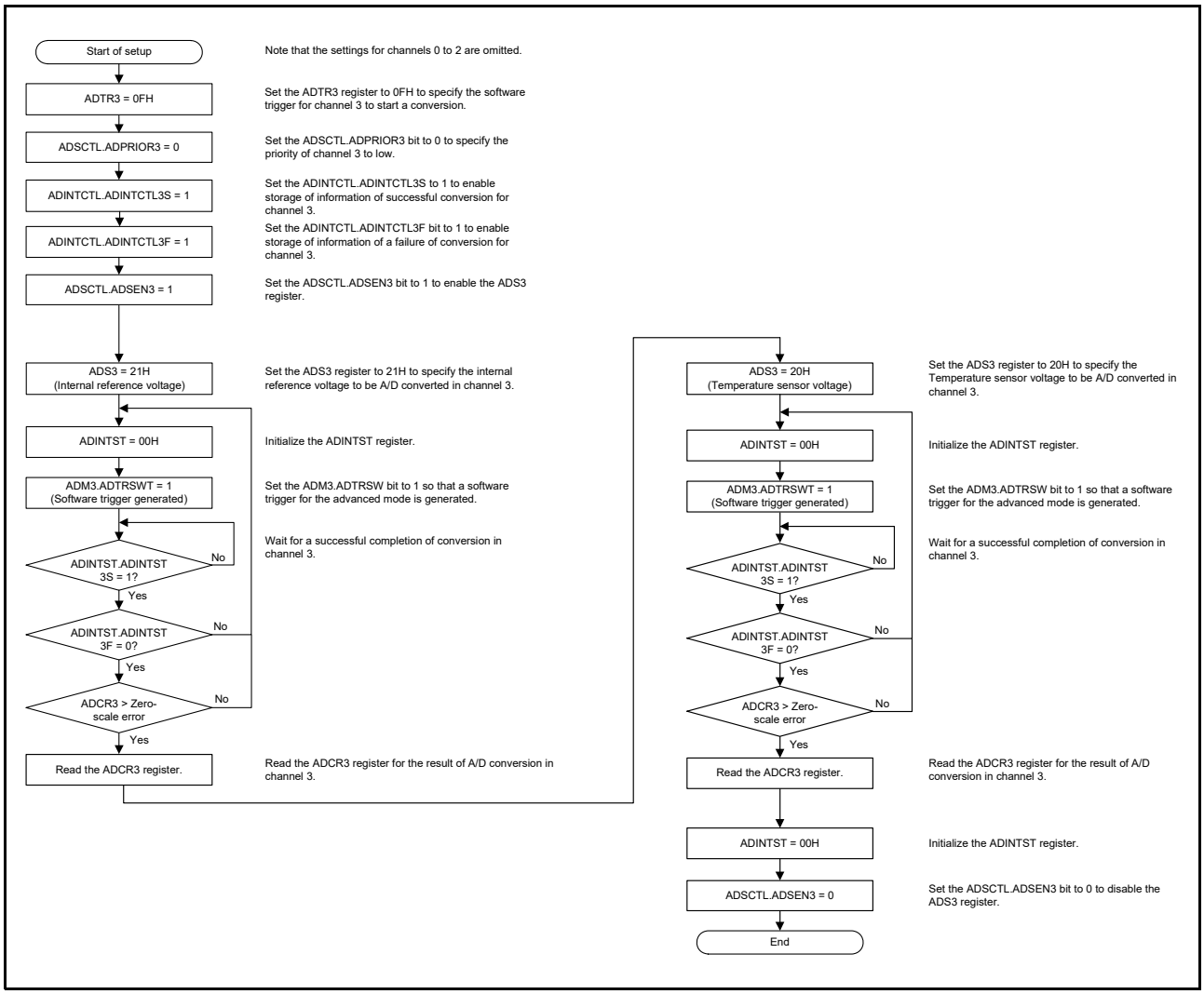


Note If the output of an interrupt signal is disabled by a flag in the ADINTCTL register, the result is still stored in the ADCRn and ADCRnH registers.

20.7.7.1 Procedure of completing the A/D conversion when A/D converting the temperature sensor output voltage or internal reference voltage in advanced mode

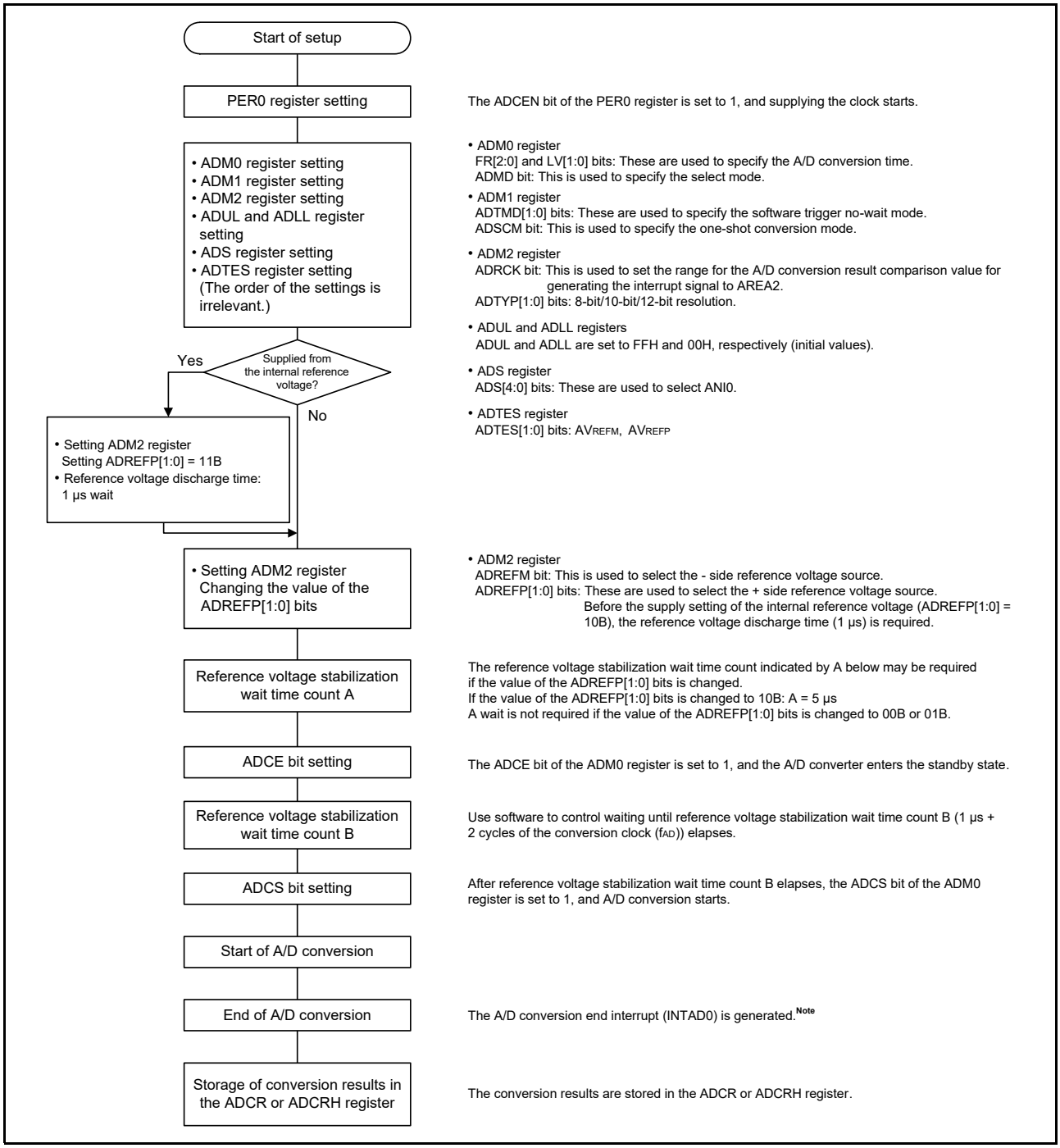
When the temperature sensor output voltage or internal reference voltage has been A/D converted, follow the steps in **Figure 20 - 51** for confirming the conversion result and for completing the conversion. **Figure 20 - 51** shows an example of cases where channel 3 measures the internal reference voltage and temperature sensor voltage while A/D conversions in channels 0 to 2 are in progress in the advanced mode.

Figure 20 - 51 Procedure of Completing the A/D Conversion When A/D Converting the Temperature Sensor Output Voltage or Internal Reference Voltage in Advanced Mode



20.7.8 Settings in test mode

Figure 20 - 52 Settings in Test Mode



Note Depending on the settings of the ADRCK bit and ADUL and ADLL registers, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR or ADCRH register.

Caution For the procedure for testing the A/D converter, see 35.3.10 Testing of the A/D converter.

20.8 Simultaneous Sampling

The RL78/G24 supports the simultaneous sampling of ANI signals on up to three channels in advanced mode. The sampled ANI signals are retained for a specified period of time, each of which is sequentially A/D converted. When the simultaneous sampling is to be used, select normal mode 1.

The ADSPSCn[1:0] bits of the ADSn register are used to disable or make settings for simultaneous sampling.
The table below shows the correspondence between the ADSPSCn[1:0] bit settings and S&H circuits.

Table 20 - 7 Channel Specification for Simultaneous Sampling

ADSPSCn1	ADSPSCn0	Channel Specification for Simultaneous Sampling
0	0	Simultaneous sampling does not proceed (by default).
0	1	The first S&H circuit is used (selected from among ANI0, ANI1, ANI4 to ANI30, and PGA output).**Note**
1	0	The second S&H circuit is used (ANI2).**Note**
1	1	The third S&H circuit is used (ANI3).**Note**

Note If you use the simultaneous sampling, the time taken for a single round of A/D conversion must be no longer than 3.3 μs. V_{DD} must be no greater than 2.7 V.

The possible use cases are as follows.
- f_{CLK} must be at least 16 MHz with no division in normal mode 1.
- f_{CLK} must be divided by 2 to be at least 32 MHz in normal mode 1.

Remark n = 0 to 3

The timing with which the INTAD0 to INTAD3 interrupt signal is generated when simultaneous sampling is in use is the same as that in normal conversion.
Making the setting for control over the generation of INTAD0 to INTAD3 and confirming its state are also possible in the same way as with normal conversion.

Caution 1. In simultaneous sampling, do not specify ANI2 or ANI3 as the target for conversion with the use of the first S&H circuit.

Caution 2. If the simultaneous sampling is to proceed continuously, an interval of at least 4 f_{AD} is required between the generation of the last INTAD0 to INTAD3 and generation of the trigger for the next simultaneous sampling.

20.8.1 Setting simultaneous sampling

The following is an example of making the setting for simultaneous sampling.

- Setting for conversion with the use of simultaneous sampling is the same as for normal conversion other than the setting of the ADSn.ADSPSCn[1:0] bits. Note that, if the ADSPSCn[1:0] bits are to be set to 10B or 11B, the setting of the ADSn[4:0] bits must be 00000B.
- In setting simultaneous sampling, the same trigger source must be set for the conversion channels specified for the first, second, and third S&H circuits.
- In setting simultaneous sampling, the priority for the conversion channels specified for the first, second, and third S&H circuits must be set to the initial value (0).

20.8.1.1 shows example of simultaneously sampling of two channels and **20.8.1.2** shows example of simultaneously sampling of three channels.

20.8.1.1 Example of simultaneously sampling of two channels

Condition	• Specifying the first S&H circuit and ANI1 in the ADS0 register • Specifying the second S&H circuit in the ADS1 register • Since this is a two-channel example, specifying the third S&H circuit is not required. • Specifying TMKB3-ch1 as the trigger source					
Control registers for each conversion channel						
	Simultaneous sampling specification		Analog input specification		Trigger source specification	
	ADSn.ADSPSCn[1:0]		ADSn.ADSn[4:0]		ADTRn.ADTRSn[3:0]	
ADS0	01B	First S&H circuit	00001B	ANI1specified	1000B	TMKB3-ch1
ADS1	10B	Second S&H circuit	00000B	Initial value	1000B	
ADS2	00B	Not used	—	Not used	—	Not used
ADS3	00B	Not used	—	Not used	—	Not used
Register for control as a single batch						
Priority	ADSCTL.ADPRIOR0 = 0 (low) ADSCTL.ADPRIOR1 = 0 (low)					

20.8.1.2 Example of simultaneously sampling of three channels

Condition	• Specifying the first S&H circuit and ANI6 in the ADS1 register • Specifying the second S&H circuit in the ADS2 register • Specifying the third S&H circuit in the ADS3 register • Specifying timer RD20 (A/D conversion trigger 0) as the trigger source					
Control registers for each conversion channel						
	ADSn.ADSPSCn[1:0]		ADSn.ADSn[4:0]		ADTRn.ADTRSn[3:0]	
ADS0	00B	Not used	—	Not used	—	Not used
ADS1	01B	First S&H circuit	00110B	ANI6 specified	1000B	Timer RD20 (A/D conversion trigger 0)
ADS2	10B	Second S&H circuit	00000B	Initial value	1000B	
ADS3	11B	Third S&H circuit	00000B	Initial value	1000B	
Register for control as a single batch						
Priority	ADSCTL.ADPRIOR1 = 0 (low) ADSCTL.ADPRIOR2 = 0 (low) ADSCTL.ADPRIOR3 = 0 (low)					

20.9 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 function, A/D conversion can be performed without operating the CPU. This is effective for reducing the operating current.
The SNOOZE mode is available in hardware trigger mode, but not in advanced mode.

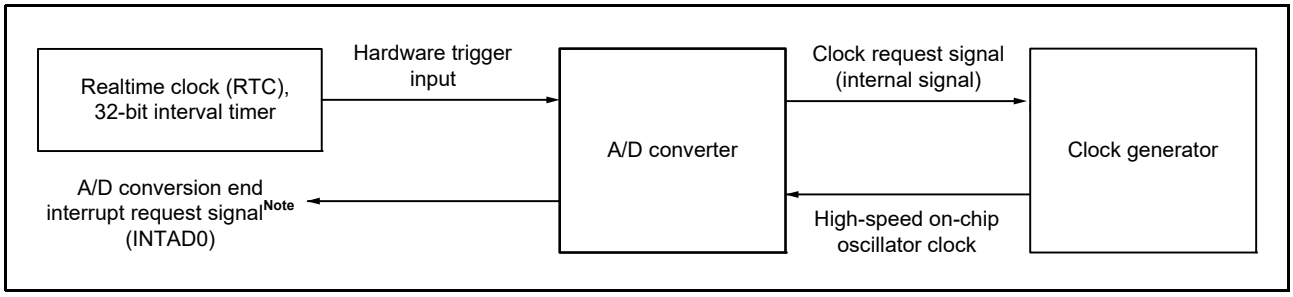
20.9.1 A/D conversion by inputting a hardware trigger

In SNOOZE mode, 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. Using this function enables power supply voltage monitoring and input key judgment based on A/D inputs.
In 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 The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock is selected for f_{CLK}.

Figure 20 - 53 Block Diagram When Using SNOOZE Mode Function in Hardware Trigger Wait Mode



When using the SNOOZE mode function, the initial setting of each register is made before switching to the STOP mode (for details about these settings, see **Figure 20 - 56 Flowchart for Settings in SNOOZE Mode (Hardware Trigger)**). Just before moving to STOP mode, set bit 2 (AWC) of A/D converter mode register 2 (ADM2) to 1. After the initial settings are made, set bit 0 (ADCE) of A/D converter mode register 0 (ADM0) 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 A/D converter 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**.

Note Depending on the setting of the A/D conversion result comparison function (ADM2.ADRCK bit, ADUL and ADLL registers), there is a possibility of no interrupt signal being generated.

Caution Use A/D converter mode register 1 (ADM1) to select the hardware trigger signal from either of the following.

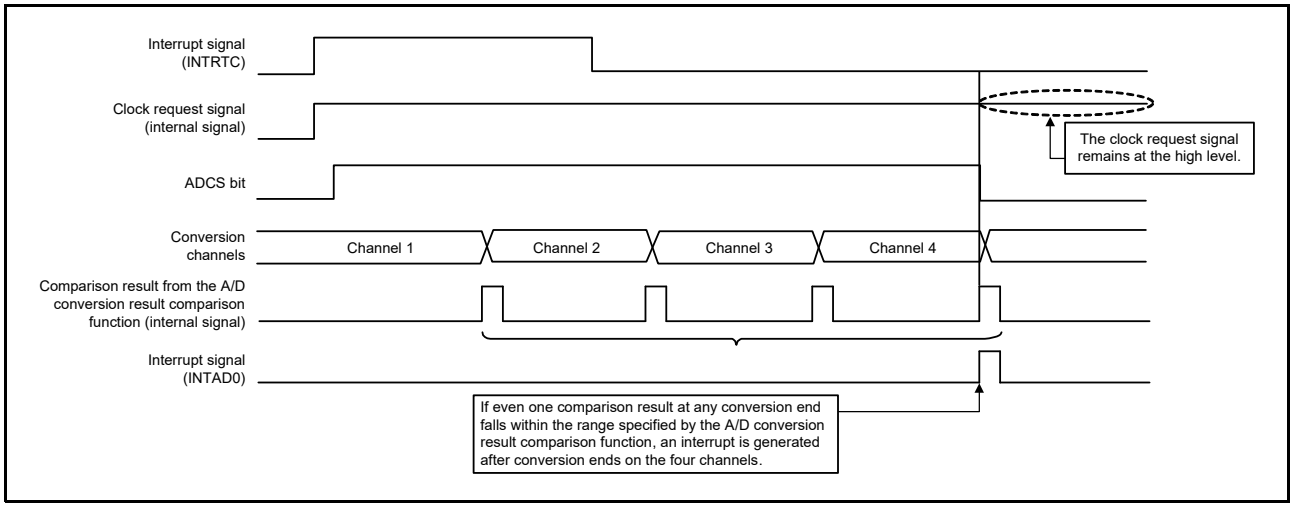
- **Realtime clock interrupt signal (INTRTC)**
- **32-bit interval timer interrupt signal (ELCTL0)**

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 ADM2.ADRCK bit and ADUL and ADLL registers), the A/D conversion end interrupt request signal (INTAD0) is generated.

   - While in the select mode
     When A/D conversion ends and an A/D conversion end interrupt request signal (INTAD0) is generated, the A/D converter returns to normal operation mode from SNOOZE mode. At this time, be sure to clear bit 2 (AWC) 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.

   - While in the scan mode
     If even one value of the A/D conversion results of the four channels falls within the range specified by the A/D conversion result comparison function, and A/D conversion end interrupt request signal (INTAD0) is generated, the A/D converter switches from the SNOOZE mode to the normal operation mode. At this time, be sure to clear bit 2 (AWC) 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 20 - 54 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 ADM2.ADRCK bit and ADUL and ADLL registers), the A/D conversion end interrupt request signal (INTAD0) is not generated.

   • While in the select mode
   If the A/D conversion end interrupt request signal (INTAD0) 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 result values of the four channels do not fall within the range specified by the A/D conversion result comparison function even once, and the A/D conversion end interrupt request signal (INTAD0) is not generated, 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 20 - 55 Operation Example When No Interrupt Is Generated After A/D Conversion Ends (While in Scan Mode)

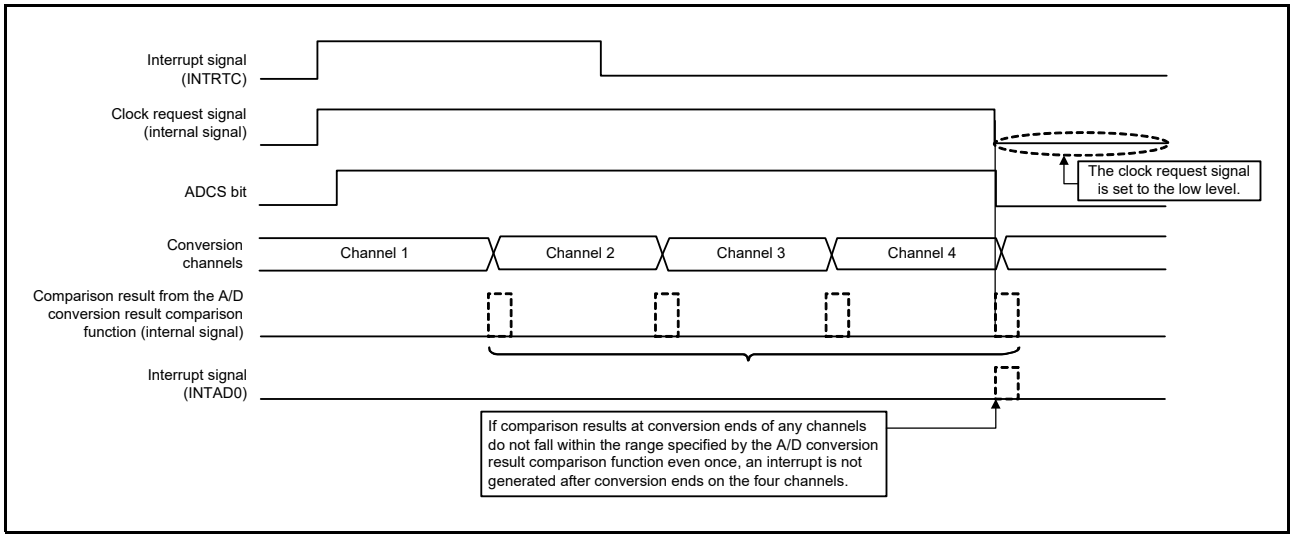
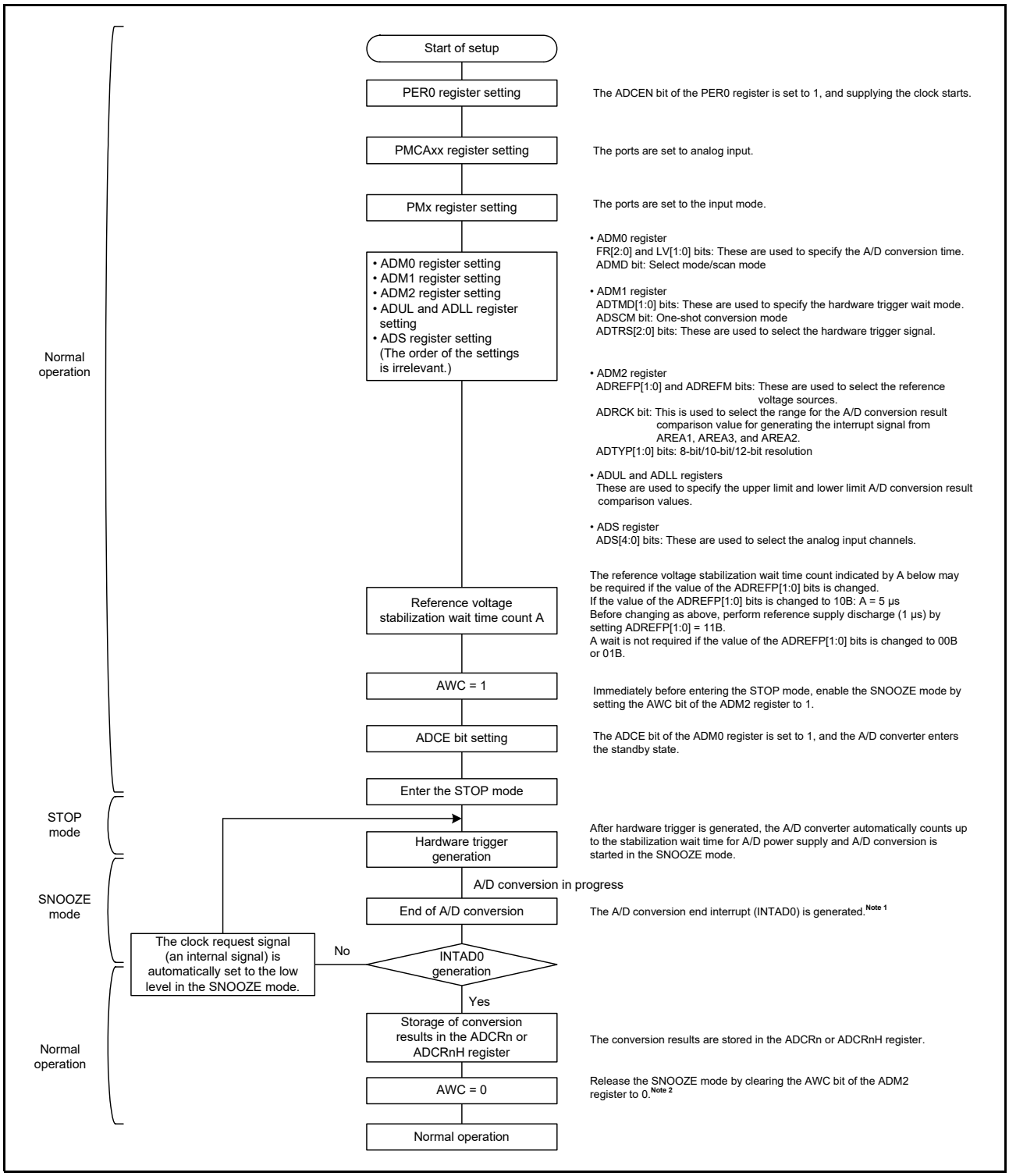


Figure 20 - 56 Flowchart for Settings in SNOOZE Mode (Hardware Trigger)



Note 1. If the A/D conversion end interrupt request signal (INTAD0) is not generated depending on the settings of the ADRCK bit and ADUL and ADLL registers, the result is not stored in the ADCRn or ADCRnH register. The A/D converter 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 the subsequent SNOOZE or normal operation mode. Be sure to clear the AWC bit to 0.

20.10 Operations When A/D Conversion Is in Contention with Another Trigger in Advanced Mode

The advanced mode allows the making of settings for multiple conversions with different trigger sources. Therefore, A/D conversion in progress may be in contention with another trigger source occurring within a certain interval.
This section describes operations when A/D conversion in progress is in contention with another trigger source.

Table 20 - 8 shows a priority list corresponding to the setting.

Table 20 - 8 Priority List Corresponding to the Setting

Priority	ADSn (n = 0 to 3)	Simultaneous Sampling Setting (ADSn.ADSPSCn[1:0])	Priority Setting (ADSCTL.ADPRIORn) (n = 0-3)
1	ADSn	1st S&H (ADSPSCn[1:0] = 01B)**Note 1**	—
2	ADSn	2nd S&H (ADSPSCn[1:0] = 10B)**Notes 1, 2**	—
3	ADSn	3rd S&H (ADSPSCn[1:0] = 11B)**Notes 1, 2**	—
4	ADS0**Note 3**	Simultaneous sampling does not proceed. (ADSPSCn[1:0] = 00B)	High (ADSCTL.ADPRIOR0 = 1)
5	ADS1**Note 3**		High (ADSCTL.ADPRIOR1 = 1)
6	ADS2**Note 3**		High (ADSCTL.ADPRIOR2 = 1)
7	ADS3**Note 3**		High (ADSCTL.ADPRIOR3 = 1)
8	ADS0**Note 3**		Low (ADSCTL.ADPRIOR0 = 0)
9	ADS1**Note 3**		Low (ADSCTL.ADPRIOR1 = 0)
10	ADS2**Note 3**		Low (ADSCTL.ADPRIOR2 = 0)
11	ADS3**Note 3**		Low (ADSCTL.ADPRIOR3 = 0)

Note 1. Do not set the same channels (first, second, and third S&H circuits) for the simultaneous sampling between the ADS0 to ADS3 registers.

Note 2. When the second and third S&H circuits are to be set for simultaneous sampling, setting of the first S&H circuit by using the ADSn register is required.

Note 3. The ADSn register with younger number has higher priority if any of them has the same priority setting.

Table 20 - 9 shows cases of the various types of contention and operations in outline for the respective cases.

Table 20 - 9 Cases of the Various Types of Contention and Operations in Outline for the Respective Cases

Priority**Note**		Operation	
Channel in Conversion	Channel Requesting a Conversion	Channel in Conversion	Channel Requesting a Conversion
One of the channels with priority 8 to 11	One of the channels with priority 1 to 7	Conversion is forcibly terminated.	A newly-requested conversion take precedence over the conversion in progress.
Other than the above		Operation continues.	Conversion is held pending.

Note The priority shown in **Table 20 - 8** applies.

20.10.1 Countermeasures for forcible termination of conversion due to contention

If the A/D conversion in progress is forcibly terminated due to contention, consider the following actions.

1. No actions are required if you do not need the result of the A/D conversion which has been forcibly terminated.
2. Change the priority setting so that you can avoid forcible termination of A/D conversion due to contention.
3. To avoid contention, adjust the timing of trigger source generation so that the trigger will not be generated.
   - For the A/D conversion time, see **Table 20 - 6 Selection of A/D Conversion Time**.
   - The completion of the conversion can be confirmed by checking the interrupt request flag register for INTAD0 to INTAD3 and the conversion interrupt status register (ADINTST).
     – INTAD0 to INTAD3 can be checked through the interrupt request flag register. For details, see **Section 29 Interrupt Functions**.
     – For details on the ADINTST register, see **20.3.14 Conversion interrupt status register (ADINTST)**.

Remark To confirm whether or not the A/D conversion is completed, check the ADINTST register as well as INTAD0 to INTAD3. This is because if a flag bit (ADINTSTnF) to indicate a failure of conversion is set in ADINTST, INTAD0 to INTAD3 will not be generated.

20.11 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 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

   1LSB is as follows when the resolution is 12 bits.

   $$1\ LSB = 1/2^{12} = 1/4096$$
   $$\approx 0.024\ \%FSR$$

   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, differential linearity errors, and 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 ±1/2LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2LSB 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 20 - 57 Overall Error

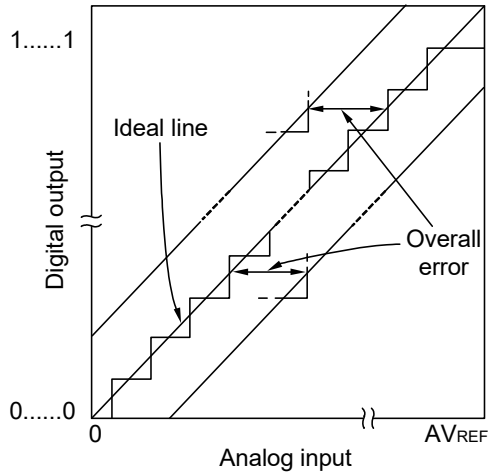
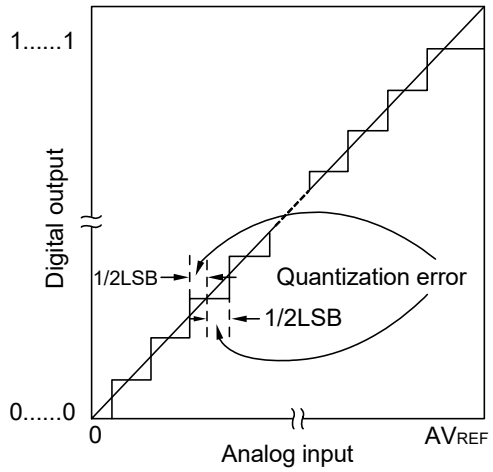


Figure 20 - 58 Quantization Error



4. Zero-scale error
   This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2LSB) 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 (3/2LSB) 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 – 3/2LSB) 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 1LSB, this indicates the difference between the actual measurement value and the ideal value of the width of output code.

Figure 20 - 59 Zero-scale Error

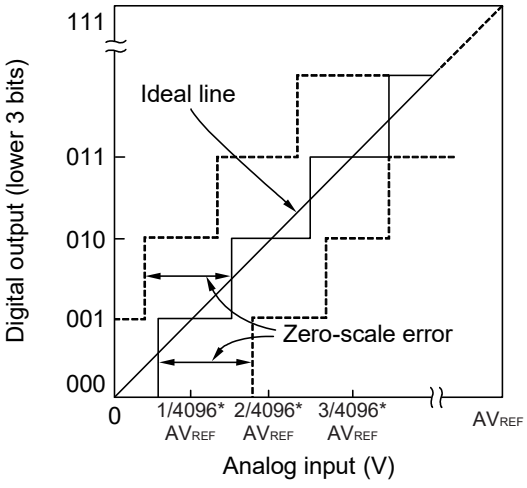


Figure 20 - 60 Full-scale Error

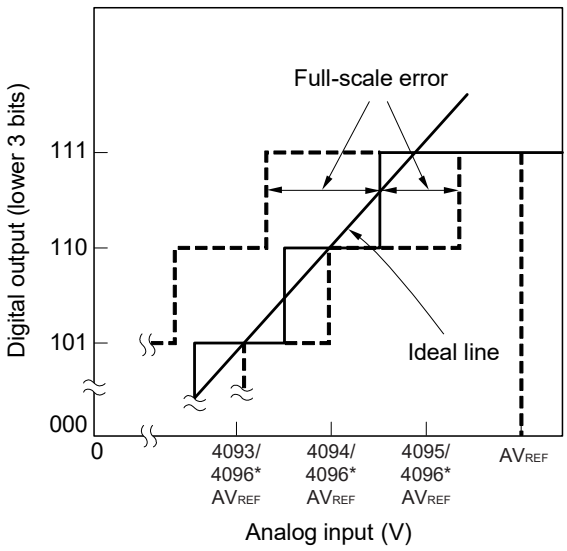


Figure 20 - 61 Integral Linearity Error

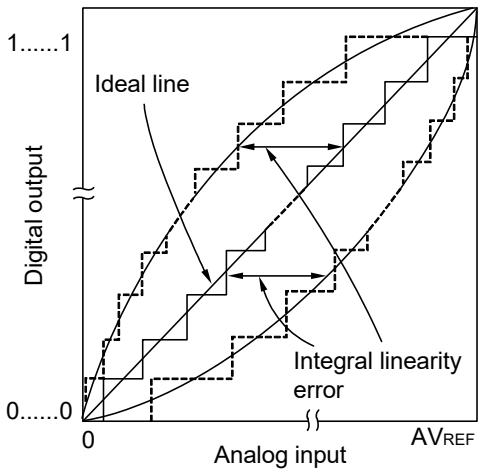
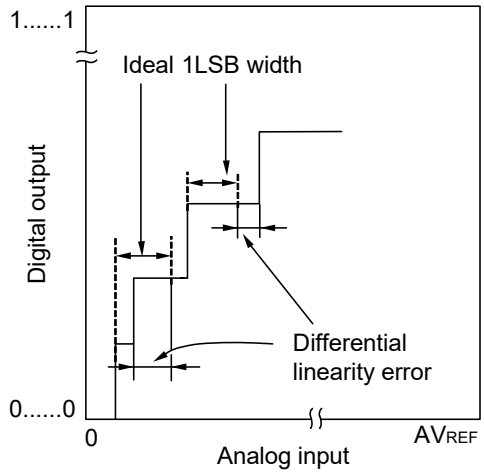
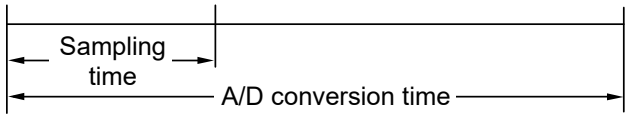


Figure 20 - 62 Differential Linearity Error



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 S&H circuit.



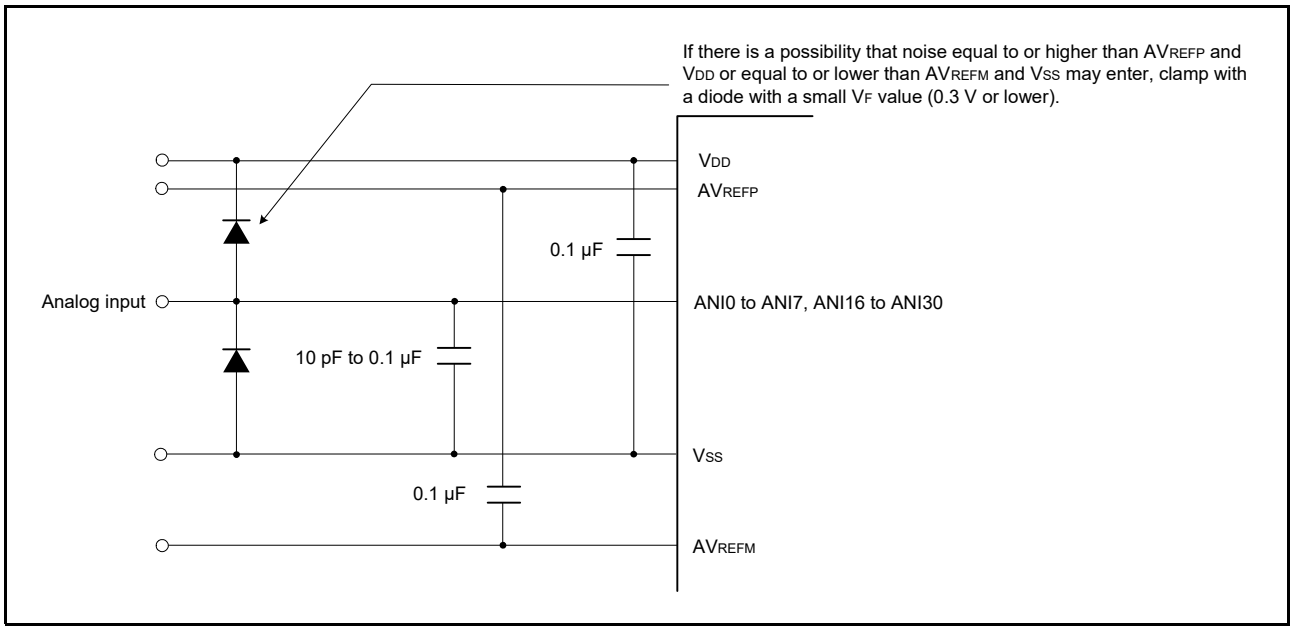
20.12 Points for Caution When the A/D Converter Is to Be Used

1. Operating current in STOP mode
   Shift to the 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 state, 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 ANI30 pins
   Observe the rated range of the input voltage on the ANI0 to ANI7 and ANI16 to ANI30 pins. If a voltage exceeding V_{DD} and AV_{REFP} or a voltage lower than 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 the internal reference voltage is selected as the reference voltage for the + side of the A/D converter, do not input a voltage higher than the internal reference voltage to a pin selected by the ADS register. However, it is no problem that a voltage higher than the internal reference voltage is input to a pin not selected by the ADS register.

   Caution For details about the internal reference voltage, see Section 43 Electrical Characteristics (T_A = –40 to +105°C) and Section 44 Electrical Characteristics (T_A = –40 to +125°C).

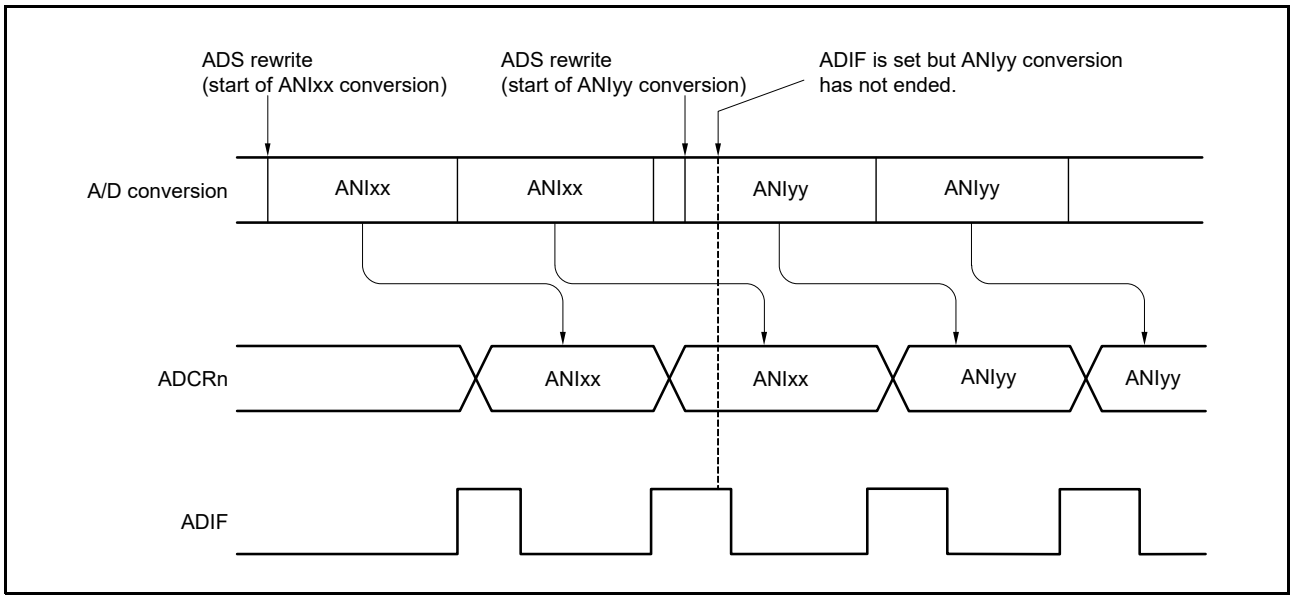
3. Conflicting operations
   <1> Conflict between the write access to the A/D conversion result register (ADCRn or ADCRnH) upon the end of conversion and the read access to the ADCRn or ADCRnH register with an instruction
   Reading from the ADCRn or ADCRnH register has the higher priority. After the read operation, the new conversion result is written to the ADCRn or ADCRnH register.
   <2> Conflict between the write access to the ADCRn or ADCRnH register for storage of the result of A/D conversion upon the end of conversion, the write access to the A/D converter mode register 0 (ADM0) with an instruction, and the write access to the analog input channel specification register (ADS) when the advanced mode is disabled
   Writing to the ADM0 and ADS registers has the higher priority. Writing to the ADCRn and ADCRnH registers is not performed, nor are the conversion end interrupt signals (INTAD0 to INTAD3) generated.
4. Noise countermeasures
   To maintain the 12-bit/10-bit resolution, attention must be paid to noise input to the AV_{REFP}, V_{DD}, ANI0 to ANI7, and ANI16 to ANI30 pins.
   <1> Connect a capacitor with a low equivalent resistance and a good frequency response (capacitance of about 0.1 µF) via the shortest possible run of relatively thick wiring to the V_{DD} and AV_{REFP} pins.
   <2> The higher the output impedance of an analog input source, the greater will be the effects of noise. To reduce the noise, connecting an external capacitor as shown in **Figure 20 - 63** 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 20 - 63 Connections of VDD, AVREFP, and Analog Input Pins



5. Analog input (ANIxx) pins
   <1> The analog input pins (ANI0 to ANI7 and ANI16 to ANI30) are also used as input port pins (P00 to P03, P10 to P17, P20 to P27, P120, P146, and P147). Do not change the output values for the port-pin functions P00 to P03, P10 to P17, P20 to P27, P120, P146, and P147 while A/D conversion of the signals on the ANI0 to ANI7 or ANI16 to ANI30 pins is selected and conversion is in progress, since doing so may lower the precision of the results of conversion.
   <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 avoid the input or output of digital signals and signals with similarly sharp transitions during conversion.
6. Input impedance of analog input (ANIxx) 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. To make sure that sampling is effective, however, we recommend using the converter with analog input sources that have output impedances no greater than 0.5 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 ANI30 to which the source is connected (see **Figure 20 - 63 Connections of VDD, AVREFP, and Analog Input Pins**). 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 sampling.
7. Interrupt request flag (ADIF)
   The interrupt request flag (ADIF) is not cleared to 0 even if the analog input channel specification register (ADS) or analog input channel specification register n (ADS0 to ADS3) 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 have been set before the ADS register is rewritten. When reading the ADIF flag immediately after rewriting to the ADS or ADS0 to ADS3 register, note that the ADIF flag is set although A/D conversion for the post-change analog input has not ended.
   When A/D conversion is stopped and then resumed, clear ADIF flag to 0 before the A/D conversion operation is resumed.

Figure 20 - 64 Timing of A/D Conversion End Interrupt Request Generation



8. Conversion results just after A/D conversion start
   While in the software trigger no-wait mode, hardware trigger no-wait mode, or advanced 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 + 2 cycles of the conversion clock (f_{AD}) after the ADCE bit was set to 1. Take measures such as polling the A/D conversion end interrupt request signal (INTAD0 to INTAD3) and removing the first conversion result.
9. A/D conversion result register (ADCRn, ADCRnH) read operation
   When a write operation is performed to A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), analog input channel specification register n (ADS0 to ADS3), or port mode control A register xx (PMCAxx), the contents of the ADCRn and ADCRnH registers may become undefined. After the completion of conversion, read the conversion result before writing to the ADM0, ADS, ADS0 to ADS3, or PMCAxx register; otherwise, an incorrect conversion result may be read.

10. Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 20 - 65 Internal Equivalent Circuit of ANIxx Pin

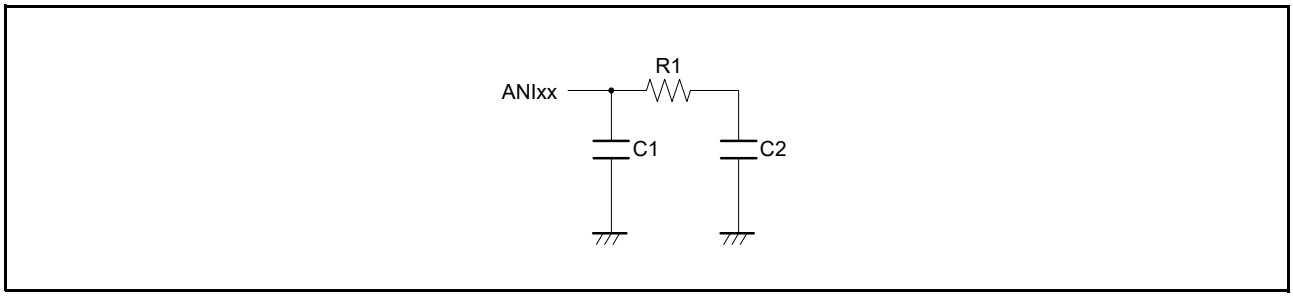


Table 20 - 10 Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{REFP}, V_{DD}	ANIxx Pin	R1 [kΩ]	C1 [pF]	C2 [pF]
4.5 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	ANI0 to ANI7	1.3	8	9
	ANI16 to ANI30	2.5	8	15
2.7 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	ANI0 to ANI7	1.9	8	9
	ANI16 to ANI30	3	8	15
2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	ANI0 to ANI7	2.2	8	9
	ANI16 to ANI30	3.5	8	15
1.8 V ≤ AV_{REFP} ≤ V_{DD} < 2.4 V	ANI0 to ANI7	6	8	9
	ANI16 to ANI30	8	8	15
1.6 V ≤ AV_{REFP} ≤ V_{DD} < 1.8 V	ANI0 to ANI7	12	8	9
	ANI16 to ANI30	15	8	15

Remark The resistance and capacitance values shown in **Table 20 - 10** are not guaranteed values.

11. Starting the A/D converter

Start the A/D converter after the AV_{REFP} and V_{DD} voltages stabilize.

Section 21 D/A Converter (DAC)

The number of channels of the D/A converter (DAC) depends on the product.

Table 21 - 1 Functions of D/A Converters

D/A Converter	When the Setting of the DACONF Bit Is 0 (10-bit Resolution Channel × 2)	When the Setting of the DACONF Bit Is 1 (10-bit Resolution Channel × 1, 8-bit Resolution Channel × 2)
DAC0	10-bit resolution	10-bit resolution
DAC1	10-bit resolution	8-bit resolution
DAC2	—	8-bit resolution

Table 21 - 2 Output Pins of D/A Converters

D/A Output Pins	20-pin Products	24- to 64-pin Products
ANO0 (DAC0 output)	✓	✓
ANO1 (DAC1 output)	✓	✓
ANO2 (DAC2 output)	—	✓

21.1 Functions of D/A Converters

The D/A converters convert digital inputs into analog signals at 8- or 10-bit resolution. They are used to control analog outputs through up to three independent channels. The D/A converters have the following features.

- When DACONF = 0: 10-bit resolution channel × 2
  When DACONF = 1: 10-bit resolution channel × 1, 8-bit resolution channel × 2
- R-2R ladder method
- Output analog voltage
  – 10-bit resolution: V_{DD} × m10/1024 (m10: Value set in the DACS0 or DACS1 register)
  – 8-bit resolution: V_{DD} × m8/256 (m8: Value set in the DACS1L or DACS2 register)
- Operation mode
  – Normal mode
  – Realtime output mode

21.2 Configuration of D/A Converters

Figures 21 - 1 and **21 - 2** show the block diagrams of D/A converter 0 (DAC0) and D/A converters 1 and 2 (DAC1 and DAC2), respectively.

Figure 21 - 1 Block Diagram of D/A Converter 0 (DAC0)

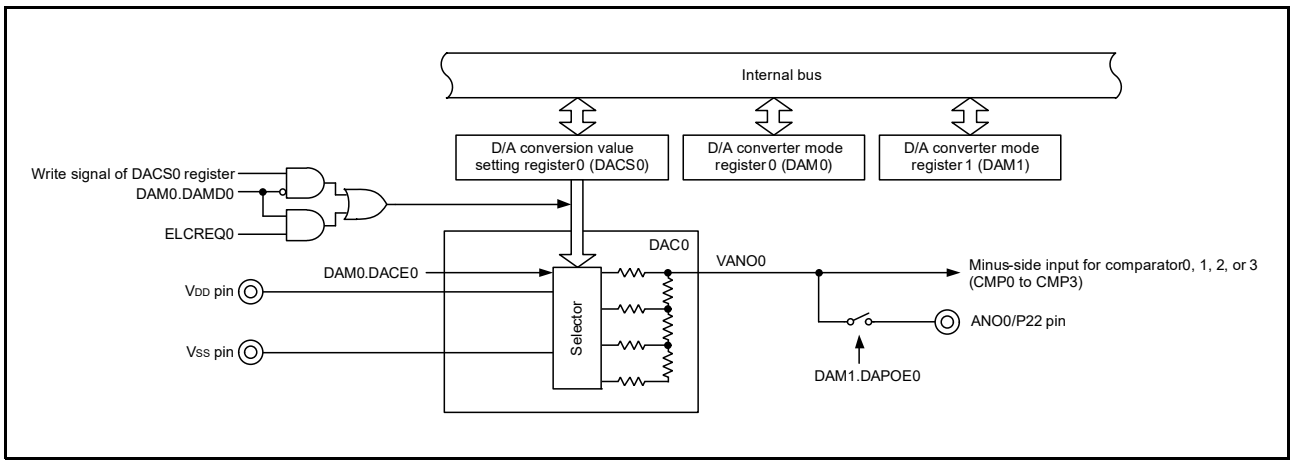
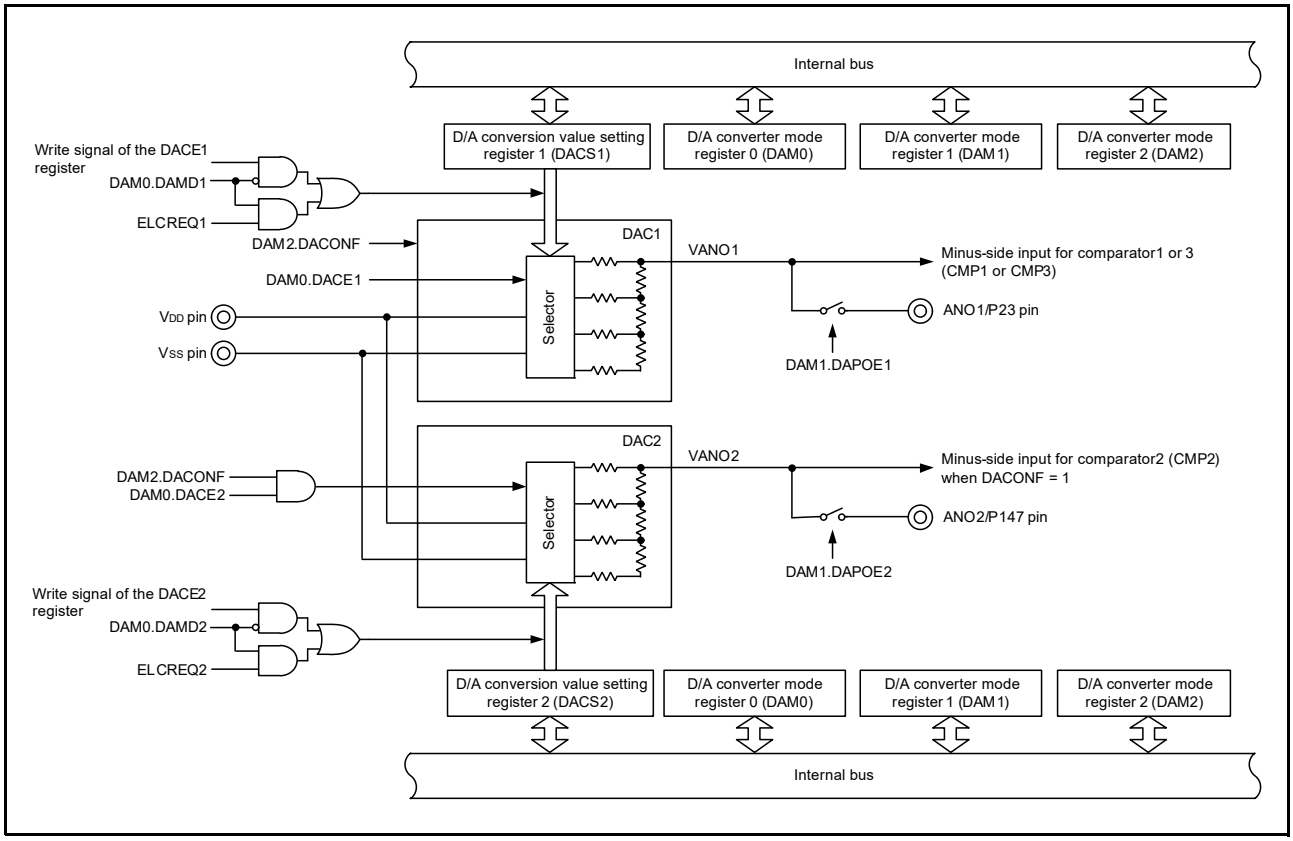


Figure 21 - 2 Block Diagram of D/A Converters 1 and 2 (DAC1 and DAC2)



Remark ELCREQ0, ELCREQ1, and ELCREQ2 are trigger signals (event signals from the ELC) for use in realtime output mode.

21.3 Registers to Control the D/A Converters

The following registers are used to control the D/A converters.

- Peripheral enable register 1 (PER1)
- Peripheral reset control register 1 (PRR1)
- D/A converter mode register 0 (DAM0)
- D/A converter mode register 1 (DAM1)
- D/A converter mode register 2 (DAM2)
- D/A conversion value setting register 0 (DACS0)
- D/A conversion value setting register 1 (DACS1)
- D/A conversion value setting register 2 (DACS2)
- Event output destination select registers n (ELSELRn) (n = 00 to 33)
- Port registers xx (Pxx) (xx = 2, 14)
- Port mode control A registers xx (PMCAxx) (xx = 2, 14)

21.3.1 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. If the D/A converters are to be used, be sure to set bit 7 (DACEN) of this register to 1. The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 21 - 3 Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	<3>	2	1	<0>
PER1	DACEN	0	PGACMPEN	TML32EN	DTCEN	0	0	DALIEN

DACEN	Control of supply of an input clock to the D/A converters
0	Stops supply of an input clock. • The SFRs used by the D/A converters cannot be written. • When an SFR used by the D/A converters is read, the value returned is 00H or 0000H.
1	Enables supply of an input clock. • The SFRs used by the D/A converters can be read and written.

Caution 1. When setting the D/A converters, be sure to set DACEN to 1 first.
If DACEN = 0, writing to the registers which control the D/A converters is ignored, and the value read is 00H (except for port mode registers 2 and 14 (PM2 and PM14) and port registers 2 and 14 (P2 and P14)).

Caution 2. Be sure to set bits 6, 2, and 1 to 0.

21.3.2 Peripheral reset control register 1 (PRR1)

The PRR1 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. To place the D/A converters in the reset state, set bit 7 (DACRES) of this register to 1. The PRR1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 21 - 4 Format of Peripheral Reset Control Register 1 (PRR1)

Address: F00FBH
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	3	2	1	<0>
PRR1	DACRES	0	PGACMPRES	TML32RES	0	0	0	DALIRES

DACRES	Control resetting of the D/A converters
0	The D/A converters are released from the reset state.
1	The D/A converters are in the reset state. • The SFRs for use with the D/A converters are initialized.

Caution Be sure to clear bits 6, 3, 2, and 1 to 0.

21.3.3 D/A converter mode register 0 (DAM0)

The DAM0 register controls the operation of the D/A converters. The DAM0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 21 - 5 Format of D/A Converter Mode Register 0 (DAM0)

Address: F0330H
After reset: 00H
R/W: R/W

Symbol	7	<6>	<5>	<4>	3	2	1	0
DAM0	0	DACE2	DACE1	DACE0	0	DAMD2	DAMD1	DAMD0

DACEi	D/A conversion operation control
0	Stops D/A conversion operation.
1	Enables D/A conversion operation.

DAMDi	D/A converter operation mode selection
0	Normal mode
1	Realtime output mode

Caution The settings of the DAMD2 and DACE2 bits have no effect when the value of the DACONF bit is 0.

Remark i = 0 to 2

21.3.4 D/A converter mode register 1 (DAM1)

The DAM1 register controls analog outputs from the D/A converters. The DAM1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 21 - 6 Format of D/A Converter Mode Register 1 (DAM1)

Address: F0331H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
DAM1	0	0	0	0	0	DAPOE2	DAPOE1	DAPOE0

DAPOEi	Control of analog outputs from the D/A converters
0	Stops outputs from the ANOi pin.
1	Enables outputs from the ANOi pin.

Caution When output of the D/A converter voltage from an ANOi pin is enabled by setting the DAPOEi bit to 1, set the corresponding port mode control A register xx (PMCAxx) bit to specify analog usage of the pin on which the ANOi pin function is multiplexed.

Remark i = 0 to 2

21.3.5 D/A converter mode register 2 (DAM2)

The DAM2 register is used to select the resolution for D/A converters 1 and 2. The DAM2 register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 21 - 7 Format of D/A Converter Mode Register 2 (DAM2)

Address: F0332H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DAM2	0	0	0	0	0	0	0	DACONF

DACONF	Selection of the resolution for D/A converter 1 (DAC1)	Selection of the resolution for D/A converter 2 (DAC2)
0	10-bit resolution	—
1	8-bit resolution	8-bit resolution

21.3.6 D/A conversion value setting register 0 (DACS0)

The DACS0 register is used to set the analog voltage value to be output to the comparator when D/A converter 0 (DAC0) is to be used. The DACS0 register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 21 - 8 Format of D/A Conversion Value Setting Register 0 (DACS0)

Address: F0334H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
DACS0	0	0	0	0	0	0	DACS09	DACS08

7	6	5	4	3	2	1	0
DACS07	DACS06	DACS05	DACS04	DACS03	DACS02	DACS01	DACS00

Remark The relation between the resolution and analog output voltage (VANO0) of the D/A converter 0 is as follows.
VANO0 = V_{DD} × (DACS0)/1024

When D/A converter 0 is not to be used, set the DACE0 bit to 0 (stopping D/A conversion operation) and set the DACS0 register to 0000H to prevent current from flowing into the R-2R resistor ladder to reduce the flow of unnecessary current.

21.3.7 D/A conversion value setting register 1 (DACS1)

The DACS1 register is used to set the analog voltage value to be output to the comparator when D/A converter 1 (DAC1) is to be used. The DACS1 register can be set by a 16-bit memory manipulation instruction. When the value of the DACONF bit is 1, the 8 lower-order bits of the DACS1 register can be handled as DACS1L, which can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 21 - 9 Format of D/A Conversion Value Setting Register 1 (DACS1)

Address: F0336H, F0337H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
DACS1	0	0	0	0	0	0	DACS19	DACS18

	7	6	5	4	3	2	1	0
(DACS1L)	DACS17	DACS16	DACS15	DACS14	DACS13	DACS12	DACS11	DACS10

Remark The relation between the resolution and analog output voltage (VANO1) of D/A converter 1 is as follows.
When the DACONF bit is set to 0: VANO1 = V_{DD} × (DACS1)/1024
When the DACONF bit is set to 1: VANO1 = V_{DD} × (DACS1L)/256

When D/A converter 1 is not to be used, set the DACE1 bit to 0 (stopping D/A conversion operation) and set the DACS1 register to 0000H to prevent current from flowing into the R-2R resistor ladder to reduce the flow of unnecessary current.

21.3.8 D/A conversion value setting register 2 (DACS2)

The DACS2 register is used to set the analog voltage value to be output to the comparator when D/A converter 2 (DAC2) is to be used. The DACS2 register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 21 - 10 Format of D/A Conversion Value Setting Register 2 (DACS2)

Address: F0333H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DACS2	DACS27	DACS26	DACS25	DACS24	DACS23	DACS22	DACS21	DACS20

Remark The relation between the resolution and analog output voltage (VANO2) of D/A converter 2 is as follows.
VANO2 = V_{DD} × (DACS2)/256

When D/A converter 2 is not to be used, set the DACE2 bit to 0 (stopping D/A conversion operation) and set the DACS2 register to 00H to prevent current from flowing into the R-2R resistor ladder to reduce the flow of unnecessary current.

21.3.9 Register to control the event output from the event link controller

When the realtime output mode of the D/A converters is used, D/A conversion is triggered by an event signal from the event link controller. For details, see **28.3.1 Event output destination select registers n (ELSELRn) (n = 00 to 33)**.

21.3.10 Registers for controlling the port functions multiplexed with the analog outputs of the D/A converter

Set the following registers to control the port functions multiplexed with the analog outputs of the D/A converter.

- Port mode registers xx (PMxx)
- Port mode control A registers xx (PMCAxx)

For details, see **7.3.1 Port mode registers xx (PMxx)** and **7.3.7 Port mode control A registers xx (PMCAxx)**.

When the pins multiplexed with ANO0 and ANO1 are to be used for analog outputs of the D/A converters, set the port mode register (PMxx) bit corresponding to each port to 1, and specify analog output using the port mode control A register xx (PMCAxx).

Remark xx = 2, 14

21.4 Operations of D/A Converters

21.4.1 Operation in normal mode

D/A conversion is handled using write operation to the DACSi register as the trigger.
The setting method is described below.

<1> Set the DACRES bit of the peripheral reset control register 1 (PRR1) to 0 to release the D/A converters from the reset state.
<2> Set the DACEN bit of the peripheral enable register 1 (PER1) to 1 to start the supply of the input clock to the D/A converters.
<3> Set the DAMDi bit of the D/A converter mode register 0 (DAM0) to 0 (normal mode).
<4> Set the DACONF bit in the D/A converter mode register 2 (DAM2).
<5> Set the analog voltage value to be output to the comparator in the D/A conversion value setting register i (DACSi).

Steps <1> to <5> above constitute the initial settings.

<6> Set the DACEi bit of the DAM0 register to 1 (enabling D/A conversion operation).
D/A conversion starts, and then, after the settling time elapses, the analog voltage set in step <5> is output to the comparator.
<7> To start subsequent D/A conversions, write to the DACSi register.

The previous D/A conversion result is retained until the next D/A conversion starts.
When the DACEi bit of the DAM0 register is set to 0 (stopping D/A conversion operation), D/A conversion stops.

Caution 1. Even if 1, 0, and then 1 are set in the DACEi bit, the analog voltage set by the DACSi register is output to the comparator when a settling time has elapsed after 1 is set for the last time.
Caution 2. If the DACSi register is rewritten during the settling time, D/A conversion is aborted and reconversion by using the rewritten values starts.
Caution 3. Set the DACRES bit of the PRR1 register to 1 to initialize all circuits of the D/A converters.

Remark i = 0 to 2

21.4.2 Operation in realtime output mode

D/A conversion is handled on each channel using the event signals from the ELC as triggers. The setting method is described below.

<1> Set the DACRES bit of the peripheral reset control register 1 (PRR1) to 0 to release the D/A converters from the reset state.
<2> Set the DACEN bit of the peripheral enable register 1 (PER1) to 1 to start the supply of the input clock to the D/A converters.
<3> Set the DAMDi bit of the D/A converter mode register 0 (DAM0) to 0 (normal mode).
<4> Set the analog voltage value to be output to the comparator in the D/A conversion value setting register i (DACSi).
<5> Set the DACEi bit of the DAM0 register to 1 (enabling D/A conversion operation).
D/A conversion starts, and then, after the settling time elapses, the analog voltage set in step <4> is output to the comparator.
<6> Use the register (ELSELRn) that controls the event signal output from the event link controller to set the trigger signal for use in realtime output mode.
<7> Set the DAMDi bit of the DAM0 register to 1 (realtime output mode).
<8> Start the operation of the event source.

Steps <1> to <8> above constitute the initial settings.

<9> Upon generation of the trigger signals for use in realtime output mode, D/A conversion starts and the analog voltage set in step <4> will be output to the comparator after a settling time has elapsed.

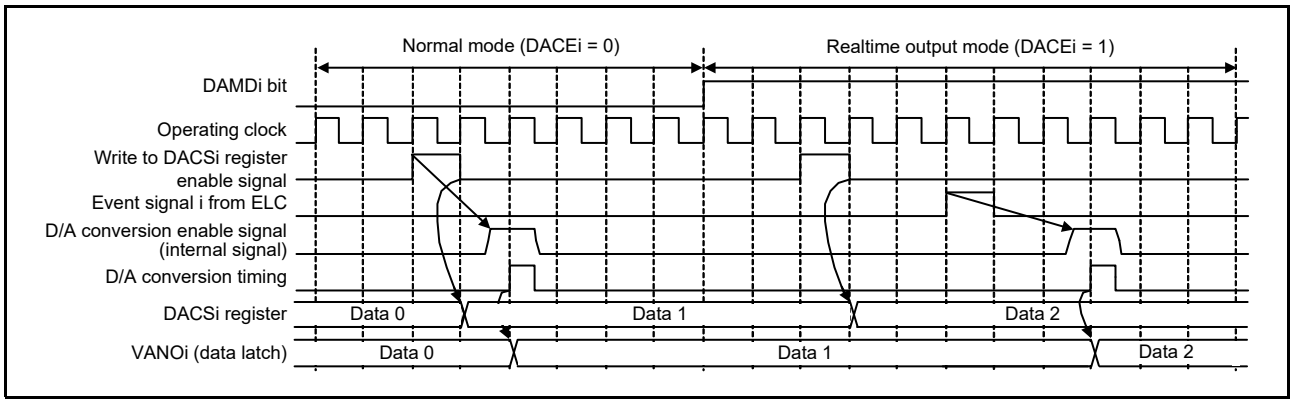
Set the analog voltage value to be output to the comparator in the DACSi register before starting the next D/A conversion (a trigger signal for use in realtime output mode is generated).
When the DACEi bit of the DAM0 register is set to 0 (stopping D/A conversion operation), D/A conversion stops.

Caution 1. Even if 1, 0, and then 1 are set in the DACEi bit, the analog voltage set by the DACSi register is output to the comparator when a settling time has elapsed after 1 is set for the last time.
Caution 2. Set the interval between each generation of the trigger signal for use in realtime output mode of the same channel to longer than the settling time. If a trigger signal for use in realtime output mode is generated during the settling time, D/A conversion is aborted and reconversion starts.
Caution 3. Set the interval between each generation of the trigger signal for use in realtime output mode of the same channel to longer than three cycles of f_{CLK}. When a trigger is generated consecutively at intervals of three or fewer f_{CLK} clock cycles, D/A conversion is started using only the first trigger.
Caution 4. Set the DACRES bit of the PRR1 register to 1 to initialize all circuits of the D/A converters.

21.4.3 Timing for outputting D/A conversion value

Figure 21 - 11 shows the timing for outputting D/A conversion value.

Figure 21 - 11 Timing for Outputting D/A Conversion Value



- Normal mode and realtime output mode (when conversion is disabled)
  The value is written to the data latch (output from VANOi) one cycle of the operating clock after the DACSi register is written.
- Realtime output mode (when conversion is enabled)
  The value is written to the data latch (output from VANOi) three cycles of the operating clock after an event signal from the ELC is accepted.

Remark i = 0 to 2

21.5 Points for Caution When the D/A Converters Are to Be Used

Observe the following cautions when using the D/A converters.

1. The operation of the D/A converters continues in the HALT and STOP modes. To lower the power consumption, therefore, clear the DACEi bit to 0, and execute the HALT or STOP instruction after stopping the operation of the D/A converters.
2. To stop the realtime output mode (including when changing to normal mode), one of the following procedures must be used:
   - Wait for at least three clock cycles after stopping the trigger output source and then set bits DACEi and DAMDi to 0.
   - After setting bits DACEi and DAMDi to 0, set the DACEN bit of the PER1 register to 0 (stopping supply of the input clock to the DAC).
     Note that setting the DACEN bit to 0 does not initialize the D/A converters.
     Use bit 7 (DACRES) of PRR1 to initialize the D/A converters and their SFRs.
3. When D/A conversion and output from an ANOi pin are enabled, do not use the analog usage of the pin on which the ANOi pin function is multiplexed.
4. In realtime output mode, set the value of the DACSi register before a trigger signal for use in realtime output mode is generated. Do not change the set value of the DACSi register while the trigger signal is being output.
5. Since the output impedance of the D/A converters is high, no current can be drawn from the ANO0 to ANO2 pins. If a load has a low input impedance, insert a follower amplifier between the load and each of the ANO0 to ANO2 pins before use. In addition, make the connection of each such pin to the load through a follower amplifier such that the wiring lengths are as short as possible since long wiring leads to a high output impedance.
   If a relatively long wiring run is not avoidable, take measures such as placing a ground pattern around the wiring area.
6. When entering STOP mode while realtime output mode is enabled, disable linking of ELC events before entering STOP mode.
7. When output of the D/A converter voltage from an ANOi pin is enabled by setting the DAPOEi bit to 1, set the corresponding port mode control A register xx (PMCAxx) bit to specify analog usage of the pin on which the ANOi pin function is multiplexed.

Remark i = 0 to 2

Section 22 Comparator Module (CMP)

The number of comparators (CMPs) depends on the number of pins as indicated in the table below.

Item	20-pin Products	24- to 64-pin Products
Comparator 0	✓	✓
Comparator 1	✓	✓
Comparator 2	✓	✓
Comparator 3	—	✓

22.1 Functions of the Comparators

The comparators have the following functions.

- The reference voltage is selectable from among external voltage inputs or D/A converter outputs.
- For position sensorless control of the motor, the motor position when the motor stops can be detected.
- For position sensorless control of the motor, the motor can be driven by zero-crossing detection of the 3-phase voltage.
- The results of comparison by comparators 0 to 3 can be output from pins VCOUT0 to VCOUT3.
- The output signals of the comparator can be used to control the PWM output from 16-bit timers KB30, KB31, and KB32 or request the restart of the timers (see **Section 15 16-bit Timers KB30, KB31, and KB32**).
- The elimination width of the noise eliminating digital filter is selectable.
- An interrupt signal can be generated upon detecting a valid edge of a comparator output.

Table 22 - 1 Functional Overview of the Comparators

Item	Description
Comparator	• 4 channels available (comparator 0 to comparator 3) • The reference voltage for the negative side is selectable: – Comparator 0: Selectable as external reference voltage IVREF0 or the output of D/A converter 0 – Comparator 1: Selected from among external reference voltage IVREF0 or the output of D/A converter 0 or 1 – Comparator 2: Selected from among external reference voltage IVREF0 or IVREF1, or the output of D/A converter 0 or 2 (when DAM2.DACONF = 1) – Comparator 3: Selected from among external reference voltage IVREF0 or IVREF1, or the output of D/A converter 0 or 1 • The positive-side input for comparator 3 is connectable to the output of the PGA. • High-level output when the positive-side input voltage is greater than the negative-side input voltage, and low-level output when the positive-side input voltage is lower less than the negative-side input voltage • The elimination width of the noise eliminating digital filter is selectable. • The output can be inverted. • Results of comparison can be output from pins (VCOUT0 to VCOUT3). • An interrupt request can be generated in response to the detection of a valid edge. • In combination with other functions, the initial position of the motor can be detected. Also, the motor position during rotation can be detected by zero-crossing detection of the 3-phase voltage of the motor. Forced cutoff of the 6-phase PWM output of the timer can be set to or released by overcurrent detection. • In combination with the TAU, the comparator output can be cut off or released by using timer window output mode.

22.2 Configuration of the Comparators

Figure 22 - 1 is a block diagram of the comparators.

Figure 22 - 1 Block Diagram of the Comparators (1/2)

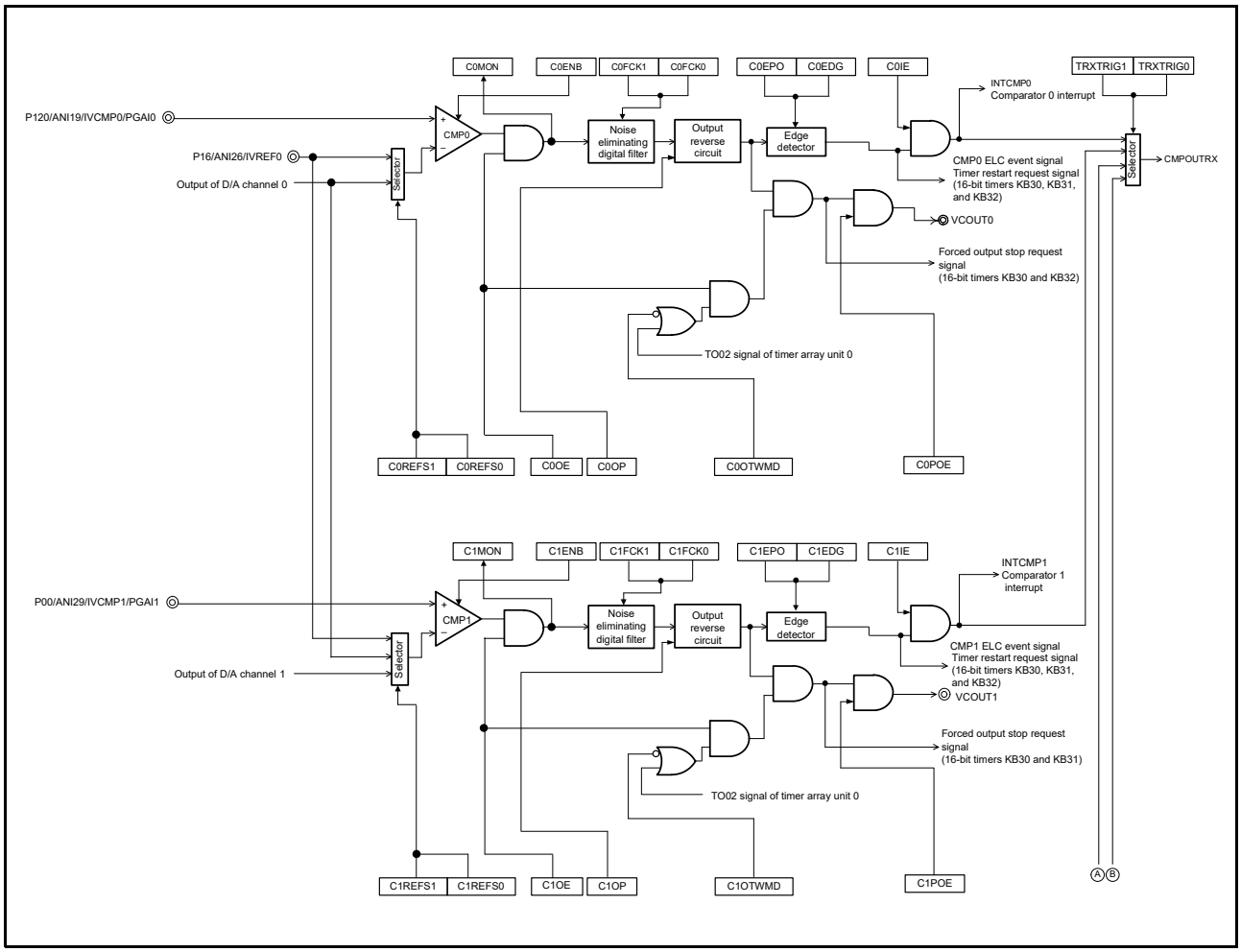
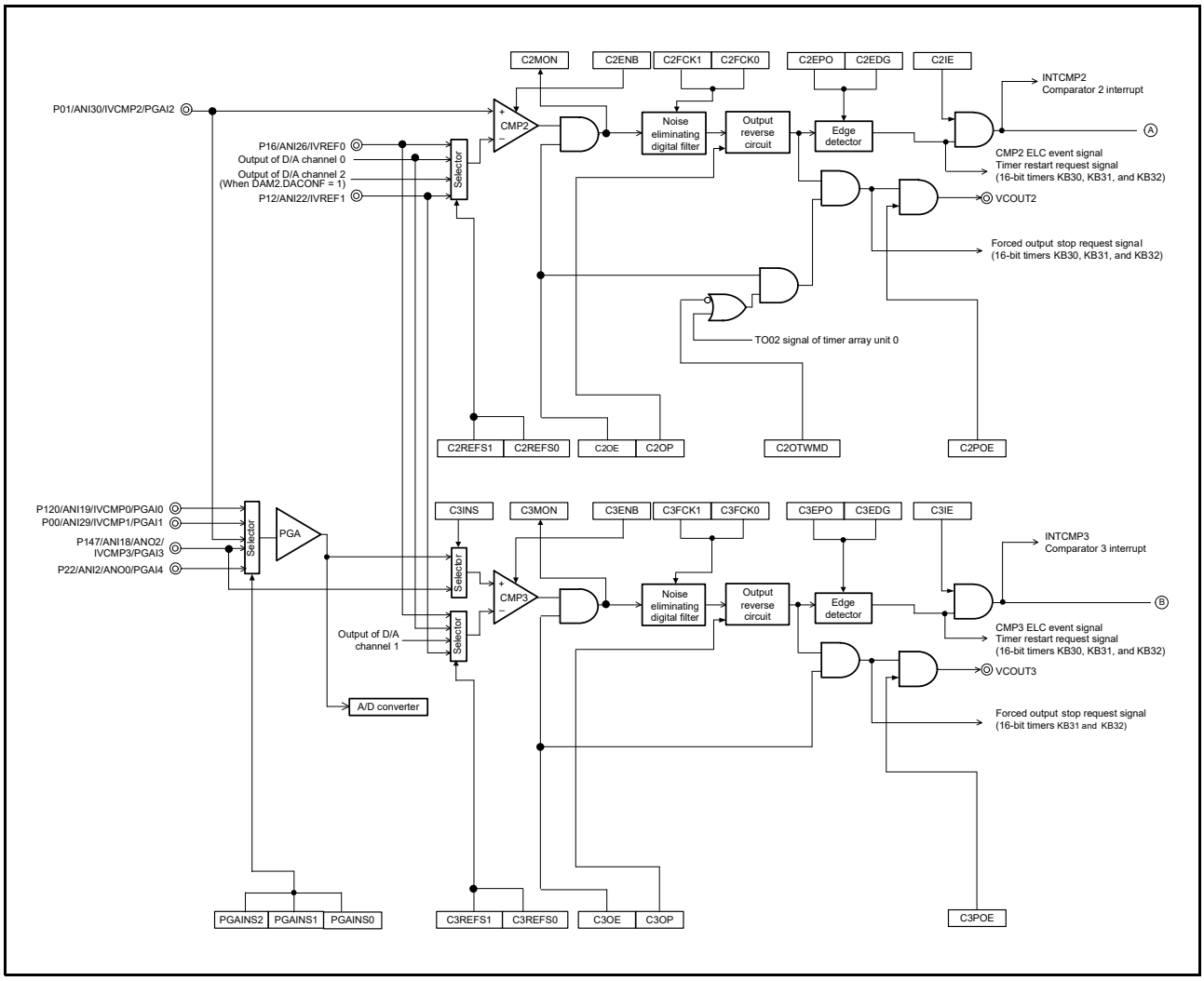


Figure 22 - 1 Block Diagram of the Comparators (2/2)



22.3 Registers to Control the Comparators

The following registers are used to control the comparators.

- Peripheral enable register 1 (PER1)
- Peripheral reset control register 1 (PRR1)
- Comparator mode setting register 0 (COMPMDR0)
- Comparator mode setting register 1 (COMPMDR1)
- Comparator filter control register 0 (COMPFIR0)
- Comparator filter control register 1 (COMPFIR1)
- Comparator output control register 0 (COMPOCR0)
- Comparator output control register 1 (COMPOCR1)
- Comparator 0 input signal selection control register (CMP0SEL)
- Comparator 1 input signal selection control register (CMP1SEL)
- Comparator 2 input signal selection control register (CMP2SEL)
- Comparator 3 input signal selection control register (CMP3SEL)
- Comparator output control register 2 (COMPOCR2)
- Port mode registers xx (PMxx) (xx = 0, 1, 3, 5, 12, 14)
- Port registers xx (Pxx) (xx = 0, 1, 3, 5, 12, 14)
- Port mode control A registers xx (PMCAxx) (xx = 0, 1, 12, 14)
- Peripheral I/O redirection registers x (PIORx) (x = 3)

22.3.1 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. Be sure to set bit 5 (PGACMPEN) of this register to 1 to use the comparator module. The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 22 - 2 Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	<3>	2	1	<0>
PER1	DACEN	0	PGACMPEN	TML32EN	DTCEN	0	0	DALIEN

PGACMPEN	Control of supply of an input clock to the PGA and comparator module
0	Stops supply of an input clock. • The SFRs used by the comparator module cannot be written.
1	Enables supply of an input clock. • The SFRs used by the comparator module can be read and written.

Caution When setting the comparator module, be sure to set the PGACMPEN bit to 1 first. If PGACMPEN = 0, writing to a control register of the comparator module is ignored, and all read values are default values (except for port mode registers 0 to 3, 5, 12, 14 (PM0 to PM3, PM5, PM12, PM14), port registers 0 to 3, 5, 12, 14 (P0 to P3, P5, P12, P14), and peripheral I/O redirection register 3 (PIOR3)).

22.3.2 Peripheral reset control register 1 (PRR1)

The PRR1 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. Set bit 5 (PGACMPRES) of this register to 1 to place the comparator module in the reset state. The PRR1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 22 - 3 Format of Peripheral Reset Control Register 1 (PRR1)

Address: F00FBH
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	3	2	1	<0>
PRR1	DACRES	0	PGACMPRES	TML32RES	0	0	0	DALIRES

PGACMPRES	Control resetting of the PGA and comparator module
0	The comparator module is released from the reset state.
1	The comparator module is in the reset state.

22.3.3 Comparator mode setting register 0 (COMPMDR0)

The COMPMDR0 register is used to enable or disable the operation of the comparators and monitor the output of the comparators. Setting the CiENB bit to 0 is prohibited when COMPOCR0.CiOE = 1. Also, setting the CiENB bit to 1 (operation enabled) is prohibited in the following cases (i = 0, 1).

- The output of D/A converter 0 is selected for the negative-side input of comparator 0, and the D/A conversion operation of D/A converter 0 is stopped (DAM0.DACE0 = 0).
- The output of D/A converter 0 is selected for the negative-side input of comparator 1, and the D/A conversion operation of D/A converter 0 is stopped (DAM0.DACE0 = 0).
- The output of D/A converter 1 is selected for the negative-side input of comparator 1, and the D/A conversion operation of D/A converter 1 is stopped (DAM0.DACE1 = 0).

The COMPMDR0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 22 - 4 Format of Comparator Mode Setting Register 0 (COMPMDR0) (1/2)

Address: F0340H
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	<3>	2	<1>	<0>
COMPMDR0	C1MON	0	C1POE	C1ENB	C0MON	0	C0POE	C0ENB

C1MON	Monitoring flag of comparator 1 Notes 1, 2
0	The input voltage of comparator 1 (IVCMP1) is lower than the reference voltage of comparator 1, or comparator 1 is stopped.
1	The input voltage of comparator 1 (IVCMP1) is greater than the reference voltage of comparator 1.

C1POE	Enabling the VCOUT1 pin output
0	Output of the result from comparator 1 on the VCOUT1 pin is disabled.
1	Output of the result from comparator 1 on the VCOUT1 pin is enabled.

C1ENB	Enabling operation of comparator 1
0	The operation of comparator 1 is disabled.
1	The operation of comparator 1 is enabled.

C0MON	Monitoring flag of comparator 0 Notes 1, 2
0	The input voltage of comparator 0 (IVCMP0) is lower than the reference voltage of comparator 0, or comparator 0 is stopped.
1	The input voltage of comparator 0 (IVCMP0) is greater than the reference voltage of comparator 0.

C0POE	Enabling the VCOUT0 pin output
0	Output of the result from comparator 0 on the VCOUT0 pin is disabled.
1	Output of the result from comparator 0 on the VCOUT0 pin is enabled.

Figure 22 - 4 Format of Comparator Mode Setting Register 0 (COMPMDR0) (2/2)

C0ENB	Enabling operation of comparator 0
0	The operation of comparator 0 is disabled.
1	The operation of comparator 0 is enabled.

Note 1. The initial value is 0 immediately after release from a reset. However, the value is undefined when both the C0ENB and C1ENB bits are set to 0 after operation of the comparator has been enabled.

Note 2. Any value written to this bit is ignored.

22.3.4 Comparator mode setting register 1 (COMPMDR1)

The COMPMDR1 register is used to enable or disable the operation of the comparators and monitor the output of the comparators. Setting the CiENB bit to 0 is prohibited when COMPOCR1.CiOE = 1. Also, setting the CiENB bit to 1 (operation enabled) is prohibited in the following cases (i = 2, 3).

- The output of D/A converter 0 is selected for the negative-side input of comparator 2, and the D/A conversion operation of D/A converter 0 is stopped (DAM0.DACE0 = 0).
- The output of D/A converter 2 is selected for the negative-side input of comparator 2, and the D/A conversion operation of D/A converter 2 is stopped (DAM0.DACE2 = 0).
- The output of D/A converter 0 is selected for the negative-side input of comparator 3, and the D/A conversion operation of D/A converter 0 is stopped (DAM0.DACE0 = 0).
- The output of D/A converter 1 is selected for the negative-side input of comparator 3, and the D/A conversion operation of D/A converter 1 is stopped (DAM0.DACE1 = 0).
- The PGA output is selected for the positive-side input of comparator 3 (CMP3SEL.C3INS = 1) while operation of the programmable gain amplifier is stopped (PGACTL.PGAEN = 0).

The COMPMDR1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 22 - 5 Format of Comparator Mode Setting Register 1 (COMPMDR1) (1/2)

Address: F0344H
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	<3>	2	<1>	<0>
COMPMDR1	C3MON	0	C3POE	C3ENB	C2MON	0	C2POE	C2ENB

C3MON	Monitoring flag of comparator 3 Notes 1, 2
0	The input voltage of comparator 3 (IVCMP3) is lower than the reference voltage of comparator 3, or comparator 3 is stopped.
1	The input voltage of comparator 3 (IVCMP3) is greater than the reference voltage of comparator 3.

C3POE	Enabling the VCOUT3 pin output
0	Output of the result from comparator 3 on the VCOUT3 pin is disabled.
1	Output of the result from comparator 3 on the VCOUT3 pin is enabled.

C3ENB	Enabling operation of comparator 3
0	The operation of comparator 3 is disabled.
1	The operation of comparator 3 is enabled.

C2MON	Monitoring flag of comparator 2 Notes 1, 2
0	The input voltage of comparator 2 (IVCMP2) is lower than the reference voltage of comparator 2, or comparator 2 is stopped.
1	The input voltage of comparator 2 (IVCMP2) is greater than the reference voltage of comparator 2.

Figure 22 - 5 Format of Comparator Mode Setting Register 1 (COMPMDR1) (2/2)

C2POE	Enabling the VCOUT2 pin output
0	Output of the result from comparator 2 on the VCOUT2 pin is disabled.
1	Output of the result from comparator 2 on the VCOUT2 pin is enabled.

C2ENB	Enabling operation of comparator 2
0	The operation of comparator 2 is disabled.
1	The operation of comparator 2 is enabled.

Note 1. The initial value is 0 immediately after release from a reset. However, the value is undefined when both the C2ENB and C3ENB bits are set to 0 after operation of the comparator has been enabled.

Note 2. Any value written to this bit is ignored.

22.3.5 Comparator filter control register 0 (COMPFIR0)

The COMPFIR0 register is used to control the digital noise filter. The COMPFIR0 register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 22 - 6 Format of Comparator Filter Control Register 0 (COMPFIR0)

Address: F0341H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
COMPFIR0	C1EDG	C1EPO	C1FCK1	C1FCK0	C0EDG	C0EPO	C0FCK1	C0FCK0

C1EDG	Selecting the edge detection for comparator 1 Note 1
0	An interrupt request is generated upon detection of one edge from comparator 1.
1	An interrupt request is generated upon detection of both edges from comparator 1.

C1EPO	Switching the edge sense of comparator 1 Note 1
0	An interrupt request is generated on the rising edge of comparator 1.
1	An interrupt request is generated on the falling edge of comparator 1.

C1FCK1	C1FCK0	Selecting the filter for comparator 1 Note 1
0	0	The filter for comparator 1 is disabled.
0	1	The filter for comparator 1 is enabled, and sampling proceeds at a cycle of f_{CLK} or f_{PLL}.
1	0	The filter for comparator 1 is enabled, and sampling proceeds at a cycle of f_{CLK} / 8 or f_{PLL}/8.
1	1	The filter for comparator 1 is enabled, and sampling proceeds in synchronization with the TAU output (TO01). Note 3

C0EDG	Selecting the edge detection for comparator 0 Note 2
0	An interrupt request is generated upon detection of one edge from comparator 0.
1	An interrupt request is generated upon detection of both edges from comparator 0.

C0EPO	Switching the edge sense of comparator 0 Note 2
0	An interrupt request is generated on the rising edge of comparator 0.
1	An interrupt request is generated on the falling edge of comparator 0.

C0FCK1	C0FCK0	Selecting the filter for comparator 0 Note 2
0	0	The filter for comparator 0 is disabled.
0	1	The filter for comparator 0 is enabled, and sampling proceeds at a cycle of f_{CLK} or f_{PLL}.
1	0	The filter for comparator 0 is enabled, and sampling proceeds at a cycle of f_{CLK}/ 8 or f_{PLL}/8.
1	1	The filter for comparator 0 is enabled, and sampling proceeds in synchronization with the TAU output (TO01). Note 3

Note 1. If the C1FCK[1:0], C1EPO, and C1EDG bits are changed, a comparator 1 interrupt request and an event signal for the ELC may be generated. Only change these bits after setting the ELSELR21 register of the ELC to 0 (no linking to the output of comparator 1). In addition, clear bit 1 (CMPIF1) in interrupt request flag register 2H (IF2H) to 0. If the C1FCK[1:0] bits are changed from 00B (comparator 1 filter disabled) to a value other than 00B (comparator 1 filter enabled), allow the period of sampling four times to elapse until updating of the filter output, and only then use the comparator 1 interrupt request or the event signal for the ELC.

Note 2. If the C0FCK[1:0], C0EPO, and C0EDG bits are changed, a comparator 0 interrupt request and an event signal for the ELC may be generated. Only change these bits after setting the ELSELR23 register of the ELC to 0 (no linking to the output of comparator 0). In addition, clear bit 0 (CMPIF0) in interrupt request flag register 2H (IF2H) to 0. If the C0FCK[1:0] bits are changed from 00B (comparator 0 filter disabled) to a value other than 00B (comparator 0 filter enabled), allow the period of sampling four times to elapse until updating of the filter output, and only then use the comparator 0 interrupt request or the event signal for the ELC.

Note 3. When bit 5 (PIOR35) of peripheral I/O redirection register 3 (PIOR3) is 0 in a 24-pin to 64-pin product, do not enable output for the TKBO20 pin in the 16-bit timer KB32.
When bit 5 (PIOR35) of peripheral I/O redirection register 3 (PIOR3) is 1 in a 20-pin to 32-pin product, do not enable output for the TKBO11 pin in the 16-bit timer KB31.

22.3.6 Comparator filter control register 1 (COMPFIR1)

The COMPFIR1 register is used to control the digital noise filter. The COMPFIR1 register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 22 - 7 Format of Comparator Filter Control Register 1 (COMPFIR1)

Address: F0345H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
COMPFIR1	C3EDG	C3EPO	C3FCK1	C3FCK0	C2EDG	C2EPO	C2FCK1	C2FCK0

C3EDG	Selecting the edge detection for comparator 3Note 1
0	An interrupt request is generated upon detection of one edge from comparator 3.
1	An interrupt request is generated upon detection of both edges from comparator 3.

C3EPO	Switching the edge sense of comparator 3Note 1
0	An interrupt request is generated on the rising edge of comparator 3.
1	An interrupt request is generated on the falling edge of comparator 3.

C3FCK1	C3FCK0	Selecting the filter for comparator 3Note 1
0	0	The filter for comparator 3 is disabled.
0	1	The filter for comparator 3 is enabled, and sampling proceeds at a cycle of f_{CLK} or f_{PLL}.
1	0	The filter for comparator 3 is enabled, and sampling proceeds at a cycle of f_{CLK} / 8 or f_{PLL}/8.
1	1	The filter for comparator 3 is enabled, and sampling proceeds in synchronization with the TAU output (TO01).Note 3

C2EDG	Selecting the edge detection for comparator 2Note 2
0	An interrupt request is generated upon detection of one edge from comparator 2.
1	An interrupt request is generated upon detection of both edges from comparator 2.

C2EPO	Switching the edge sense of comparator 2Note 2
0	An interrupt request is generated on the rising edge of comparator 2.
1	An interrupt request is generated on the falling edge of comparator 2.

C2FCK1	C2FCK0	Selecting the filter for comparator 2Note 2
0	0	The filter for comparator 2 is disabled.
0	1	The filter for comparator 2 is enabled, and sampling proceeds at a cycle of f_{CLK} or f_{PLL}.
1	0	The filter for comparator 2 is enabled, and sampling proceeds at a cycle of f_{CLK}/ 8 or f_{PLL}/8.
1	1	The filter for comparator 2 is enabled, and sampling proceeds in synchronization with the TAU output (TO01).Note 3

Note 1. If the C3FCK[1:0], C3EPO, and C3EDG bits are changed, a comparator 3 interrupt request and an event signal for the ELC may be generated. Only change these bits after setting the ELSELR26 register of the ELC to 0 (no linking to the output of comparator 3). In addition, clear bit 4 (CMPIF3) in interrupt request flag register 2H (IF2H) to 0. If the C3FCK[1:0] bits are changed from 00B (comparator 3 filter disabled) to a value other than 00B (comparator 3 filter enabled), allow the period of sampling four times to elapse until updating of the filter output, and only then use the comparator 3 interrupt request or the event signal for the ELC.

Note 2. If the C2FCK[1:0], C2EPO, and C2EDG bits are changed, a comparator 2 interrupt request and an event signal for the ELC may be generated. Only change these bits after setting the ELSELR25 register of the ELC to 0 (no linking to the output of comparator 2). In addition, clear bit 3 (CMPIF2) in interrupt request flag register 2H (IF2H) to 0. If the C2FCK[1:0] bits are changed from 00B (comparator 2 filter disabled) to a value other than 00B (comparator 2 filter enabled), allow the period of sampling four times to elapse until updating of the filter output, and only then use the comparator 2 interrupt request or the event signal for the ELC.

Note 3. When bit 5 (PIOR35) of peripheral I/O redirection register 3 (PIOR3) is 0 in a 24-pin to 64-pin product, do not enable output for the TKBO20 pin in the 16-bit timer KB32.
When bit 5 (PIOR35) of peripheral I/O redirection register 3 (PIOR3) is 1 in a 20-pin to 32-pin product, do not enable output for the TKBO11 pin in the 16-bit timer KB31.

22.3.7 Comparator output control register 0 (COMPOCR0)

The COMPOCR0 register is used to control the logical sense of the comparators output, enable or disable the output, and enable or disable the interrupt output. Setting the CiOE bit to 1 (operation enabled) is prohibited in the following cases (i = 0, 1).

- Operation of comparator i is disabled (COMPMDR0.CiENB = 0)
- The output of D/A converter 0 is selected for the negative-side input of comparator 2, and the D/A conversion operation of D/A converter 0 is stopped (DAM0.DACE0 = 0).
- The output of D/A converter 2 is selected for the negative-side input of comparator 2, and the D/A conversion operation of D/A converter 2 is stopped (DAM0.DACE2 = 0) (when DAM2.DACONF = 1).
- The output of D/A converter 0 is selected for the negative-side input of comparator 3, and the D/A conversion operation of D/A converter 0 is stopped (DAM0.DACE0 = 0).
- The output of D/A converter 1 is selected for the negative-side input of comparator 3, and the D/A conversion operation of D/A converter 1 is stopped (DAM0.DACE1 = 0).
- The PGA output is selected for the positive-side input of comparator 3 (CMP3SEL.C3INS = 1) while operation of the programmable gain amplifier is stopped (PGACTL.PGAEN = 0).

The COMPOCR0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 22 - 8 Format of Comparator Output Control Register 0 (COMPOCR0) (1/2)

Address: F0342H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
COMPOCR0	C1OTWMD	C1OP	C1OE	C1IE	C0OTWMD	C0OP	C0OE	C0IE

C1OTWMD	Controlling timer window output mode for comparator 1[Note 1]
0	Normal output mode for comparator 1 (controlled by the C1OE bit)
1	Timer window output mode for comparator 1 (controlled by both TAU output (TO02) and the C1OE bit)

C1OP	Selecting the logical sense of the VCOUT1 output
0	The output of comparator 1 is output to VCOUT1.
1	The inverted output of comparator 1 is output to VCOUT1.

C1OE	Enabling the VCOUT1 pin output[Note 2]
0	Output of the result from comparator 1 is disabled.
1	Output of the result from comparator 1 is enabled.

C1IE	Enabling the interrupt request from comparator 1[Note 3]
0	The interrupt request from comparator 1 is disabled.
1	The interrupt request from comparator 1 is enabled.

Figure 22 - 8 Format of Comparator Output Control Register 0 (COMPOCR0) (2/2)

C0OTWMD	Controlling timer window output mode for comparator 0[Note 1]
0	Normal output mode for comparator 0 (controlled by the C0OE bit)
1	Timer window output mode for comparator 0 (controlled by both TAU output (TO02) and the C0OE bit)

C0OP	Selecting the logical sense of the VCOUT0 output
0	The output of comparator 0 is output to VCOUT0.
1	The inverted output of comparator 0 is output to VCOUT0.

C0OE	Enabling the VCOUT0 pin output[Note 4]
0	Output of the result from comparator 0 is disabled.
1	Output of the result from comparator 0 is enabled.[Notes 3, 5]

C0IE	Enabling the interrupt request from comparator 0[Note 5]
0	The interrupt request from comparator 0 is disabled.
1	The interrupt request from comparator 0 is enabled.

Note 1. When comparators 0 and 1 are to be used in the timer window output mode, be sure to set bit 3 (C0EDG) and bit 7 (C1EDG) of the COMPFIR0 register to 0, respectively. The C0OE bit and C0OTWMD bit or the C1OE bit and C1OTWMD bit cannot be set simultaneously. Only set the C0OE and C1OE bits to 1 after having set the C0OTWMD and C1OTWMD bits, respectively.

Note 2. When the C1OE bit is changed, a comparator 1 interrupt request and an event signal for the ELC may be generated. Only change this bit after setting the ELSELR24 register of the ELC to 0 (no linking to the output of comparator 1). Also, initialize bit 1 (CMPIF1) in interrupt request flag register 2H (IF2H) (no interrupt request) after the change.

Note 3. When the C0OE bit is changed, a comparator 0 interrupt request and an event signal for the ELC may be generated. Only change this bit after setting the ELSELR23 register of the ELC to 0 (no linking to the output of comparator 0). Also, initialize bit 0 (CMPIF0) in interrupt request flag register 2H (IF2H) (no interrupt request) after the change.

Note 4. If the C1IE bit is changed from 0 (interrupt request disabled) to 1 (interrupt request enabled), since bit 1 (CMPIF1) in interrupt request flag register 2H (IF2H) may be set to 1 (interrupt requested), initialize bit 1 (CMPIF1) in interrupt request flag register 2H (IF2H) (no interrupt request) before using the interrupt.

Note 5. If the C0IE bit is changed from 0 (interrupt request disabled) to 1 (interrupt request enabled), since bit 0 (CMPIF0) in interrupt request flag register 2H (IF2H) may be set to 1 (interrupt requested), initialize bit 0 (CMPIF0) in interrupt request flag register 2H (IF2H) (no interrupt request) before using the interrupt.

22.3.8 Comparator output control register 1 (COMPOCR1)

The COMPOCR1 register is used to control the logical sense of the comparators output, enable or disable the output, and enable or disable the interrupt output. Setting the CiOE bit to 1 (operation enabled) is prohibited in the following cases (i = 2, 3).

- Operation of comparator i is disabled (COMPMDR1.CiENB = 0)
- The output of D/A converter 0 is selected for the negative-side input of comparator 2, and the D/A conversion operation of D/A converter 0 is stopped (DAM0.DACE0 = 0).
- The output of D/A converter 2 is selected for the negative-side input of comparator 2, and the D/A conversion operation of D/A converter 2 is stopped (DAM0.DACE2 = 0) (when DAM2.DACONF = 1).
- The output of D/A converter 0 is selected for the negative-side input of comparator 3, and the D/A conversion operation of D/A converter 0 is stopped (DAM0.DACE0 = 0).
- The output of D/A converter 1 is selected for the negative-side input of comparator 3, and the D/A conversion operation of D/A converter 1 is stopped (DAM0.DACE1 = 0).
- The PGA output is selected for the positive-side input of comparator 3 (CMP3SEL.C3INS = 1) while operation of the programmable gain amplifier is stopped (PGACTL.PGAEN = 0).

The COMPOCR1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 22 - 9 Format of Comparator Output Control Register 1 (COMPOCR1) (1/2)

Address: F0346H
After reset: 00H
R/W: R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
COMPOCR1	0	C3OP	C3OE	C3IE	C2OTWMD	C2OP	C2OE	C2IE

C3OP	Selecting the logical sense of the VCOUT3 output[Note 4]
0	The output of comparator 3 is output to VCOUT3.
1	The inverted output of comparator 3 is output to VCOUT3.

C3OE	Enabling the VCOUT3 pin output[Notes 2, 4]
0	Output of the result from comparator 3 is disabled.
1	Output of the result from comparator 3 is enabled.

C3IE	Enabling the interrupt request from comparator 3[Note 3]
0	The interrupt request from comparator 3 is disabled.
1	The interrupt request from comparator 3 is enabled.

C2OTWMD	Controlling timer window output mode for comparator 2[Note 1]
0	Normal output mode for comparator 2 (controlled by the C2OE bit)
1	Timer window output mode for comparator 2 (controlled by both TAU output (TO02) and the C2OE bit)

C2OP	Selecting the logical sense of the VCOUT2 output
0	The output of comparator 2 is output to VCOUT2.
1	The inverted output of comparator 2 is output to VCOUT2.

Figure 22 - 9 Format of Comparator Output Control Register 1 (COMPOCR1) (2/2)

C2OE	Enabling the VCOUT2 pin output[Note 5]
0	Output of the result from comparator 2 is disabled.
1	Output of the result from comparator 2 is enabled.[Notes 3, 6]

C2IE	Enabling the interrupt request from comparator 2[Note 6]
0	The interrupt request from comparator 2 is disabled.
1	The interrupt request from comparator 2 is enabled.

Note 1. When comparator 2 is to be used in the timer window output mode, be sure to set bit 3 (C2EDG) of the COMPFIR1 register to 0. The C2OE bit and C2OTWMD bit cannot be set simultaneously. Only set the C2OE bit to 1 after having set the C2OTWMD bit.

Note 2. When the C3OE bit is changed, a comparator 3 interrupt request and an event signal for the ELC may be generated. Only change this bit after setting the ELSELR26 register of the ELC to 0 (no linking to the output of comparator 3). Also, initialize bit 4 (CMPIF3) in interrupt request flag register 2H (IF2H) (no interrupt request) after the change.

Note 3. When the C2OE bit is changed, a comparator 2 interrupt request and an event signal for the ELC may be generated. Only change this bit after setting the ELSELR25 register of the ELC to 0 (no linking to the output of comparator 2). Also, initialize bit 3 (CMPIF2) in interrupt request flag register 2H (IF2H) (no interrupt request) after the change.

Note 4. The C3OE bit and C3OP bit are controlled so that the result of comparator 3 is input to the PWM option unit to allow forced cutoff of the PWM output.

Note 5. If the C3IE bit is changed from 0 (interrupt request disabled) to 1 (interrupt request enabled), since bit 4 (CMPIF3) in interrupt request flag register 2H (IF2H) may be set to 1 (interrupt requested), initialize bit 4 (CMPIF3) in interrupt request flag register 2H (IF2H) (no interrupt request) before using the interrupt.

Note 6. If the C2IE bit is changed from 0 (interrupt request disabled) to 1 (interrupt request enabled), since bit 3 (CMPIF2) in interrupt request flag register 2H (IF2H) may be set to 1 (interrupt requested), initialize bit 3 (CMPIF2) in interrupt request flag register 2H (IF2H) (no interrupt request) before using the interrupt.

22.3.9 Comparator 0 input signal selection control register (CMP0SEL)

The CMP0SEL register is used to select the input signal on the negative side of comparator 0. The CMP0SEL register should only be rewritten while comparator operation is stopped (C0ENB = 0). The CMP0SEL register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 22 - 10 Format of Comparator 0 Input Signal Selection Control Register (CMP0SEL)

Address: F034AH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
CMP0SEL	0	0	0	0	0	0	C0REFS1	C0REFS0

C0REFS1	C0REFS0	Selecting the input signal on the negative side of comparator 0
0	0	The output of D/A converter 0 is selected.
0	1	The external reference voltage (IVREF0) is selected.
1	0	Setting prohibited
1	1	Setting prohibited

Caution Set the switching interval to at least 3 µs for stable operation when switching the analog input of comparator 0.

22.3.10 Comparator 1 input signal selection control register (CMP1SEL)

The CMP1SEL register is used to select the input signal on the negative side of comparator 1. The CMP1SEL register should only be rewritten while comparator operation is stopped (C1ENB = 0). The CMP1SEL register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 22 - 11 Format of Comparator 1 Input Signal Selection Control Register (CMP1SEL)

Address: F034BH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
CMP1SEL	0	0	0	0	0	0	C1REFS1	C1REFS0

C1REFS1	C1REFS0	Selecting the input signal on the negative side of comparator 1
0	0	The output of D/A converter 0 is selected.
0	1	The output of D/A converter 1 is selected.
1	0	The external reference voltage (IVREF0) is selected.
1	1	Setting prohibited

Caution Set the switching interval to at least 3 µs for stable operation when switching the analog input of comparator 1.

22.3.11 Comparator 2 input signal selection control register (CMP2SEL)

The CMP2SEL register is used to select the input signal on the negative side of comparator 2. The CMP2SEL register should only be rewritten while comparator operation is stopped (C2ENB = 0). The CMP2SEL register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 22 - 12 Format of Comparator 2 Input Signal Selection Control Register (CMP2SEL)

Address: F034CH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
CMP2SEL	0	0	0	0	0	0	C2REFS1	C2REFS0

C2REFS1	C2REFS0	Selecting the input signal on the negative side of comparator 2
0	0	The output of D/A converter 0 is selected.
0	1	The output of D/A converter 2 is selected.**Note**
1	0	The external reference voltage (IVREF0) is selected.
1	1	The external reference voltage (IVREF1) is selected.

Note Cannot be selected when DAM2.DACONF = 0.

Caution **Set the switching interval to at least 3 µs for stable operation when switching the analog input of comparator 2.**

22.3.12 Comparator 3 input signal selection control register (CMP3SEL)

The CMP3SEL register is used to select the input signals on the positive and negative sides of comparator 3. The CMP3SEL register should only be rewritten while comparator operation is stopped (C3ENB = 0). The CMP3SEL register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 22 - 13 Format of Comparator 3 Input Signal Selection Control Register (CMP3SEL)

Address: F034DH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
CMP3SEL	C3INS	0	0	0	0	0	C3REFS1	C3REFS0

C3INS	Selecting the input signal on the positive side of comparator 3
0	An external pin (IVCMP3) is selected.
1	PGAO (PGA output) is selected.

C3REFS1	C3REFS0	Selecting the input signal on the negative side of comparator 3
0	0	The output of D/A converter 0 is selected.
0	1	The output of D/A converter 1 is selected.
1	0	The external reference voltage (IVREF0) is selected.
1	1	The external reference voltage (IVREF1) is selected.

Caution Set the switching interval to at least 3 µs for stable operation when switching the analog input of comparator 3.

22.3.13 Comparator output control register 2 (COMPOCR2)

The COMPOCR2 register is used to select the trigger signal (CMPOUTRX) that is output from a comparator for use with timer RX. The COMPOCR2 register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 22 - 14 Format of Comparator Output Control Register 2 (COMPOCR2)

Address: F034EH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
COMPOCR2	0	0	0	0	0	0	TRXTRIG1	TRXTRIG0

TRXTRIG1	TRXTRIG0	Selecting the trigger signal output from a comparator for use with timer RX
0	0	Interrupt output signal from comparator 0
0	1	Interrupt output signal from comparator 1
1	0	Interrupt output signal from comparator 2
1	1	Interrupt output signal from comparator 3

Caution When switching the interrupt output signal, interrupt requests must be disabled (CnIE = 0) in the comparator beforehand. Interrupt requests must also be disabled (TRXMK = 1) in timer RX.

22.3.14 Registers for controlling the port functions multiplexed with the analog inputs and outputs of the comparators

Set the following registers to control the port functions multiplexed with the analog inputs and outputs of the comparators.

- Port mode registers xx (PMxx)
- Port registers xx (Pxx)
- Port mode control A registers xx (PMCAxx)

For details, see **7.3.1 Port mode registers xx (PMxx)**, **7.3.2 Port registers xx (Pxx)**, and **7.3.7 Port mode control A registers xx (PMCAxx)**.

When the port pins multiplexed with IVCMP0 to IVCMP3, IVREF0, and IVREF1 are to be used for analog inputs of the comparators, set the port mode register xx (PMxx) and port mode control A register xx (PMCAxx) bits corresponding to the target ports to 1.

When the port pins multiplexed with VCOUT0 to VCOUT3 are to be used for digital outputs of the comparators, set the port register xx (Pxx) and port mode register xx (PMxx) bits corresponding to the target ports to 0.

Remark xx = 0, 1, 3, 5, 12, 14
Note that PMCA3 and PMCA5 are not present in the RL78/G24 products.

22.4 Operation

Comparators 0 to 3 are capable of independent operation. The setting methods and operations are the same for each of them. Only comparator 3 is usable with the output of the PGA as an input. The procedure for setting the individual or interlocked operation of the comparators is described in **Table 22 - 2**.

Table 22 - 2 Procedure for Setting the Registers to Control the Comparators

Step	Register	Bit	Value to Be Set
1	PER1	PGACMPEN, DACEN[Note 6]	1 (input clock is supplied)
2	PMCAxx	PMCA120, PMCA00, PMCA01, PMCA147, PMCA22, PMCA16, PMCA12	Select the function of the IVCMPi, IVREFx, and PGAIi pins (set the PMCA120, PMCA00, PMCA01, PMCA147, PMCA22, PMCA16, and PMCA12 bits to 1 (analog input)).[Note 1]
3	PMxx	PM120, PM00, PM01, PM147, PM22, PM16, PM12	Select the function of the IVCMPi, IVREFx, and PGAIi pins (PM120, PM00, PM01, PM147, PM22, PM16, and PM12 bits are set to 1 (input mode)).[Note 1]
4	PGACTL	PGAVG[1:0]	Select the amplification rate.[Notes 3, 4]
5	PGACTL	PVRVS	0 (V_{SS} pin is selected)[Notes 3, 4] 1 (PGAGND pin is selected)[Notes 3, 4]
6	PGACTL	PGAEN	1 (operation is enabled)[Notes 3, 4]
7	Wait for the PGA stabilization time (at least 5 μs).		
8	CMP3SEL (comparator 3)	C3INS (comparator 3)	Select the positive-side input of comparator 3.
9	CMPiSEL	CiREFS[1:0]	Select the negative-side input of comparator i.
10	DACSn		(D/A converted value is set)[Note 4]
11	DAM0	DACEn	(D/A conversion operation is enabled)[Note 4]
12	Wait for the D/A voltage stabilization time (at least 1 s).		
13	Select the function of the IVCMP0 to IVCMP3, IVREF0, IVREF1, and PGAIi[Note 3] pins (input).		
14	COMPMDRx	CiENB	1 (operation is enabled)
15	Wait for the comparator stabilization time (at least 3 μs).		
16	COMPFIRx	CiFCK[1:0]	Select whether the digital filter is to be used or not and the sampling clock.
		CiEPO, CiEDG	Select the edge detection condition for an interrupt request (rising edge, falling edge, or both edges).
17	COMPOCRx	CiOP	Set the output of comparator i (selecting the logical sense).
		CnOTWMD	Make the timer window output mode setting for comparators 0 to 2.
18	COMPOCRx	CiOE	Set the output of comparator i (enabling output).
19	COMPMDRx	CiPOE	Set the output of the VCOUTi pins (enabling output).
20	COMPOCR2	TRXTRIGx	Select the trigger signal output from a comparator and use with timer RX.[Note 5]
21	COMPOCRx	CiIE	Enable or disable the interrupt request output.
22	PR02H, PR12H	CMPPR0i, CMPPR1i	When an interrupt is to be used: Select the interrupt priority level.
23	MK2H	CMPMKi	When an interrupt is to be used: Select the interrupt mask.
24	IF2H	CMPIFi	When an interrupt is to be used: 0 (no interrupt request: for initialization)[Note 2]

Note 1. Setting is required when the IVCMPi or IVREFx pin is to be used as an input signal for comparator i.

(Notes, Caution, and Remark are listed on the next page.)

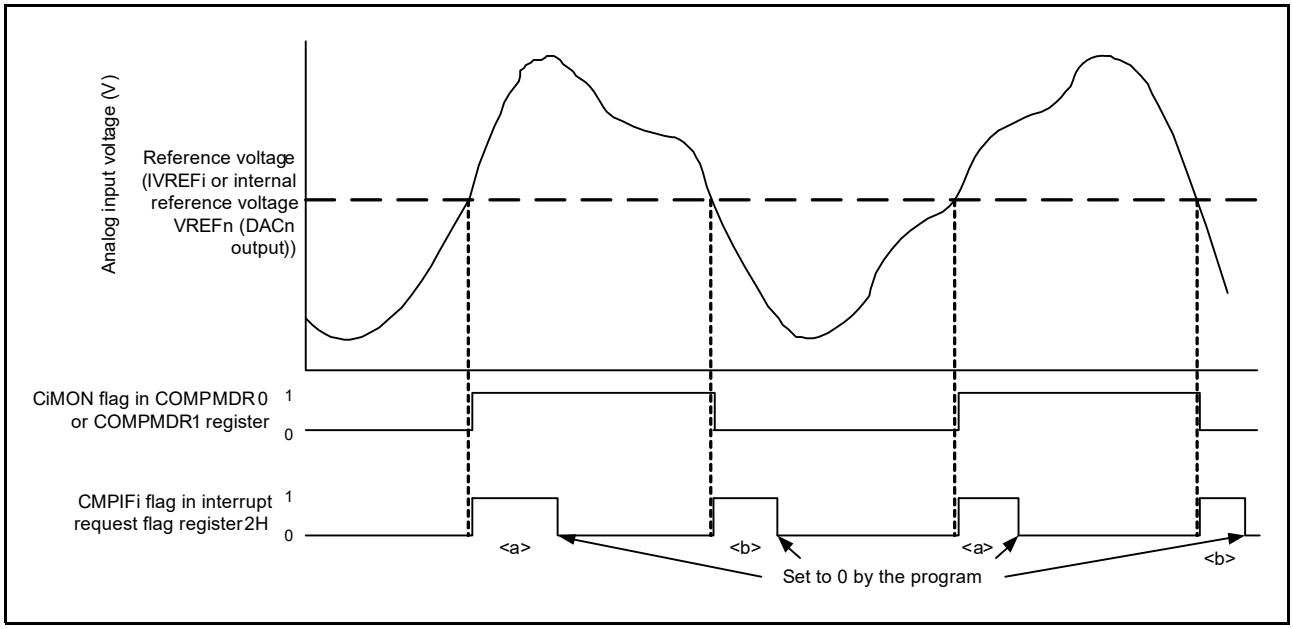
Note 2. After a comparator has been set up, an unwanted interrupt may occur before operation has become stable. Therefore, the interrupt flags have to be initialized.
Note 3. Setting is required for the interlocked operation of comparator 3 and PGA.
Note 4. For details on settings of the D/A converter (DAC) and programmable gain amplifier (PGA), see **Section 21 D/A Converter (DAC)** and **Section 23 Programmable Gain Amplifier (PGA)**.
Note 5. Setting is required when a comparator is to be used to provide a trigger function for use with timer RX.
Note 6. Setting is required when the D/A converter (DAC) is used for the input signal on the negative side of comparator i.

Caution **When initializing all circuits of comparator i, set the PGACMPRES bit in the PRR1 register to 1.**

Remark i = 0 to 3; n = 0 to 2; x = 0, 1

Figure 22 - 15 shows an example of operation for comparator i (i = 0 to 3; in normal output mode). The CiMON flag in the COMPMDR0 or COMPMDR1 register is set to 1 when the voltage of the analog input is higher than the reference voltage, and the CiMON flag is set to 0 when the voltage of the analog input is lower than the reference voltage.
When using an interrupt from comparator i, set the CiIE bit in the COMPOCR0 or COMPOCR1 register to 1 (enabling interrupt requests). If the result of comparison changes at this timing, comparator i will generate an interrupt request. For details on interrupt requests, see **22.4.1 Digital filter for comparator i (i = 0 to 3)**.

Figure 22 - 15 Example of Operation for Comparator i (i = 0-3) (Normal Output Mode)



Caution **The above diagram shows an example of operation when the CiFCK[1:0] bits in the COMPFIR0 or COMPFIR1 register are 00B (filter for comparator i is disabled) and the CiEDG bit is 1 (both-edge detection). (When CiEDG = 0 and CiEPO = 0 (rising edge), the CMPIFi flag only changes as shown by <a>. When CiEDG = 0 and CiEPO = 1 (falling edge), the CMPIFi flag only changes as shown by <b>.)**

Remark n = 0 to 2

22.4.1 Digital filter for comparator i (i = 0 to 3)

Comparator i contains a digital filter. The sampling clock can be selected by the CiFCK[1:0] bits in the COMPFIR0 or COMPFIR1 register. The output signal of comparator i is sampled at 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 22 - 16 shows the configuration of the digital filter for comparator i and edge detection. **Figure 22 - 17** shows the digital filter for comparator i (i = 0 to 3) and an example of interrupt operation.

Figure 22 - 16 Configuration of the Digital Filter of Comparator i and Edge Detection

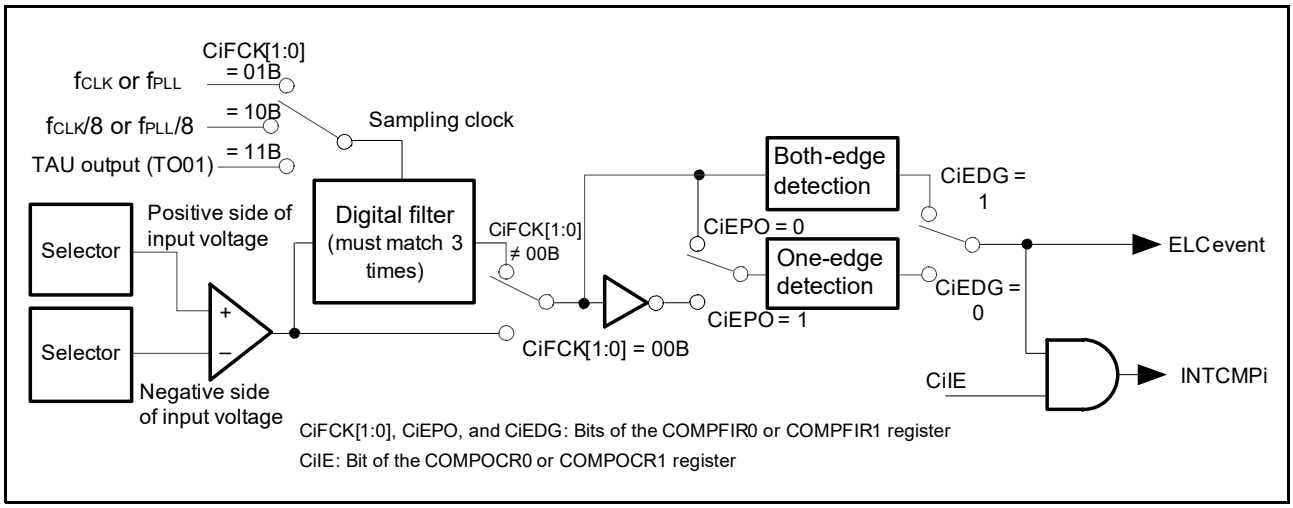
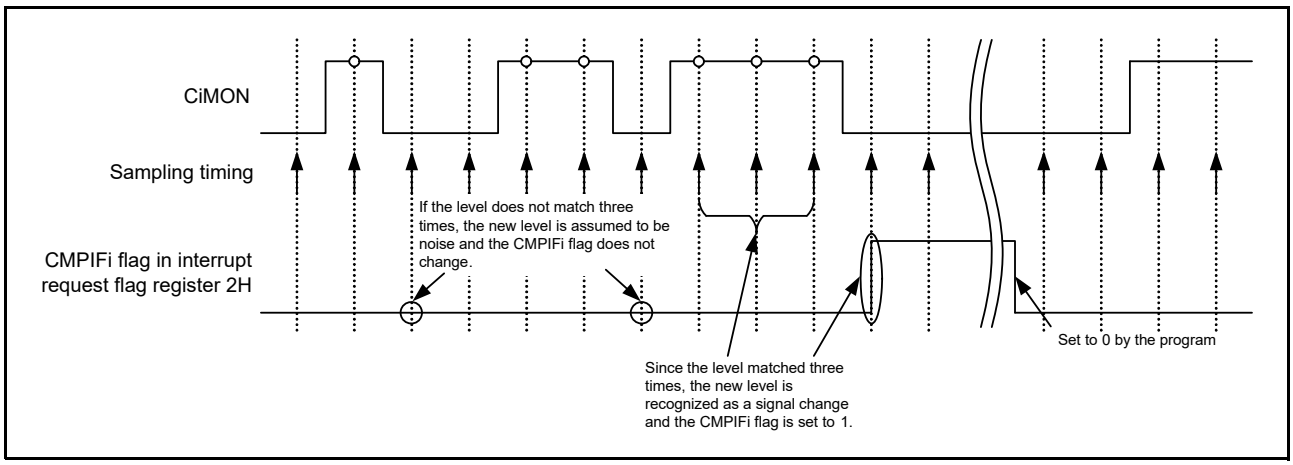


Figure 22 - 17 Digital Filter of Comparator i (i = 0 to 3) and an Example of Interrupt Operation



Caution **The above diagram shows an example of operation in which the C0OTWMD, C1OTWMD, or C2OTWMD bit in the COMPOCR0 or COMPOCR1 register is 0 and the CiFCK[1:0] bits in the COMPFIR0 or COMPFIR1 register are 01B, 10B, or 11B (digital filter is enabled).**

22.4.2 Interrupts of comparator i (i = 0 to 3)

The comparator module generates four interrupt requests: from the comparator 0 interrupt to the comparator 3 interrupt. Each comparator i interrupt uses a priority level specification flag, an interrupt mask flag, an interrupt request flag, and an interrupt vector.

When using a comparator i interrupt, set the CiIE bit in the COMPOCR0 or COMPOCR1 register to 1 (enabling output of the comparator i interrupt request). The condition for generating an interrupt request can be set in the COMPFIR0 or COMPFIR1 register. A comparator output can also be passed through a digital filter. Three different sampling clocks are selectable for the digital filters.

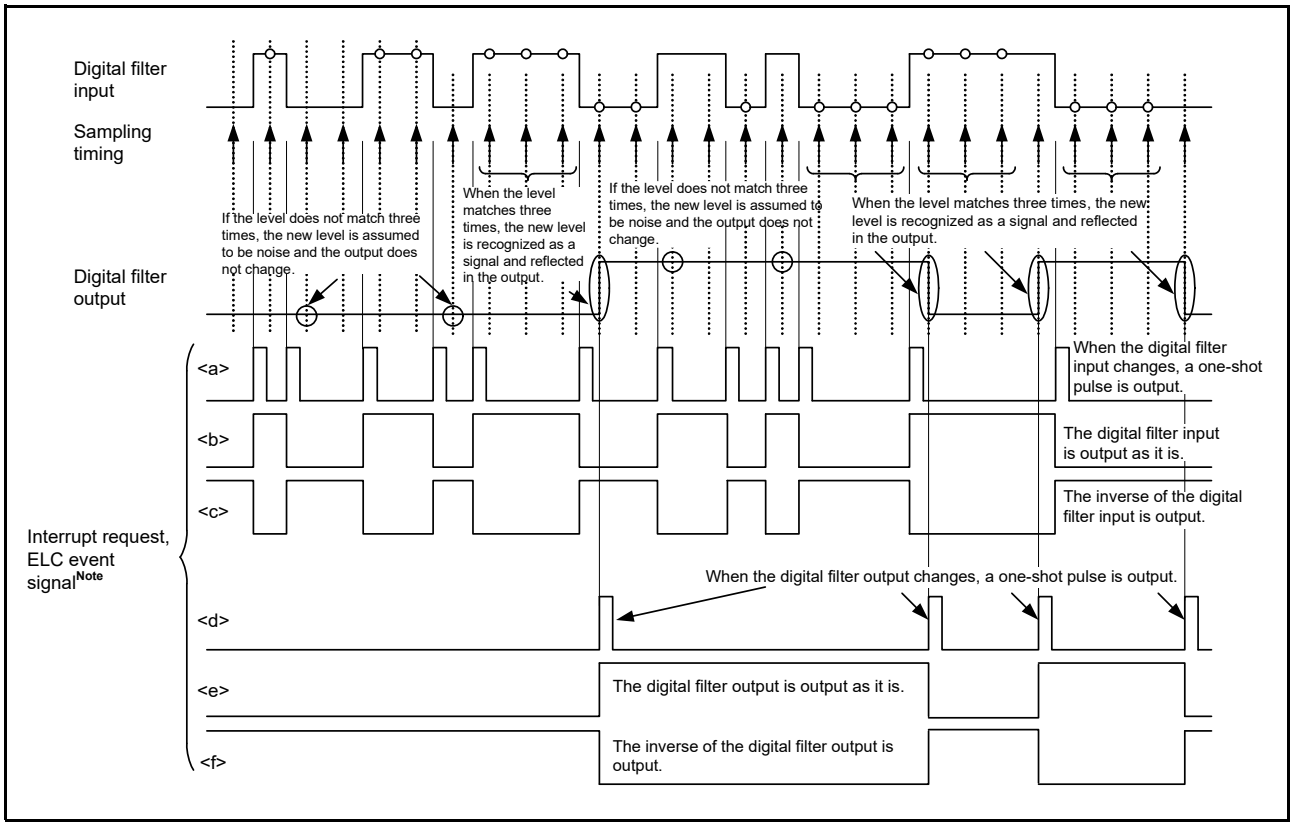
For the correspondence between setting registers and generating interrupt requests, see the following descriptions of the registers.

- **22.3.5 Comparator filter control register 0 (COMPFIR0)**
- **22.3.6 Comparator filter control register 1 (COMPFIR1)**
- **22.3.7 Comparator output control register 0 (COMPOCR0)**
- **22.3.8 Comparator output control register 1 (COMPOCR1)**

22.4.3 Event signal output to the event link controller (ELC)

An event signal for the ELC is generated on detection of the edge for the digital filter output that was set in the COMPFIR0 or COMPFIR1 register, in the same way as for the condition for generating an interrupt request. However, unlike interrupt requests, event signals for the ELC are always output regardless of the setting of the CiIE bit in the COMPOCR0 or COMPOCR1 register. The ELSELR23 to ELSELR26 registers of the ELC can be used to select where to output the event signals and to stop the linking of events.

Figure 22 - 18 Digital Filter and Operations of Generating Interrupt Requests and Output of Event Signals for the ELC



Note When the CiIE bit (i = 0 to 3) is 1, the same waveform is generated for an interrupt request and an event signal for the ELC. When the CiIE bit (i = 0 to 3) is 0, the value is fixed to 0 for only an interrupt request.

The waveforms <a>, <b>, and <c> are shown for an example of operation in which the CiFCK[1:0] bits (i = 0 to 3) in the COMPFIR0 or COMPFIR1 register are 00B (digital filter is disabled).
The waveforms <d>, <e>, and <f> are shown for an example of operation in which the CiFCK[1:0] bits (i = 0 to 3) in the COMPFIR0 or COMPFIR1 register are 01B, 10B, or 11B (digital filter is enabled).
<a> and <d> are the cases where the CiEDG bit is set to 1 (both edges), <b> and <e> are the cases where the CiEDG bit is 0 and the CiEPO bit is 0 (rising edges), and <c> and <f> are the cases where the CiEDG bit is 0 and the CiEPO bit is 1 (falling edges).

22.4.4 Outputs of comparator i (i = 0 to 3)

The result of comparison by a comparator can be output to an external pin. The CiOP and CiOE bits in the COMPOCR0 or COMPOCR1 register can be used to set the logical sense of the output (non-inverted or inverted) and the CiPOE bit in the COMPMDR0 or COMPMDR1 register can be used to enable or disable the output. For the correspondence between register settings and comparator output, see the following descriptions of the registers.

- **22.3.3 Comparator mode setting register 0 (COMPMDR0)**
- **22.3.4 Comparator mode setting register 1 (COMPMDR1)**
- **22.3.7 Comparator output control register 0 (COMPOCR0)**
- **22.3.8 Comparator output control register 1 (COMPOCR1)**

To output the result of comparison by a comparator to the VCOUTi output pin, use the following procedure to set up the pin. (Note that the pins are set as inputs following a reset.)

<1> Set the registers of the comparators (see **Table 22 - 2 Procedure for Setting the Registers to Control the Comparators**).
<2> Set the VCOUTi output for the comparator (select the logical sense and enable the output in the COMPOCR0 or COMPOCR1 register).
<3> Set the COMPMDR0 or COMPMDR1 register to enable pin output.
<4> Set the port mode control A register bit corresponding to the VCOUTi output pin to 0.
<5> Set the port register bit corresponding to the VCOUTi output pin to 0.
<6> Set the port mode register bit corresponding to the VCOUTi output pin to output (start output from the pin).

22.4.5 Stopping or supplying the clock of the comparators

To stop the clock of a comparator by setting peripheral enable register 1 (PER1), use the following procedure.

<1> Set the CiENB bit in the COMPMDR0 or COMPMDR1 register to 0 to stop a comparator.
<2> Set the CMPIFi flag in interrupt request flag register 2H (IF2H) to 0 to clear any unwanted interrupts before stopping a comparator.
<3> Set the PGACMPEN bit in the PER1 register to 0.

When the clock is stopped by the setting of the PER1 register, all the internal registers in the comparators are initialized. To use the comparators again, follow the procedure in **Table 22 - 2** to set the registers.

Caution **When DTC activation is enabled in either of the following states, a DTC transfer is started and an interrupt request is generated on completion of the transfer. Therefore, enable DTC activation after confirming the comparator i monitoring flag (CiMON) as required.**

- **State 1: All of the following conditions have been satisfied.**
  - **The setting is made to generate an interrupt request on one-edge detection in comparator i (CiEDG = 0).**
  - **The setting is made to generate an interrupt request on the rising edge of comparator i (CiEPO = 0).**
  - **The input voltage on the positive side is greater than the input voltage on the negative side.**
- **State 2: All of the following conditions have been satisfied.**
  - **The setting is made to generate an interrupt request on one-edge detection in comparator i (CiEDG = 0).**
  - **The setting is made to generate an interrupt request on the falling edge of comparator i (CiEPO = 1).**
  - **The input voltage on the positive side is lower than the input voltage on the negative side.**

Remark i = 0 to 3; n = 0 to 2

22.4.6 Notes on using interlocking of the comparators with the 16-bit timers KB30, KB31, and KB32

The comparators can be used to output triggers for the functions interlocked with 16-bit timers KB30, KB31, and KB32 (forced output stop 1 and 2, and restart functions) as well as to request interrupts.
When using the functions interlocked with 16-bit timers KB30, KB31, and KB32, use comparator filter control registers 0 and 1 (COMPFIR0 and COMPFIR1) to specify the valid edges on the comparator signals.
Note that the width of the active signal required to make a given function operate differs between functions. Set the registers with reference to **Table 22 - 3** and **Figure 22 - 19** and configure external circuits to ensure the required active signal width.

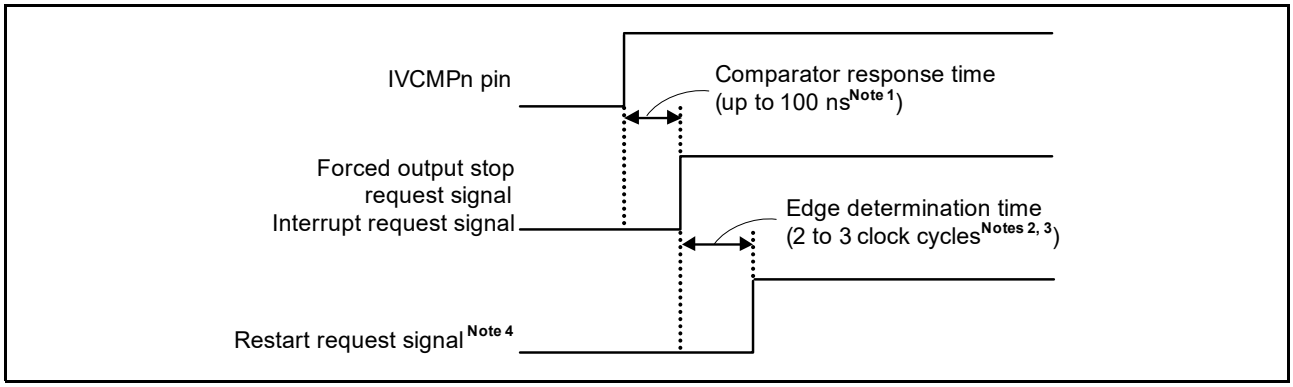
Table 22 - 3 Relationship between Functions of Comparator n, Register Settings, and Active Signal Width

Function	Setting of Bits in the Registers for Setting the Valid Edges	Active Signal Width Required for Operation of the Function		
		Comparator n Interrupt	Forced Output Stop of 16-bit Timers KB30, KB31, and KB32	Restart of 16-bit Timers KB30, KB31, and KB32
External interrupt (can drive release from STOP mode**Note 1**)	CnEDG, CnEPO	Up to 100 ns**Note 1** 2 to 3 clock cycles**Notes 2, 3**	—	—
Forced output stop of 16-bit timers KB30, KB31, and KB32	— **Note 4**	Up to 100 ns**Note 1** 2 to 3 clock cycles**Notes 2, 5**	Up to 100 ns**Notes 1, 5**	—
Restart of 16-bit timers KB30, KB31, and KB32	CnEDG, CnEPO	Up to 100 ns**Note 1** 2 to 3 clock cycles**Notes 2, 3**	—	Up to 100 ns**Note 1** 2 to 3 clock cycles**Notes 2, 3**

Note 1. These are the values when the digital filter is enabled (CnFCK[1:0] = 00B) by comparator filter control register n (COMPFIRn). When the digital filter setting is not 00B, the specified elimination width is added to these values.
Note 2. The clock is f_{CLK} or f_{PLL} when bit 0 (DSCON) of the PLL control register (DSCCTL) is 1.
Note 3. After the timer restart request signal has been received, it takes 1 clock cycle until the timer restart function operates and it takes an additional delay time of 10 to 40 ns until the state of the output pin changes.
Note 4. Forced output stop signals 1 and 2 are active high.
Note 5. After forced output stop signals 1 and 2 have been triggered, it takes a delay time of 10 to 40 ns until the state of the output pin changes.

Remark n = 0 to 3

Figure 22 - 19 Timing for Generating the Forced Output Stop Request Signal and Timer Restart Request Signal by Comparator n



Note 1. This is the value when the digital filter is enabled (CnFCK[1:0] = 00B) by comparator filter control register n (COMPFIRn). When the digital filter setting is not 00B, the specified elimination width is added to this value.

Note 2. The clock is f_{CLK} or f_{PLL} when bit 0 (DSCON) of the PLL control register (DSCCTL) is 1.

Note 3. After the timer restart request signal has been received, it takes 1 clock cycle until the timer restart function operates and it takes an additional delay time of 10 to 40 ns until the state of the output pin changes.

Note 4. After forced output stop signals 1 and 2 have been triggered, it takes a delay time of 10 to 40 ns until the state of the output pin changes.

Remark n = 0 to 3

For interlocking of external interrupts INTPm with 16-bit timers KB30, KB31, and KB32, refer to **15.10 Notes on Using Interlocking of External Interrupts INTPm with 16-bit Timers KB30, KB31, and KB32**.

Section 23 Programmable Gain Amplifier (PGA)

The RL78/G24 incorporates a programmable gain amplifier.
To use PGAGND as the ground of the feedback resistor for the programmable gain amplifier, apply the same voltage to it as that on Vss.

Item	20-pin	24- to 64-pin
Analog input channels	4 ch (PGAI0 to PGAI2, PGAI4)	5 ch (PGAI0 to PGAI4)
Ground of the feedback resistor for the programmable gain amplifier	Vss/PGAGND	Vss/PGAGND
Output pin for voltage output by the programmable gain amplifier	PGAO	PGAO

23.1 Functions of Programmable Gain Amplifier

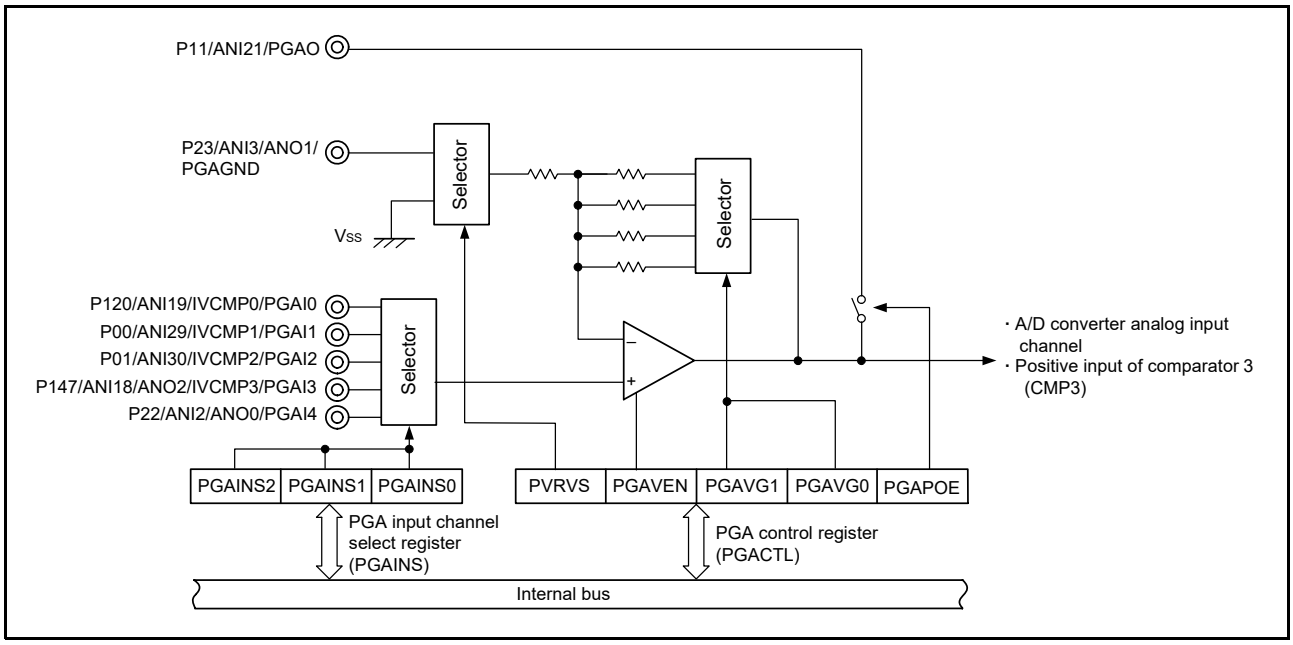
The programmable gain amplifier is provided with the following functions.

- Input to the programmable gain amplifier is selectable from among the PGAI0 to PGAI4 pins.
- Gain is selectable from among four values.
- The output signal of a programmable gain amplifier can be set as the analog input of the A/D converter and the positive input signal of comparator 3 (CMP3).

23.2 Configuration of Programmable Gain Amplifier

The programmable gain amplifier includes the following hardware.

Figure 23 - 1 Block Diagram of Programmable Gain Amplifier



Remark The pins in the above diagram are those of the 64-pin products.

23.3 Registers to Control Programmable Gain Amplifier

The following registers are used to control the programmable gain amplifier.

- Peripheral enable register 1 (PER1)
- Peripheral reset control register 1 (PRR1)
- PGA control register (PGACTL)
- PGA input channel select register (PGAINS)
- Port mode registers xx (PMxx) (xx = 0 to 2, 12, 14)
- Port mode control A registers xx (PMCAxx) (xx = 0 to 2, 12, 14)

23.3.1 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. If the programmable gain amplifier and comparator are to be used, be sure to set bit 5 (PGACMPEN) of this register to 1. The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 23 - 2 Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	<3>	2	1	<0>
PER1	DACEN	0	PGACMPEN	TML32EN	DTCEN	0	0	DALIEN

PGACMPEN[Note]	Control of supply of an input clock to the comparator or programmable gain amplifier
0	Stops supply of an input clock. • The SFRs used by the comparator or programmable gain amplifier cannot be written.
1	Enables supply of an input clock. • The SFRs used by the comparator or programmable gain amplifier can be read and written.

Note If bits C3FCK[1:0], C3EPO, and C3EDG of the COMPFIR1 register in the comparator are changed, a comparator detection 3 interrupt request and an event signal to the ELC may be generated. Change these bits only after setting the ELSELR26 register for the ELC to 0 (not linked to comparator 3 output). In addition, clear bit 4 (CMPIF3) in interrupt request flag register 2H (IF2H) to 0.
If bits C3FCK[1:0] are changed from 00B (no comparator 3 filter) to a value other than 00B (comparator 3 filter enabled), allow the time for sampling four times to elapse until the filter output is updated, and then use the comparator 3 interrupt request or the event signal to the ELC.

Caution 1. When setting the comparator or programmable gain amplifier, be sure to set the PGACMPEN bit to 1 first.
If PGACMPEN = 0, writing to the registers which control the comparator or programmable gain amplifier is ignored, and all read values are default values (except for port mode registers xx (PMxx) and port registers xx (Pxx)).

Caution 2. Be sure to set bits 6, 2, and 1 to 0.

23.3.2 Peripheral reset control register 1 (PRR1)

The PRR1 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. To place the programmable gain amplifier in the reset state, set bit 5 (PGACMPRES) of this register to 1. The PRR1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 23 - 3 Format of Peripheral Reset Control Register 1 (PRR1)

Address: F00FBH
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	3	2	1	<0>
PRR1	DACRES	0	PGACMPRES	TML32RES	0	0	0	DALIRES

PGACMPRES	Control resetting of the comparator and programmable gain amplifier
0	The comparator and programmable gain amplifier are released from the reset state.
1	The comparator and programmable gain amplifier are in the reset state.

Caution Be sure to set bits 6 and 3 to 1 to 0.

23.3.3 PGA control register (PGACTL)

The PGACLT register is used to enable or stop operation of the programmable gain amplifier and set its gain. The PGACTL register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 23 - 4 Format of PGA Control Register (PGACTL)

Address: F0347H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	5	4	3	2	1	0
PGACTL	PGAEN	PGAPOE	—	—	PVRVS	—	PGAVG1	PGAVG0

PGAEN	Control of programmable gain amplifier operation
0	Stops programmable gain amplifier operation.
1	Enables programmable gain amplifier operation.

PGAPOE	Control over output pin for voltage output by the programmable gain amplifier
0	The output voltage from the programmable gain amplifier is not output from the PGAO pin.
1	The output voltage from the programmable gain amplifier is output from the PGAO pin.

PVRVS	Selection of the ground pin of the feedback resistor for the programmable gain amplifier
0	Selects Vss.
1	Selects PGAGND.

PGAVG1	PGAVG0	Programmable gain amplifier gain selection
0	0	×4
0	1	×8
1	0	×16
1	1	×32

Caution 1. Be sure to set bits 5, 4, and 2 to 0.
Caution 2. Rewrite the bits of the PGACTL register other than PGAEN while operation of the programmable gain amplifier is stopped (PGAEN = 0).
Caution 3. For the programmable gain amplifier, a PGA operation stabilization wait time of 5 μs when the gain is 4 or 8, or of 10 μs when the gain is 16 or 32 is required after setting the PGAEN bit to 1.
Caution 4. Current cannot be drawn from the PGAO pin. The gain of the voltage from the pin thus cannot be externally adjusted.

23.3.4 PGA input channel select register (PGAINS)

The PGAINS register is used to select the input channel of the programmable gain amplifier. The PGAINS register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 23 - 5 Format of PGA Input Channel Select Register (PGAINS)

Address: F0348H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PGAINS	0	0	0	0	0	PGAINS2	PGAINS1	PGAINS0

PGAINS2	PGAINS1	PGAINS0	Analog input channel of the programmable gain amplifier
0	0	0	The PGAI0 pin is selected.
0	0	1	The PGAI1 pin is selected.
0	1	0	The PGAI2 pin is selected.
0	1	1	The PGAI3 pin is selected.
1	0	0	The PGAI4 pin is selected.
Other than above			Setting prohibited

Caution 1. Be sure to set bits 7 to 3 to 0.

Caution 2. When the setting of the PGAINS register is to be changed while the programmable gain amplifier is operating (PGACTL.PGAEN = 1), adjust the timing of starting the A/D conversion, and control the comparator 3 output (e.g. C3IE = 0, C3POE = 0) as required to suit the setting of the slew rate of the PGA.

23.3.5 Registers for controlling the port functions multiplexed with the inputs and outputs of the programmable gain amplifier

Set the following registers to control the port functions multiplexed with the analog inputs and outputs of the programmable gain amplifier.

- Port mode registers xx (PMxx)
- Port mode control A registers xx (PMCAxx)

For details, see **7.3.1 Port mode registers xx (PMxx)** and **7.3.7 Port mode control A registers xx (PMCAxx)**.

When the pins multiplexed with PGAI0 to PGAI4 and PGAGND are to be used for analog inputs of the programmable gain amplifier, set the bits corresponding to the selected multiplexed port pin in port mode register xx (PMxx) and port mode control A register xx (PMCAxx) to 1.
When the pin multiplexed with PGAO is to be used for analog output of the programmable gain amplifier, set the bits corresponding to the selected multiplexed port pin in port mode register xx (PMxx) and port mode control A register xx (PMCAxx) to 1.

Remark xx = 0 to 2, 12, 14

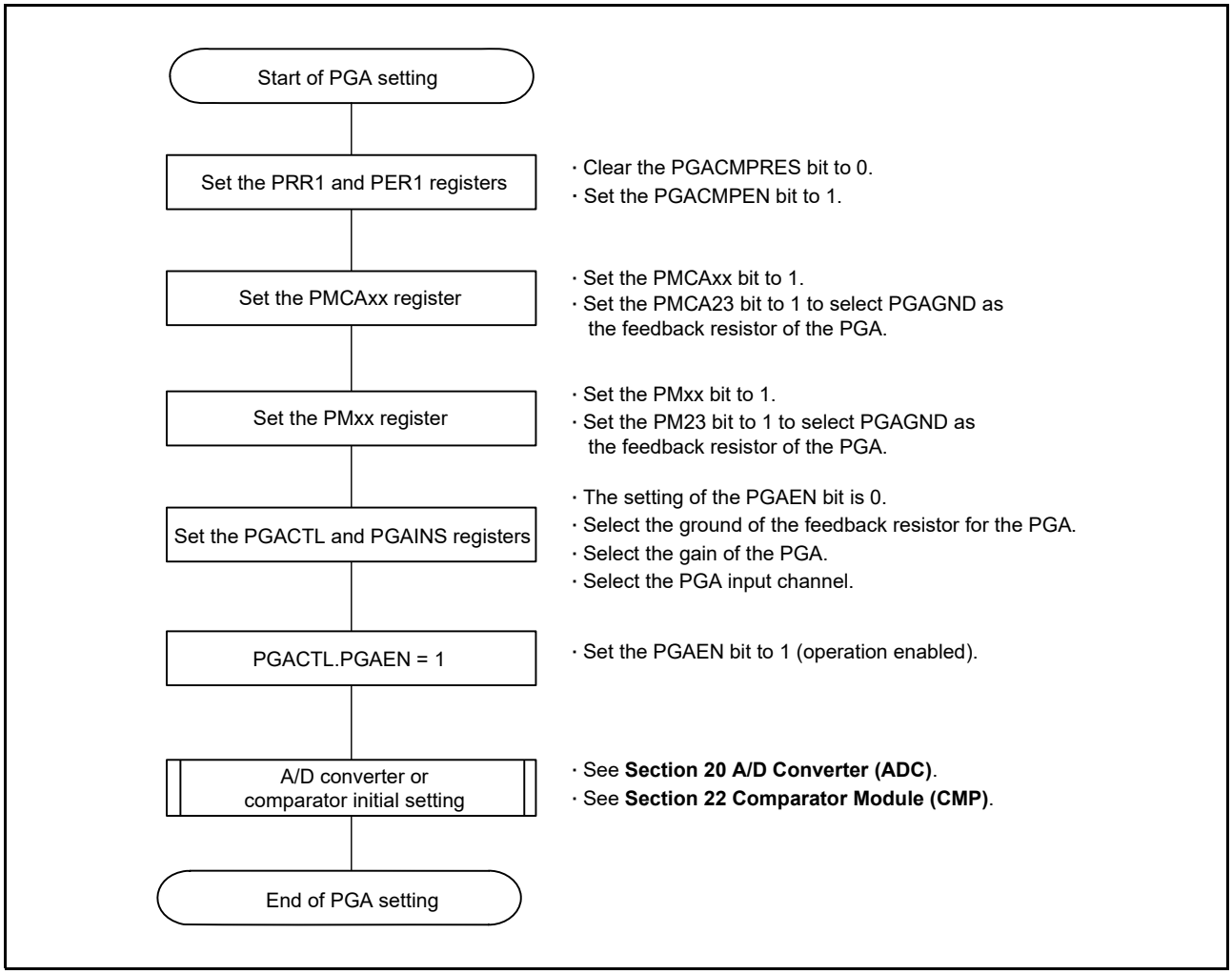
23.4 Operation of Programmable Gain Amplifier

The analog voltage inputs from the PGAI0 to PGAI4 pins are amplified within the microcontroller. The gain can be selected from four types (×4, ×8, ×16, and ×32).
The amplified voltage can be used as an analog input of the A/D converter and the positive input signal for comparator 3 (CMP3).
The procedure for starting operation of the programmable gain amplifier is described below.

<1> Set the PGACMPEN bit of the PER1 register to supply an input clock to the programmable gain amplifier.
<2> Use the PMCA0, PMCA2, PMCA12, and PMCA14 registers to set the pins (PGAI0 to PGAI4) to be used in the programmable gain amplifier as analog inputs.
<3> Use the PM0, PM2, PM12, and PM14 registers to set the pins (PGAI0 to PGAI4) to be used in the programmable gain amplifier to input mode.
<4> Use the PGAVG[1:0] bits in the PGACTL register to select the gain (×4, ×8, ×16, and ×32).
<5> Select the pin input to the programmable gain amplifier by the setting of the PGAINS[2:0] bits in the PGAINS register.
<6> To use a programmable gain amplifier output as the positive input signal for comparator 3, set the C3INS bit of the CMP3SEL register to select the input from the PGA.
<7> Set the PGAEN bit to 1 and enable operation of the programmable gain amplifier.

23.4.1 Setting procedure for starting the programmable gain amplifier

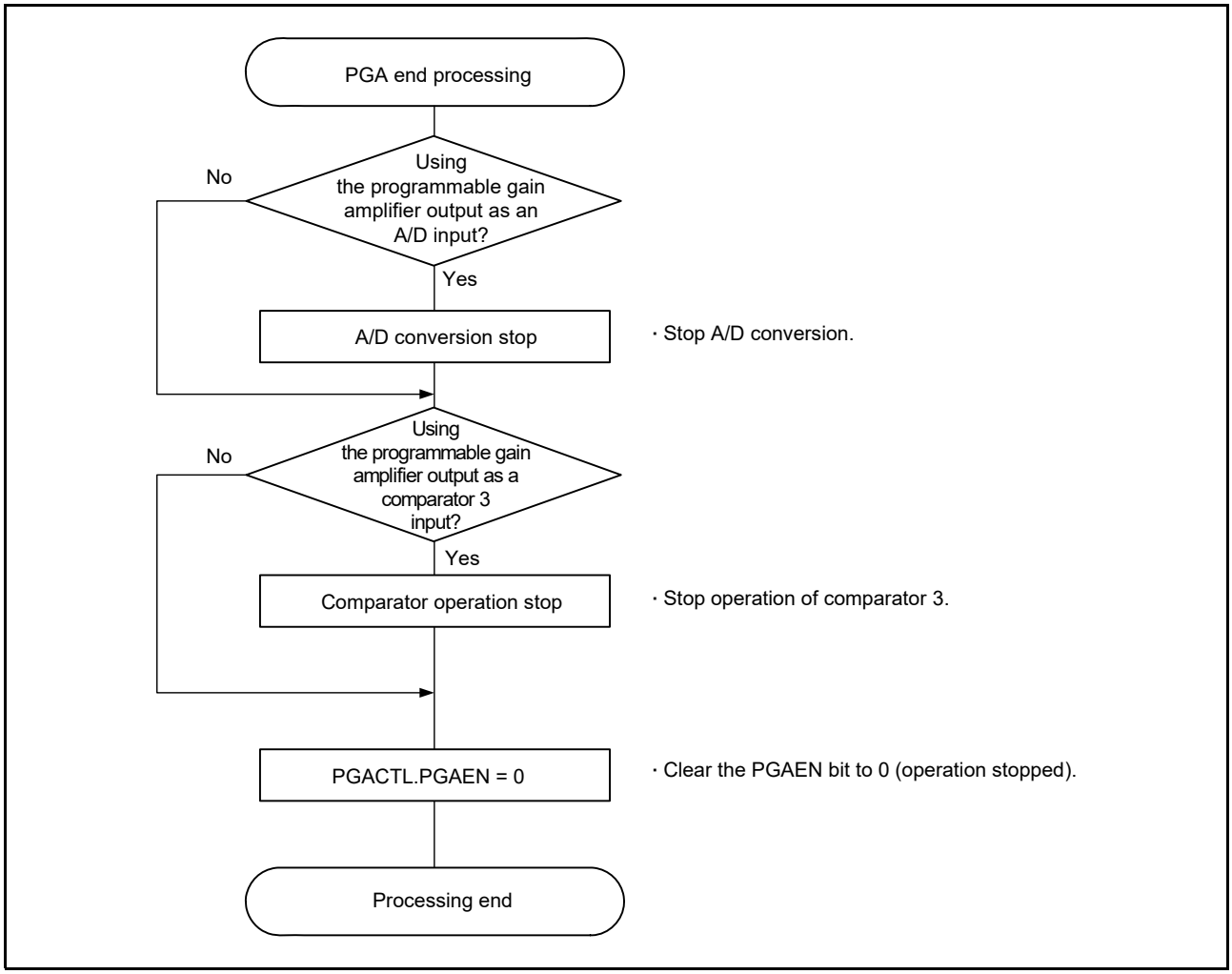
Figure 23 - 6 Operation Setting Flow of Programmable Gain Amplifier



Caution After setting the PGAEN bit to 1, start A/D conversion after a PGA operation stabilization wait time of 5 µs when the gain is 4 or 8, or of 10 µs when the gain is 16 or 32.

23.4.2 Setting procedure for stopping the programmable gain amplifier

Figure 23 - 7 Operation Stopping Flow of Programmable Gain Amplifier



Caution 1. When restarting the programmable gain amplifier and A/D converter or comparator, start the function after 5 µs when the gain is 4 or 8, or 10 µs when the gain is 16 or 32 have elapsed as the PGA operation stabilization wait time after having set the PGAEN bit to 1.

Caution 2. The A/D conversion pins and comparators to which the programmable gain amplifier output is not connected can be used even when operation of the programmable gain amplifier is stopped.

Section 24 Serial Array Unit (SAU)

This product has two serial array units. Serial array units 0 and 1 respectively have four and two serial channels. All channels can handle 3-wire serial (simplified SPI or CSI[Note]), UART, and simplified I2C communications. See the tables below for assignment of the function in each channel.

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.

<20-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I2C
0	0	—	—	—
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

<24- and 25-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I2C
0	0	—	UART2 (supporting LIN-bus)[Note]	—
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

<30- and 32-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I2C
0	0	CSI00 (supporting slave select input)	UART0 (supporting LIN-bus)	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

<40- and 44-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00 (supporting slave select input)	UART0 (supporting LIN-bus)	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	CSI21		IIC21

<48- and 52-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00 (supporting slave select input)	UART0 (supporting LIN-bus)	IIC00
	1	CSI01		IIC01
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	CSI21		IIC21

<64-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00 (supporting slave select input)	UART0 (supporting LIN-bus)	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	CSI21		IIC21

Note This function can be used when the setting of bit 1 (PIOR01) of peripheral I/O redirection register 0 (PIOR0) is 1.

When UART0 is used for channels 0 and 1 of unit 0, CSI00 and CSI01 cannot be used, but CSI10, UART1, or IIC10 in channels 2 and 3 can be used.

Caution Most of the following descriptions in this section use the units and channels of the 64-pin products as an example.

24.1 Functions of Serial Array Unit

Each serial interface supported by the RL78/G24 has the following features.

24.1.1 Simplified SPIs (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21)

Data is transmitted or received in synchronization with the serial clock (SCK) output from the master.
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 **24.5 Operation of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) 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_{CLK}/2 (CSI00 only)
  Max. f_{CLK}/4
  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. In the SNOOZE mode, data can be received without CPU processing upon detecting SCK input in the STOP mode.

Note Set up the transfer rate within a range satisfying the SCK cycle time (t_{KCY}). For details, see **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** or **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.

24.1.2 UART (UART0 to UART2)

The UART functions are for asynchronous communications and each UART uses two lines: serial data transmission (TxD) and serial data reception (RxD). These two lines are used to send and receive data asynchronously (using an internal baud rate) to and from the other-party device per single data frame, each consisting of a start bit, data, parity bit, and stop bit. Full-duplex UART communications can be realized by using two channels, one for use in transmission (even-numbered channel) and the other for use in reception (odd-numbered channel). The LIN bus can also be supported by the combination of a UART, a timer array unit, and an external interrupt (INTP0).
For details about the settings, see **24.6 Operation of UART (UART0 to UART2) Communication**.

[Data transmission/reception]

- Data length of 7, 8, or 9 bits**Note**
- MSB/LSB first selectable
- 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 reception supports the SNOOZE mode. In the SNOOZE mode, data can be received without CPU processing upon detecting RxD input in the STOP mode. The SNOOZE mode is only available in UART0, which support the reception baud rate adjustment function.

- 30- to 64-pin products: UART0
- 80- to 128-pin products: UART0, UART2

UART0 (channels 0 and 1 of unit 0) supports the LIN-bus interface.

[LIN-bus functions]

- Wakeup signal detection
- Break field (BF) detection
- Sync field measurement, baud rate calculation

} Using the external interrupt (INTP0) and timer array unit

Note UART0 only supports the 9-bit data length.

24.1.3 Simplified I2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21)

Simplified I2C is a clock synchronous communications function for use in communicating with two or more devices over two lines: serial clock (SCL) and serial data (SDA). This simplified I2C is designed for single communications with devices such as an EEPROM, flash memory, or an A/D converter, and therefore, it only functions as a master. Start and stop conditions, along with control register operations, should be handled by software in compliance with the AC specifications.

For details about the settings, see **24.8 Operation of Simplified I2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) 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]

- ACK error, or overrun error

* [Functions not supported by simplified I2C]

- Slave transmission, slave reception
- Arbitration loss detection
- 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 **24.8.3**, **2. Processing flow** for details.

Remark 1. To use an I2C bus of full function, see **Section 25 Serial Interface IICA (IICA)**.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

24.2 Configuration of Serial Array Unit

The serial array unit includes the following registers, and input and output pins.

Table 24 - 1 Configuration of Serial Array Unit

Item	Configuration
Shift register	8 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, SCK01, SCK10, SCK11, SCK20, and SCK21 pins (for simplified SPI), SCL00, SCL01, SCL10, SCL11, SCL20, and SCL21 pins (for simplified I^2C)
Serial data input	SI00, SI01, SI10, SI11, SI20, and SI21 pins (for simplified SPI), RxD0 (for UART supporting LIN-bus), RxD1, and RxD2 pins
Serial data output	SO00, SO01, SO10, SO11, SO20, SO21, SO30, SO31 pins (for simplified SPI), TxD0 (for UART supporting LIN-bus), TxD1, and TxD2 pins
Serial data I/O	SDA00, SDA01, SDA10, SDA11, SDA20, and SDA21 pins (for simplified I^2C)
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Peripheral reset control register 0 (PRR0) • Serial clock select registers m (SPSm) (m = 0, 1) • Serial channel enable status registers m (SEm) (m = 0, 1) • Serial channel start registers m (SSm) (m = 0, 1) • Serial channel stop registers m (STm) (m = 0, 1) • Serial output enable registers m (SOEm) (m = 0, 1) • Serial output registers m (SOm) (m = 0, 1) • Serial output level registers m (SOLm) (m = 0, 1) • Serial standby control register m (SSCm) (m = 0) • Input switch control register (ISC) • Noise filter enable register 0 (NFEN0)
	<Registers of each channel> • Serial data registers mn (SDRmn) (mn = 00 to 03, 10, 11) • Serial mode registers mn (SMRmn) (mn = 00 to 03, 10, 11) • Serial communication operation setting registers mn (SCRmn) (mn = 00 to 03, 10, 11) • Serial status registers mn (SSRmn) (mn = 00 to 03, 10, 11) • Serial flag clear trigger registers mn (SIRmn) (mn = 00 to 03, 10, 11)
	• Port mode registers xx (PMxx) (xx = 0, 1, 3, 5 to 7) • Port registers xx (Pxx) (xx = 0, 1, 3, 5, 7) • Port input mode registers xx (PIMxx) (xx = 0, 1, 3, 5, 7) • Port output mode registers xx (POMxx) (xx = 0, 1, 3, 5, 7) • Port mode control A registers xx (PMCAxx) (xx = 0, 1) • UART loopback select register (ULBS)

Note 1. The number of bits used as the shift register and buffer register differs depending on the unit and channel.
- 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, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)
q: UART number (q = 0 to 2), r: IIC number (r = 00, 01, 10, 11, 20, 21)

Figure 24 - 1 shows the block diagram of serial array unit 0 (SAU0).

Figure 24 - 1 Block Diagram of Serial Array Unit 0 (SAU0)

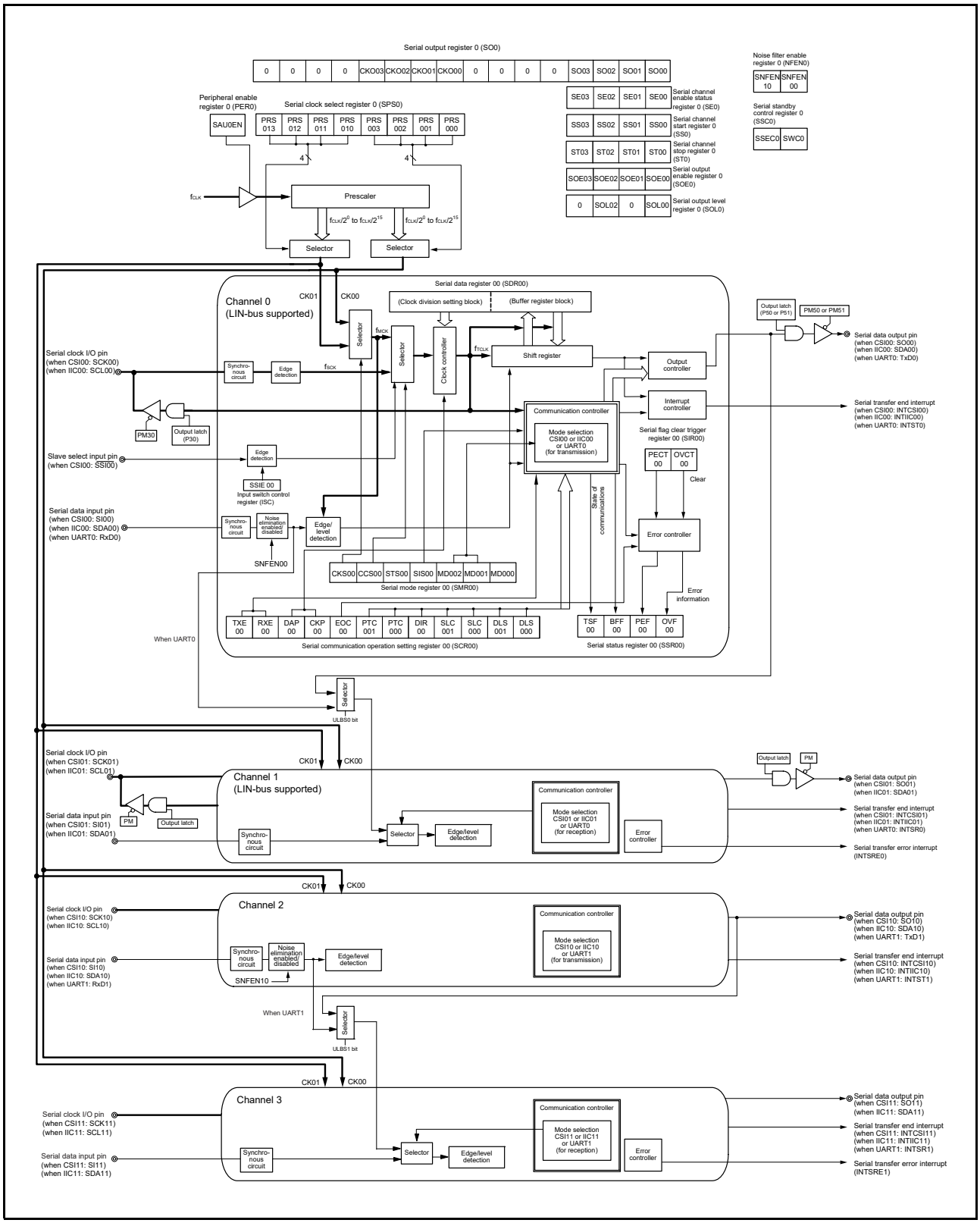
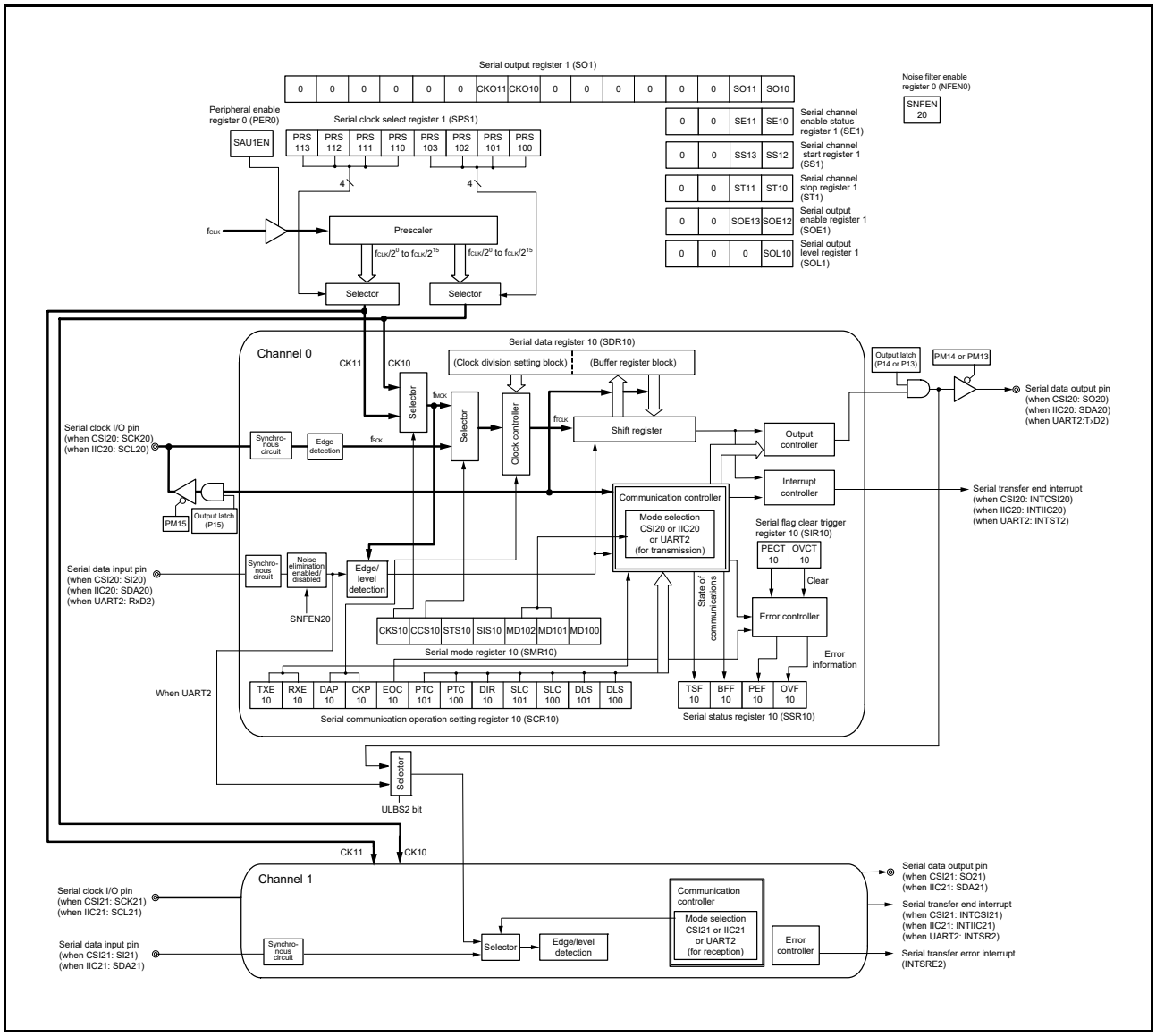


Figure 24 - 2 shows the block diagram of serial array unit 1 (SAU1).

Figure 24 - 2 Block Diagram of Serial Array Unit 1 (SAU1)



24.2.1 Shift register

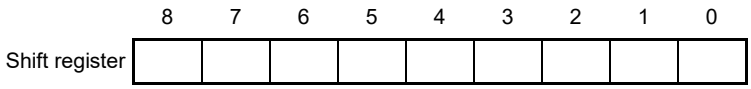
This is a 9-bit register that converts parallel data into serial data or vice versa.

In case of the UART communication of nine bits of data, nine bits (bits 0 to 8) are used[Note].

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 or 9 bits of serial data register mn (SDRmn).



Note UART0 only supports the 9-bit data length.

24.2.2 Lower 8 or 9 bits of the serial data register mn (SDRmn)

The SDRmn 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 or 9 bits. When data is to be transmitted, set transmit data to be transferred to the shift register to the lower 8 or 9 bits.

The data stored in the lower 8 or 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 the SDRmn register)
- 8-bit data length (stored in bits 0 to 7 of the SDRmn register)
- 9-bit data length (stored in bits 0 to 8 of the SDRmn register)[Note 1]

The SDRmn register can be read or written in 16-bit units.

The lower 8 or 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)

The value of each SDRmn register following a reset is 0000H.

Note 1. UART0 only supports the 9-bit data length.

Note 2. When operation is stopped (SEmn = 0), do not rewrite the SDRmn[7:0] bits by an 8-bit memory manipulation instruction (the SDRmn[15:9] bits 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.

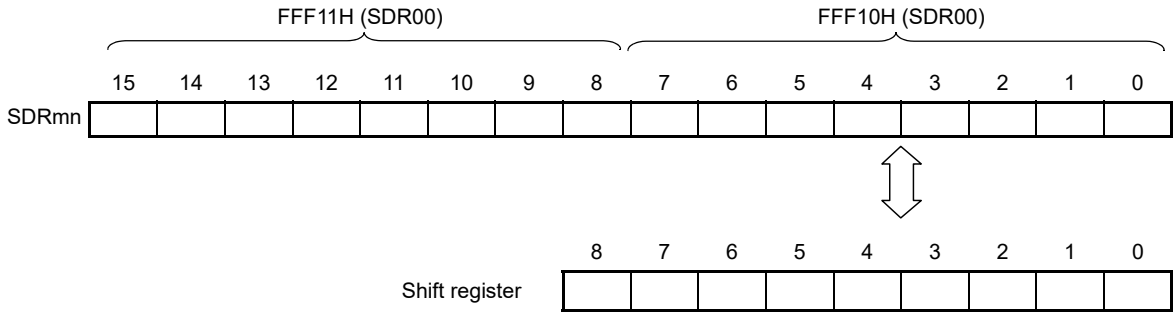
Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)
q: UART number (q = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21)

Figure 24 - 3 Format of Serial Data Register mn (SDRmn) (mn = 00, 01, 10, 11)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01)
FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11)

After reset: 0000H

R/W: R/W



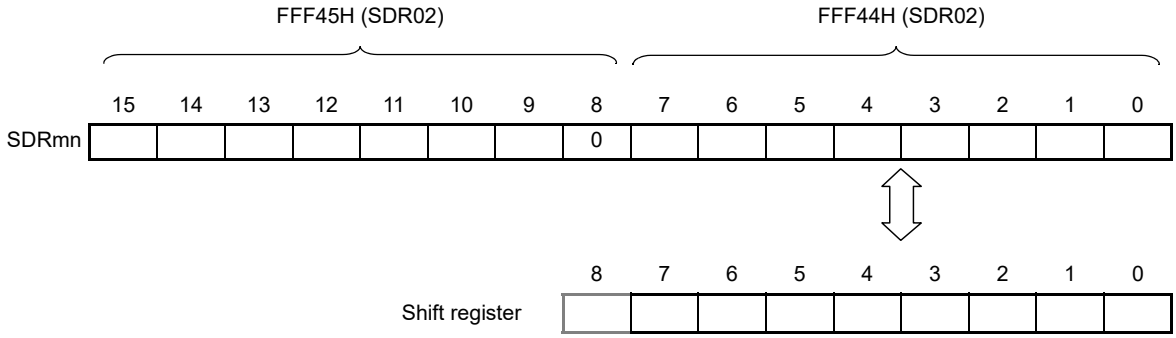
Remark For the function of the higher 7 bits of the SDRmn register, see **24.3 Registers to Control the Serial Array Unit**.

Figure 24 - 4 Format of Serial Data Register mn (SDRmn) (mn = 02, 03, 10, 11)

Address: FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03),
FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11)
FFF14H, FFF15H (SDR12), FFF16H, FFF17H (SDR13)

After reset: 0000H

R/W: R/W



Caution **Be sure to clear bit 8 to 0.**

Remark For the function of the higher 7 bits of the SDRmn register, see **24.3 Registers to Control the Serial Array Unit**.

24.3 Registers to Control the Serial Array Unit

The following registers are used to control the serial array unit.

- Peripheral enable register 0 (PER0)
- Peripheral reset control register 0 (PRR0)
- Serial clock select registers m (SPSm) (m = 0, 1)
- Serial mode registers mn (SMRmn) (mn = 00 to 03, 10, 11)
- Serial communication operation setting registers mn (SCRmn) (mn = 00 to 03, 10, 11)
- Serial data registers mn (SDRmn) (mn = 00 to 03, 10, 11)
- Serial flag clear trigger registers mn (SIRmn) (mn = 00 to 03, 10, 11)
- Serial status registers mn (SSRmn) (mn = 00 to 03, 10, 11)
- Serial channel start registers m (SSm) (m = 0, 1)
- Serial channel stop registers m (STm) (m = 0, 1)
- Serial channel enable status registers m (SEm) (m = 0, 1)
- Serial output enable registers m (SOEm) (m = 0, 1)
- Serial output registers m (SOm) (m = 0, 1)
- Serial output level registers m (SOLm) (m = 0, 1)
- Serial standby control register m (SSCm) (m = 0)
- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port mode registers xx (PMxx) (xx = 0, 1, 3, 5 to 7)
- Port registers xx (Pxx) (xx = 0, 1, 3, 5, 7)
- Port input mode registers xx (PIMxx) (xx = 0, 1, 3, 5, 7)
- Port output mode registers xx (POMxx) (xx = 0, 1, 3, 5, 7)
- Port mode control A registers xx (PMCAxx) (xx = 0, 1)
- UART loopback select register (ULBS)

24.3.1 Peripheral enable register 0 (PER0)

The PER0 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. Set bits 2 (SAU0EN) and 3 (SAU1EN) of this register to 1 to use serial array units 0 and 1, respectively. The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of the PER0 register following a reset is 00H.

Figure 24 - 5 Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

SAUmEN	Control of supply of an input clock to serial array unit m
0	Stops supply of an input clock. • The SFRs used by serial array unit m cannot be written. • When an SFR used by serial array unit m is read, the value returned is 00H or 0000H.
1	Enables supply of an input clock. • The SFRs used by serial array unit m can be read and written.

Caution 1. When setting serial array unit m, start by setting the following registers while the setting of the SAUmEN bit is 1.
- **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)**

If the setting of the SAUmEN bit is 0, writing to the registers which control serial array unit m is ignored. Note, however, writing to the following registers is valid.
- **Input switch control register (ISC)**
- **Noise filter enable register 0 (NFEN0)**
- **Port input mode registers 0, 1, 3, 5, 7 (PIM0, PIM1, PIM3, PIM5, PIM7)**
- **Port output mode registers 0, 1, 3, 5, 7 (POM0, POM1, POM3, POM5, POM7)**
- **Port mode control A registers 0, 1 (PMCA0, PMCA1)**
- **Port mode registers 0, 1, 3, 5, 7 (PM0, PM1, PM3, PM5, PM7)**
- **Port registers 0, 1, 3, 5, 7 (P0, P1, P3, P5, P7)**

Caution 2. Be sure to set bits 6 and 1 to 0.

24.3.2 Peripheral reset control register 0 (PRR0)

The PRR0 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. Set bits 2 (SAU0RES) and 3 (SAU1RES) of this register to 1 to place serial array units 0 and 1 in the reset state, respectively. The PRR0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of the PRR0 register following a reset is 00H.

Figure 24 - 6 Format of Peripheral Reset Control Register 0 (PRR0)

Address: F00F1H
After reset: 00H
R/W: R/W

Symbol	7	6	<5>	<4>	<3>	<2>	1	<0>
PRR0	0	0	ADCRES	IICA0RES	SAU1RES	SAU0RES	0	TAU0RES

SAUmRES	Control resetting of serial array unit m
0	Serial array unit m is released from the reset state.
1	Serial array unit m is in the reset state. • The SFRs for use with serial array unit m are initialized.

Caution Be sure to set bits 7, 6, and 1 to 0.

24.3.3 Serial clock select registers m (SPSm) (m = 0, 1)

The SPSm 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 respective channel is in operation (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 as the SPSmL register by an 8-bit memory manipulation instruction. The value of each SPSm register following a reset is 0000H.

Figure 24 - 7 Format of Serial Clock Select Register m (SPSm)

Address: F0126H, F0127H (SPS0), F0166H, F0167H (SPS1)Note
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SPSm	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
PRSm13	PRSm12	PRSm11	PRSm10	PRSm03	PRSm02	PRSm01	PRSm00

PRS mk3	PRS mk2	PRS mk1	PRS mk0		Selection of operation clock (CKmk)Note					
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 32 MHz	f_{CLK} = 48 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz	32 MHz	Setting prohibited
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz	24 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz	12 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	4 MHz	6 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz	2 MHz	3 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz	1 MHz	1.5 MHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 kHz	750 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	250 kHz	375 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz	188 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	62.5 kHz	93.8 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz	46.9 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	15.6 kHz	23.4 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz	11.7 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz	3.91 kHz	5.86 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz	2.93 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	153 Hz	305 Hz	610 Hz	977 Hz	1.46 kHz

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).

Caution Be sure to set bits 15 to 8 to 0.

Remark 1. f_{CLK}: CPU/peripheral hardware clock frequency
Remark 2. m: Unit number (m = 0, 1)
Remark 3. k = 0, 1

24.3.4 Serial mode registers mn (SMRmn) (mn = 00 to 03, 10, 11)

The SMRmn register is used to set 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, the operating mode (as simplified SPI or CSI, UART, or simplified I^2C), 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 respective channel is in operation (SEmn = 1). However, the MDmn0 bit can be rewritten during operation. The SMRmn register can be set by a 16-bit memory manipulation instruction. The value of each SMRmn register following a reset is 0020H.

Figure 24 - 8 Format of Serial Mode Register mn (SMRmn) (1/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03),
F0150H, F0151H (SMR10), F0152H, F0153H (SMR11)

After reset: 0020H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSmn

7	6	5	4	3	2	1	0
0	SISmn0	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 or 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 or CSI, UART transmission, and simplified I^2C).
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.	

SISmn0**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.

Figure 24 - 8 Format of Serial Mode Register mn (SMRmn) (2/2)

MDmn2	MDmn1	Setting of operation mode of channel n
0	0	Simplified SPI (CSI) mode
0	1	UART mode
1	0	Simplified I^2C mode
1	1	Setting prohibited

MDmn0	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 continuous transmission, set this bit to 1 and write the next transmit data when SDRmn data has run out.	

Note Only provided for the SMR01, SMR03, and SMR11 registers.

Caution **Be sure to clear bits 13 to 9, 7, 4, and 3 (or bits 13 to 6, 4, and 3 for the SMR00, SMR02, or SMR10 register) to 0. Be sure to set bit 5 to 1.**

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21), q: UART number (q = 0 to 2), r: IIC number (r = 00, 01, 10, 11, 20, 21)

24.3.5 Serial communication operation setting registers mn (SCRmn) (mn = 00 to 03, 10, 11)

The SCRmn 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 respective channel is in operation (SEmn = 1). The SCRmn register can be set by a 16-bit memory manipulation instruction. The value of each SCRmn register following a reset is 0087H.

Figure 24 - 9 Format of Serial Communication Operation Setting Register mn (SCRmn) (1/2)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03),
F0158H, F0159H (SCR10), F015AH, F015BH (SCR11)

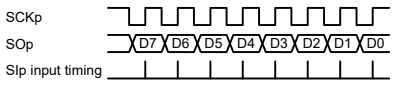
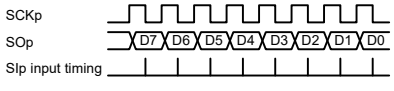
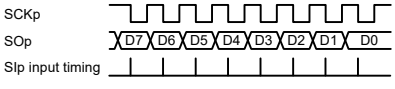
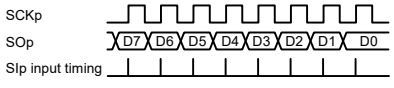
After reset: 0087H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn	0	EOCmn	PTCmn1	PTCmn0

7	6	5	4	3	2	1	0
DIRmn	0	SLCmn1	SLCmn0	0	1	DLSmn1	DLSmn0

TXEmn	RXEmn	Setting of operation mode of channel n
0	0	Disable communication.
0	1	Reception only
1	0	Transmission only
1	1	Transmission/reception

DAPmn	CKPmn	Selection of data and clock phase in simplified SPI (CSI) mode	Type
0	0		1
0	1		2
1	0		3
1	1		4
Set DAPmn and CKPmn to 00B in the UART mode and simplified I^2C mode.			

EOCmn	Mask control of error interrupt signal (INTSREx (x = 0 to 2))
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**.	

Figure 24 - 9 Format of Serial Communication Operation Setting Register mn (SCRmn) (2/2)

PTCmn1	PTCmn0	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 4**.	No parity judgment
1	0	Outputs even parity.	Handles as even parity.
1	1	Outputs odd parity.	Handles as odd parity.
Set PTCmn[1:0] to 00B in the simplified SPI (CSI) mode and simplified I^2C mode.			

DIRmn	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.
Set DIRmn = 0 in the simplified I^2C mode.	

SLCmn1 **Note 1**	SLCmn0	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, 10 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 (SLCmn[1:0] = 01B) during UART reception and in the simplified I^2C mode. Set no stop bit (SLCmn[1:0] = 00B) in the simplified SPI (CSI) mode. Set 1 bit (SLCmn[1:0] = 01B) or 2 bits (SLCmn[1:0] = 10B) during UART transmission.		

DLSmn1 **Note 2**	DLSmn0	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)
0	0	Setting prohibited
Set DLSmn[1:0] to 11B in the simplified I^2C mode.		

Note 1. Only provided for the SCR00, SCR02, and SCR10 registers.
Note 2. Only provided for the SCR00 and SCR01 registers. This bit is fixed to 1 for the other registers.
Note 3. The setting of EOCmn to 1 in CSImn may cause an error interrupt (INTSREn).
Note 4. 0 is always appended regardless of the data contents.

Caution **Set bits 11, 6, and 3 to 0 (also set bit 5 of the SCR01, SCR03, and SCR11 register to 0). Set bit 2 to 1.**

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)

24.3.6 Serial data registers mn (SDRmn) (mn = 00 to 03, 10, 11)

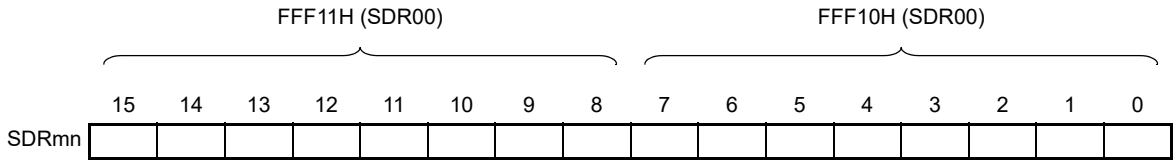
The SDRmn is a 16-bit transmit/receive data register of channel n. Bits 8 to 0 (lower 9 bits) of the SDR00, SDR01, SDR10, and SDR11 registers or bits 7 to 0 (lower 8 bits) of the SDR02, SDR03, SDR10, and SDR11 registers function as a transmit/receive buffer register, and bits 15 to 9 (higher 7 bits) 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 operation clock by bits 15 to 9 (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 the SDR00, SDR01, SDR10, and SDR11 registers to 0000000B. The input clock f_{SCK} (slave transfer in simplified SPI or CSI mode) from the SCKp pin is used as the transfer clock. The lower 8 or 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 or 9 bits, and during transmission, the data to be transmitted to the shift register is set to the lower 8 or 9 bits. The SDRmn register can be read or written in 16-bit units. However, the higher 7 bits can only be written or read when the operation is stopped (SEmn = 0). During operation (SEmn = 1), a value is written only to the lower 8 or 9 bits of the SDRmn register. When the SDRmn register is read during operation, the higher 7 bits are always read as 0. The value of each SDRmn register following a reset is 0000H.

Figure 24 - 10 Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01)
FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11)

After reset: 0000H

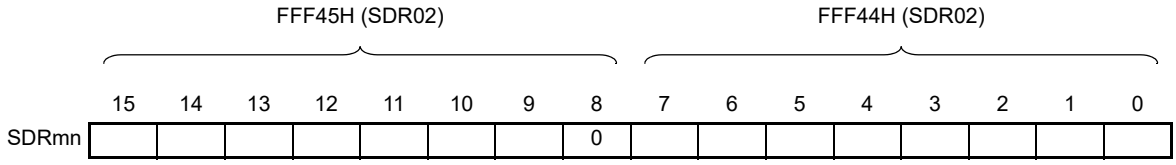
R/W: R/W



Address: FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03)
FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11)

After reset: 0000H

R/W: R/W



SDRmn[15:9]							Transfer clock setting by dividing the operation 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

Caution 1. Set bit 8 of the SDR02, SDR03, SDR10, and SDR11 registers to 0.

Caution 2. Setting SDRmn[15:9] to 0000000B or 0000001B is prohibited when UART is used.

Caution 3. Setting SDRmn[15:9] to 0000000B is prohibited when simplified I^2C is used. Set SDRmn[15:9] to 0000001B or greater.

Caution 4. When operation is stopped (SEmn = 0), do not rewrite the SDRmn[7:0] bits by an 8-bit memory manipulation instruction (the SDRmn[15:9] bits are all cleared to 0).

Remark 1. For the function of the lower 8 or 9 bits of the SDRmn register, see **24.2 Configuration of Serial Array Unit**.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

24.3.7 Serial flag clear trigger registers mn (SIRmn) (mn = 00 to 03, 10, 11)

The SIRmn 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 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 as the SIRmnL register by an 8-bit memory manipulation instruction. The value of each SIRmn register following a reset is 0000H.

Figure 24 - 11 Format of Serial Flag Clear Trigger Register mn (SIRmn)

Address: F0108H, F0109H (SIR00) to F010EH, F010FH (SIR03),
F0148H, F0149H (SIR10), F014AH, F014BH (SIR11)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SIRmn	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	FECTmn	PECTmn	OVCTmn

FECTmn **Note**	Clear trigger of framing error flag of channel n
0	Not cleared
1	Clears the FEFmn bit of the SSRmn register to 0.

PECTmn	Clear trigger of parity error flag of channel n
0	Not cleared
1	Clears the PEFmn bit of the SSRmn register to 0.

OVCTmn	Clear trigger of overrun error flag of channel n
0	Not cleared
1	Clears the OVFmn bit of the SSRmn register to 0.

Note Only provided for the SIR01, SIR03, and SIR11 registers.

Caution **Set bits 15 to 3 (or bits 15 to 2 for the SIR00, SIR02, and SIR10 registers) to 0.**

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Remark 2. When the SIRmn register is read, 0000H is always read.

24.3.8 Serial status registers mn (SSRmn) (mn = 00 to 03, 10, 11)

The SSRmn register indicates the state of communications and occurrence of errors for 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 read as the SSRmnL register by an 8-bit memory manipulation instruction. The value of each SSRmn register following a reset is 0000H.

Figure 24 - 12 Format of Serial Status Register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03),
F0140H, F0141H (SSR10), F0142H, F0143H (SSR11)

After reset: 0000H

R/W: R

Symbol	15	14	13	12	11	10	9	8
SSRmn	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
	0	TSFmn	BFFmn	0	0	FEFmn	PEFmn	OVFmn

TSFmn	Flag indicating the state of communications for 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.	

BFFmn	Flag indicating the state of the buffer register for 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 condition> • Transmit data is written to the SDRmn register while the TXEmn bit of the SCRmn register is set to 1 (transmission or transmission/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/reception mode in each communication mode). • A reception error occurs.	

Figure 24 - 12 Format of Serial Status Register mn (SSRmn) (2/2)

FEFmn[Note]	Framing error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception).
<Clear conditions> • 1 is written to the FECTmn bit of the SIRmn register. <Set condition> • A stop bit is not detected when UART reception ends.	

PEFmn	Parity/ACK error detection flag of channel n
0	No error occurs.
1	Parity error occurs (during UART reception) or ACK is not detected (during I^2C transmission).
<Clear conditions> • 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 at the ACK reception timing during I^2C transmission (ACK is not detected).	

OVFmn	Overrun error detection flag of channel n
0	No error occurs.
1	An error occurs
<Clear conditions> • 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/reception mode in each communication mode). • Transmit data is not ready for slave transmission or transmission/reception in simplified SPI (CSI) mode.	

Note Only provided for the 3SSR01, SSR03, and SSR11 registers.

Caution 1. 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 (OVFmn = 1) is detected.

Caution 2. When the simplified SPI (CSI) is performing reception operations in the SNOOZE mode (SWCm = 1), the OVFmn flag will not change.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

24.3.9 Serial channel start registers m (SSm) (m = 0, 1)

The SSm is a trigger register that is used to enable starting communication/count by each channel. When 1 is written to a bit (SSmn) of this register, 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 as the SSmL register by an 1-bit or 8-bit memory manipulation instruction. The value of each SSm register following a reset is 0000H.

Figure 24 - 13 Format of Serial Channel Start Register m (SSm)

Address: F0122H, F0123H (SS0)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SS0	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	SS03	SS02	SS01	SS00

Address: F0162H, F0163H (SS1)Note
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SS1	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	SS11	SS10

SSmn	Operation start trigger of channel n
0	No trigger operation
1	Set the SEmn bit to 1 to place the channel in the communications waiting state.Note

Note Setting an SSmn bit to 1 during communications stops communications through channel n and places the channel in the waiting state. At this time, the values of the control registers and shift register, the states of the SCKmn and SOmn pins, and the values of the FEFmn, PEFmn, and OVFmn flags are retained.

Caution 1. Set bits 15 to 4 of the SS0 register, and bits 15 to 2 of the SS1 registers 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 at least 4 fMCK clock cycles have elapsed.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
Remark 2. When the SSm register is read, 0000H is always read.

24.3.10 Serial channel stop registers m (STm) (m = 0, 1)

The STm is a trigger register that is used to enable stopping communication/count by each channel. When 1 is written to a bit (STmn) of this register, 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 be set by a 16-bit memory manipulation instruction. The lower 8 bits of the STm register can be set as the STmL register by a 1-bit or 8-bit memory manipulation instruction. The value of each STm register following a reset is 0000H.

Figure 24 - 14 Format of Serial Channel Stop Register m (STm)

Address: F0124H, F0125H (ST0)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
ST0	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	ST03	ST02	ST01	ST00

Address: F0164H, F0165H (ST1)Note
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
ST1	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	ST11	ST10

STmn	Operation stop trigger of channel n
0	No trigger operation
1	Clears the SEmn bit to 0 and stops the communication operationNote.

Note The values of the control registers and shift register, the states of the SCKmn and SOmn pins, and the values of the FEFmn, PEFmn, and OVFmn flags are retained.

Caution Set bits 15 to 4 of the ST0 register, and bits 15 to 2 of the ST1 register to 0.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
Remark 2. When the STm register is read, 0000H is always read.

24.3.11 Serial channel enable status registers m (SEm) (m = 0, 1)

The SEm register indicates whether data transmission/reception operation of each channel is enabled or stopped. When 1 is written to a bit of serial channel start register m (SSm), the corresponding bit of this register is set to 1. When 1 is written to a bit of serial channel stop register m (STm), the corresponding bit is cleared to 0. For channel n whose operation is enabled, the value of the CKOmn bit of serial output register m (SOm) to be described later cannot be rewritten by software, and a value reflected by a communication operation is output from the serial clock pin. For channel n whose operation is stopped, the value of the CKOmn bit of the SOm register can be set by software and is output 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 read as the SEmL register by a 1-bit or 8-bit memory manipulation instruction. The value of each SEm register following a reset is 0000H.

Figure 24 - 15 Format of Serial Channel Enable Status Register m (SEm)

Address: F0120H, F0121H (SE0)
After reset: 0000H
R/W: R

Symbol	15	14	13	12	11	10	9	8
SE0	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	SE03	SE02	SE01	SE00

Address: F0160H, F0161H (SE1)
After reset: 0000H
R/W: R

Symbol	15	14	13	12	11	10	9	8
SE1	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	SE11	SE10

SEmn	Indication of whether operation of channel n is enabled or stopped.
0	Operation stops
1	Operation is enabled.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

24.3.12 Serial output enable registers m (SOEm) (m = 0, 1)

The SOEm register is used to enable or stop output of the serial communication operation of each channel. For channel n whose serial output is enabled, the value of the SOmn bit of serial output register m (SOm) to be described later cannot be rewritten by software, 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, such as that of a start condition/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 as the SOEmL register by a 1-bit or 8-bit memory manipulation instruction. The value of each SOEm register following a reset is 0000H.

Figure 24 - 16 Format of Serial Output Enable Register m (SOEm)

Address: F012AH, F012BH (SOE0)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SOE0	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	SOE03	SOE02	SOE01	SOE00

Address: F016AH, F016BH (SOE1)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SOE1	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	SOE11	SOE10

SOEmn	Serial output enable/stop of channel n
0	Stops output by serial communication operation.
1	Enables output by serial communication operation.

Caution **Set bits 15 to 4 of the SOE0 register, and bits 15 to 2 of the SOE1 register to 0.**

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

24.3.13 Serial output registers m (SOm) (m = 0, 1)

The SOm 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 the pin for serial interface as a port function pin, set the corresponding CKOmn and SOmn bits to 1. The SOm register can be set by a 16-bit memory manipulation instruction. The value of each SOm register following a reset is 0F0FH.

Figure 24 - 17 Format of Serial Output Register m (SOm)

Address: F0128H, F0129H (SO0)
After reset: 0F0FH
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SO0	0	0	0	0	CKO03	CKO02	CKO01	CKO00

7	6	5	4	3	2	1	0
0	0	0	0	SO03	SO02	SO01	SO00

Address: F0168H, F0169H (SO1)
After reset: 0303H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SO1	0	0	0	0	0	0	CKO11	CKO10

7	6	5	4	3	2	1	0
0	0	0	0	0	0	SO11	SO10

CKOmn	Serial clock output of channel n
0	Serial clock output value is 0.
1	Serial clock output value is 1.

SOmn	Serial data output of channel n
0	Serial data output value is 0.
1	Serial data output value is 1.

Caution **Set bits 15 to 12, and 7 to 4 of the SO0 register to 0.**
Set bits 15 to 10, and 7 to 2 of the SO1 register to 0.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

24.3.14 Serial output level registers m (SOLm) (m = 0, 1)

The SOLm register is used to set inversion of the data output level of each channel. This register can be set only in the UART mode. Set 0 for the bit corresponding the channel used in the simplified SPI (CSI) mode or 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 respective channel is in operation (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 as the SOLmL register by an 8-bit memory manipulation instruction. The value of each SOLm register following a reset is 0000H.

Figure 24 - 18 Format of Serial Output Level Register m (SOLm)

Address: F0134H, F0135H (SOL0)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SOL0	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	SOL02	0	SOL00

Address: F0174H, F0175H (SOL1)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SOL1	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	SOL10

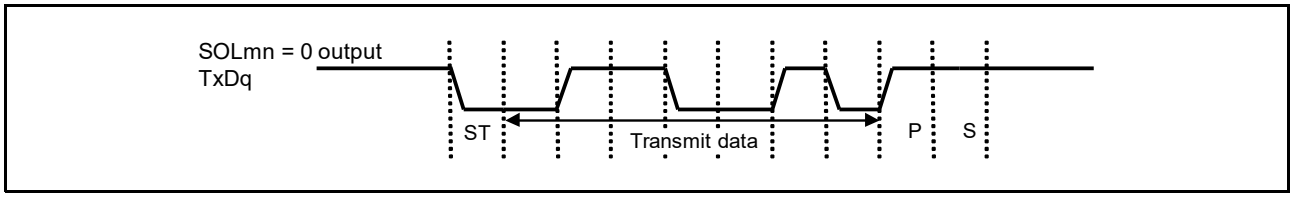
SOLmn	Selects inversion of the level of the transmit data of channel n in UART mode
0	Communication data is output as is.
1	Communication data is inverted and output.

Caution Set bits 15 to 3, and 1 of the SOL0 register, and bits 15 to 1 of the SOL1 register to 0.

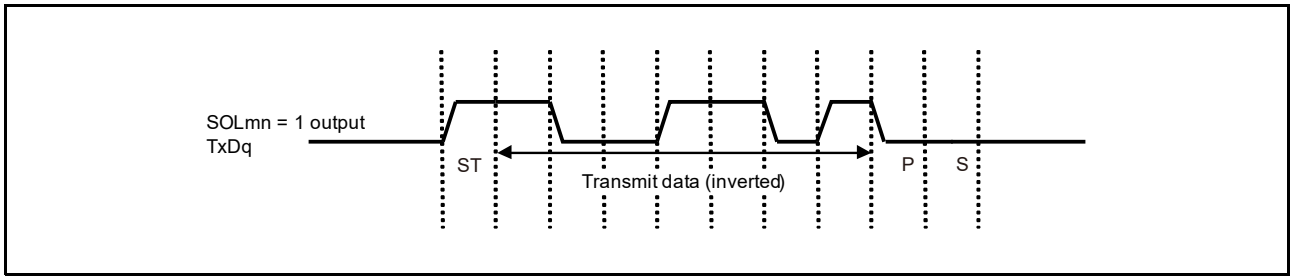
Figure 24 - 19 shows examples in which the level of transmit data is reversed during UART transmission.

Figure 24 - 19 Examples of Reverse Transmit Data

a) Non-reverse Output (SOLmn = 0)



b) Reverse Output (SOLmn = 1)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

24.3.15 Serial standby control register m (SSCm) (m = 0)

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 as the SSCmL register by an 8-bit memory manipulation instruction. The value of each SSCm register following a reset is 0000H.

Caution The maximum transfer rate in the SNOOZE mode is as follows.

- **When using CSI00: Up to 1 Mbps**
- **When using UART0: Up to 115.2 kbps (when setting FWKUP = 1, $f_{CLK} = f_{IH}$ (32 MHz))**

Figure 24 - 20 Format of Serial Standby Control Register m (SSCm)

Address: F0138H, F0139H (SSC0)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SSCm	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	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/INTSRE2).
1	Disable the generation of error interrupts (INTSRE0/INTSRE2).

- 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 or medium-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 immediately before shifting to STOP mode.
  Also, be sure to change the SWCm bit to 0 after returning from STOP mode to normal operation mode.

Table 24 - 2 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.	INTSREx is generated.
1	1	INTSRx is generated.	No interrupt is generated.

Remark m: Unit number (m = 0), n: Channel number (n = 0)

24.3.16 Input switch control register (ISC)

The ISC[1:0] bits of the ISC register are used to realize a LIN-bus communication operation by UART2 in coordination with an external interrupt and the timer array unit. When bit 0 is set to 1, the input signal of the serial data input (RxD0) pin is selected as an external interrupt (INTP0) that can be used to detect a wakeup signal. When bit 1 is set to 1, the input signal of the serial data input (RxD0) pin is selected as a timer input, so that wake up signal can be detected, the low width of the break field, and the pulse width of the sync field can be measured by the timer. The SSIE00 bit is used to control the $\overline{\text{SSI00}}$ input of channel 0 in the communications through CSI00 in the slave mode. Reception and transmission do not proceed even if the serial clock is input while the $\overline{\text{SSI00}}$ pin is being driven high. Reception and transmission proceed in response to an input of the serial clock according to the mode setting while the $\overline{\text{SSI00}}$ pin is being driven low. The ISC register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of the ISC register following a reset is 00H.

Figure 24 - 21 Format of Input Switch Control Register (ISC)

Address: F0073H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ISC	SSIE00 **Note**	0	0	0	0	0	ISC1	ISC0

SSIE00	Setting of the $\overline{\text{SSI00}}$ input of channel 0 in the communications through CSI00 in the slave mode
0	The $\overline{\text{SSI00}}$ input is disabled.
1	The $\overline{\text{SSI00}}$ input is enabled.

ISC1	Switching channel 3 input of timer array unit 0
0	Uses the input signal of the TI03 pin as a timer input (normal operation).
1	Input signal of the RxD0 pin is used as timer input (detects the wakeup signal and measures the low width of the break field and the pulse width of the sync field).

ISC0	Switching external interrupt (INTP0) input
0	Uses the input signal of the INTP0 pin as an external interrupt (normal operation).
1	Uses the input signal of the RxD0 pin as an external interrupt (wakeup signal detection).

Note In the 20- to 25-pin products, set this bit to 0.

Caution Set bits 6 to 2 to 0.

24.3.17 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, after synchronization is performed with the operation clock (f_{MCK}) of the target channel, 2-clock match detection is performed. When the noise filter is disabled, only synchronization is performed with the operation clock (f_{MCK}) of the target channel. The NFEN0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of the NFEN0 register following a reset is 00H.

Figure 24 - 22 Format of Noise Filter Enable Register 0 (NFEN0)

Address: F0070H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
NFEN0	0	0	0	SNFEN20	0	SNFEN10	0	SNFEN00

SNFEN20	Use of noise filter of RxD2 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN20 to 1 to use the RxD2 pin. Clear SNFEN20 to 0 to use the other than RxD2 pin.	

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 SNFEN10 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 SNFEN00 to 0 to use the other than RxD0 pin.	

24.3.18 Registers for controlling the port functions multiplexed with the inputs and outputs of the serial array unit

Set the following registers to control the port functions multiplexed with the inputs and outputs of the serial array unit.

- Port mode registers xx (PMxx)
- Port registers xx (Pxx)
- Port input mode registers xx (PIMxx)
- Port output mode registers xx (POMxx)
- Port mode control A registers xx (PMCAxx)

For details, see the following sections.

- **7.3.1 Port mode registers xx (PMxx)**
- **7.3.2 Port registers xx (Pxx)**
- **7.3.4 Port input mode registers xx (PIMxx)**
- **7.3.5 Port output mode registers xx (POMxx)**
- **7.3.7 Port mode control A registers xx (PMCAxx)**

When the pins multiplexed with SOp, SCKp, SCLr, SDAr, and TxD0 to TxD2 are to be used for serial data outputs or serial clock outputs, set the corresponding bits in the given port mode control A registers xx (PMCAxx) and port mode registers xx (PMxx) to 0 and those in port registers xx (Pxx) to 1.

When using the port pin in N-ch open-drain output (withstand voltage of V_{DD}[Note 1]/withstand voltage of EV_{DD}[Note 2]) mode, set the corresponding bit in port output mode register xx (POMxx) to 1. When connecting an external device operating at a different voltage (1.8 V, 2.5 V or 3 V), see **7.4.5 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers**.

Example: When P02/ANI17/SO10/TxD1 is to be used for serial data output

- Set the PMCA02 bit of port mode control A register 0 to 0.
- Set the PM02 bit of port mode register 0 to 0.
- Set the P02 bit of port register 0 to 1.

When the pins multiplexed with SIp, SCKp, and RxD0 to RxD2 are to be used for serial data inputs or serial clock inputs, set the corresponding bits in the given port mode registers xx (PMxx) to 1 and those in port mode control A registers xx (PMCAxx) to 0. At this time, the bits in port registers xx (Pxx) may be 0 or 1.

Example: When P03/ANI16/SI10/RxD1/SDA10 is to be used for serial data input

- Set the PMCA03 bit of port mode control A register 0 to 0.
- Set the PM03 bit of port mode register 0 to 1.
- Set the P03 bit of port register 0 to 0 or 1.

Note 1. This applies to the 20- to 52-pin products.

Note 2. This applies to the 64-pin products.

Remark xx = 0, 1, 3, 5 to 7

Note that PIM6, POM6, PMCA3, and PMCA5 to PMCA7 are not present in the RL78/G24 products.

24.3.19 UART loopback select register (ULBS)

The ULBS register is used to enable the UART loopback function. This register has bits to individually control UART channels. When the bit corresponding to each channel is set to 1, the UART loopback function is selected, and output from the transmission shift register is looped back to the reception shift register. The ULBS register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of the ULBS register following a reset is 00H.

Figure 24 - 23 Format of UART Loopback Select Register (ULBS)

Address: F0079H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
ULBS	0	0	0	0	0	ULBS2	ULBS1	ULBS0

ULBSn	Selection of the UART loopback function
0	Inputs the state of the RxDn pin of serial array unit n to the reception shift register.
1	Loops back output from the transmission shift register to the reception shift register.

Caution **Set bits 7 to 3 to 0.**

Remark n = 0 to 2

24.4 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.

24.4.1 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 the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. To stop the operation of serial array unit 0, set bit 2 (SAU0EN) to 0. To stop the operation of serial array unit 1, set bit 3 (SAU1EN) to 0.

Figure 24 - 24 Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units

a) Peripheral enable register 0 (PER0): Set only the bit of SAUm to be stopped to 0.

	7	6	5	4	3	2	1	0
PER0	RTCWEN ×	0	ADCEN ×	IICA0EN ×	SAU1EN 0/1	SAU0EN 0/1	0	TAU0EN ×

Control of SAUm input clock
0: Stops supply of input clock
1: Supplies input clock

Caution 1. If SAUmEN = 0, writing to the registers which control serial array unit m is ignored, and, even if the register is read, only the initial 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, 1, 3, 5, 7 (PIM0, PIM1, PIM3, PIM5, PIM7)**
- **Port output mode registers 0, 1, 3, 5, 7 (POM0, POM1, POM3, POM5, POM7)**
- **Port mode control A registers 0, 1 (PMCA0, PMCA1)**
- **Port mode registers 0, 1, 3, 5, 7 (PM0, PM1, PM3, PM5, PM7)**
- **Port registers 0, 1, 3, 5, and 7 (P0, P1, P3, P5, and P7)**

Caution 2. Set bits 6 and 1 to 0.

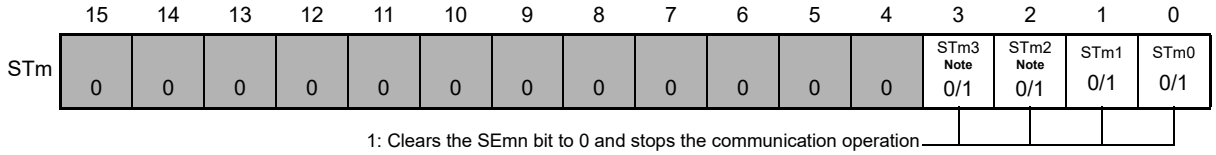
Remark ×: Bits not used with serial array units (depending on the settings of other peripheral functions)
0/1: Set to 0 or 1 depending on the usage of the user

24.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

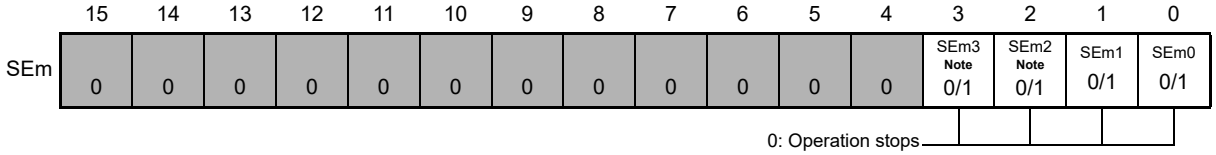
Figure 24 - 25 Each Register Setting When Stopping the Operation by Channels

a) Serial channel stop register m (STm): The STm is a trigger register that is used to enable stopping communication/count by each channel.



* 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.

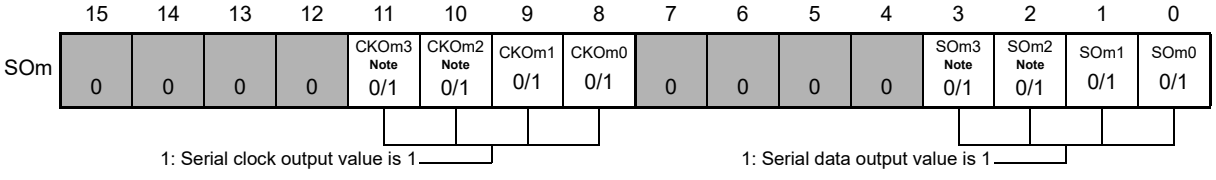


* The SEm is a read-only status register, whose operation is stopped by using the STm register.
With a channel whose operation is stopped, the value of the CKOmn bit of the SOm register can be set by software.

c) Serial output enable register m (SOEm): This register is used to enable or stop output of the serial communication operation of each channel.

* 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): The SOm is a buffer register for serial output of each channel.



* When using pins corresponding to each channel as port function pins, set the corresponding CKOmn, SOmn bits to 1.

Note These bits are only present in serial array unit 1.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Remark 2. ■: Setting disabled (fixed by hardware), 0/1: Set to 0 or 1 depending on the usage of the user

24.5 Operation of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) Communication

Simplified SPI is a clock synchronous communications 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[Note] During master communication: Max. f_{CLK}/2 (CSI00 only)
  Max. f_{CLK}/4
  During slave communication: Max. f_{MCK}/6

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

CSI00 and CSI20 support the SNOOZE mode. In the SNOOZE mode, data can be received without CPU processing upon detecting SCK input in the STOP mode. The SNOOZE mode is only available in CSI00.

Note Set up the transfer rate within a range satisfying the SCK cycle time (t_{KCY}). For details, see **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** or **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.

Caution Use a general-purpose port pin to send a chip select signal when required.

The channels supporting simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) are channels 0 to 3 of SAU0 and channels 0 and 1 of SAU1.

<20-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I2C
0	0	—	—	—
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

<24- and 25-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I2C
0	0	—	UART0 (supporting LIN-bus)Note	—
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

<30- and 32-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I2C
0	0	CSI00 (supporting slave select input)	UART0 (supporting LIN-bus)	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

<40- and 44-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I2C
0	0	CSI00 (supporting slave select input)	UART0 (supporting LIN-bus)	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	CSI21		IIC21

<48- and 52-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00 (supporting slave select input)	UART0 (supporting LIN-bus)	IIC00
	1	CSI01		IIC01
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	CSI21		IIC21

<64-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00 (supporting slave select input)	UART0 (supporting LIN-bus)	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	CSI21		IIC21

Note This function can be used when the setting of bit 1 (PIOR01) in the peripheral I/O redirection register 0 (PIOR0).

Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) handles the following seven types of communications.

- Master transmission (See **24.5.1**.)
- Master reception (See **24.5.2**.)
- Master transmission/reception (See **24.5.3**.)
- Slave transmission (See **24.5.4**.)
- Slave reception (See **24.5.5**.)
- Slave transmission/reception (See **24.5.6**.)
- SNOOZE mode function (See **24.5.7**.)

24.5.1 Master transmission

Master transmission is that the RL78 microcontroller outputs a transfer clock and transmits data to another device.

Simplified SPI	CSI00	CSI01	CSI10	CSI11	CSI20	CSI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCK00, SO00	SCK01, SO01	SCK10, SO10	SCK11, SO11	SCK20, SO20	SCK21, SO21
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21
	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. $f_{CLK}/2$ [Hz] (CSI00 only), $f_{CLK}/4$ [Hz] 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 cycle 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 specified in the electrical characteristics. For details, see **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** or **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.

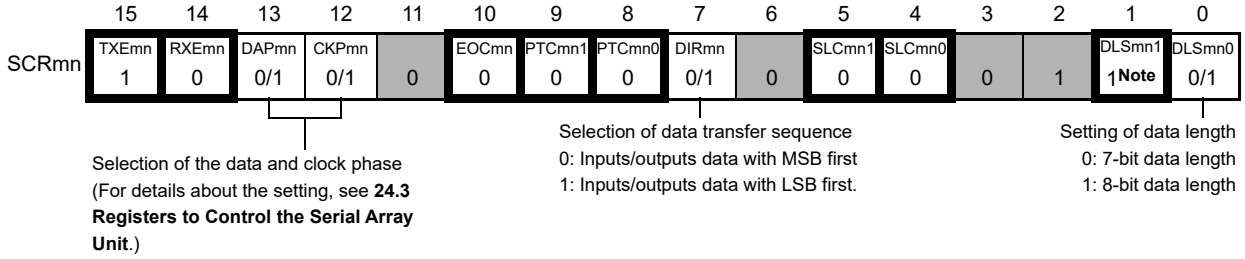
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

1. Register setting

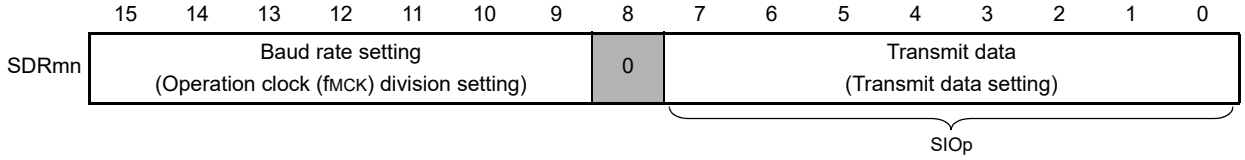
Figure 24 - 26 Example of Contents of Registers for Master Transmission of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21)

a) Serial mode register mn (SMRmn)

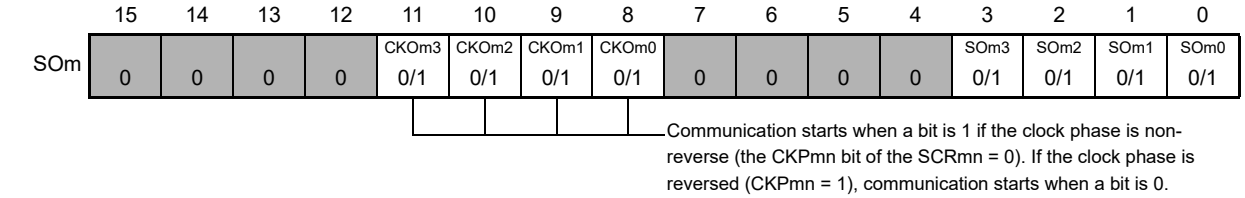
b) Serial communication operation setting register mn (SCRmn)



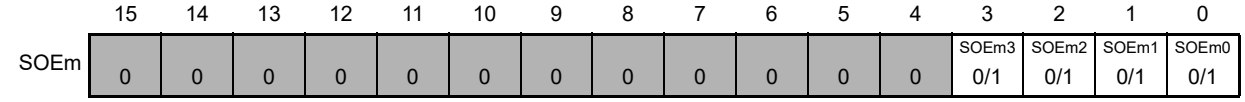
c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



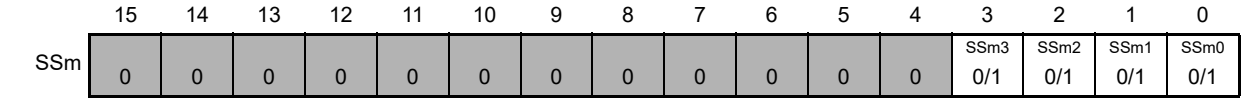
d) Serial output register m (SOm): Set only the bit of the target channel.



e) Serial output enable register m (SOEm): Set only the bit of the target channel to 1.



f) Serial channel start register m (SSm): Set only the bit of the target channel to 1.



Note Only provided for the SCR00 and SCR01 registers. This bit is fixed to 1 for the other registers.

(Remarks are listed on the next page.)

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

Remark 2. ■: Setting is fixed in the simplified SPI (CSI) master transmission mode

■: Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

2. Operation procedure

Figure 24 - 27 Initial Setting Procedure for Master Transmission

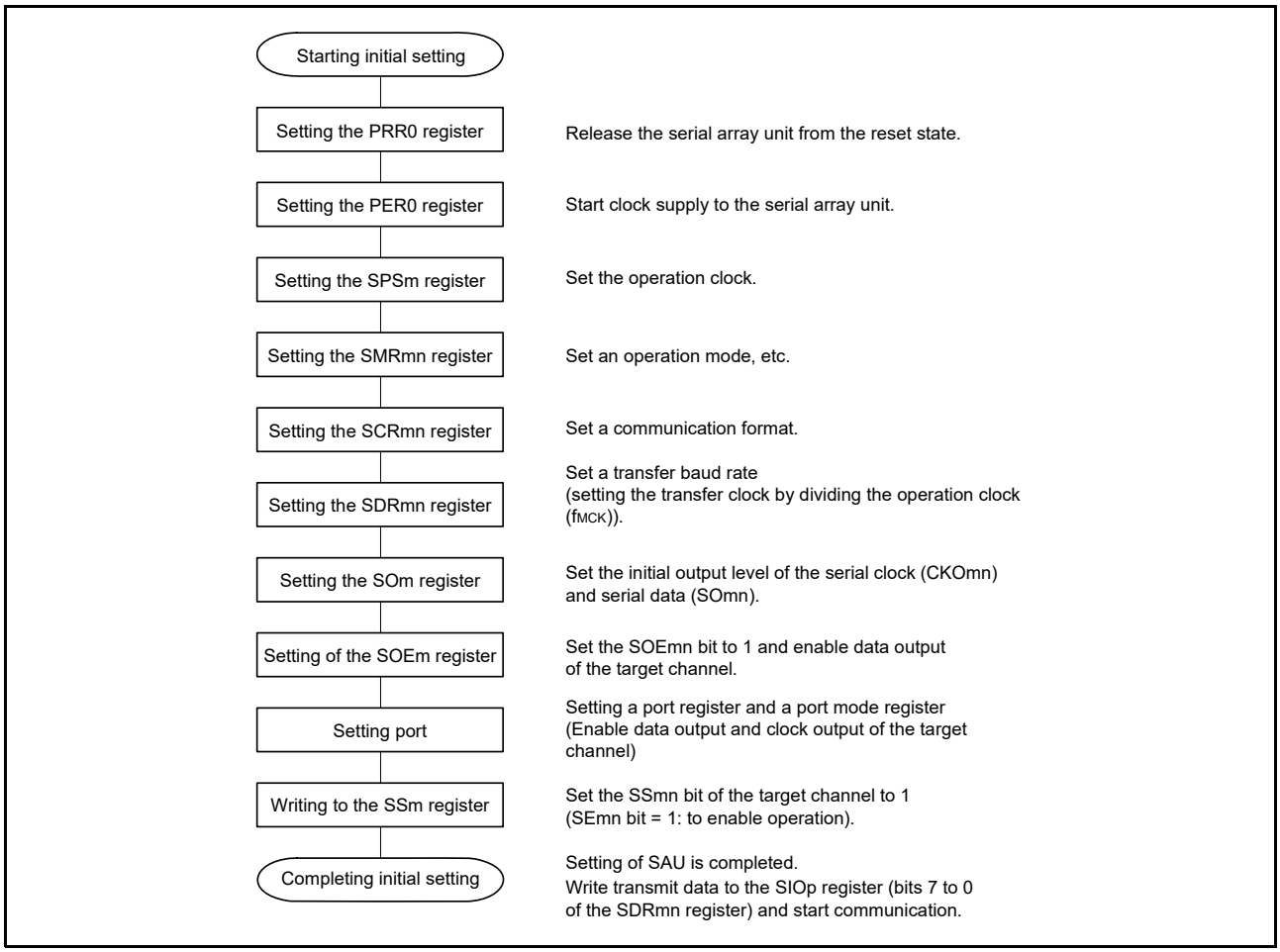


Figure 24 - 28 Procedure for Stopping Master Transmission

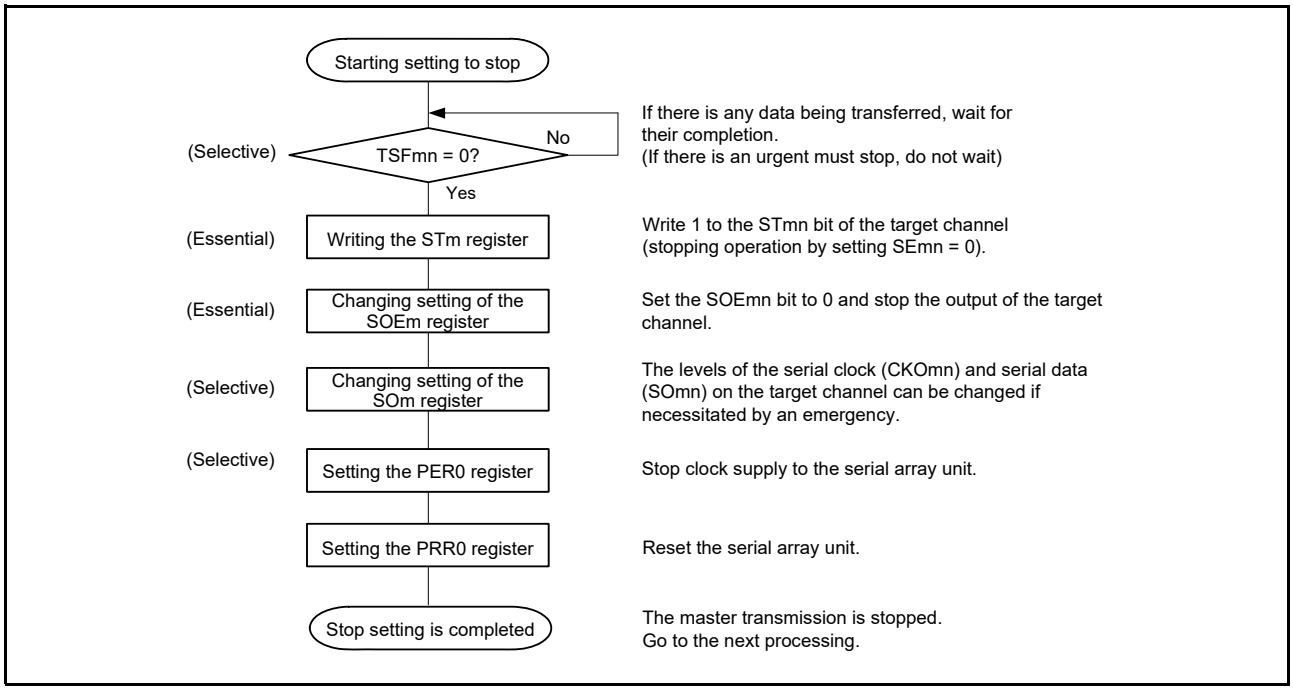
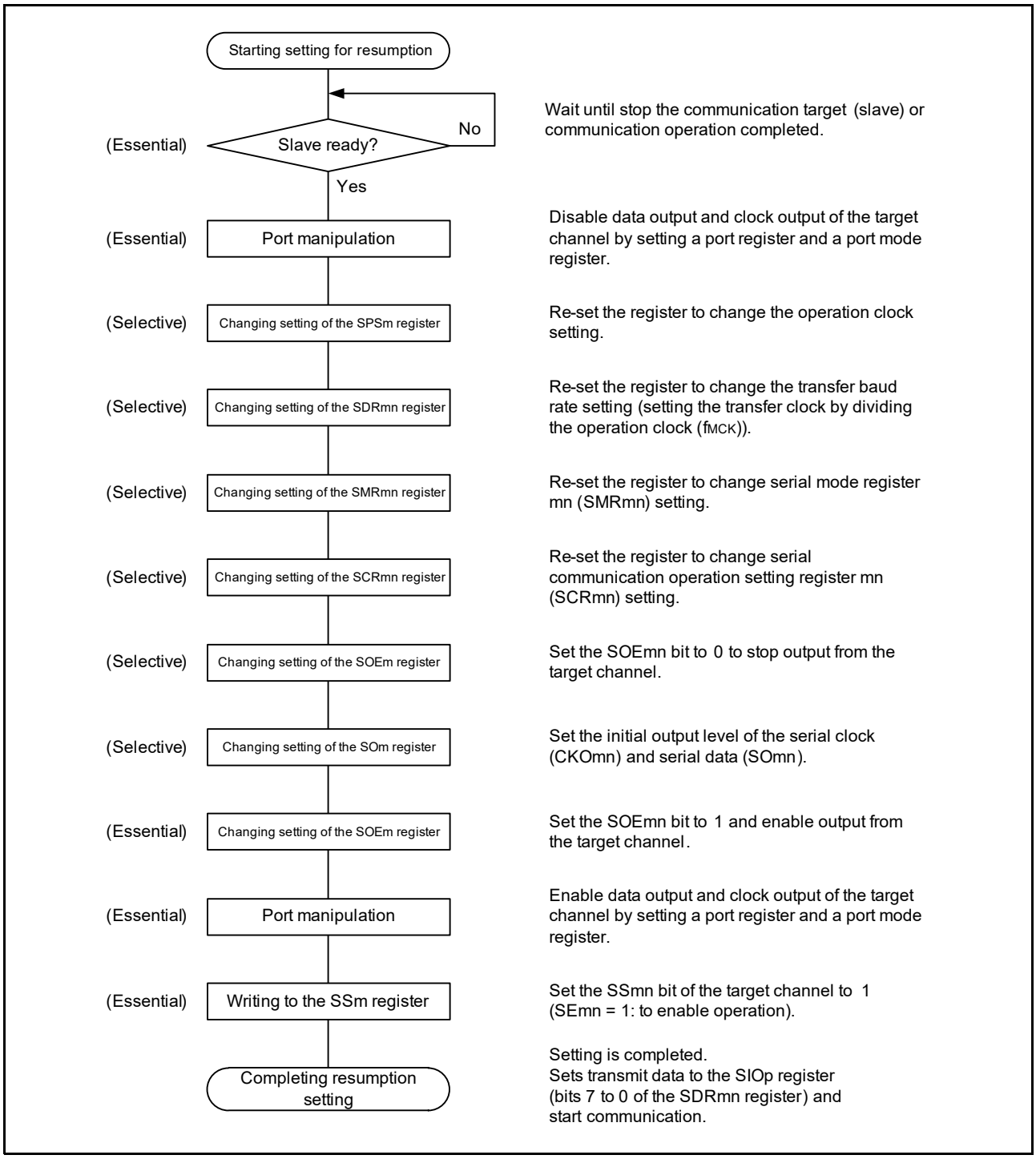


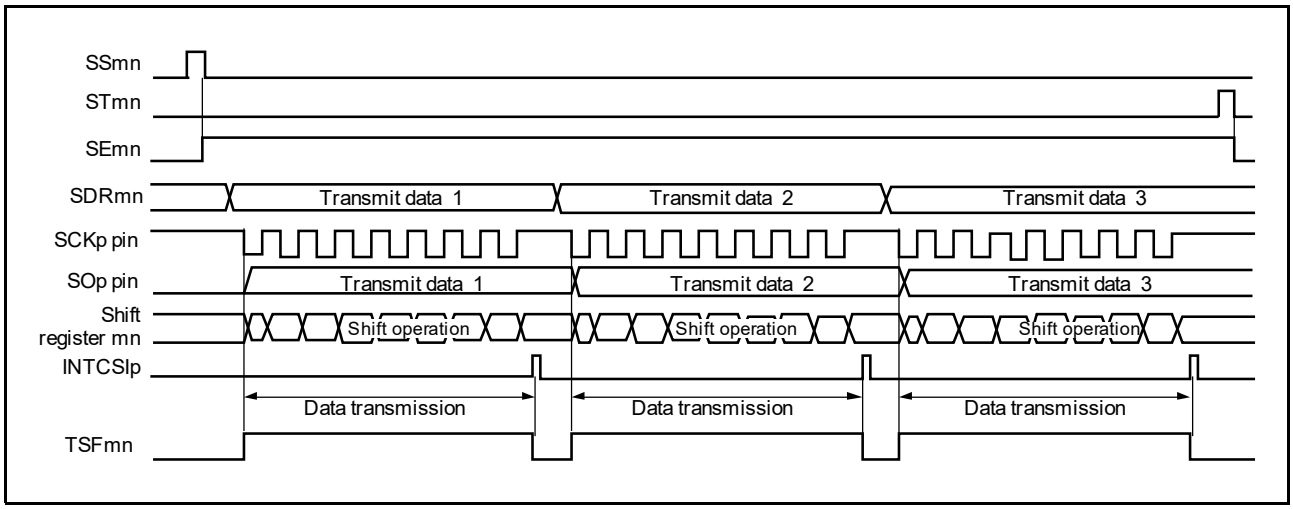
Figure 24 - 29 Procedure for Resuming Master Transmission



Remark If PRR0 is rewritten while stopping the communication to reset the serial array unit, wait until the communication target (slave) stops or communication finishes, and then perform initialization instead of restarting the communication.

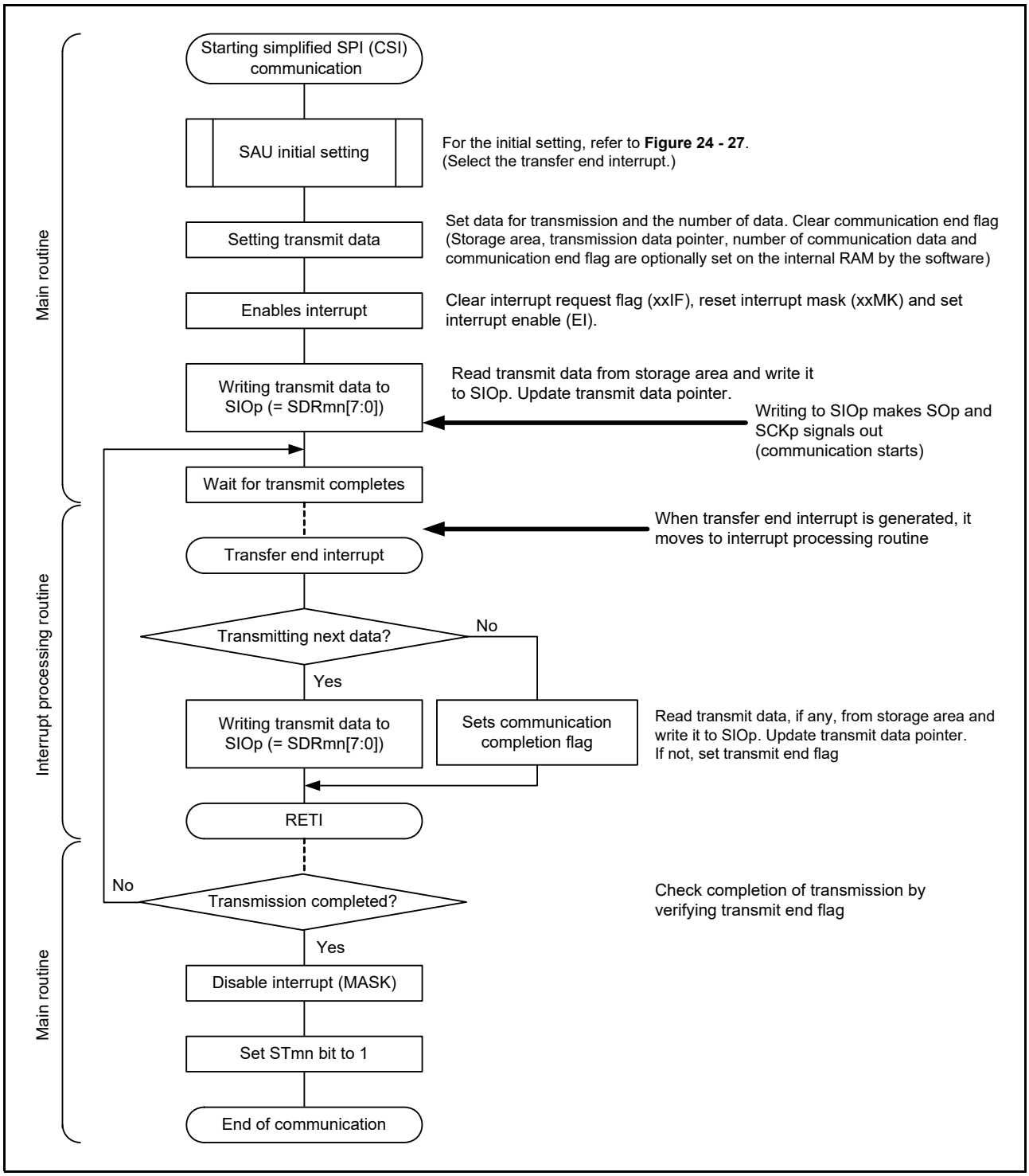
3. Processing flow (in single-transmission mode)

Figure 24 - 30 Timing Chart of Master Transmission (in Single-transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)



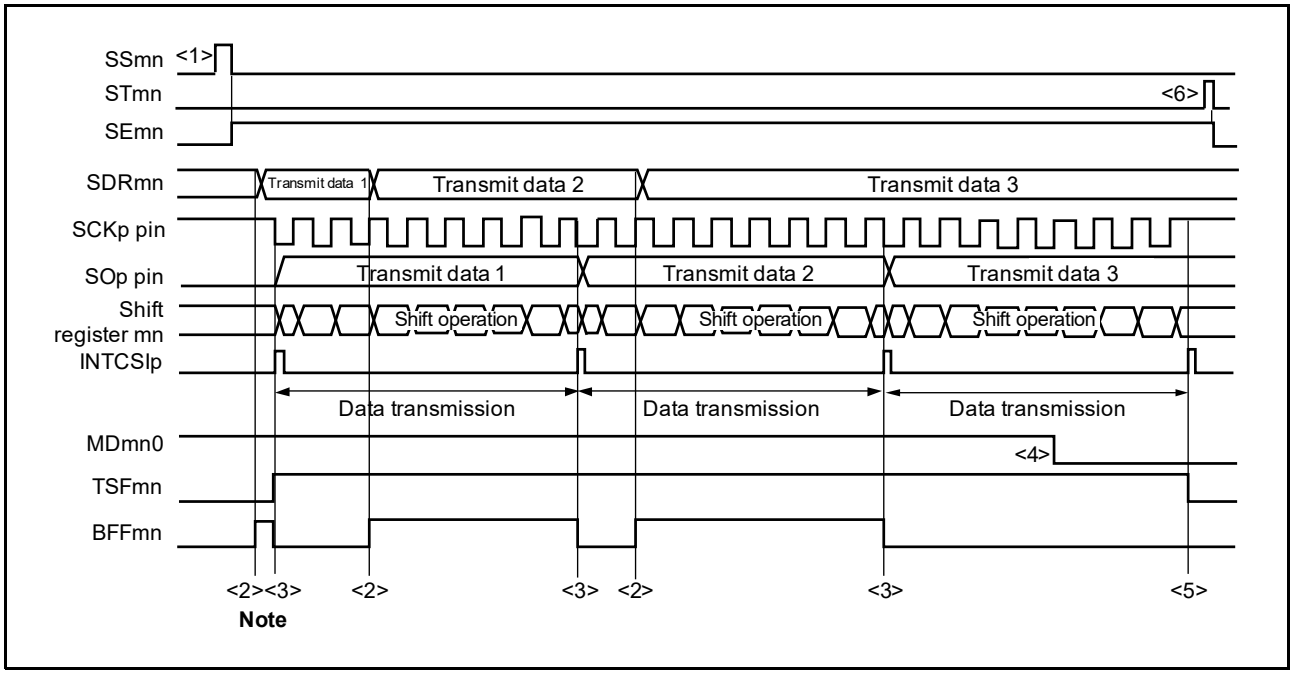
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

Figure 24 - 31 Flowchart of Master Transmission (in Single-transmission Mode)



4. Processing flow (in continuous transmission mode)

Figure 24 - 32 Timing Chart of Master Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)

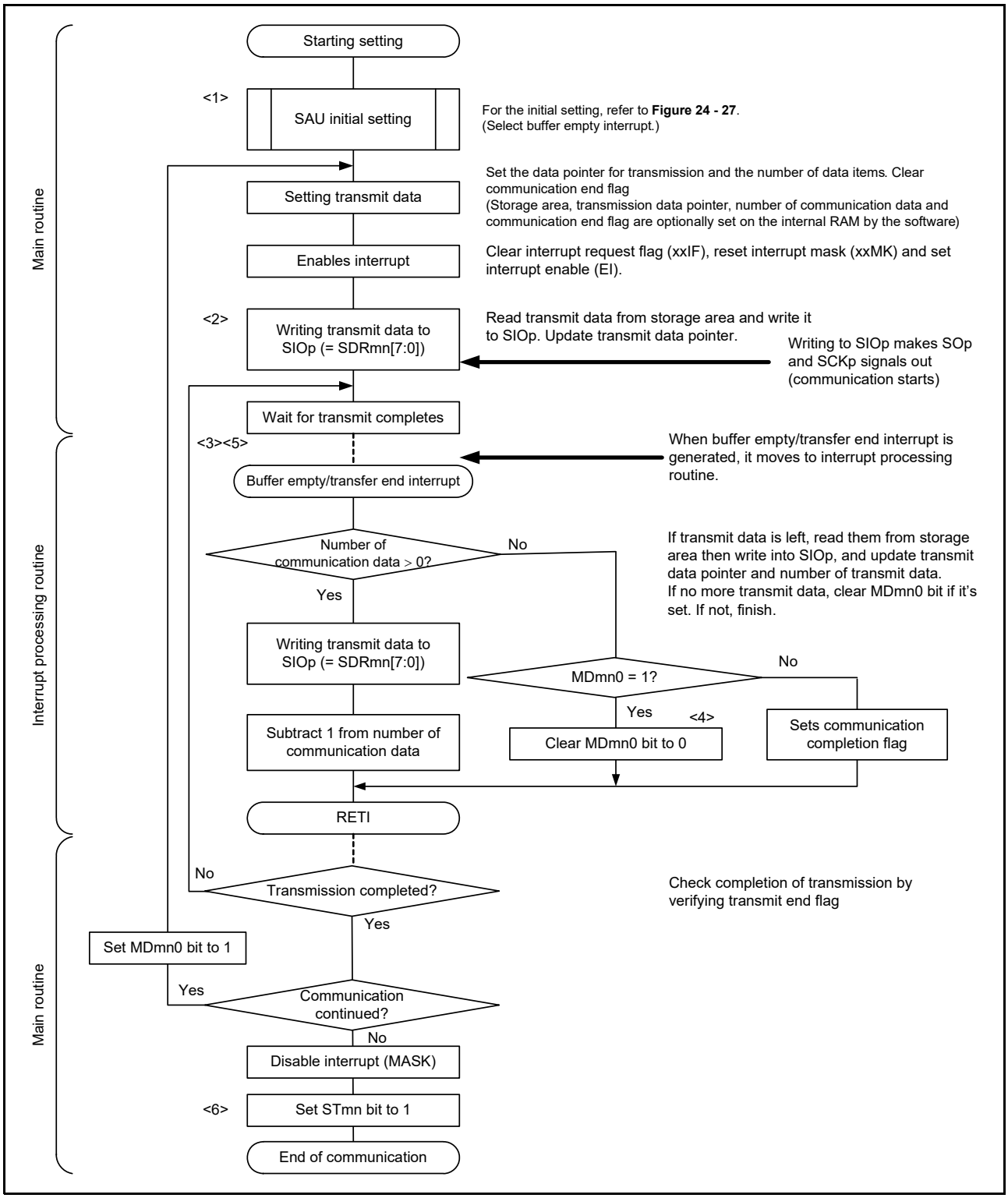


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

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 has been rewritten before the transfer end interrupt of the last transmit data.**

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

Figure 24 - 33 Flowchart of Master Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 24 - 32 Timing Chart of Master Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)**.

24.5.2 Master reception

Master reception is that the RL78 microcontroller outputs a transfer clock and receives data from another device.

Simplified SPI	CSI00	CSI01	CSI10	CSI11	CSI20	CSI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCK00, SI00	SCK01, SI01	SCK10, SI10	SCK11, SI11	SCK20, SI20	SCK21, SI21
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21
	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. $f_{CLK}/2$ [Hz] (CSI00 only), $f_{CLK}/4$ [Hz] 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 cycle 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 specified in the electrical characteristics. For details, see **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** or **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.

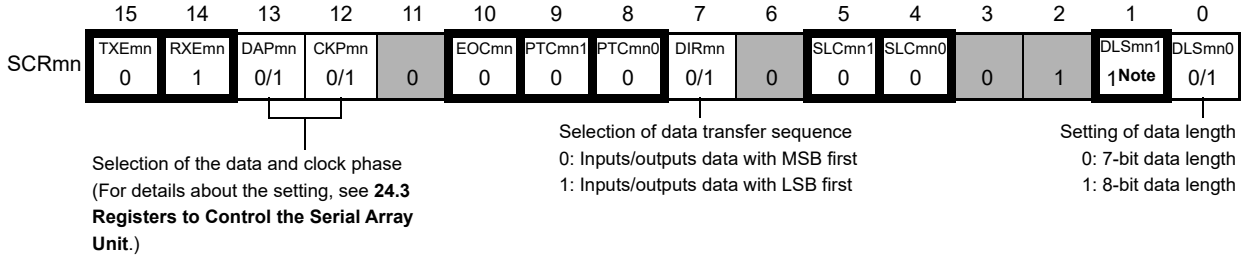
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

1. Register setting

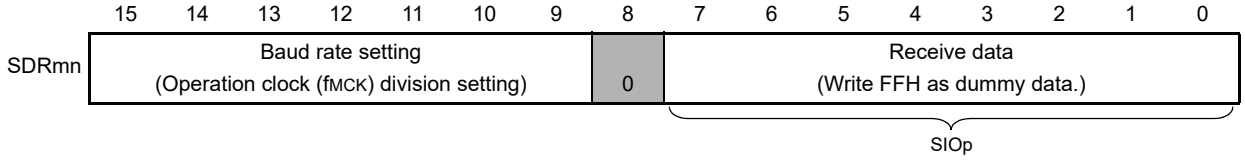
Figure 24 - 34 Example of Contents of Registers for Master Reception of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21)

a) Serial mode register mn (SMRmn)

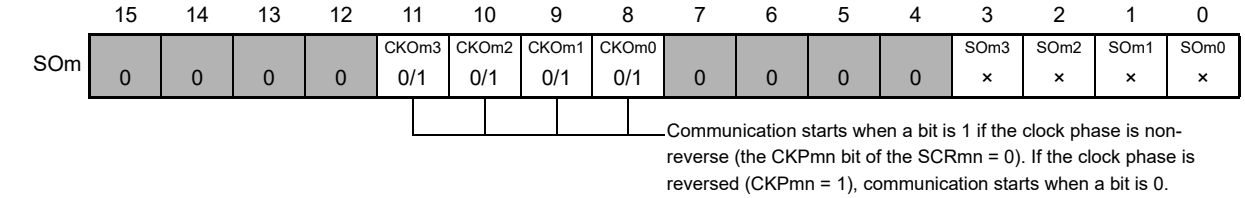
b) Serial communication operation setting register mn (SCRmn)



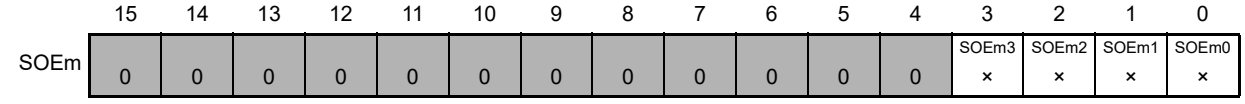
c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



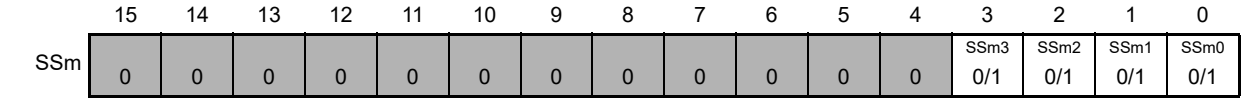
d) Serial output register m (SOm): Set only the bit of the target channel.



e) Serial output enable register m (SOEm): This register is not used in this mode.



f) Serial channel start register m (SSm): Set only the bit of the target channel to 1.



Note Only provided for the SCR00 and SCR01 registers. This bit is fixed to 1 for the other registers.

(Remarks are listed on the next page.)

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

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 24 - 35 Initial Setting Procedure for Master Reception

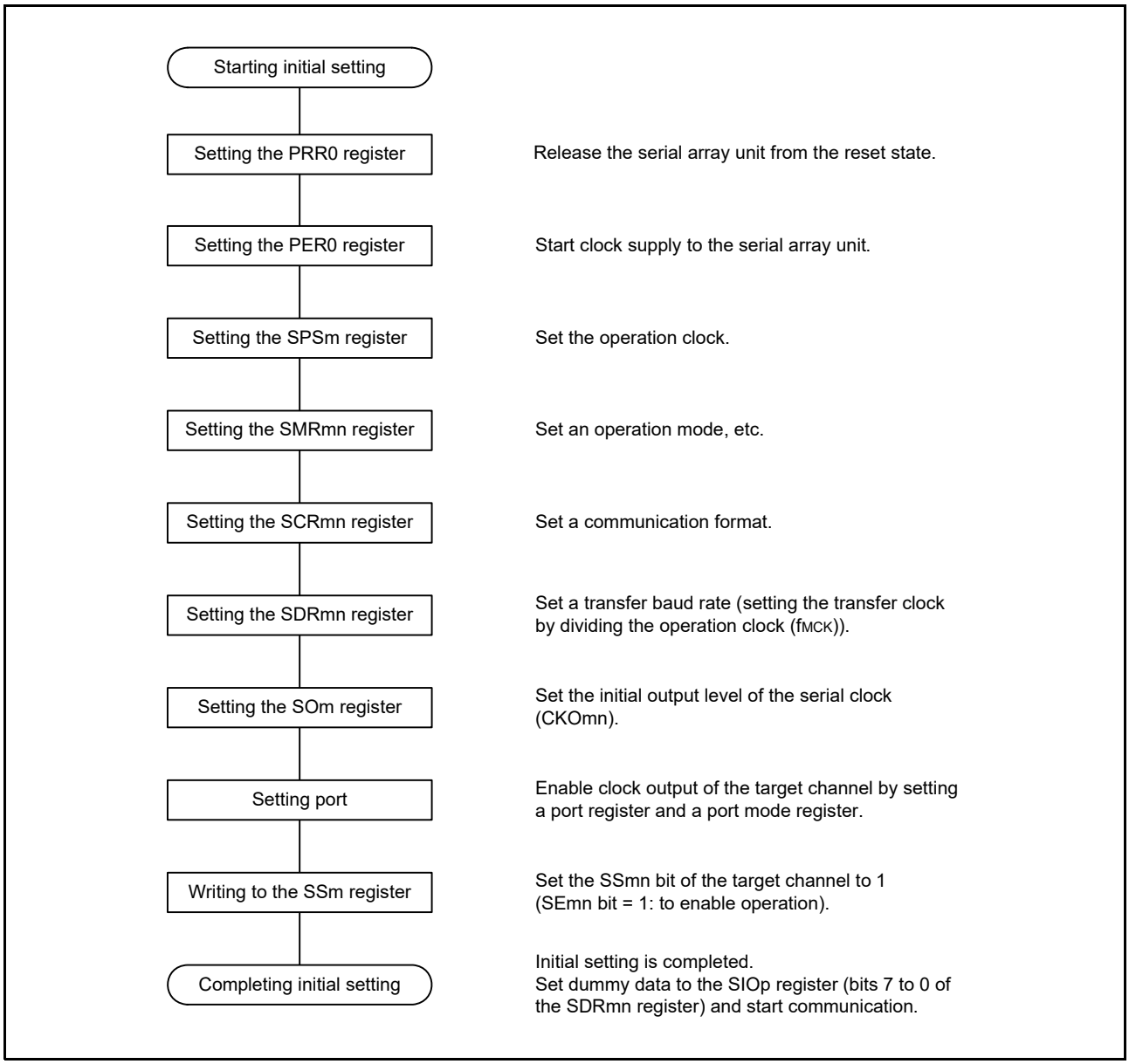


Figure 24 - 36 Procedure for Stopping Master Reception

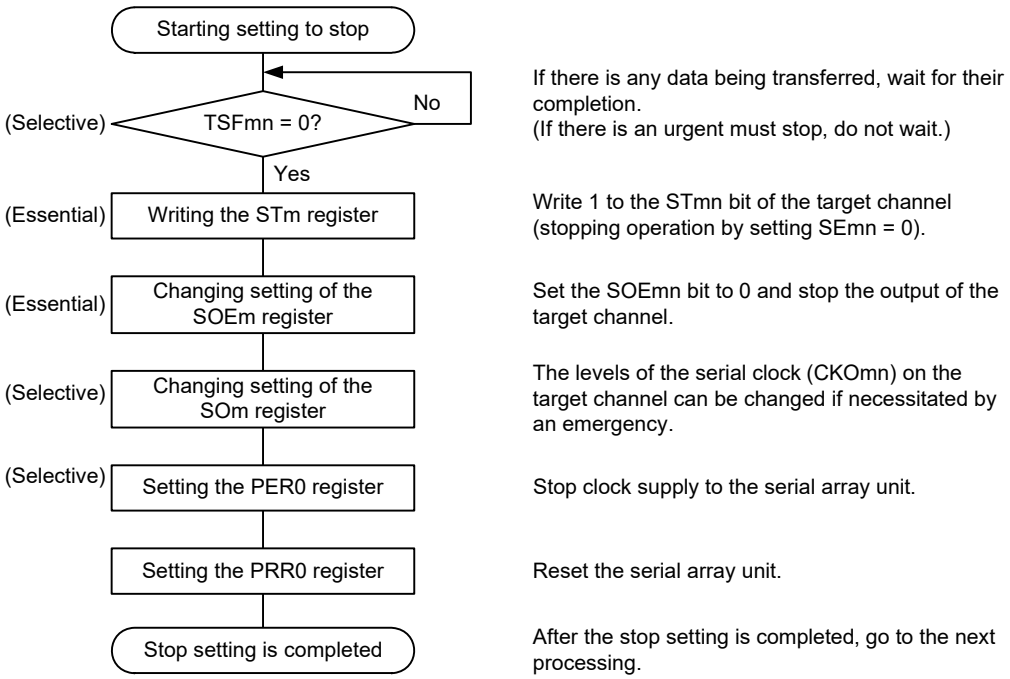
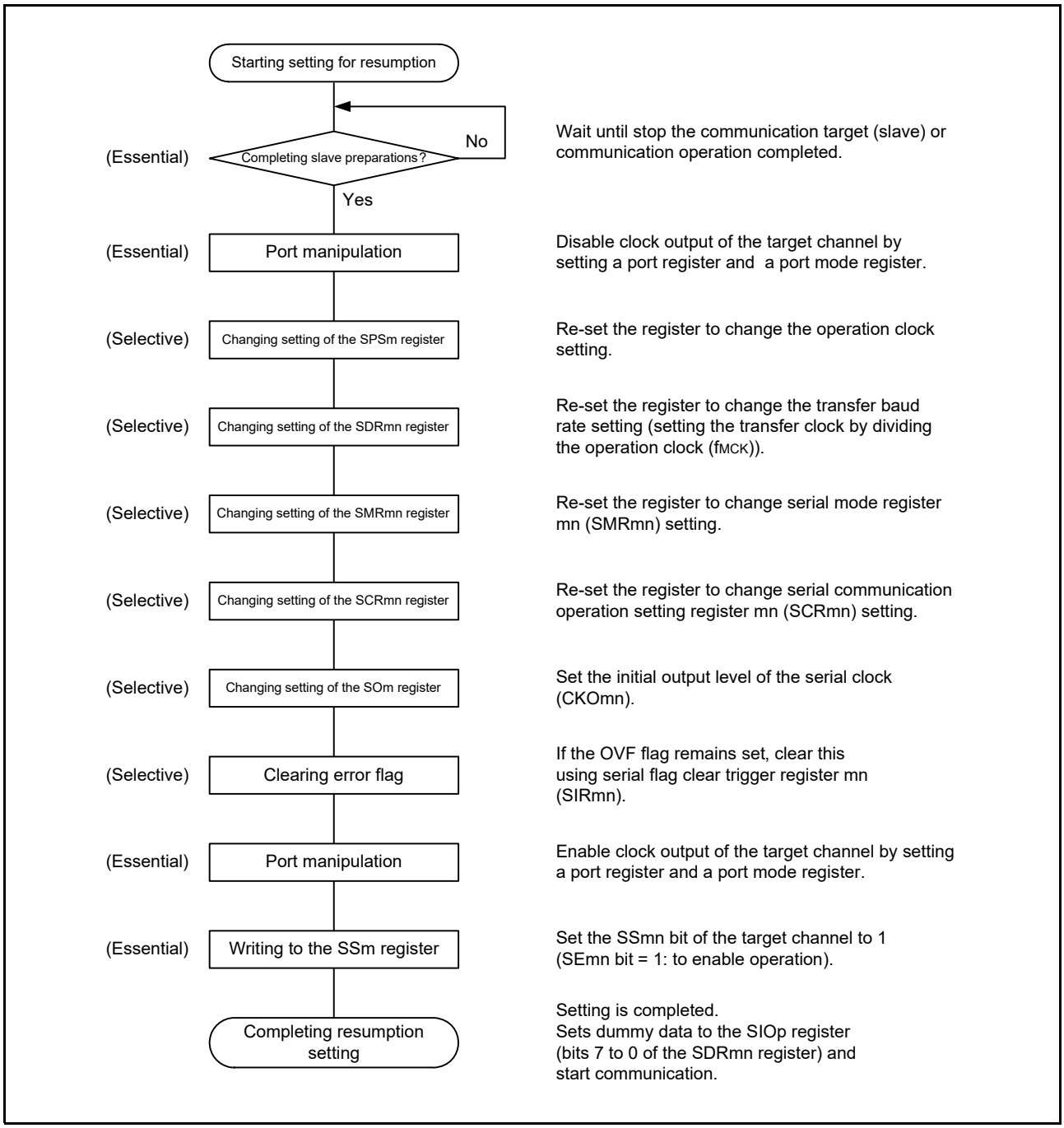


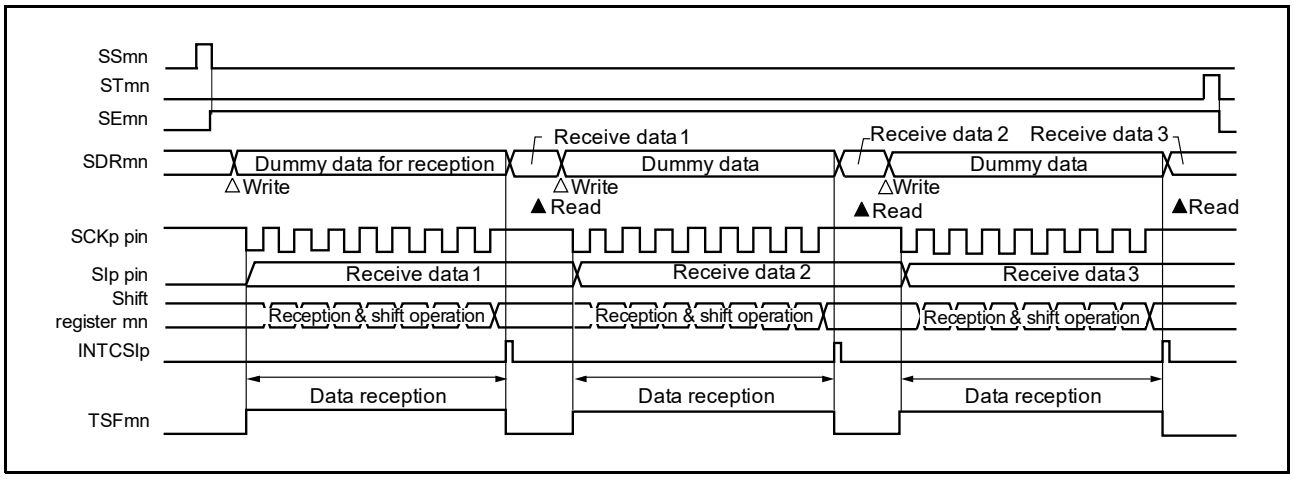
Figure 24 - 37 Procedure for Resuming Master Reception



Remark If PRR0 is rewritten while stopping the communication to reset the serial array unit, wait until the communication target (slave) stops or communication finishes, and then perform initialization instead of restarting the communication.

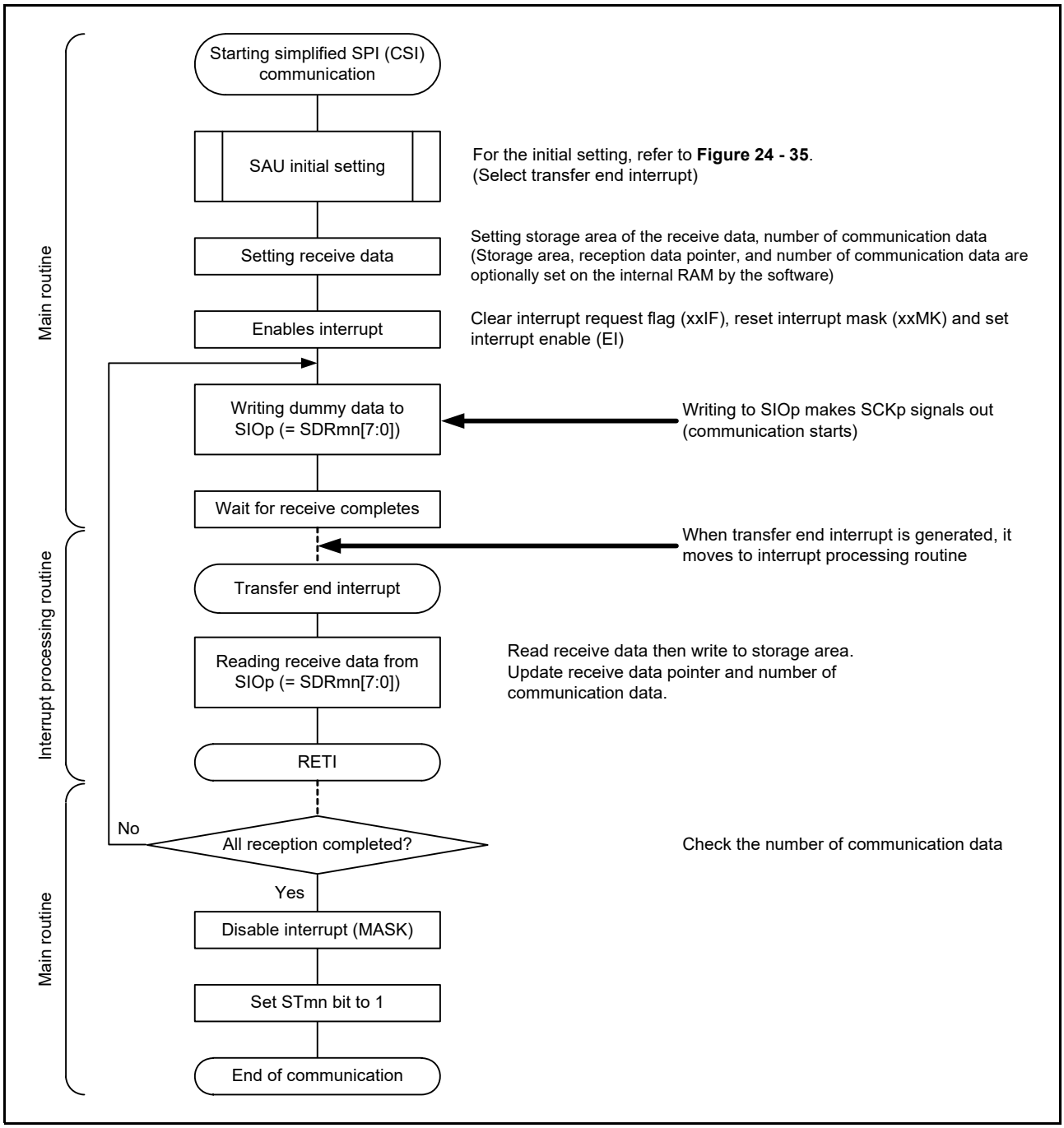
3. Processing flow (in single-reception mode)

Figure 24 - 38 Timing Chart of Master Reception (in Single-reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



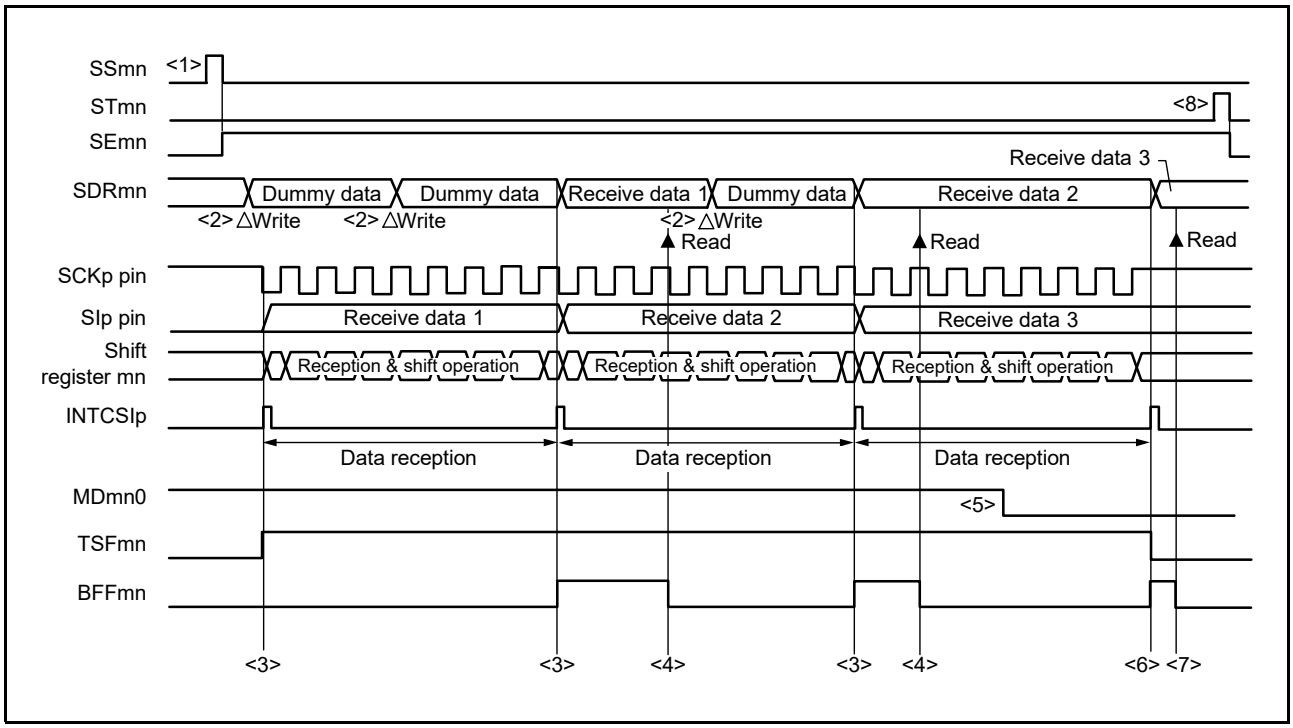
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

Figure 24 - 39 Flowchart of Master Reception (in Single-reception Mode)



4. Processing flow (in continuous reception mode)

Figure 24 - 40 Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)

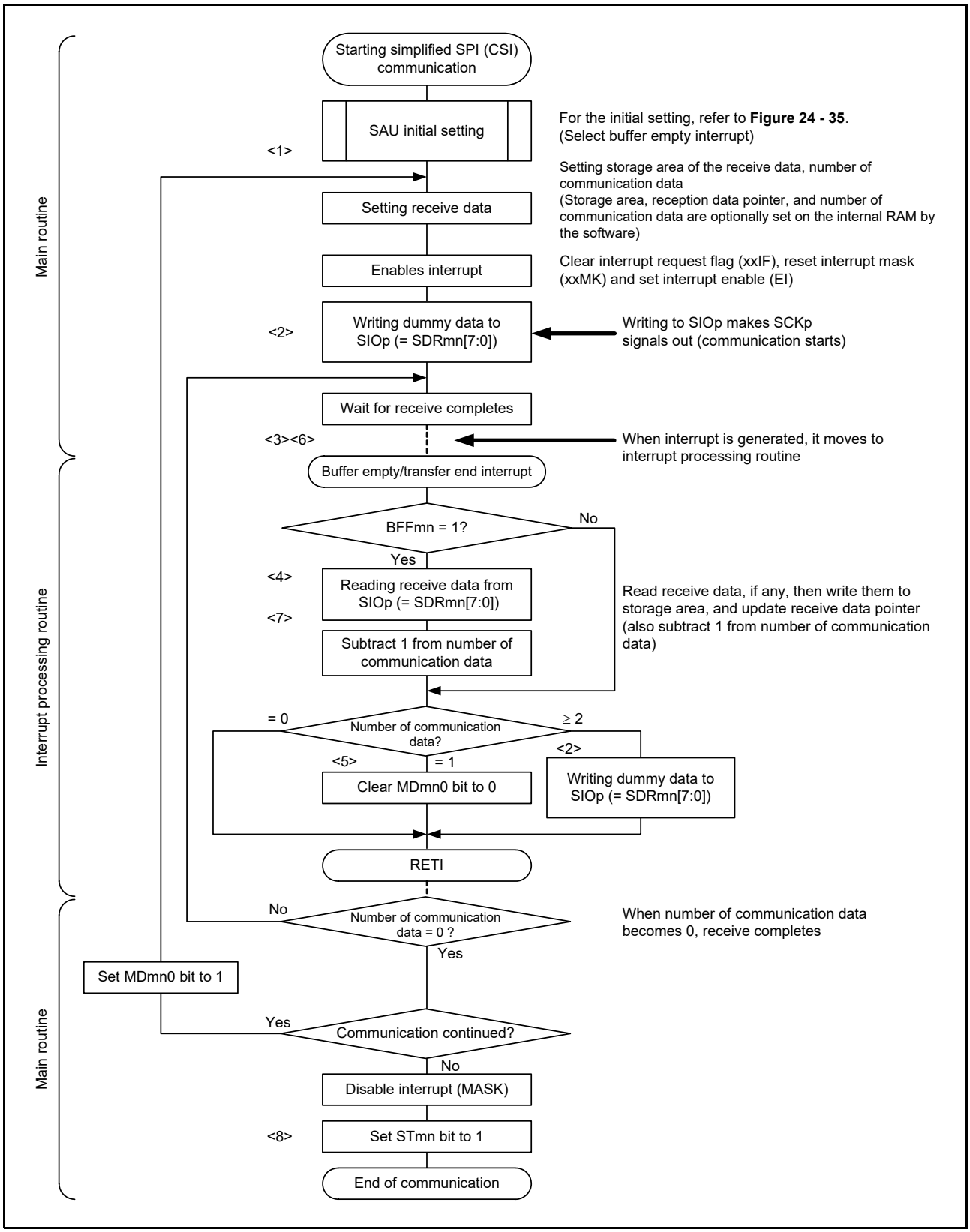


Caution **The MDmn0 bit can be rewritten even during operation.**
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 24 - 41 Flowchart of Master Reception (in Continuous Reception Mode)**.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

Figure 24 - 41 Flowchart of Master Reception (in Continuous Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 24 - 40 Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)**.

24.5.3 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	CSI01	CSI10	CSI11	CSI20	CSI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK10, SI10, SO10	SCK11, SI11, SO11	SCK20, SI20, SO20	SCK21, SI21, SO21
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21
	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. $f_{CLK}/2$ [Hz] (CSI00 only), $f_{CLK}/4$ [Hz] 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 cycle 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 specified in the electrical characteristics. For details, see **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** or **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.

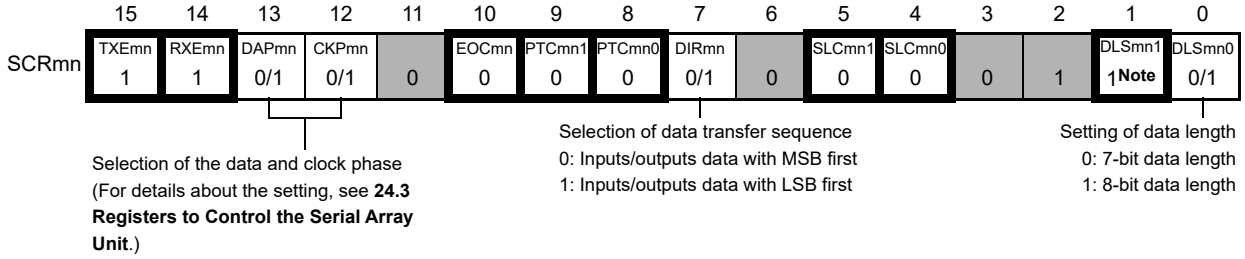
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

1. Register setting

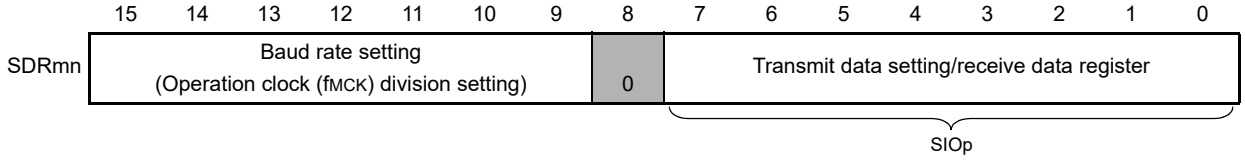
Figure 24 - 42 Example of Contents of Registers for Master Transmission/Reception of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21)

a) Serial mode register mn (SMRmn)

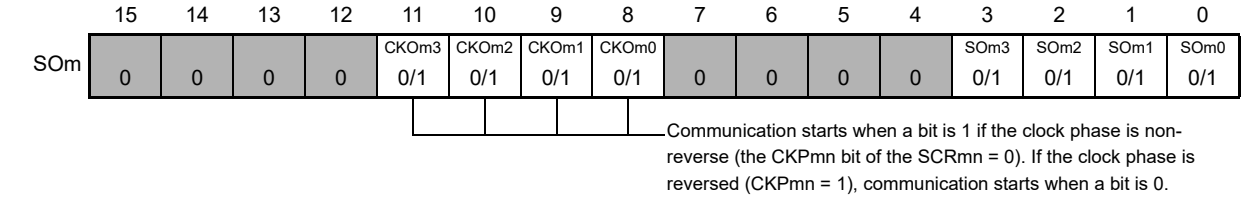
b) Serial communication operation setting register mn (SCRmn)



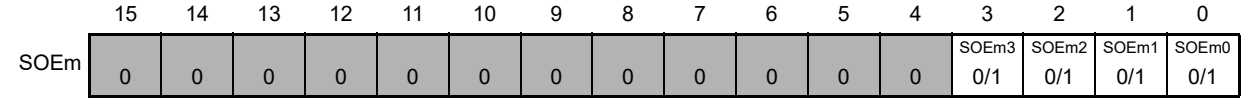
c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



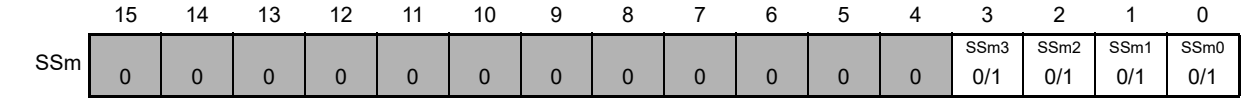
d) Serial output register m (SOm): Set only the bit of the target channel.



e) Serial output enable register m (SOEm): Set only the bit of the target channel to 1.



f) Serial channel start register m (SSm): Set only the bit of the target channel to 1.



Note Only provided for the SCR00 and SCR01 registers. This bit is fixed to 1 for the other registers.

(Remarks are listed on the next page.)

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

Remark 2. ■: Setting is fixed in the simplified SPI (CSI) master transmission/reception mode

■: Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

2. Operation procedure

Figure 24 - 43 Initial Setting Procedure for Master Transmission/Reception

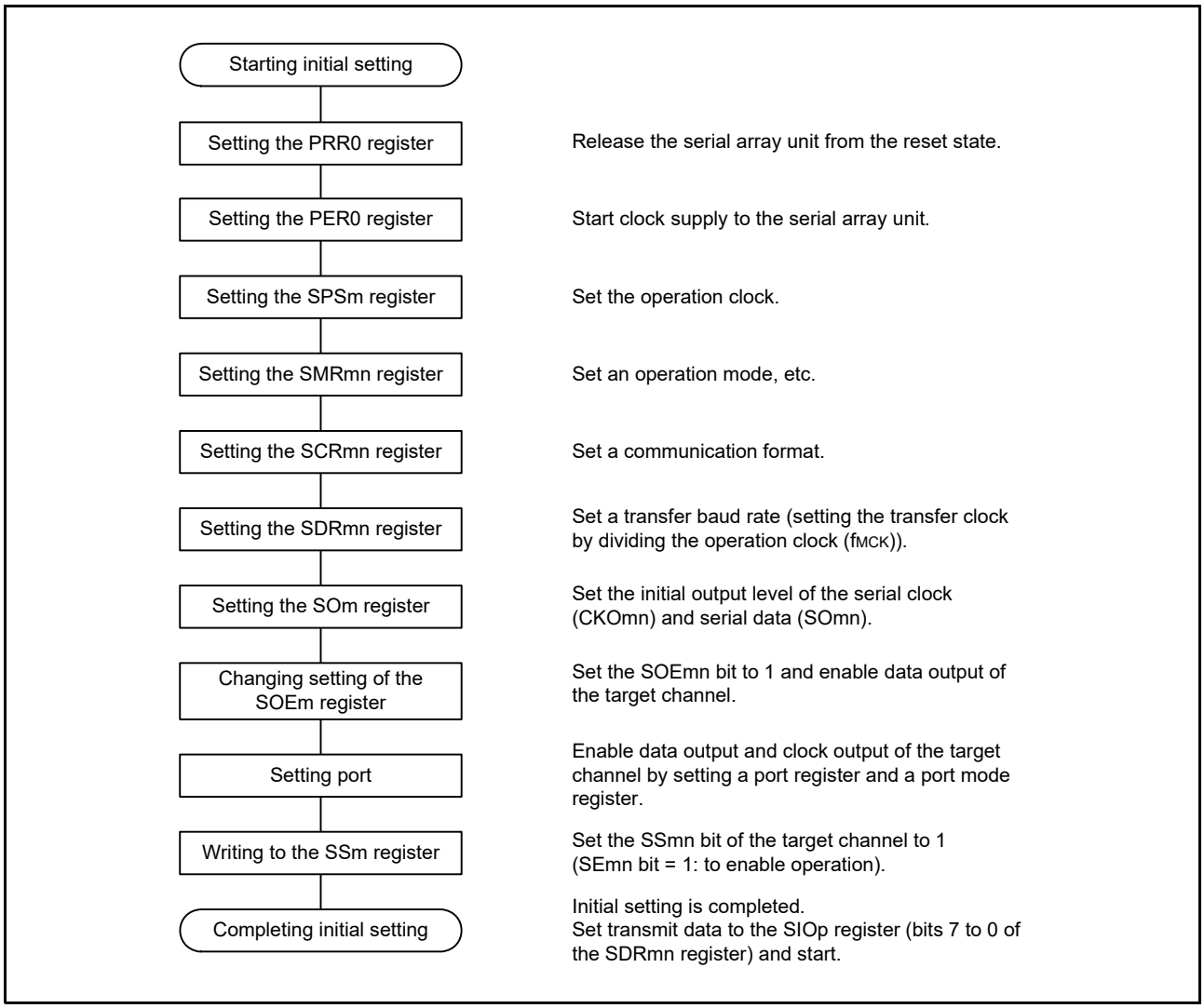


Figure 24 - 44 Procedure for Stopping Master Transmission/Reception

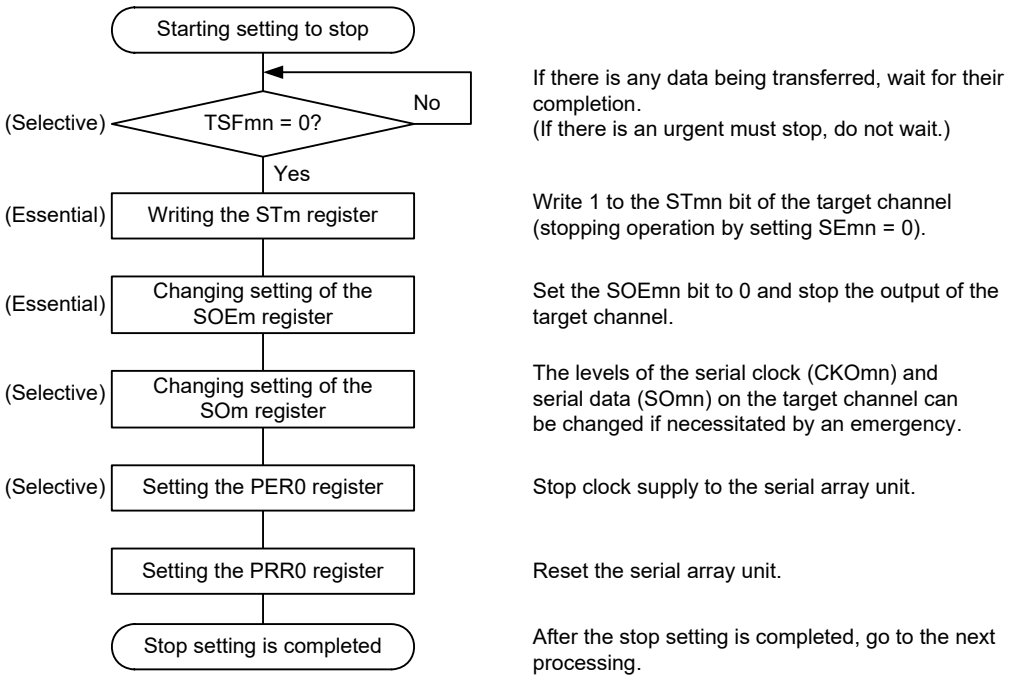
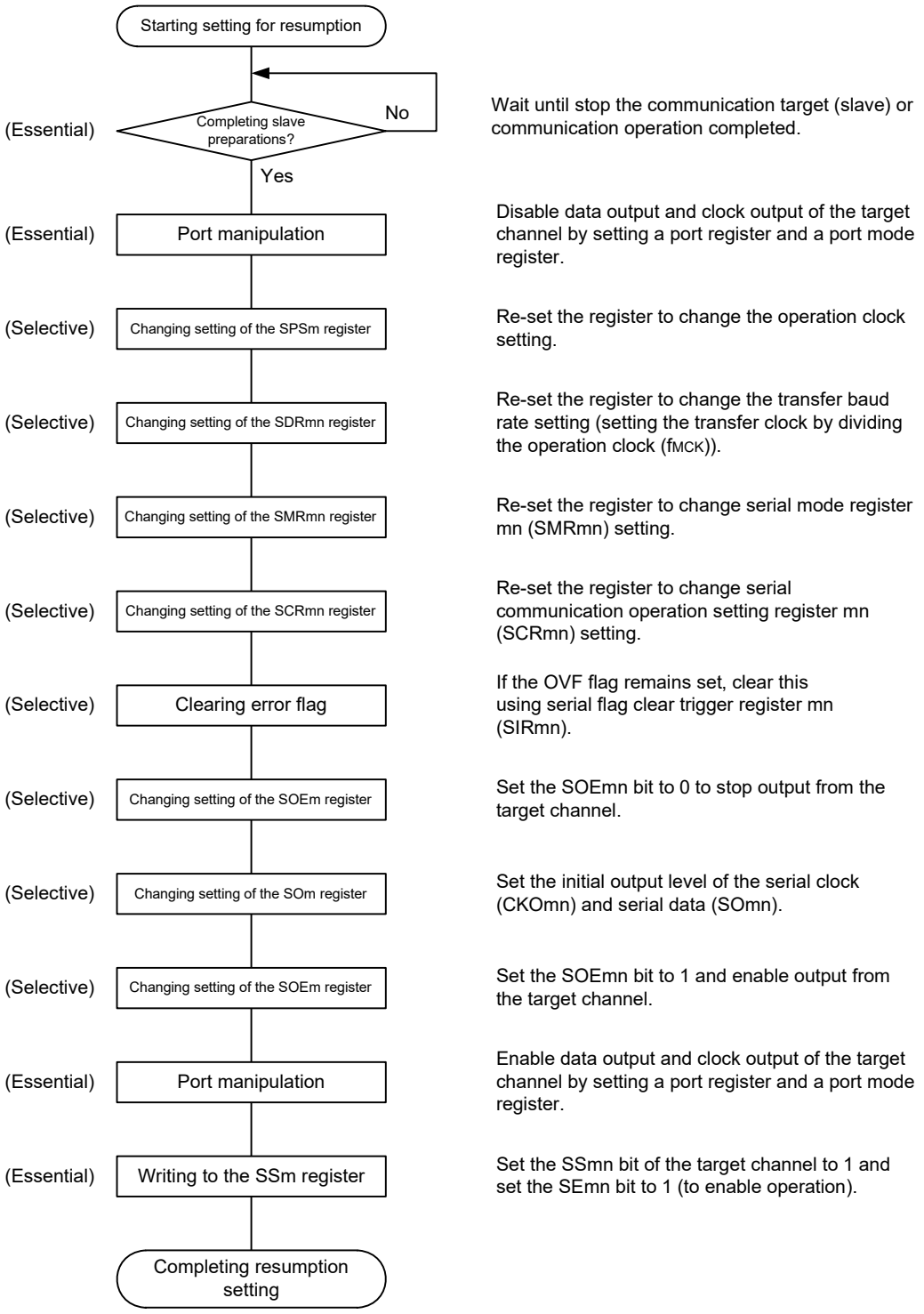
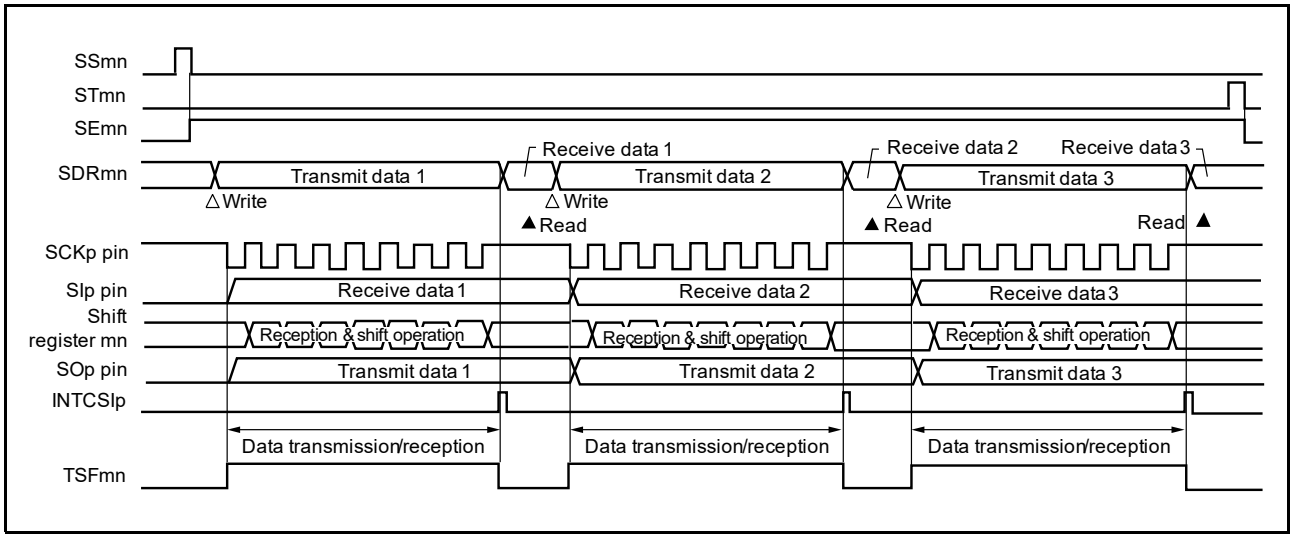


Figure 24 - 45 Procedure for Resuming Master Transmission/Reception



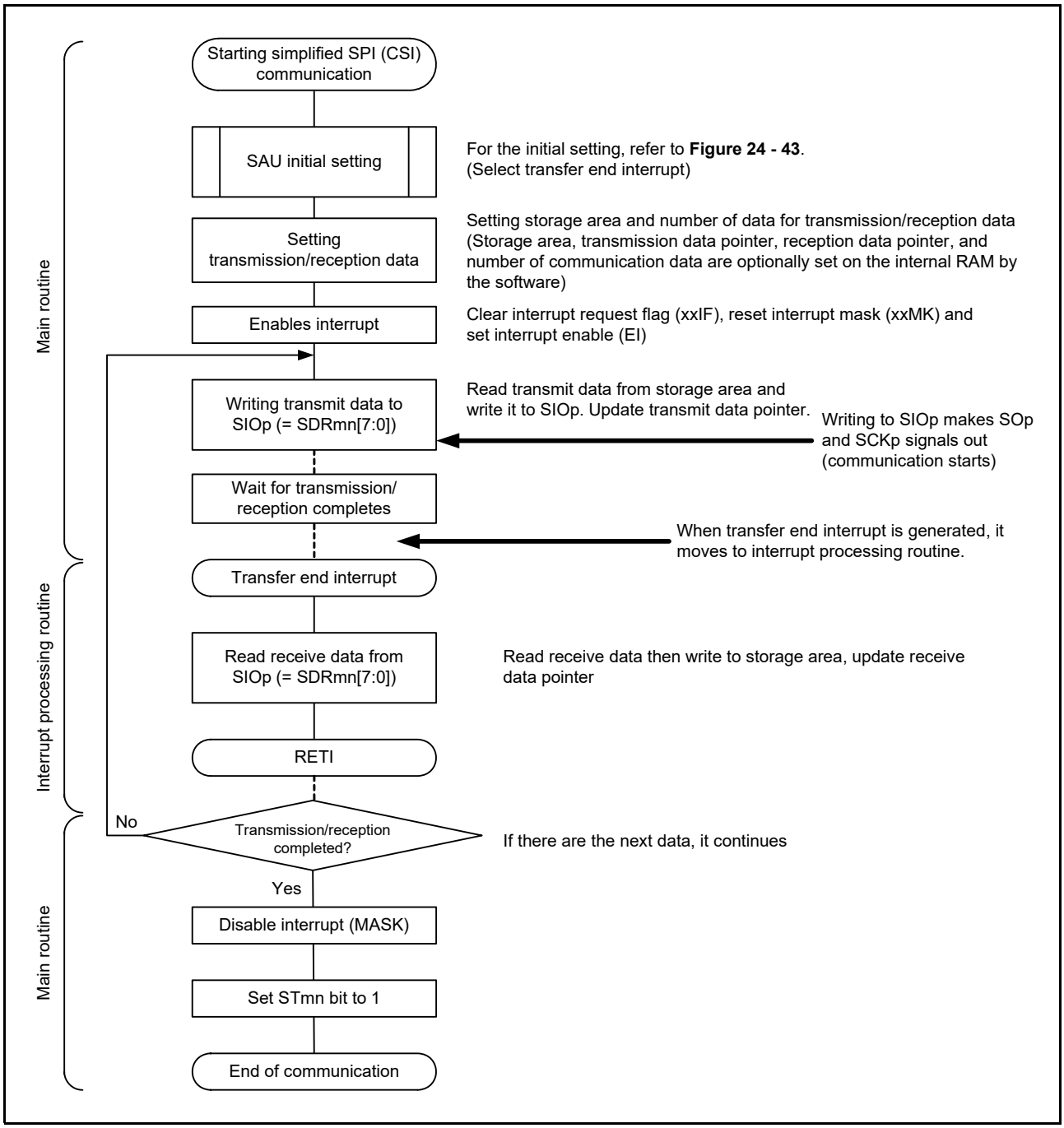
3. Processing flow (in single-transmission/reception mode)

Figure 24 - 46 Timing Chart of Master Transmission/Reception (in Single-transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

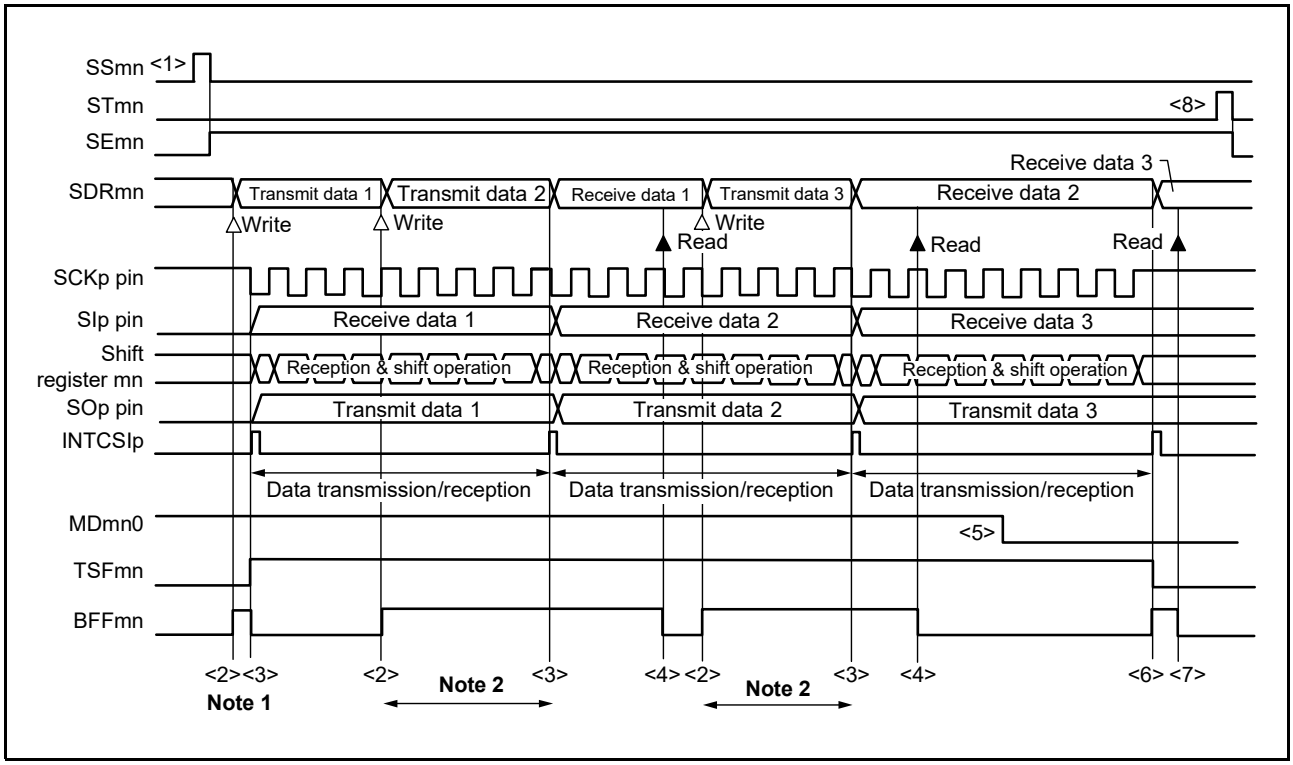
Figure 24 - 47 Flowchart of Master Transmission/Reception (in Single-transmission/Reception Mode)



4. Processing flow (in continuous transmission/reception mode)

Figure 24 - 48 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)

<R>



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.

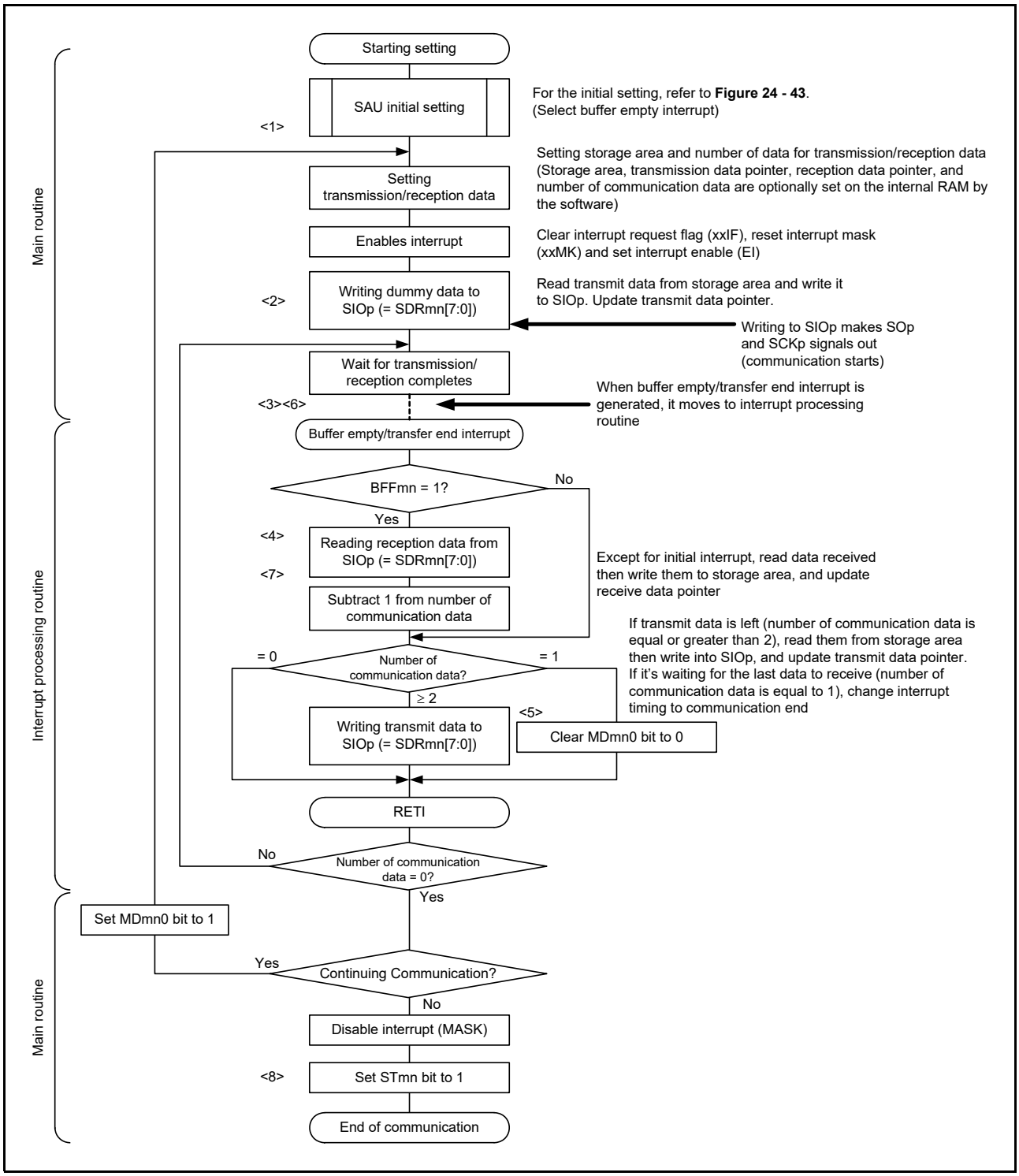
Note 2. The transmit data can be read by reading the SDRmn register during this period. Reading this register does not affect the transfer operation.

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 24 - 49 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

Figure 24 - 49 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 24 - 48 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)**.

24.5.4 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	CSI01	CSI10	CSI11	CSI20	CSI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCK00, SO00	SCK01, SO01	SCK10, SO10	SCK11, SO11	SCK20, SO20	SCK21, SO21
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21
	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 cycle 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, SCK01, SCK10, SCK11, SCK20, and SCK21 pins 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 specified in the electrical characteristics. For details, see **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** or **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.

Remark 1. f_{MCK}: Operation clock frequency of target channel
f_{SCK}: Serial clock frequency

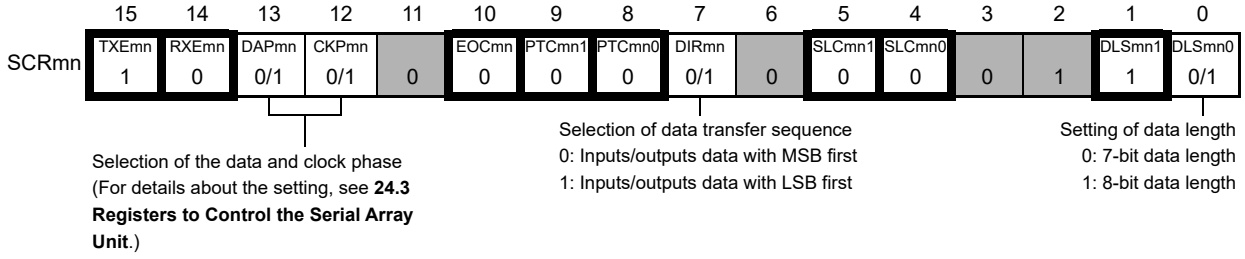
Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

1. Register setting

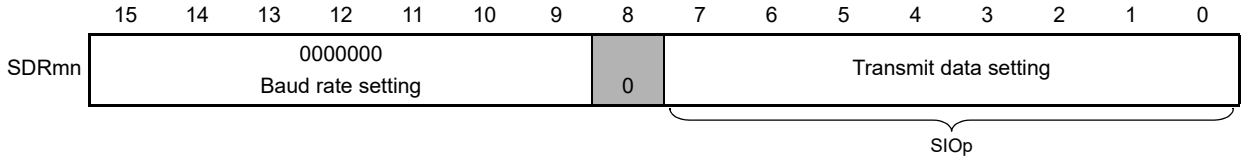
Figure 24 - 50 Example of Contents of Registers for Slave Transmission of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21)

a) Serial mode register mn (SMRmn)

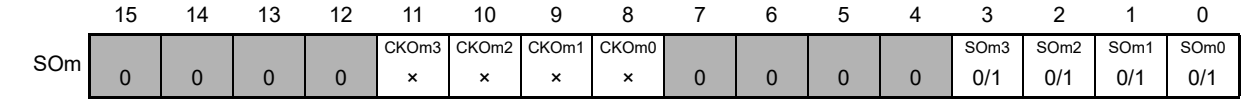
b) Serial communication operation setting register mn (SCRmn)



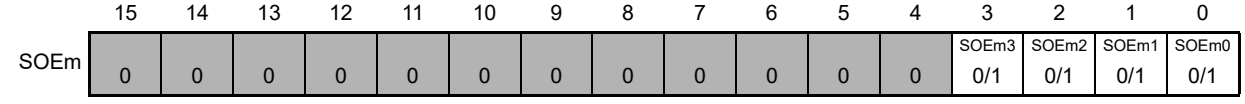
c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



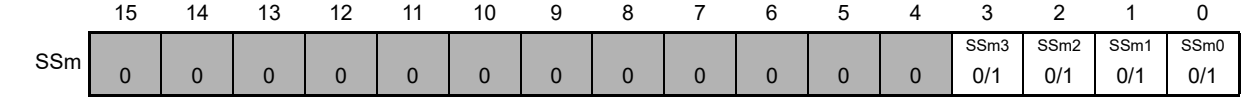
d) Serial output register m (SOm): Set only the bit of the target channel.



e) Serial output enable register m (SOEm): Set only the bit of the target channel to 1.



f) Serial channel start register m (SSm): Set only the bit of the target channel to 1.



(Remarks are listed on the next page.)

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

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 24 - 51 Initial Setting Procedure for Slave Transmission

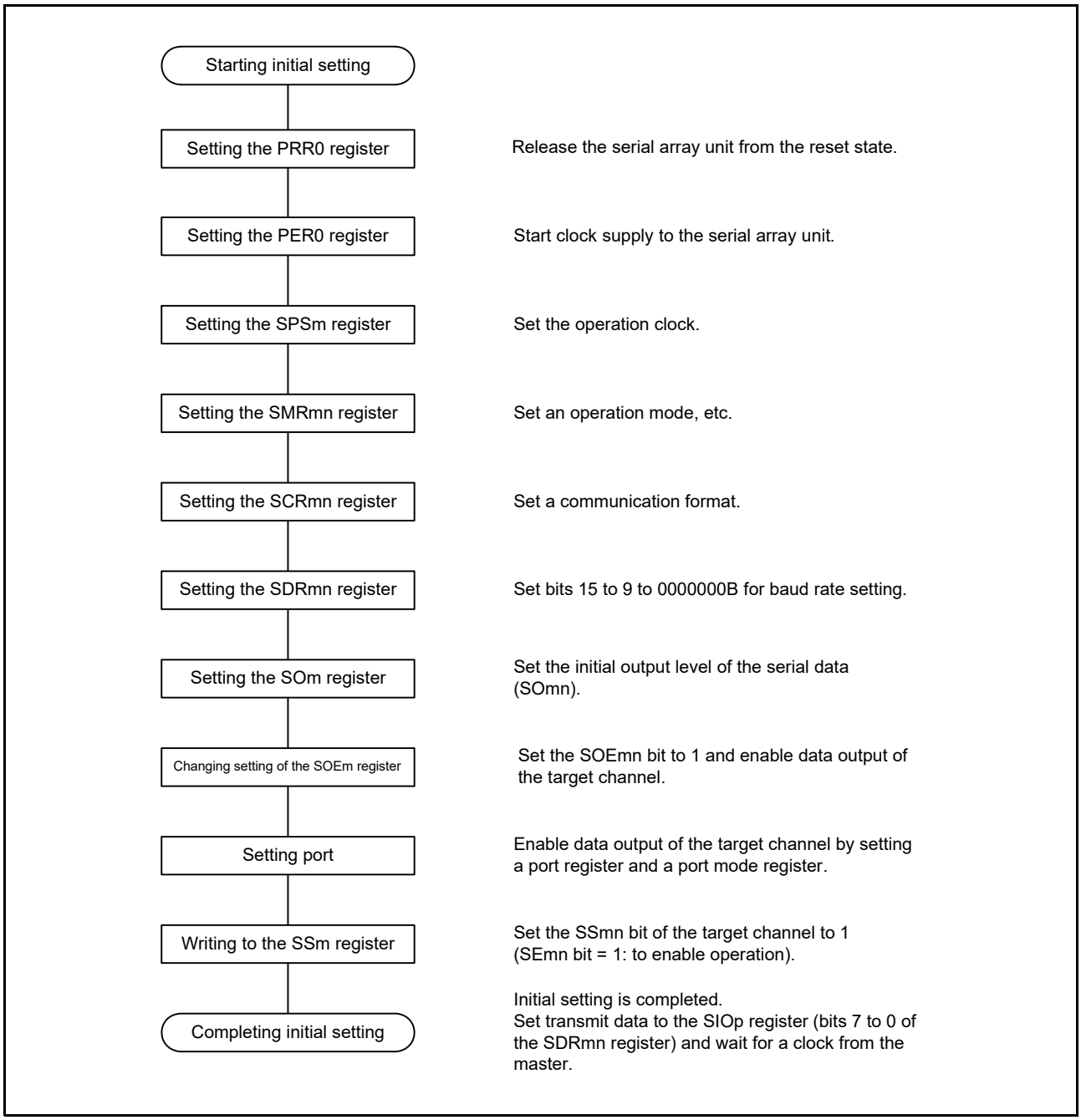


Figure 24 - 52 Procedure for Stopping Slave Transmission

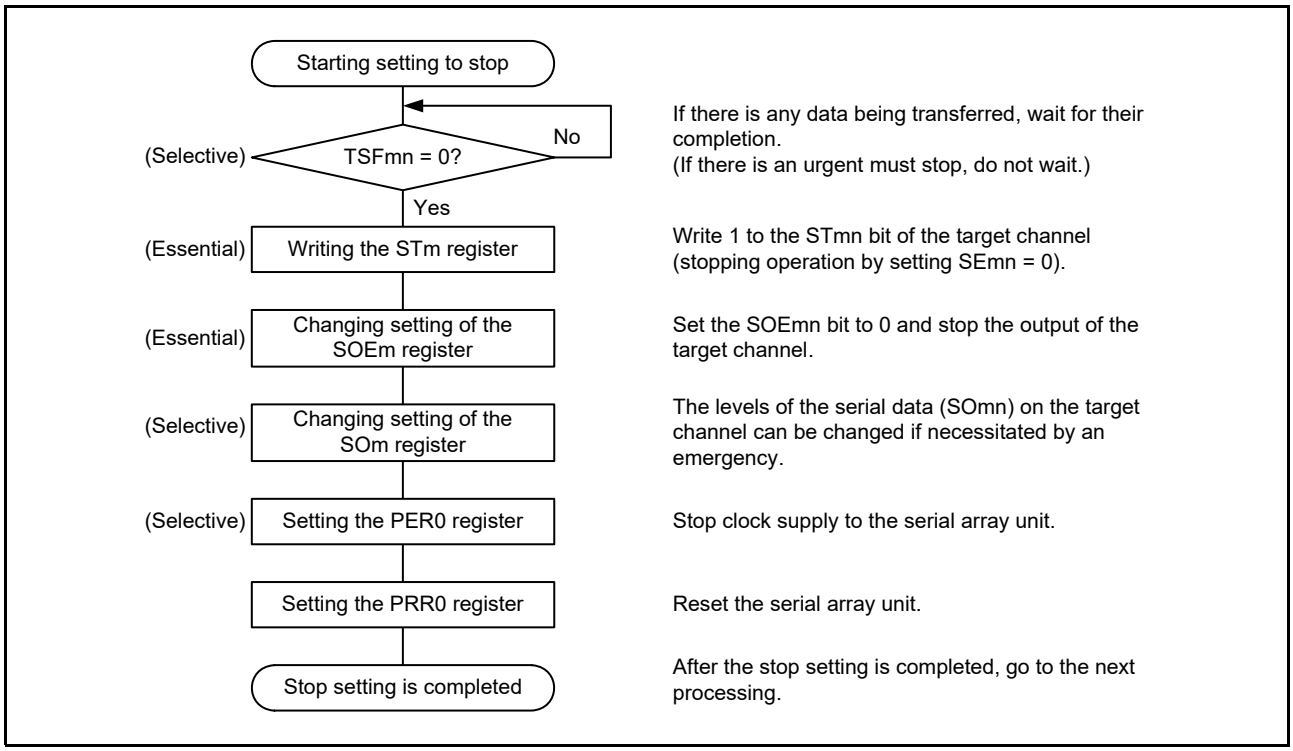
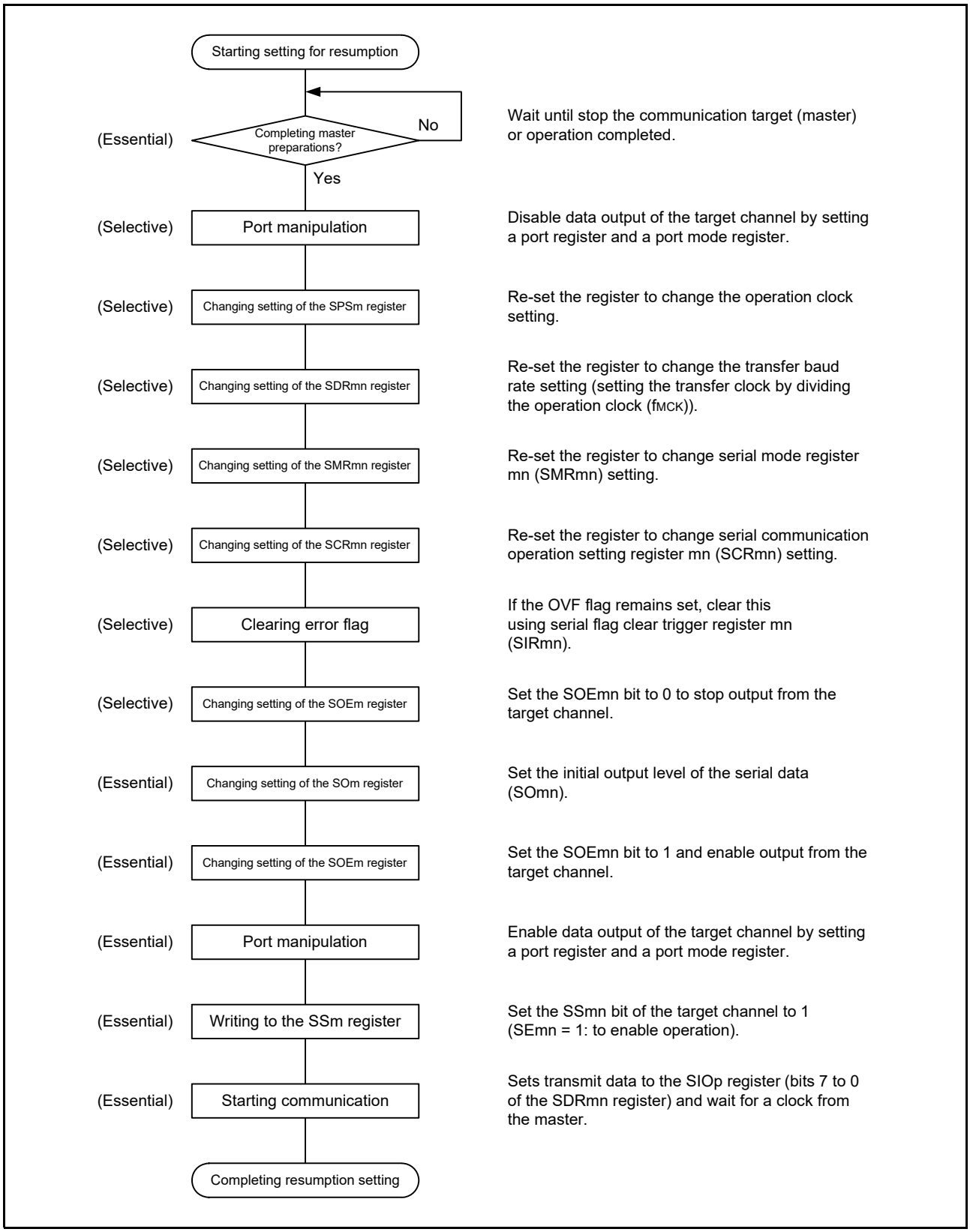


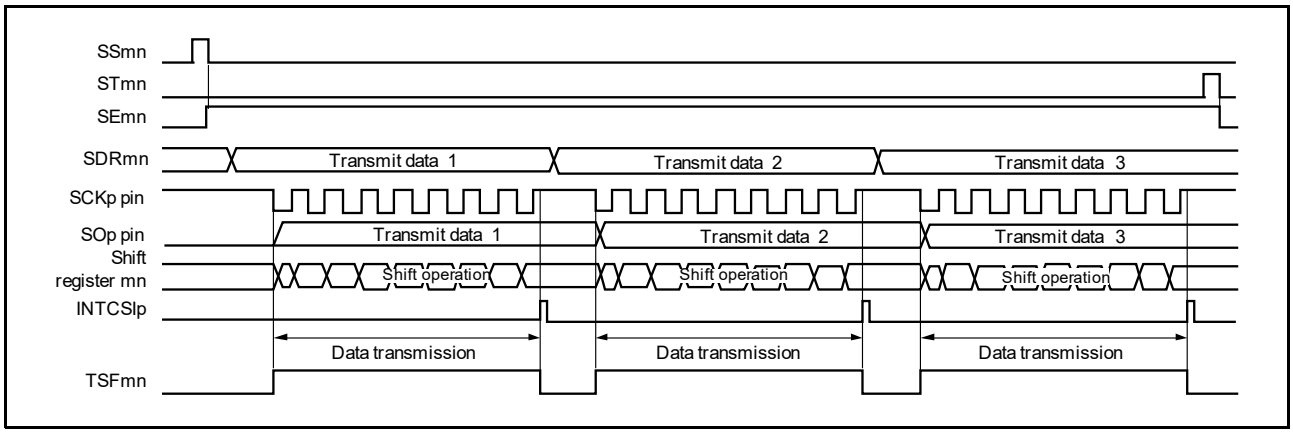
Figure 24 - 53 Procedure for Resuming Slave Transmission



Remark If PRR0 is rewritten while stopping the communication to reset the serial array unit, wait until the communication target (master) stops or communication finishes, and then perform initialization instead of restarting the communication.

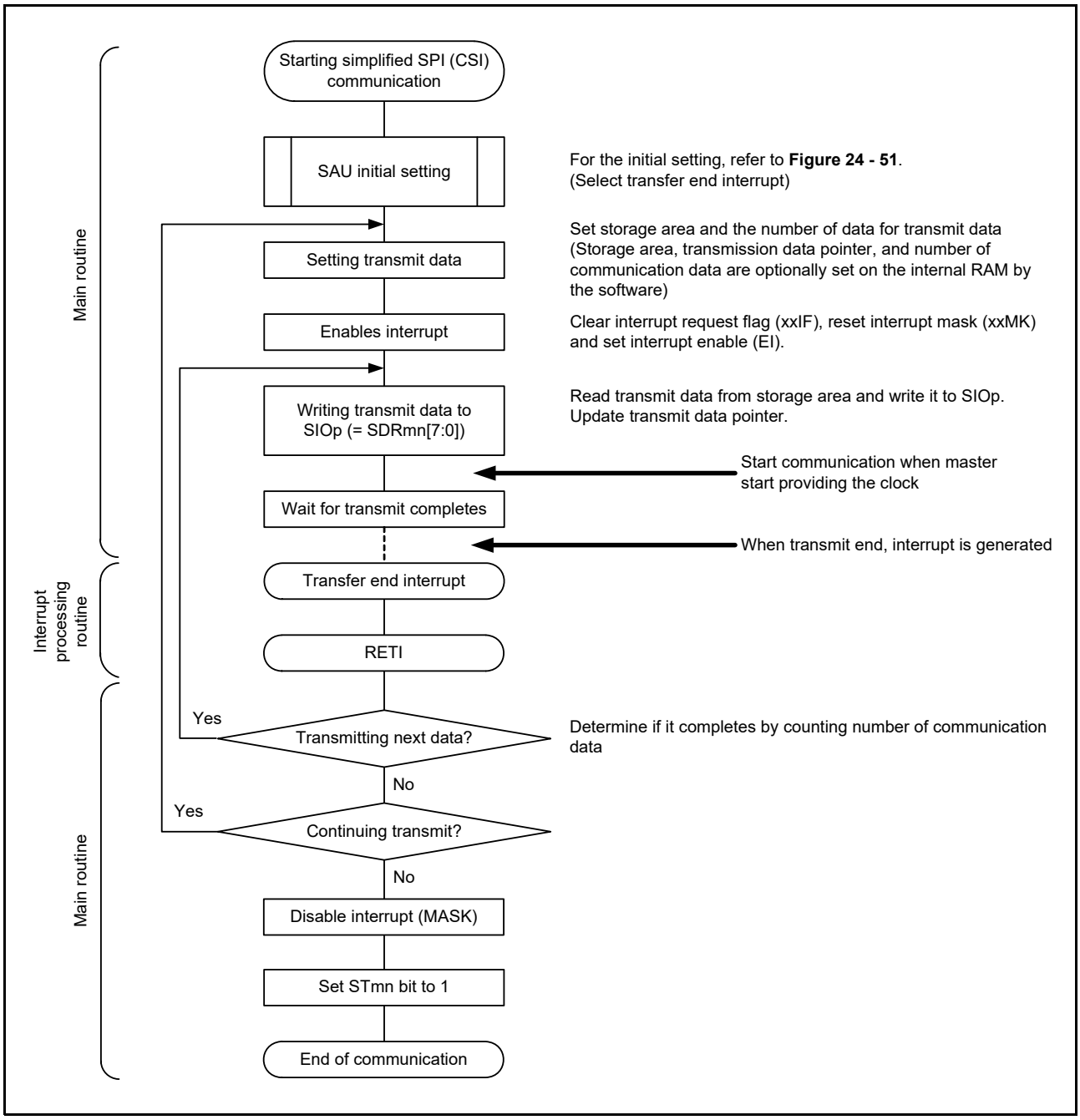
3. Processing flow (in single-transmission mode)

Figure 24 - 54 Timing Chart of Slave Transmission (in Single-transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)



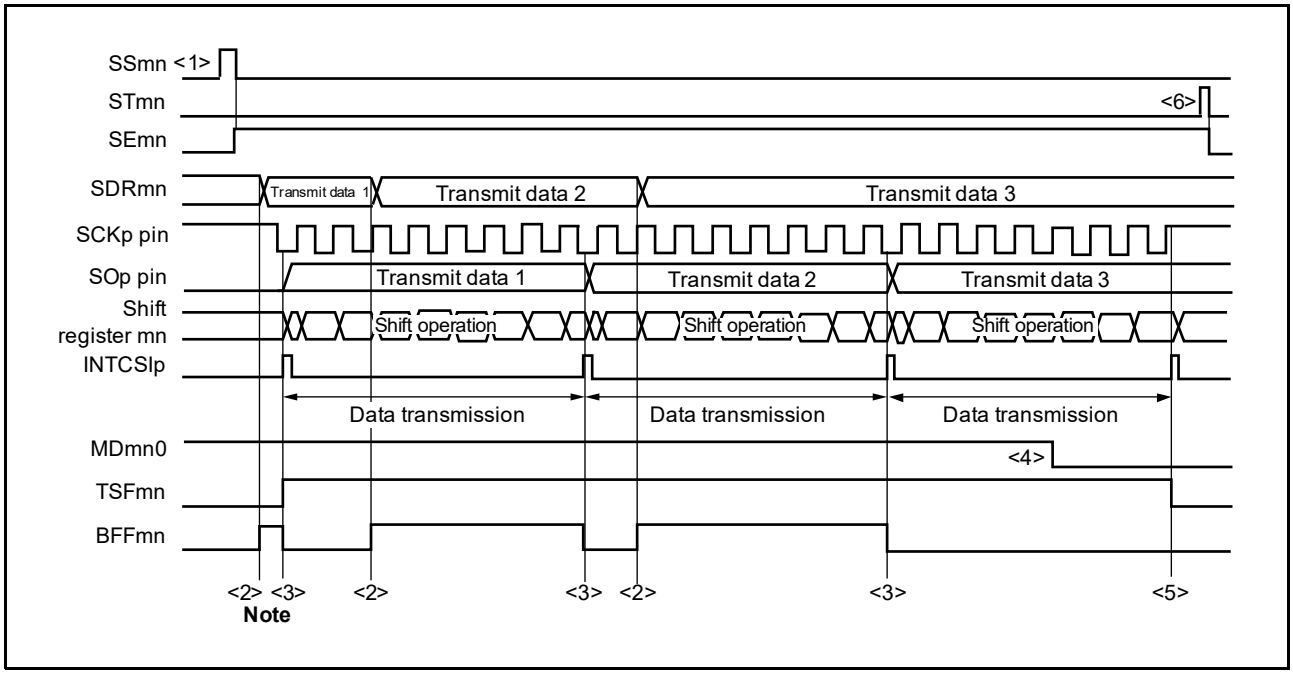
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

Figure 24 - 55 Flowchart of Slave Transmission (in Single-transmission Mode)



4. Processing flow (in continuous transmission mode)

Figure 24 - 56 Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)

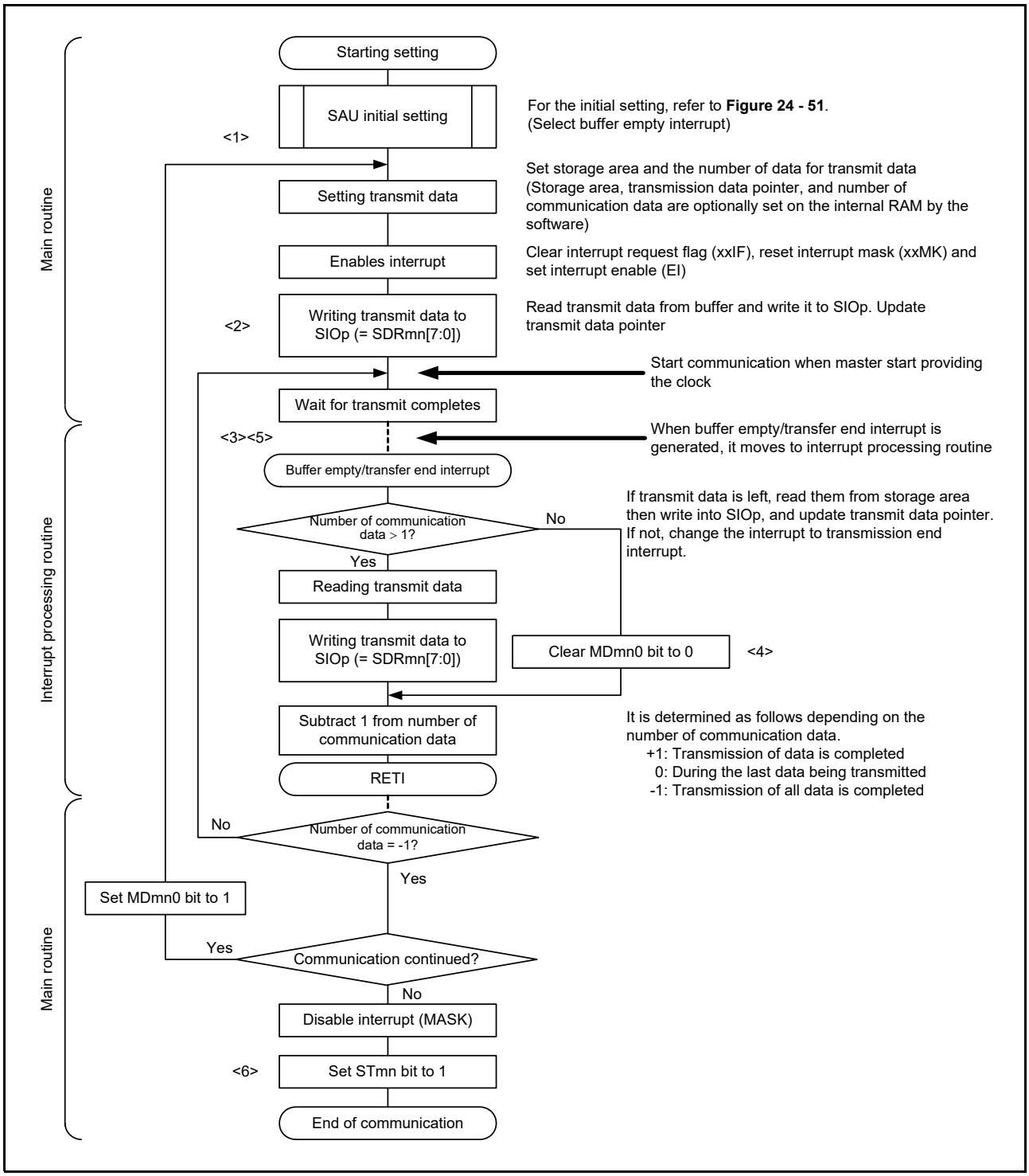


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

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, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

Figure 24 - 57 Flowchart of Slave Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 24 - 56 Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)**.

24.5.5 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	CSI01	CSI10	CSI11	CSI20	CSI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCK00, SI00	SCK01, SI01	SCK10, SI10	SCK11, SI11	SCK20, SI20	SCK21, SI21
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21
	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 cycle 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, SCK01, SCK10, SCK11, SCK20, and SCK21 pins 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 specified in the electrical characteristics. For details, see **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** or **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.

Remark 1. f_{MCK}: Operation clock frequency of target channel
f_{SCK}: Serial clock frequency

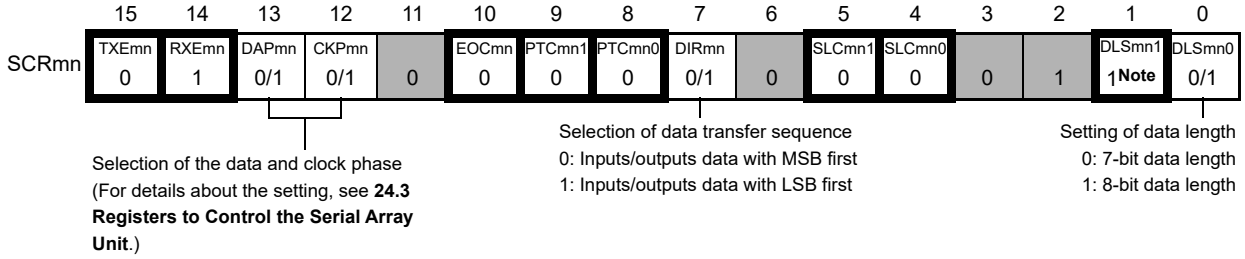
Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

1. Register setting

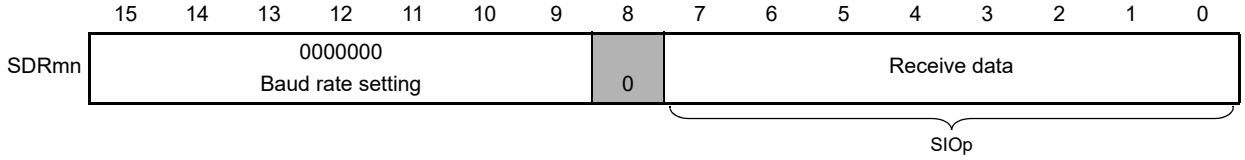
Figure 24 - 58 Example of Contents of Registers for Slave Reception of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21)

a) Serial mode register mn (SMRmn)

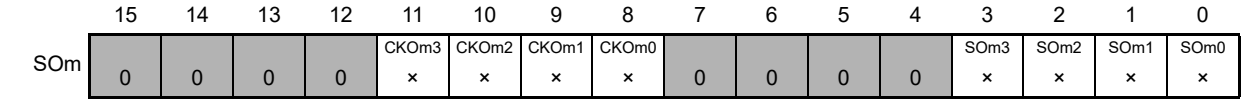
b) Serial communication operation setting register mn (SCRmn)



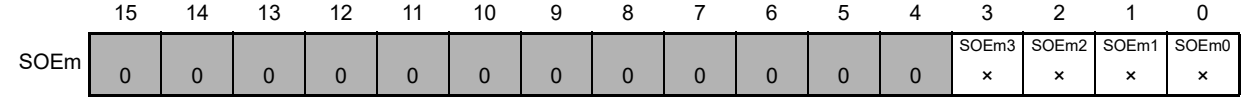
c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



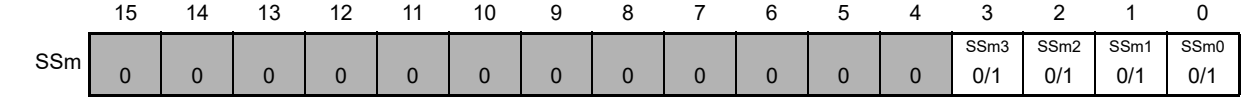
d) Serial output register m (SOm): This register is not used in this mode.



e) Serial output enable register m (SOEm): This register is not used in this mode.



f) Serial channel start register m (SSm): Set only the bit of the target channel to 1.



Note Only provided for the SCR00 and SCR01 registers. This bit is fixed to 1 for the other registers.

(Remarks are listed on the next page.)

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

Remark 2. ■: Setting is fixed in the 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 24 - 59 Initial Setting Procedure for Slave Reception

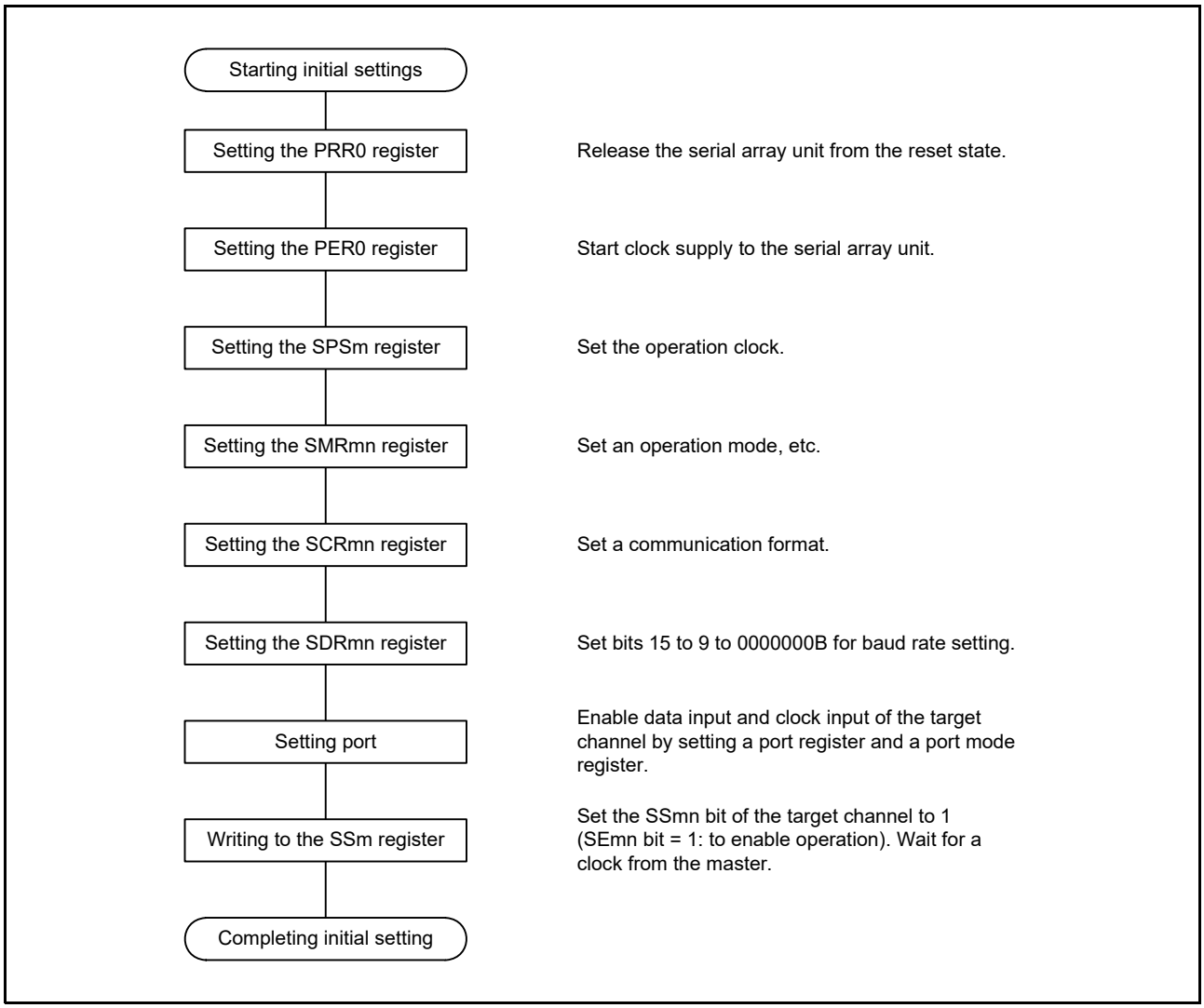


Figure 24 - 60 Procedure for Stopping Slave Reception

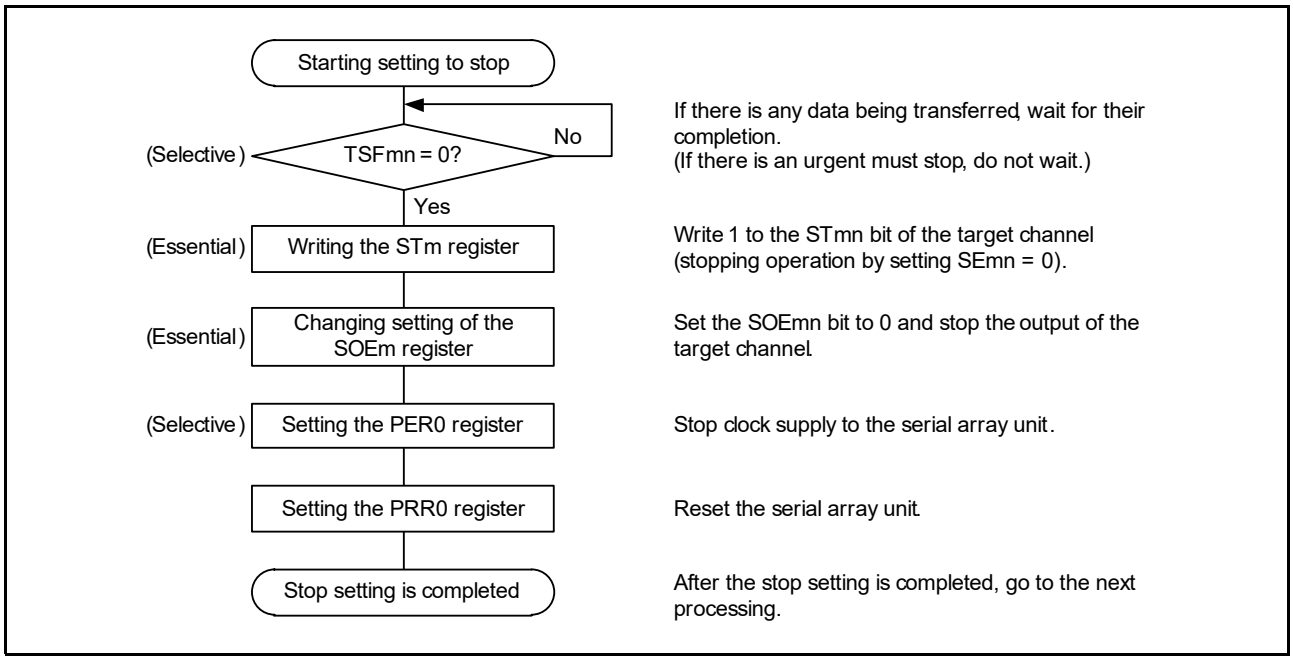
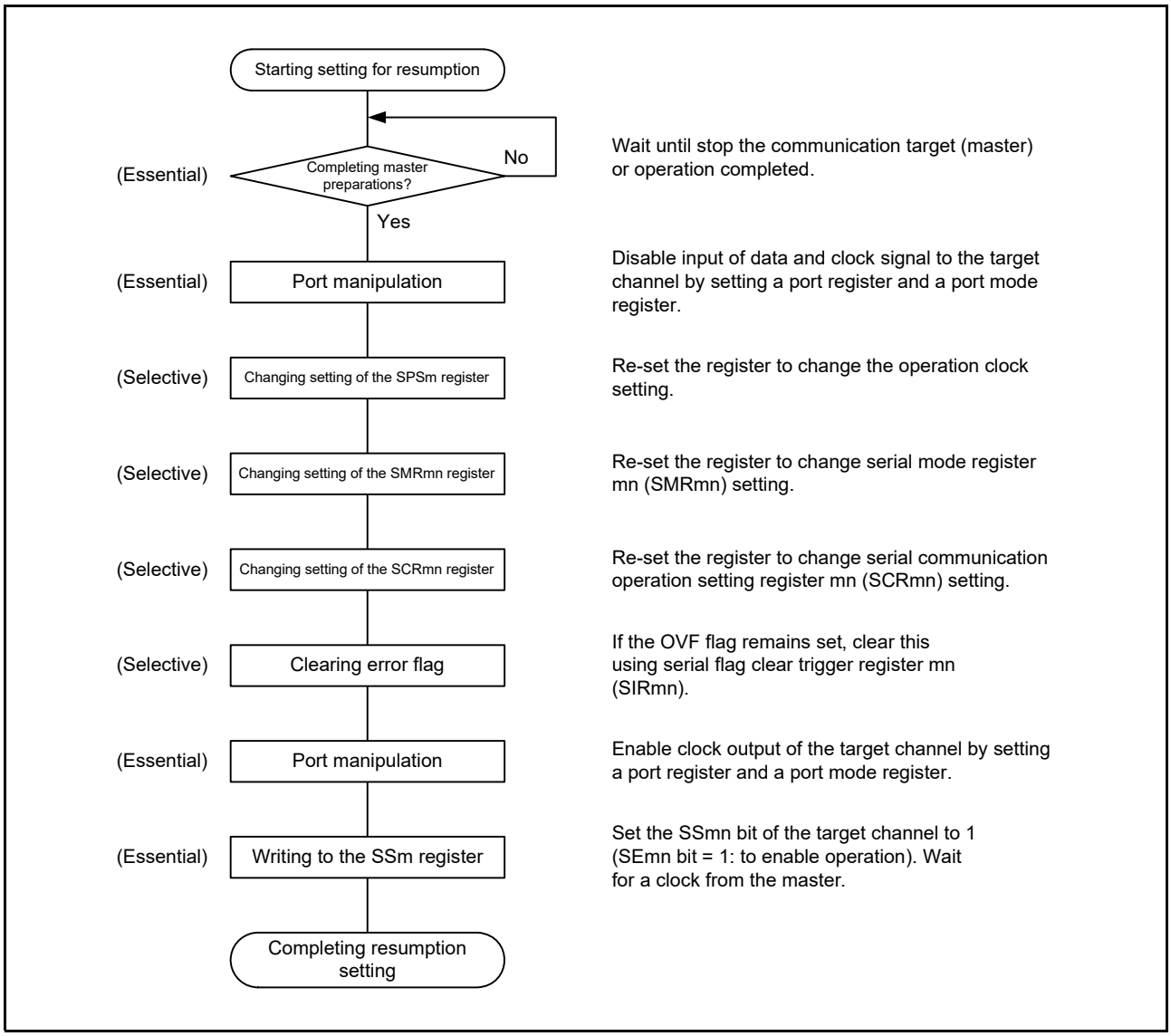


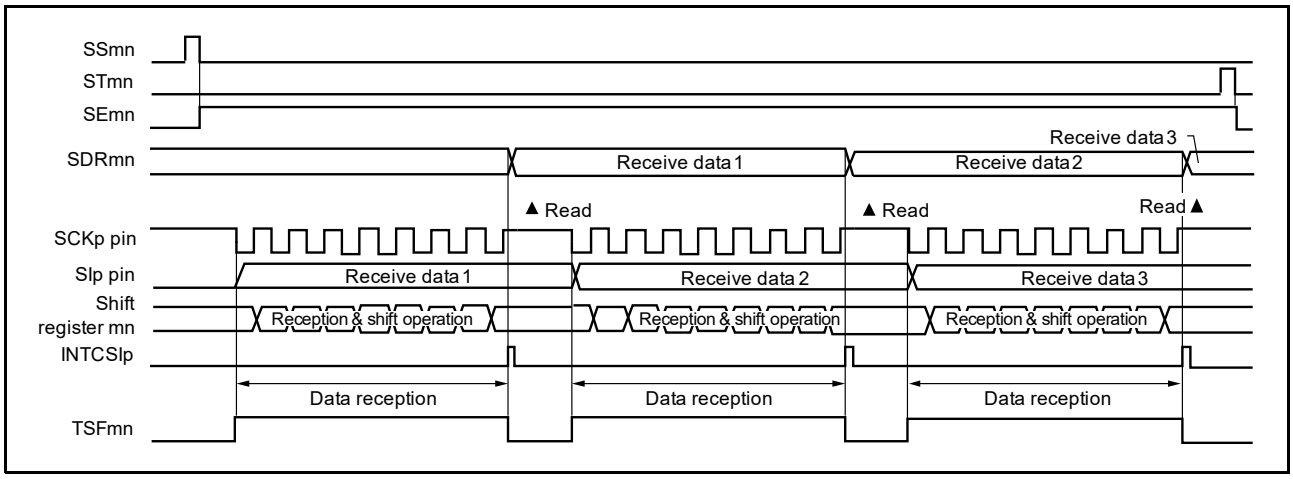
Figure 24 - 61 Procedure for Resuming Slave Reception



Remark If PRR0 is rewritten while stopping the communication to reset the serial array unit, wait until the communication target (master) stops or communication finishes, and then perform initialization instead of restarting the communication.

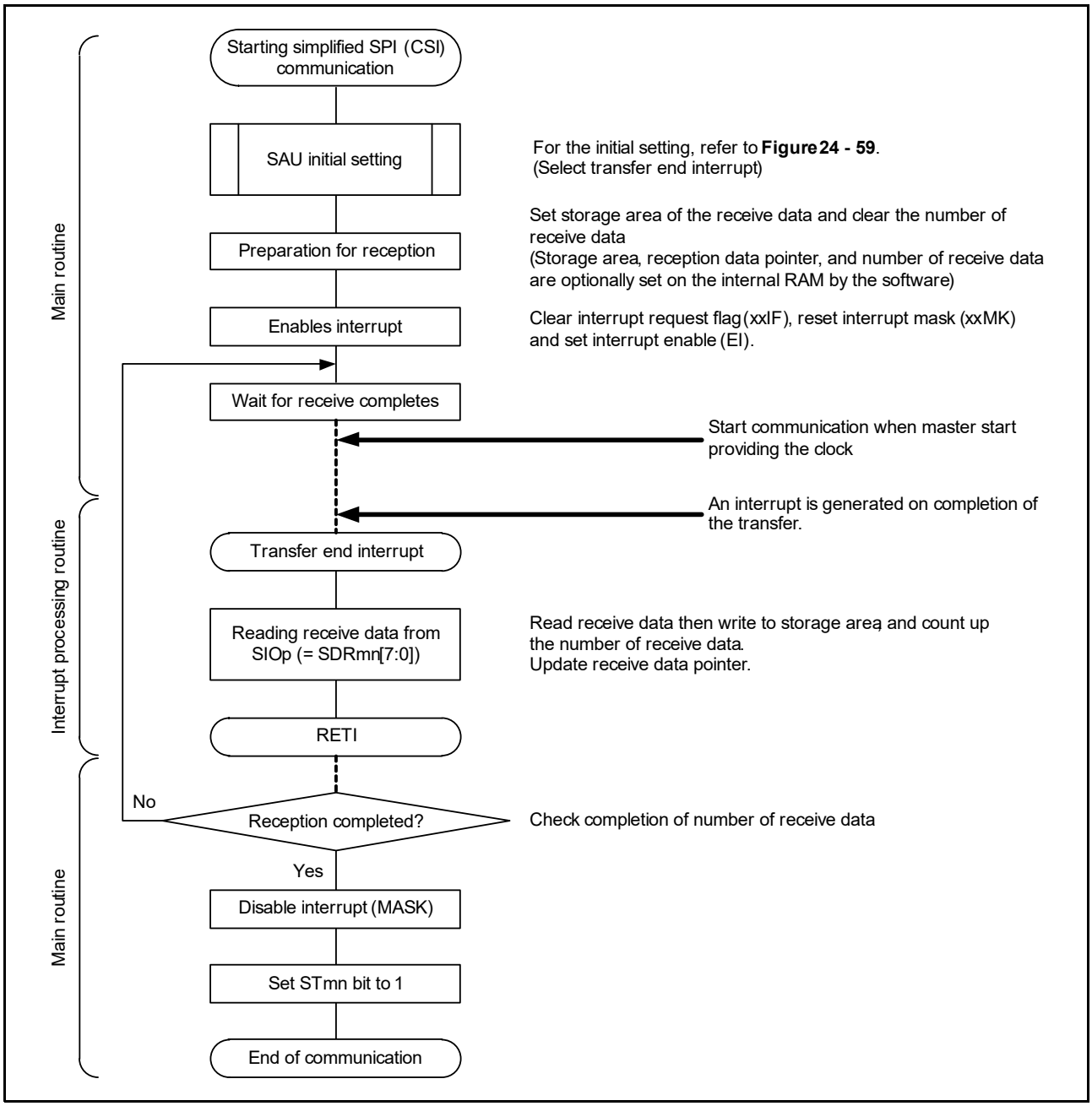
3. Processing flow (in single-reception mode)

Figure 24 - 62 Timing Chart of Slave Reception (in Single-reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

Figure 24 - 63 Flowchart of Slave Reception (in Single-reception Mode)



24.5.6 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	CSI01	CSI10	CSI11	CSI20	CSI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK10, SI10, SO10	SCK11, SI11, SO11	SCK20, SI20, SO20	SCK21, SI21, SO21
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21
	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 at the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock cycle 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, SCK01, SCK10, SCK11, SCK20, and SCK21 pins 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 specified in the electrical characteristics. For details, see **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** or **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.

Remark 1. f_{MCK}: Operation clock frequency of target channel
f_{SCK}: Serial clock frequency

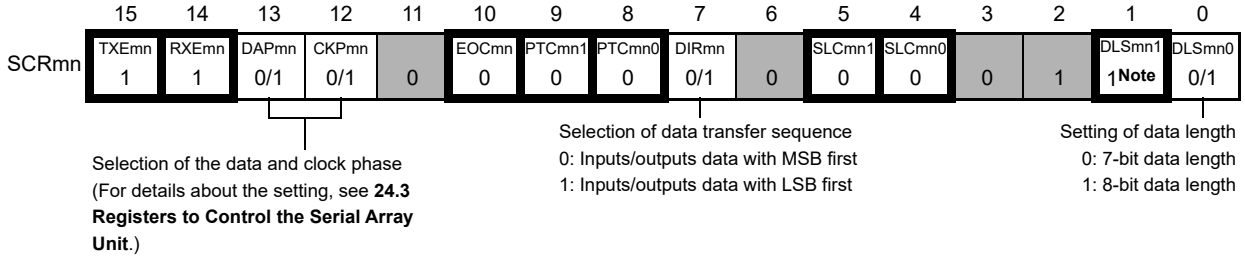
Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

1. Register setting

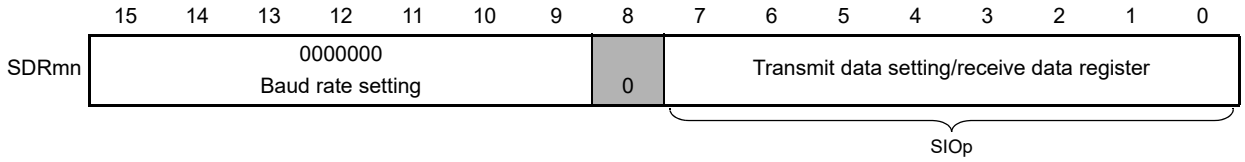
Figure 24 - 64 Example of Contents of Registers for Slave Transmission/Reception of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21)

a) Serial mode register mn (SMRmn)

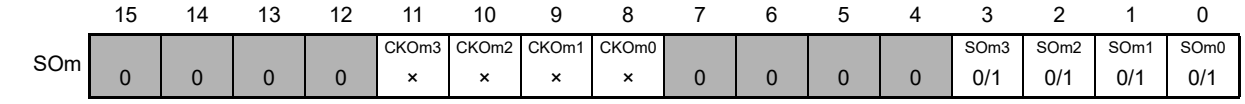
b) Serial communication operation setting register mn (SCRmn)



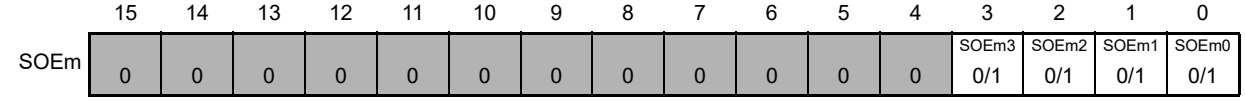
c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



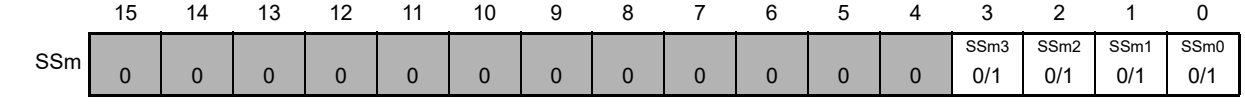
d) Serial output register m (SOm): Set only the bit of the target channel.



e) Serial output enable register m (SOEm): Set only the bit of the target channel to 1.



f) Serial channel start register m (SSm): Set only the bit of the target channel to 1.



Note Only provided for the SCR00 and SCR01 registers. This bit is fixed to 1 for the other registers.

Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

(Remarks are listed on the next page.)

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

Remark 2. ■: Setting is fixed in the simplified SPI (CSI) slave transmission/reception mode

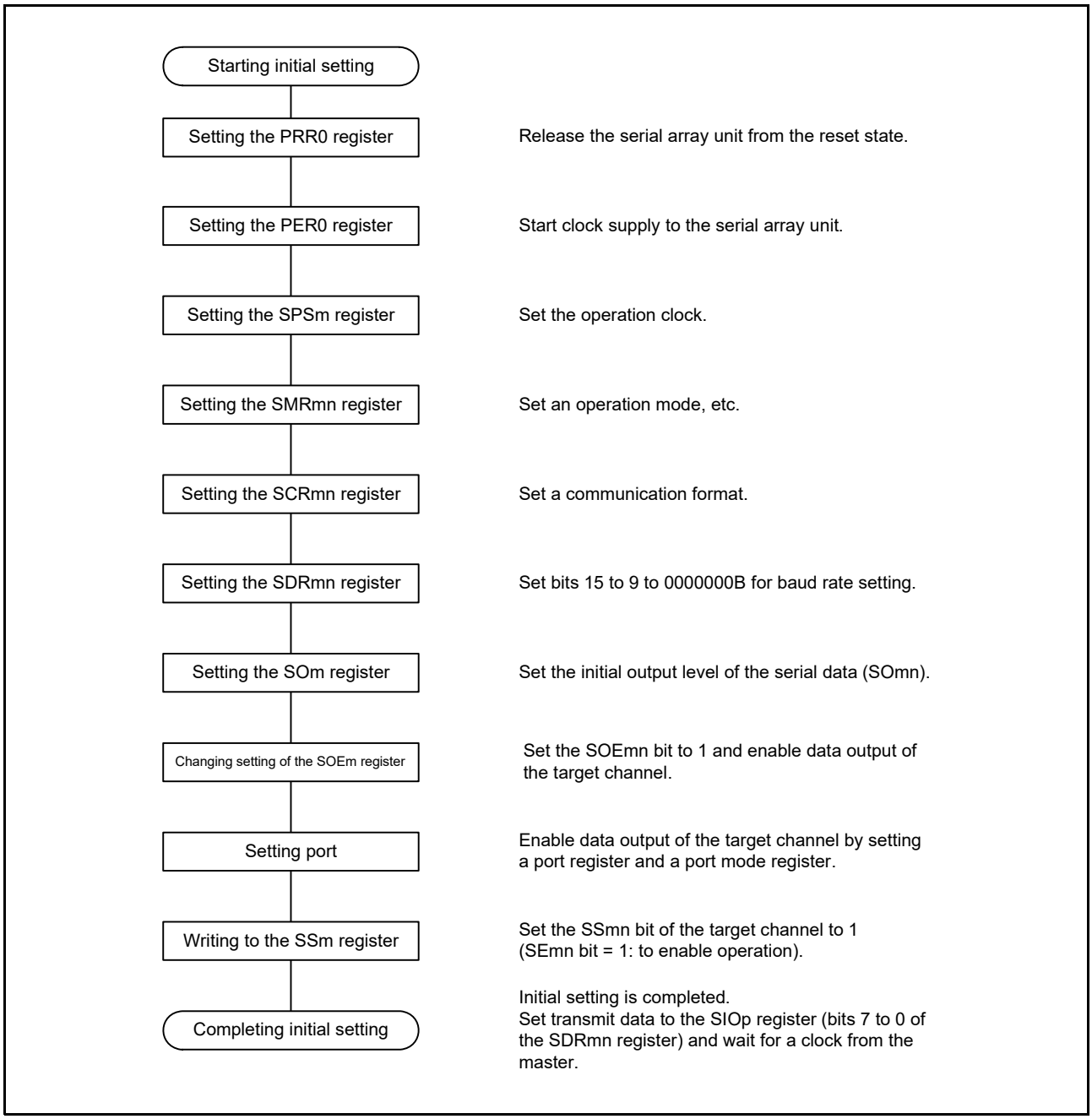
■: Setting disabled (set to the initial value)

×: 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

2. Operation procedure

Figure 24 - 65 Initial Setting Procedure for Slave Transmission/Reception



Caution Set transmit data to the SIOp register before the clock from the master is started.

Figure 24 - 66 Procedure for Stopping Slave Transmission/Reception

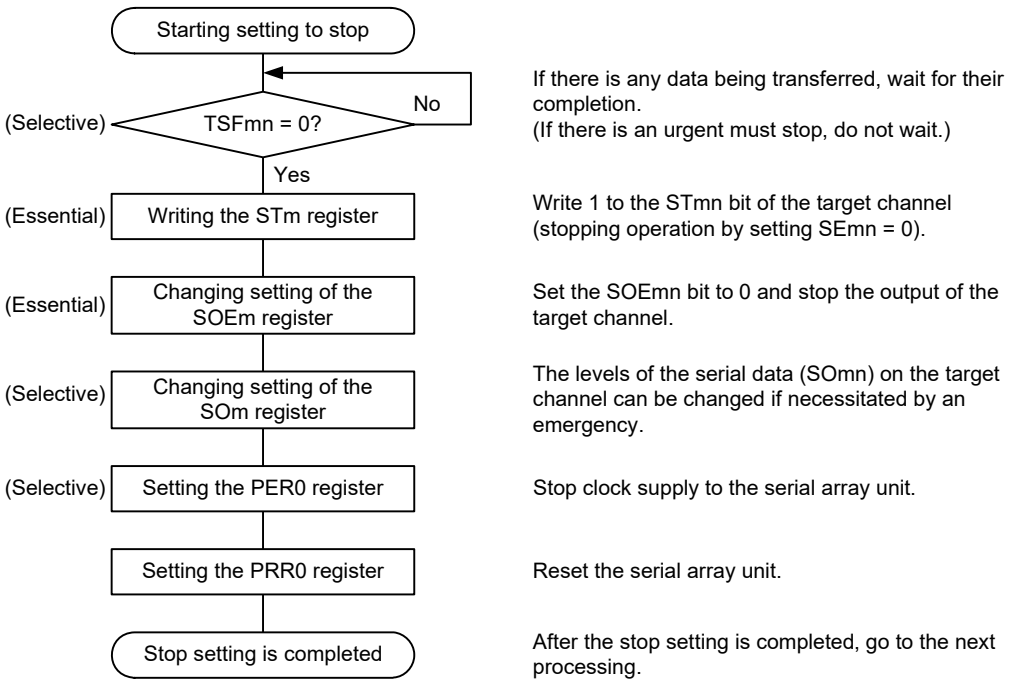
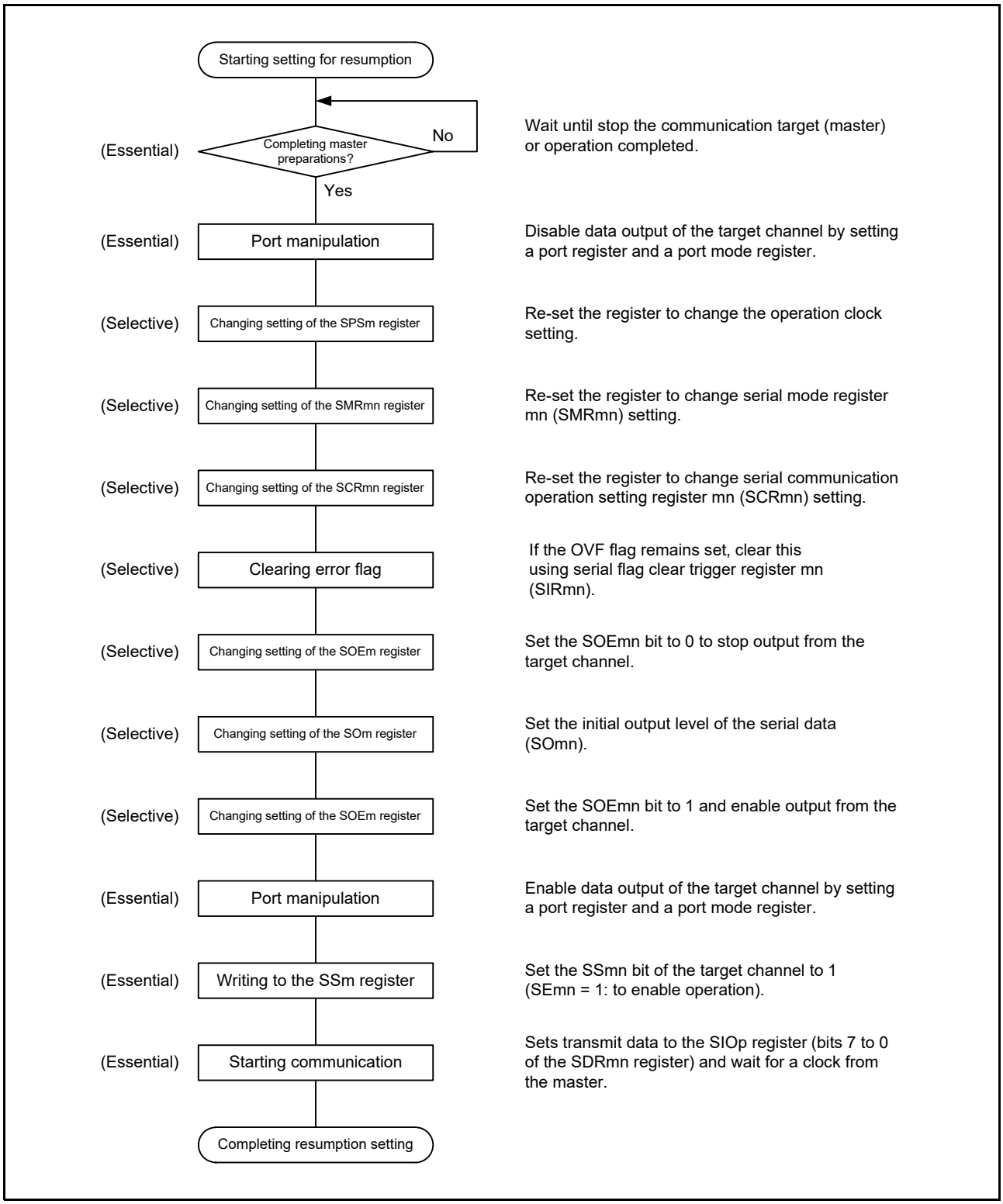


Figure 24 - 67 Procedure for Resuming Slave Transmission/Reception

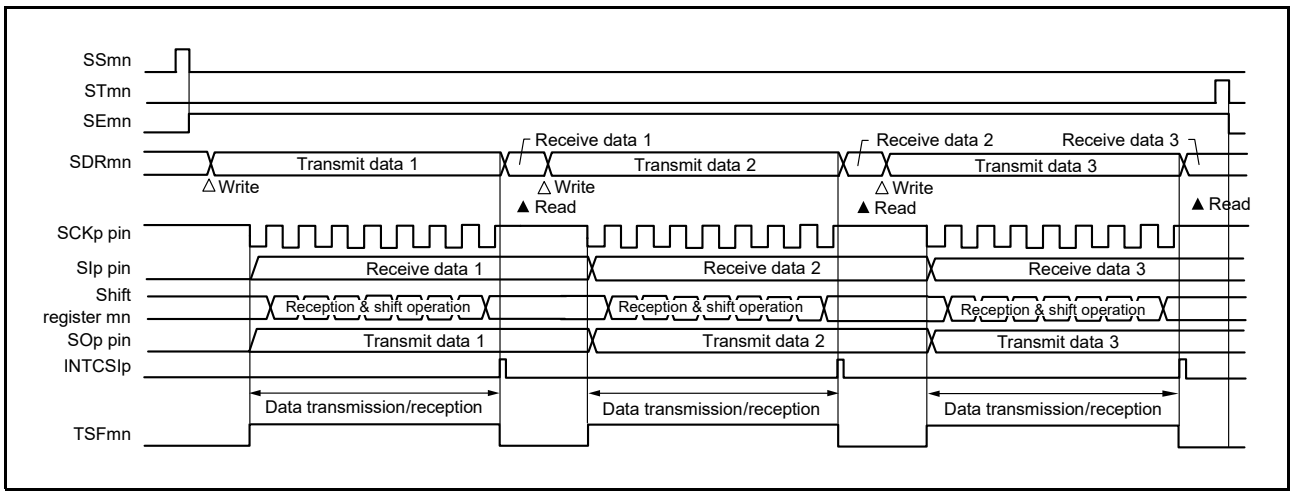


Caution 1. Set transmit data to the SIOp register before the clock from the master is started.

Caution 2. If PRR0 is rewritten while stopping the communication to reset the serial array unit, wait until the communication target (master) stops or communication finishes, and then perform initialization instead of restarting the communication.

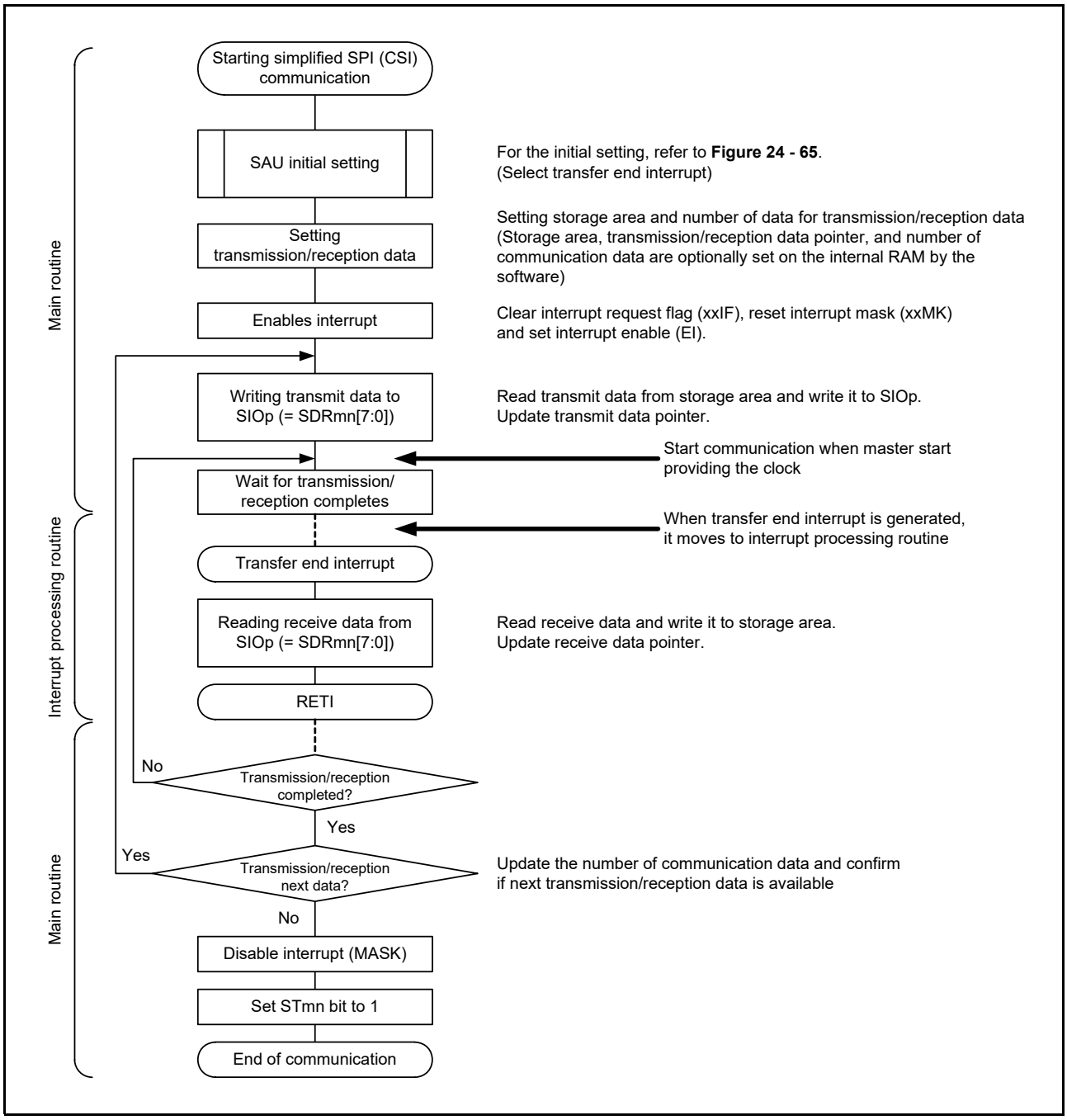
3. Processing flow (in single-transmission/reception mode)

Figure 24 - 68 Timing Chart of Slave Transmission/Reception (in Single-transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

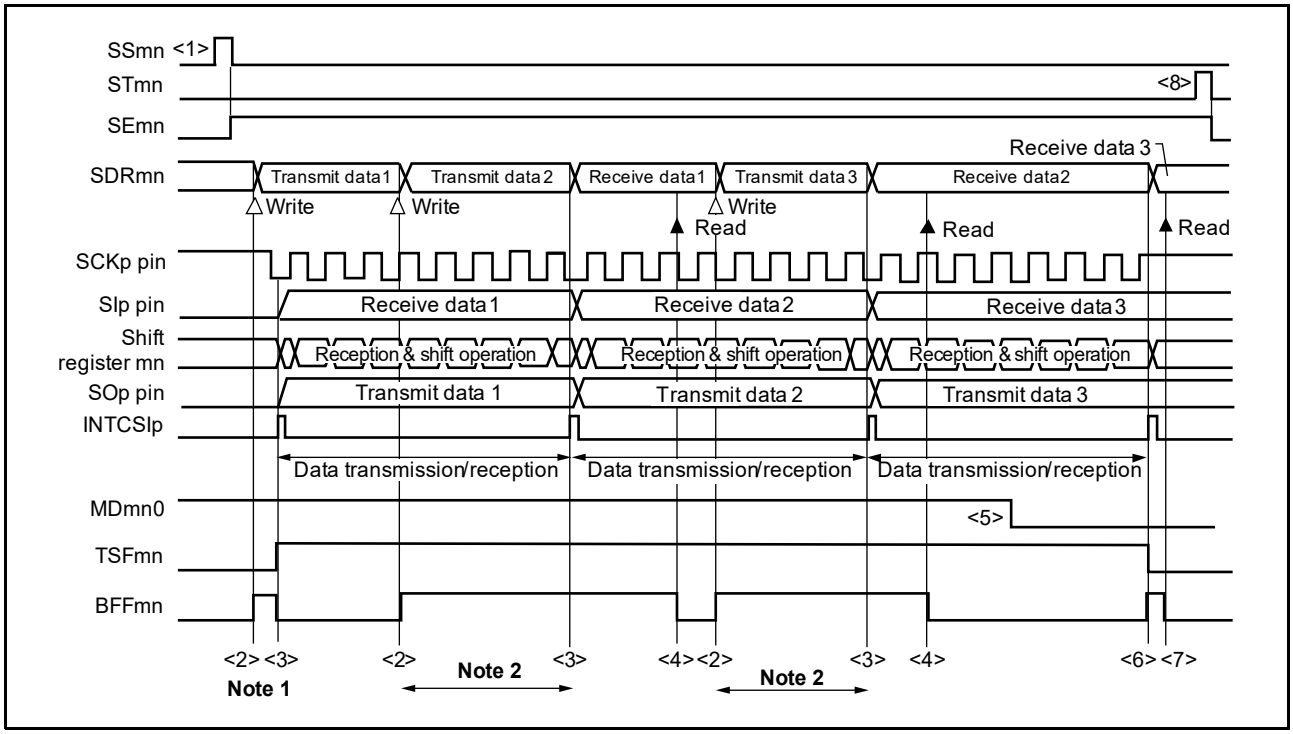
Figure 24 - 69 Flowchart of Slave Transmission/Reception (in Single-transmission/Reception Mode)



Caution Set transmit data to the SIOp register before the clock from the master is started.

4. Processing flow (in continuous transmission/reception mode)

Figure 24 - 70 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.

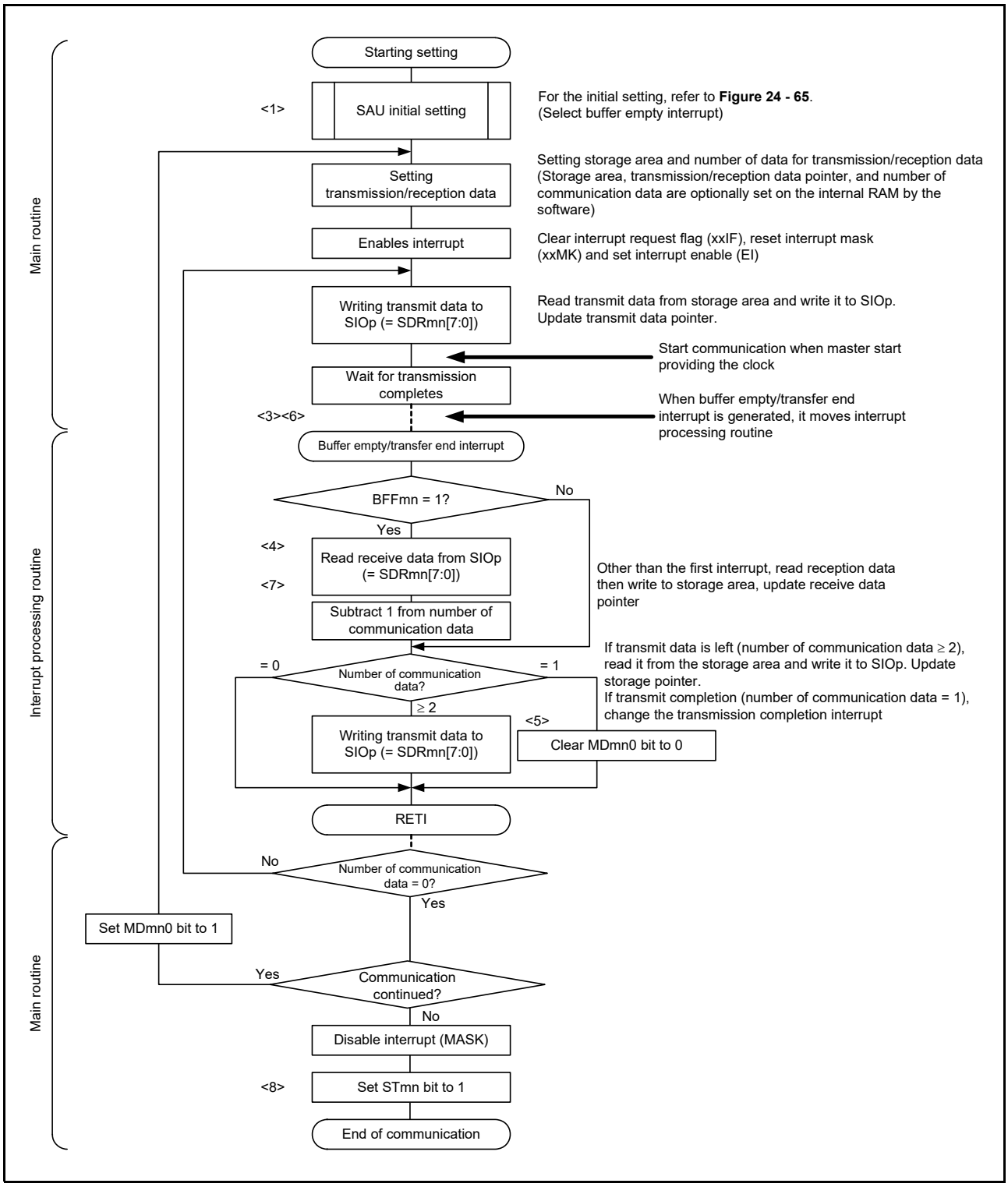
Note 2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

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 24 - 71 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

Figure 24 - 71 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution 1. Set transmit data to the SIOp register before the clock from the master is started.

Caution 2. When the interrupt on compare match with ITLCMP01 is selected as a capture trigger, the compare match detection flag for channel 2 (ITF02) and capture detection flag (ITF0C) are set on capture of the counter value. When only using the capture detection flag, set the ITLMKF0 register to mask the compare match detection flag for channel 2.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 24 - 70 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)**.

24.5.7 SNOOZE mode function

The SNOOZE mode makes the simplified SPI (CSI) perform reception operations upon SCKp pin input detection while in the STOP mode. Normally the simplified SPI (CSI) stops communication in the STOP mode. However, using the SNOOZE mode enables the simplified SPI (CSI) to perform reception operations without CPU operation upon detection of the SCKp pin input. Only CSI00 can be set to the SNOOZE mode.

When using the simplified SPI (CSI) in SNOOZE mode, make the following setting before switching to the STOP mode (See **Figure 24 - 73 Flowchart of SNOOZE Mode Operation (Once Startup)** and **Figure 24 - 75 Flowchart of SNOOZE Mode Operation (Continuous Startup)**.)

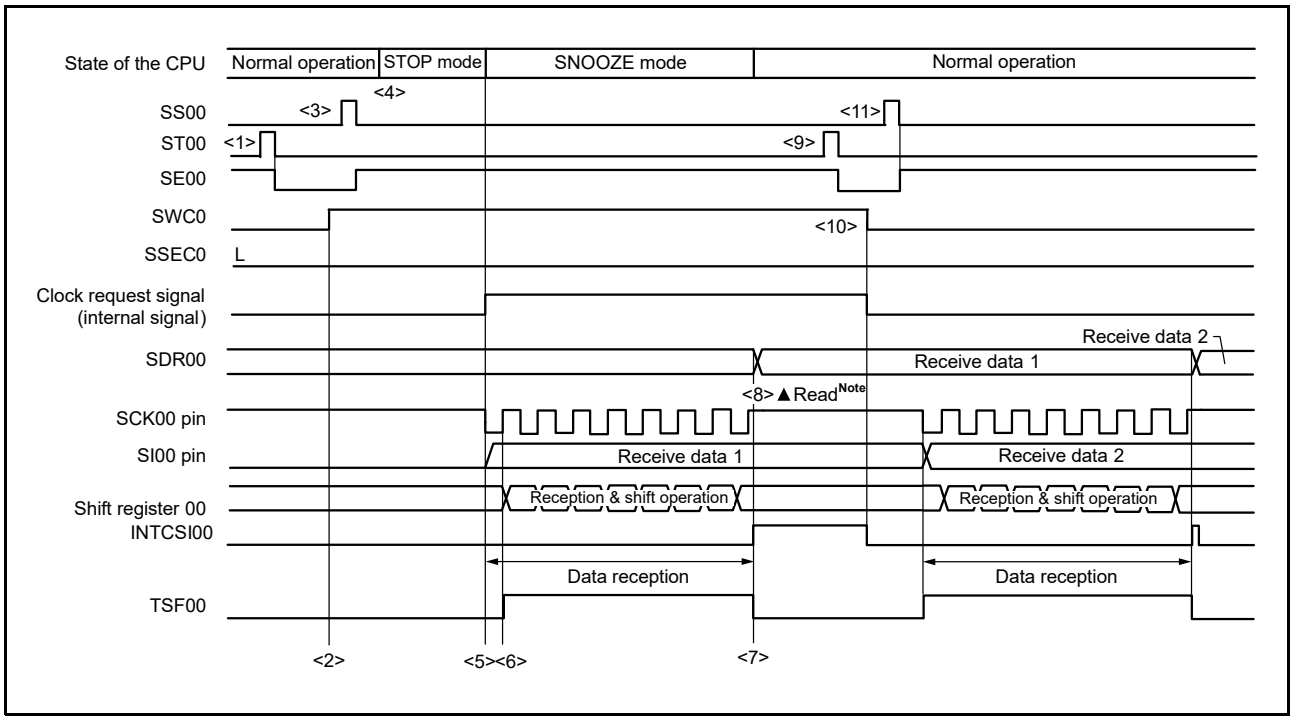
- When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 immediately before switching to the STOP mode. After the initial setting has been completed, set the SSm0 bit of serial channel start register m (SSm) to 1.
- The CPU shifts to the SNOOZE mode on 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 or medium-speed on-chip oscillator clock is selected for f_{CLK}.

Caution 2. The maximum transfer rate when using CSIp in the SNOOZE mode is 1 Mbps.

1. SNOOZE mode operation (once startup)

Figure 24 - 72 Timing Chart of SNOOZE Mode Operation (Once 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.

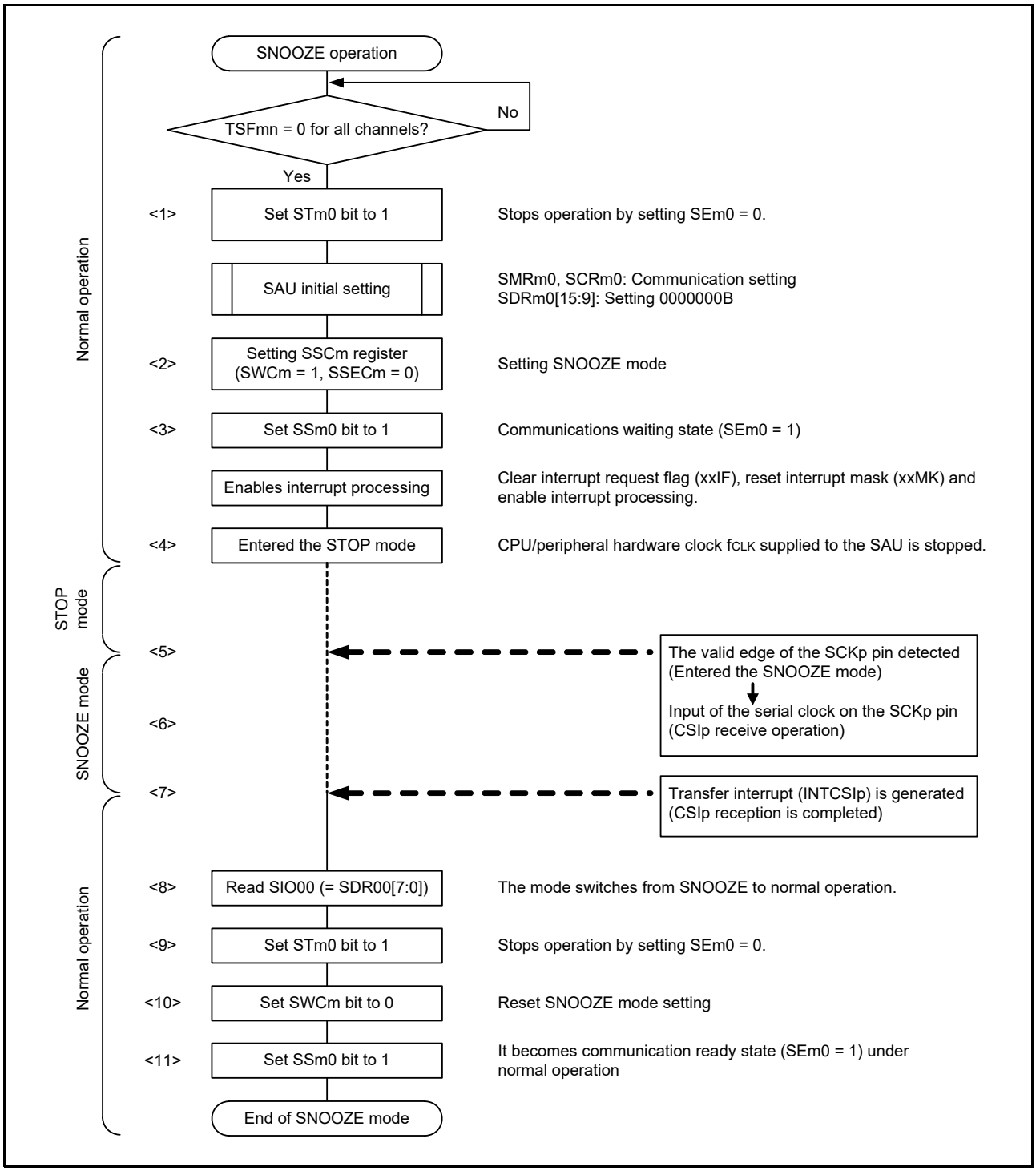
Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (the SEm0 bit is cleared and the operation stops). After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).

Caution 2. When SWCm = 1, the BFFm0 and OVFm0 flags will not change.

Remark 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 24 - 73 Flowchart of SNOOZE Mode Operation (Once Startup)**.

Remark 2. m = 0; p = 00

Figure 24 - 73 Flowchart of SNOOZE Mode Operation (Once Startup)



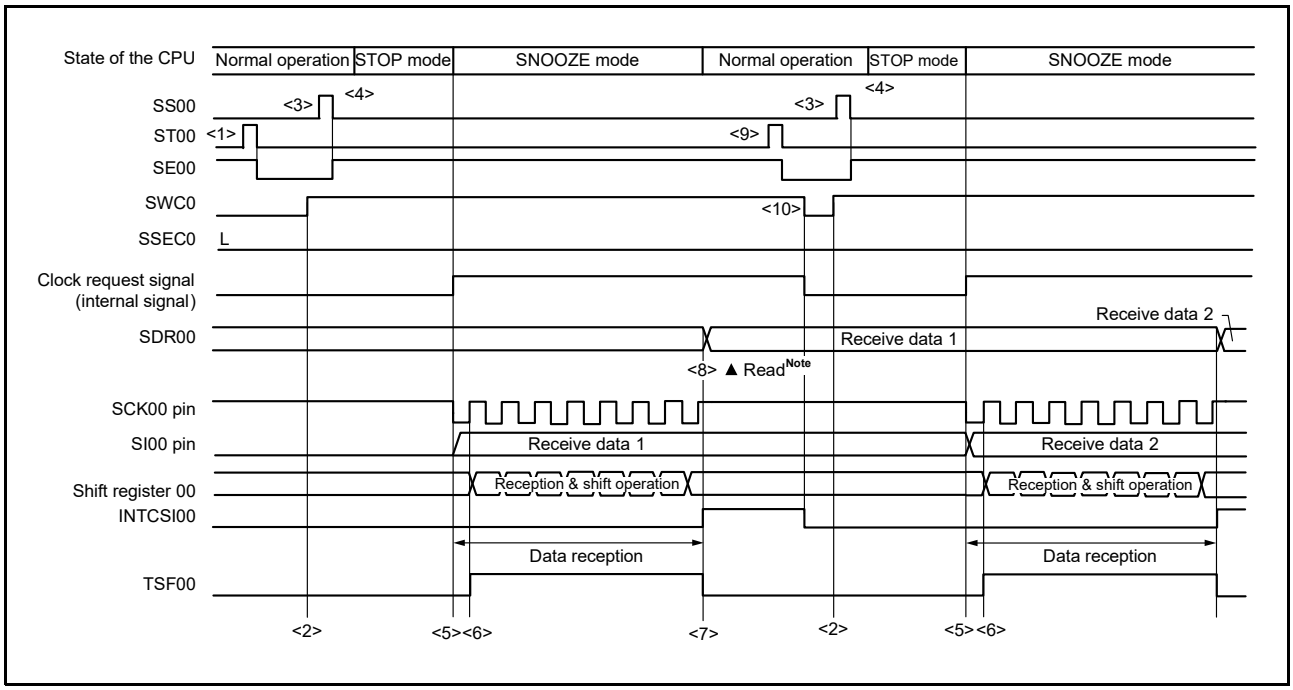
<R>

Remark 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 24 - 72 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 24 - 74 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.

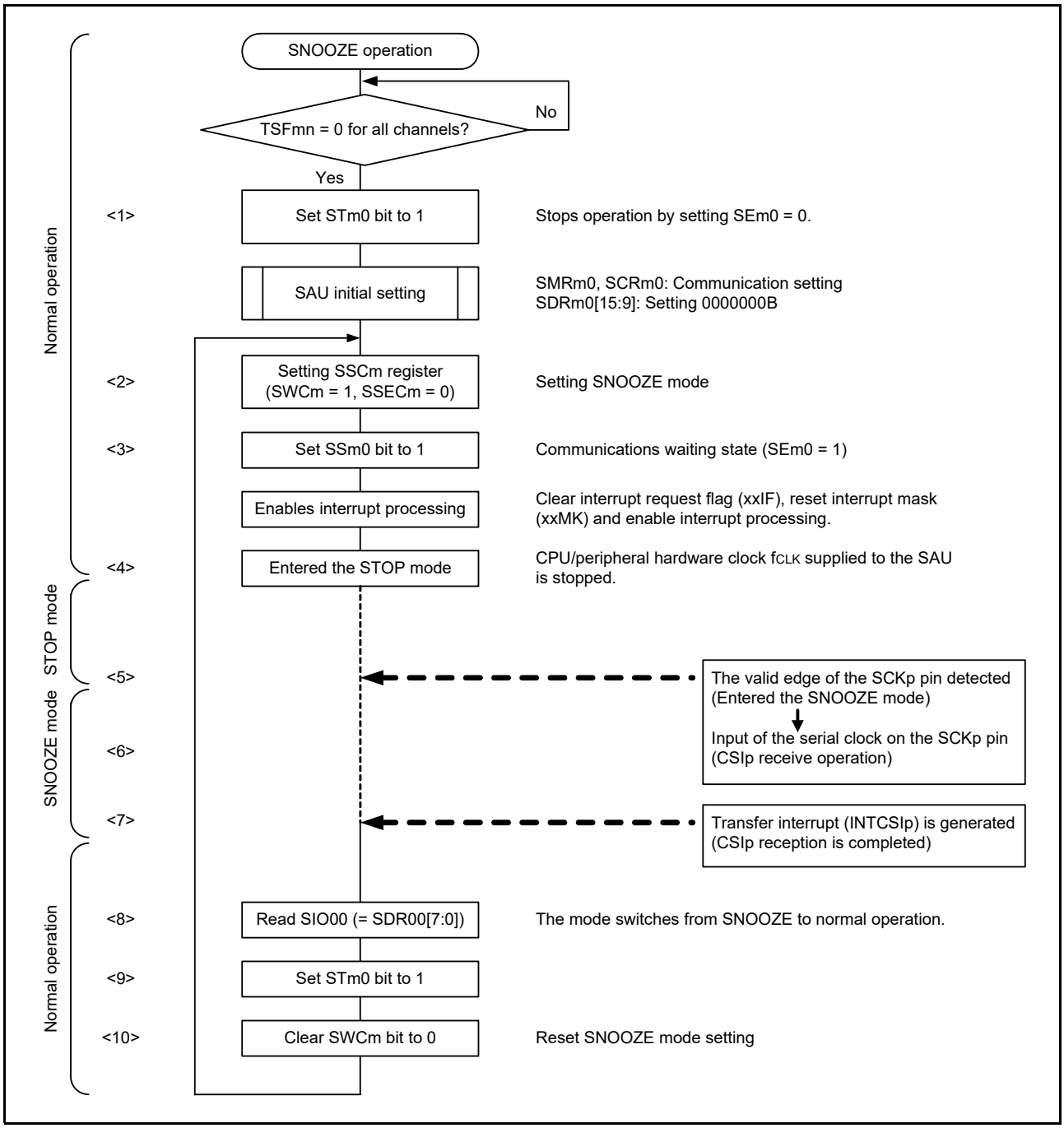
Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (the SEm0 bit is cleared and the operation stops). After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).

Caution 2. When SWCm = 1, the BFFm0 and OVFm0 flags will not change.

Remark 1. <1> to <10> in the figure correspond to <1> to <10> in **Figure 24 - 75 Flowchart of SNOOZE Mode Operation (Continuous Startup)**.

Remark 2. m = 0, p = 00

Figure 24 - 75 Flowchart of SNOOZE Mode Operation (Continuous Startup)



Remark 1. <1> to <10> in the figure correspond to <1> to <10> in **Figure 24 - 74 Timing Chart of SNOOZE Mode Operation (Continuous Startup) (Type 1: DAPmn = 0, CKPmn = 0)**.

Remark 2. m = 0; p = 00

24.5.8 Calculating transfer clock frequency

The transfer clock frequency for simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) 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 24 - 3 Selection of Operation Clock for Simplified SPI (1/2)

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} = 32 MHz	f_{CLK} = 48 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	32 MHz	Setting prohibited
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	16 MHz	24 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	8 MHz	12 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	4 MHz	6 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	2 MHz	3 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	1 MHz	1.5 MHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	500 kHz	750 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	250 kHz	375 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	125 kHz	188 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz	93.8 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz	46.9 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz	23.4 kHz
	×	×	×	×	1	1	0	0	$f_{CLK}/2^{12}$	7.81 kHz	11.7 kHz
	×	×	×	×	1	1	0	1	$f_{CLK}/2^{13}$	3.91 kHz	5.86 kHz
	×	×	×	×	1	1	1	0	$f_{CLK}/2^{14}$	1.95 kHz	2.93 kHz
	×	×	×	×	1	1	1	1	$f_{CLK}/2^{15}$	977 Hz	1.46 kHz

Table 24 - 3 Selection of Operation Clock for Simplified SPI (2/2)

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} = 32 MHz	f_{CLK} = 48 MHz
1	0	0	0	0	×	×	×	×	f_{CLK}	32 MHz	Setting prohibited
	0	0	0	1	×	×	×	×	$f_{CLK}/2$	16 MHz	24 MHz
	0	0	1	0	×	×	×	×	$f_{CLK}/2^2$	8 MHz	12 MHz
	0	0	1	1	×	×	×	×	$f_{CLK}/2^3$	4 MHz	6 MHz
	0	1	0	0	×	×	×	×	$f_{CLK}/2^4$	2 MHz	3 MHz
	0	1	0	1	×	×	×	×	$f_{CLK}/2^5$	1 MHz	1.5 MHz
	0	1	1	0	×	×	×	×	$f_{CLK}/2^6$	500 kHz	750 kHz
	0	1	1	1	×	×	×	×	$f_{CLK}/2^7$	250 kHz	375 kHz
	1	0	0	0	×	×	×	×	$f_{CLK}/2^8$	125 kHz	188 kHz
	1	0	0	1	×	×	×	×	$f_{CLK}/2^9$	62.5 kHz	93.8 kHz
	1	0	1	0	×	×	×	×	$f_{CLK}/2^{10}$	31.25 kHz	46.9 kHz
	1	0	1	1	×	×	×	×	$f_{CLK}/2^{11}$	15.63 kHz	23.4 kHz
	1	1	0	0	×	×	×	×	$f_{CLK}/2^{12}$	7.81 kHz	11.7 kHz
	1	1	0	1	×	×	×	×	$f_{CLK}/2^{13}$	3.91 kHz	5.86 kHz
	1	1	1	0	×	×	×	×	$f_{CLK}/2^{14}$	1.95 kHz	2.93 kHz
	1	1	1	1	×	×	×	×	$f_{CLK}/2^{15}$	977 Hz	1.46 kHz

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, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

24.5.9 Procedure for processing errors that occurred during simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) communication

The procedure for processing errors that occurred during simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) communication is described in **Table 24 - 4**.

Table 24 - 4 Processing Procedure in Case of Overrun Error

Software Manipulation	State of the Hardware	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.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

24.6 Operation of UART (UART0 to UART2) Communication

The UART functions are for asynchronous communications and each UART uses two lines: serial data transmission (TxD) and serial data reception (RxD). These two lines are used to send and receive data asynchronously (using an internal baud rate) to and from the other-party device per single data frame, each consisting of a start bit, data, parity bit, and stop bit. Full-duplex UART communications can be realized by using two channels, one for use in transmission (even-numbered channel) and the other for use in reception (odd-numbered channel). The LIN bus can also be supported by the combination of UART0, timer array unit 0 (channel 3), and an external interrupt (INTP0).

[Data transmission/reception]

- Data length of 7, 8, or 9 bits**Note**
- MSB/LSB first selectable
- Level setting of transmit/receive data (selecting whether to reverse the level)
- Parity bit appending and parity check functions
- Stop bit appending, stop bit check function

[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 reception supports the SNOOZE mode. In the SNOOZE mode, data can be received without CPU processing upon detecting RxD input in the STOP mode. The SNOOZE mode is only available in UART0, which support the reception baud rate adjustment function.

The LIN-bus is accepted in UART0 (channels 0 and 1 of unit 0).

[LIN-bus functions]

- Wakeup signal detection
- Break field (BF) detection
- Sync field measurement, baud rate calculation

} Using the external interrupt (INTP0) and the timer array unit 0 (channel 3)

Note Only UART0 supports the 9-bit data length.

When the medium-speed on-chip oscillator clock (f_{IM}) or low-speed on-chip oscillator clock (f_{IL}) is selected for f_{CLK}, use the medium-speed on-chip oscillator trimming register (MIOTRM) or the low-speed on-chip oscillator trimming register (LIOTRM) to correct the accuracy of the oscillation frequency.

- UART0 uses channels 0 and 1 of SAU0.
- UART1 uses channels 2 and 3 of SAU0.
- UART2 uses channels 0 and 1 of SAU1.

<20-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	—	—	—
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

<24- and 25-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	—	UART0 (supporting LIN-bus)Note	—
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

<30- and 32-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00 (supporting slave select input)	UART0 (supporting LIN-bus)	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

<40- and 44-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00 (supporting slave select input)	UART0 (supporting LIN-bus)	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	CSI21		IIC21

<48- and 52-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00 (supporting slave select input)	UART0 (supporting LIN-bus)	IIC00
	1	CSI01		IIC01
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	CSI21		IIC21

<64-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00 (supporting slave select input)	UART0 (supporting LIN-bus)	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	CSI21		IIC21

Note This function can be used when the setting of bit 1 (PIOR01) in peripheral I/O redirection register 0 (PIOR0).

Select a single function for each channel. Only the selected function is possible. If UART0 is selected for channels 0 and 1 of unit 0, for example, the CSI00 and CSI01 functions cannot be used. At this time, however, channel 2 or 3 of the same unit can be used for a function other than UART0, such as CSI10, UART1, and IIC10.

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 four types of communication operations.

- UART transmission (See **24.6.1**.)
- UART reception (See **24.6.2**.)
- LIN transmission (UART0 only) (See **24.7.1**.)
- LIN reception (UART0 only) (See **24.7.2**.)

24.6.1 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	UART2
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	TxD0	TxD1	TxD2
Interrupt	INTST0	INTST1	INTST2
	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**Note 2**	Max. f_{MCK}/6 [bps] (SDRmn[15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]		
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 bit		
Data direction	MSB or LSB first		

Note 1. Only UART0 supports the 9-bit data length.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** or **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.

Remark 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

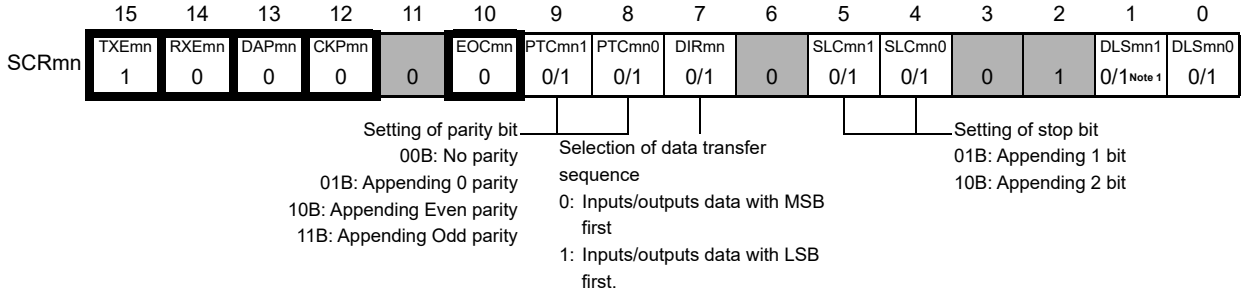
Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10

1. Register setting

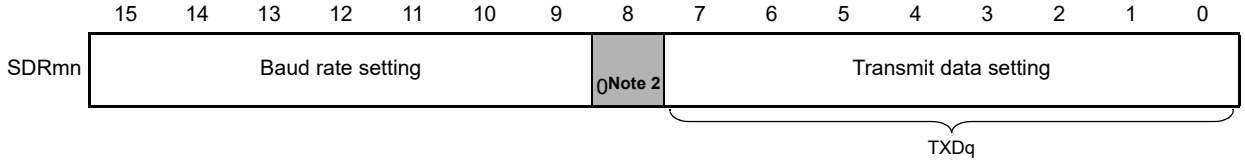
Figure 24 - 76 Example of Contents of Registers for UART Transmission of UART (UART0 to UART2)

a) Serial mode register mn (SMRmn)

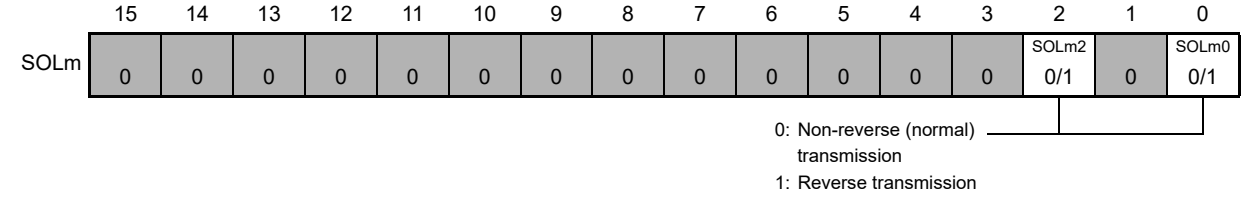
b) Serial communication operation setting register mn (SCRmn)



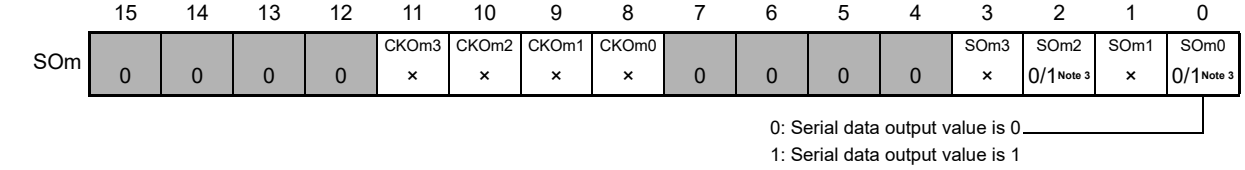
c) Serial data register mn (SDRmn) (lower 8 bits: TXDq)



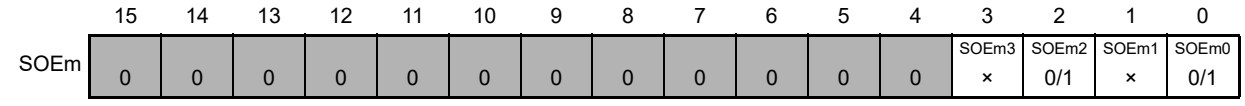
d) Serial output level register m (SOLm): Set only the bit of the target channel.



e) Serial output register m (SOm): Set only the bit of the target channel.



f) Serial output enable register m (SOEm): Set only the bit of the target channel to 1.



g) Serial channel start register m (SSm): Set only the bit 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 1. Only provided for the SCR00 register. This bit is fixed to 1 for the other registers.

Note 2. When UART0 performs 9-bit communication, bits 0 to 8 of the SDRm0 register are used as the transmission data specification area. Only UART0 supports the 9-bit data length.

Note 3. Before transmission is started, 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, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 2),
mn = 00, 02, 10

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

2. Operation procedure

Figure 24 - 77 Initial Setting Procedure for UART Transmission

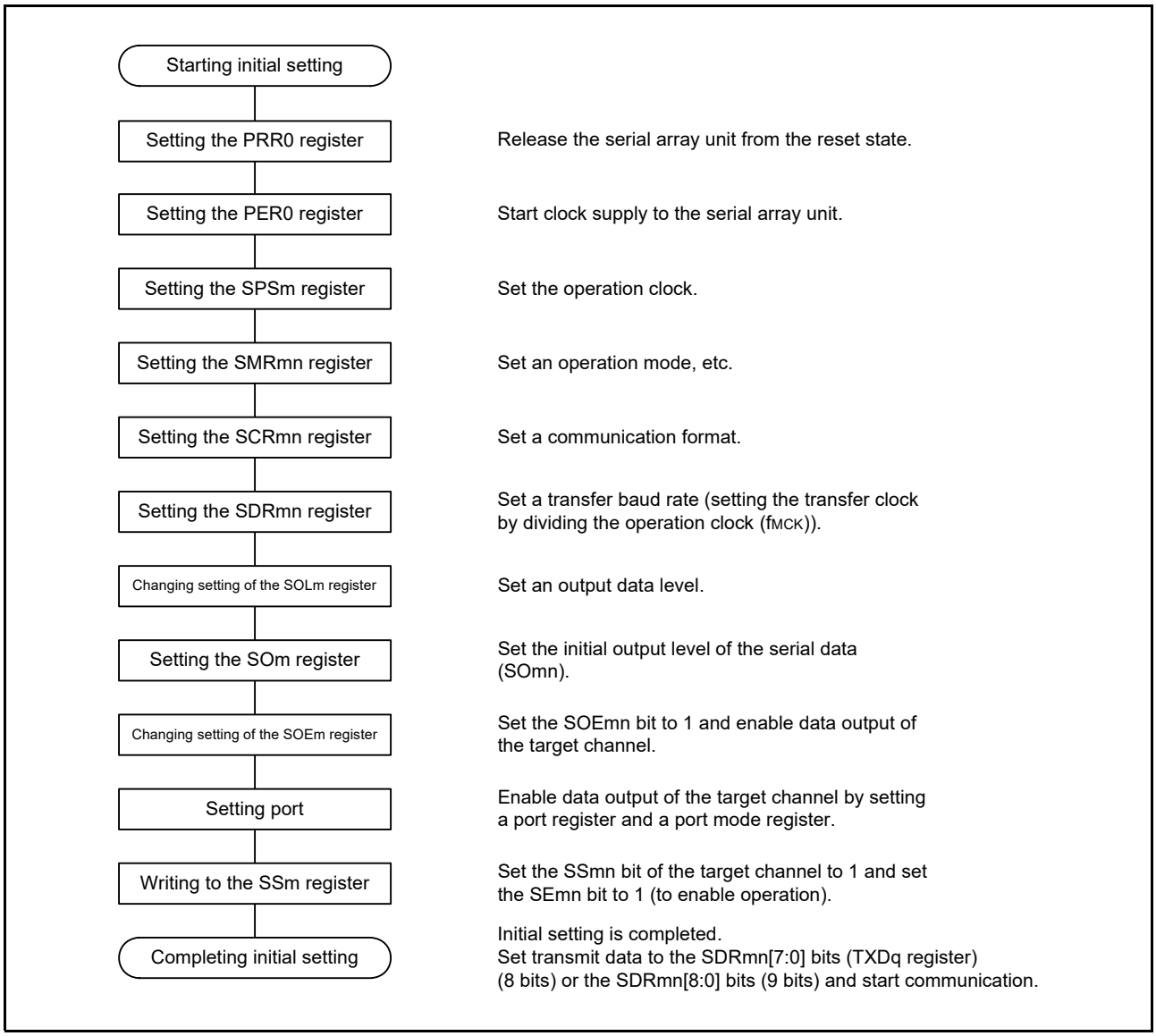


Figure 24 - 78 Procedure for Stopping UART Transmission

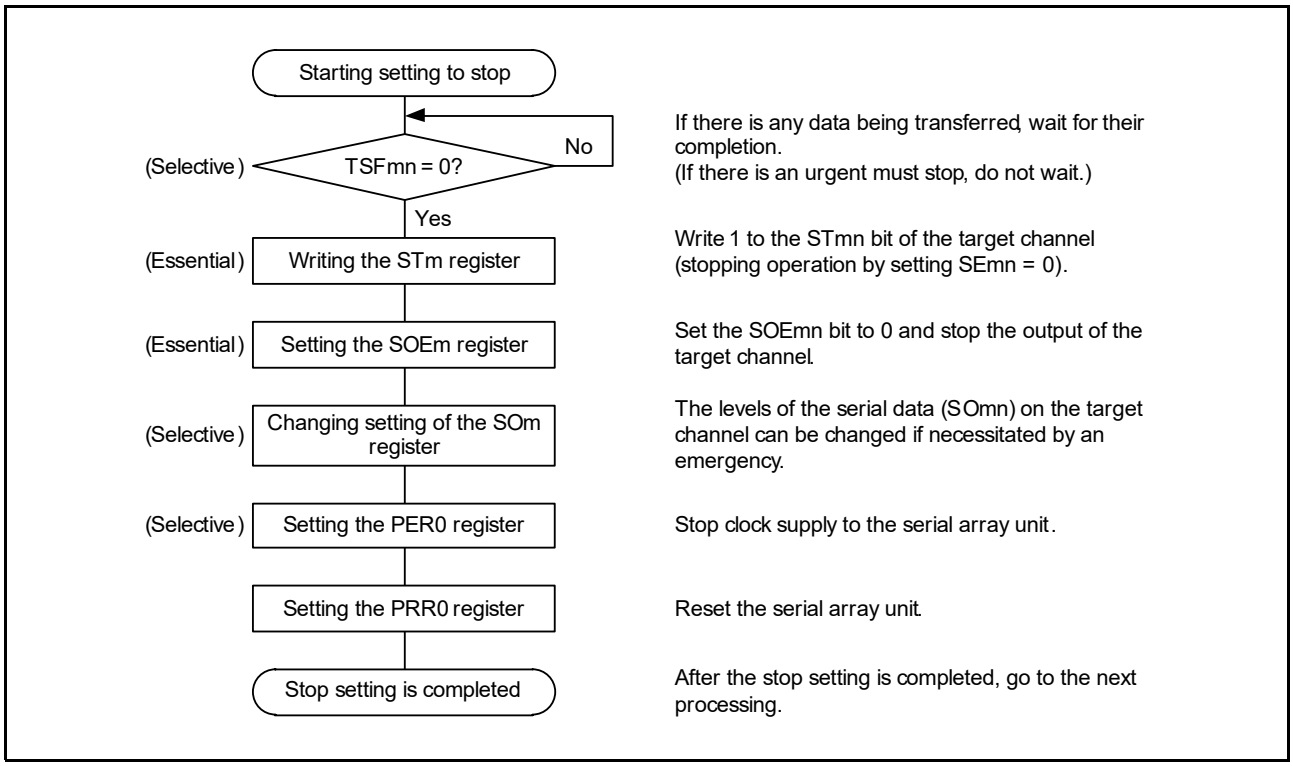
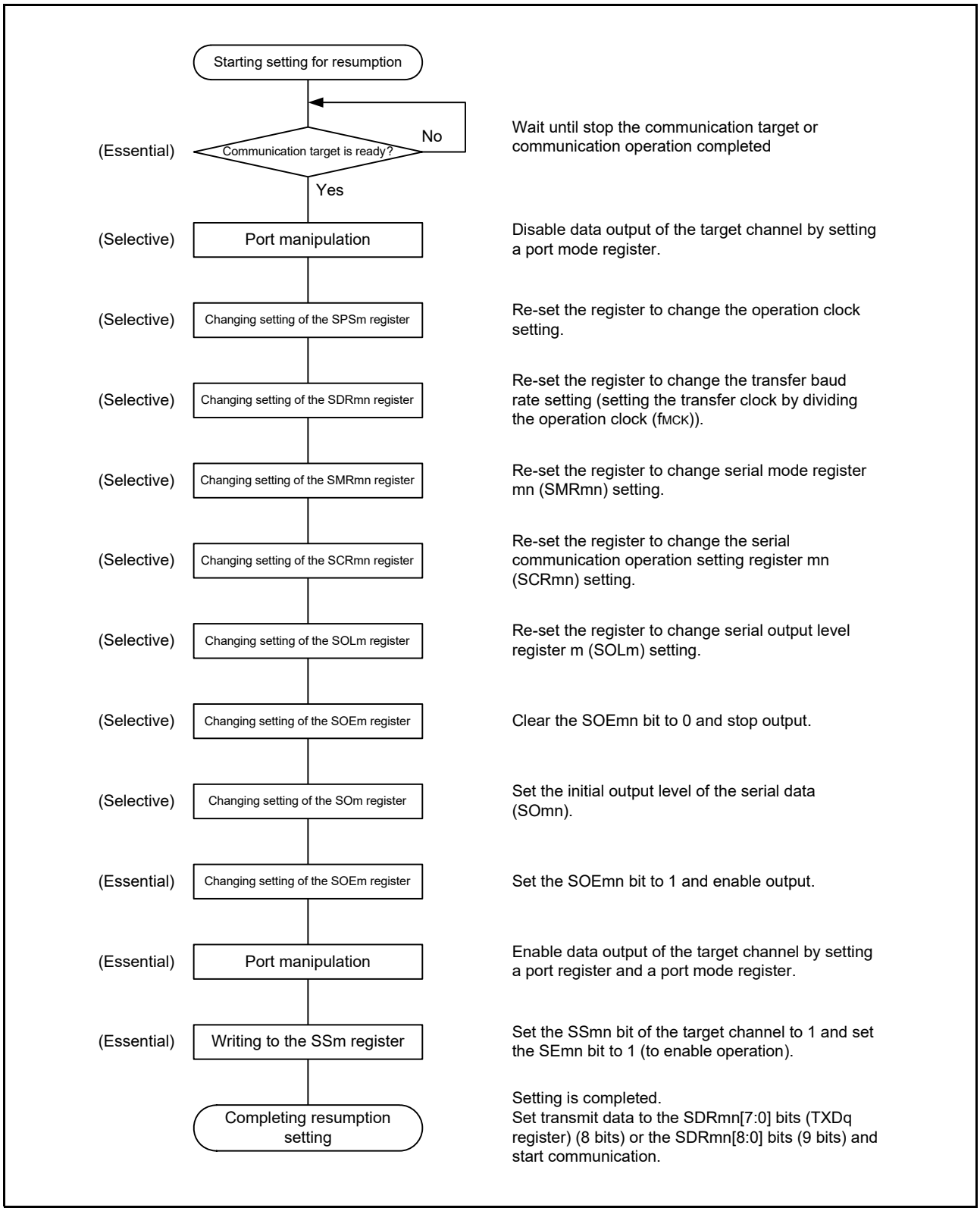


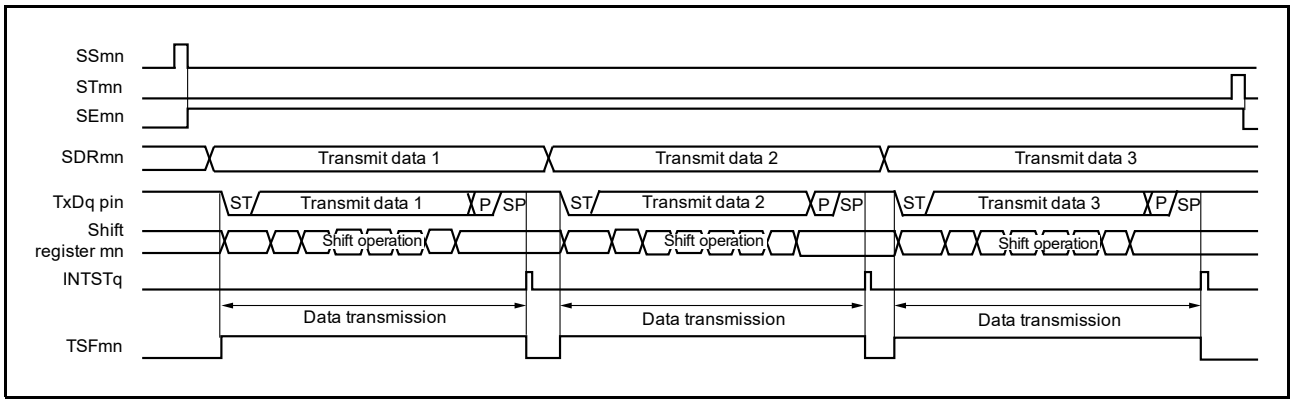
Figure 24 - 79 Procedure for Resuming UART Transmission



Remark If PRR0 is rewritten while stopping the communication to reset the serial array unit, wait until the communication target stops or communication finishes, and then perform initialization instead of restarting the communication.

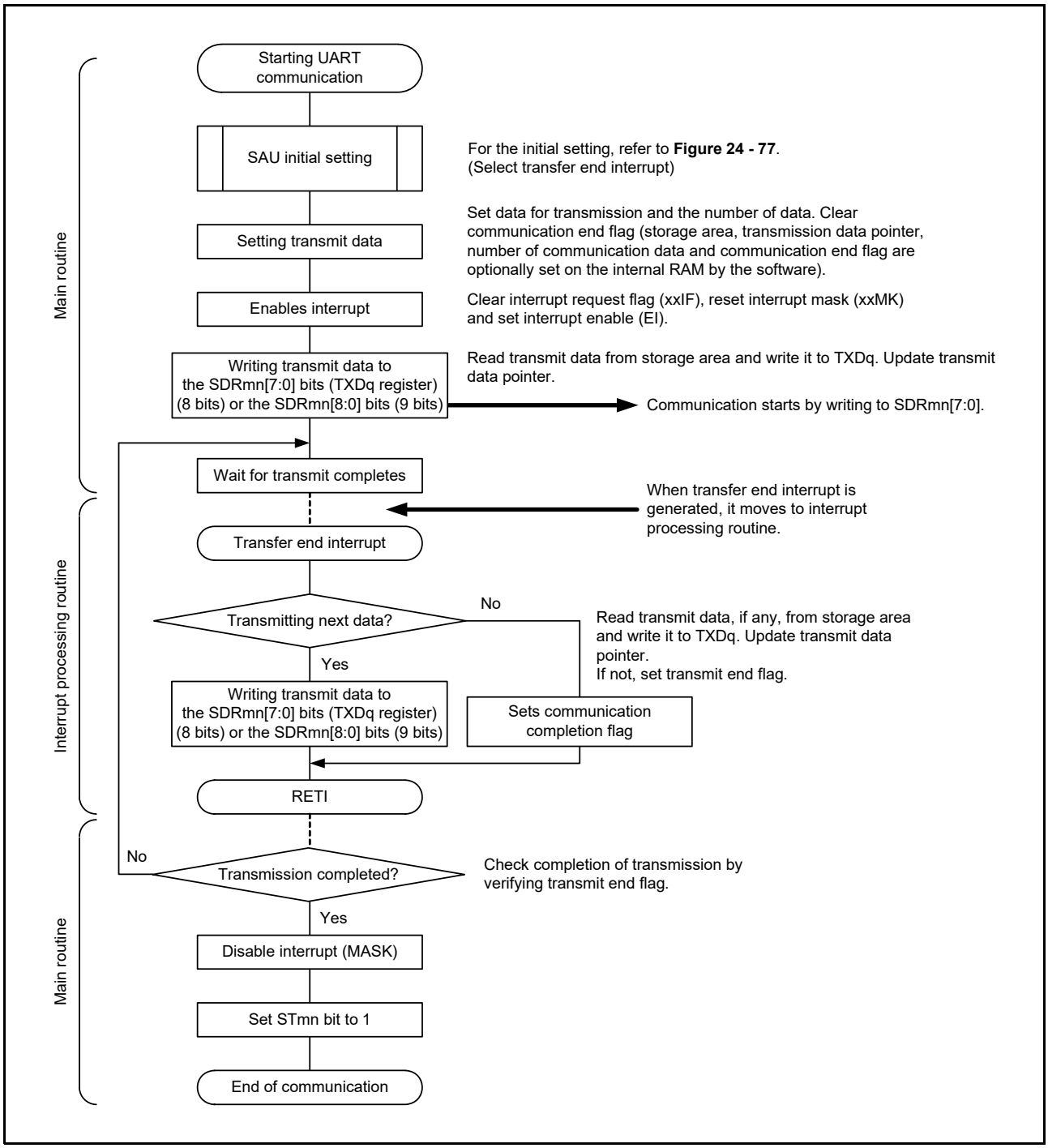
3. Processing flow (in single-transmission mode)

Figure 24 - 80 Timing Chart of UART Transmission (in Single-transmission Mode)



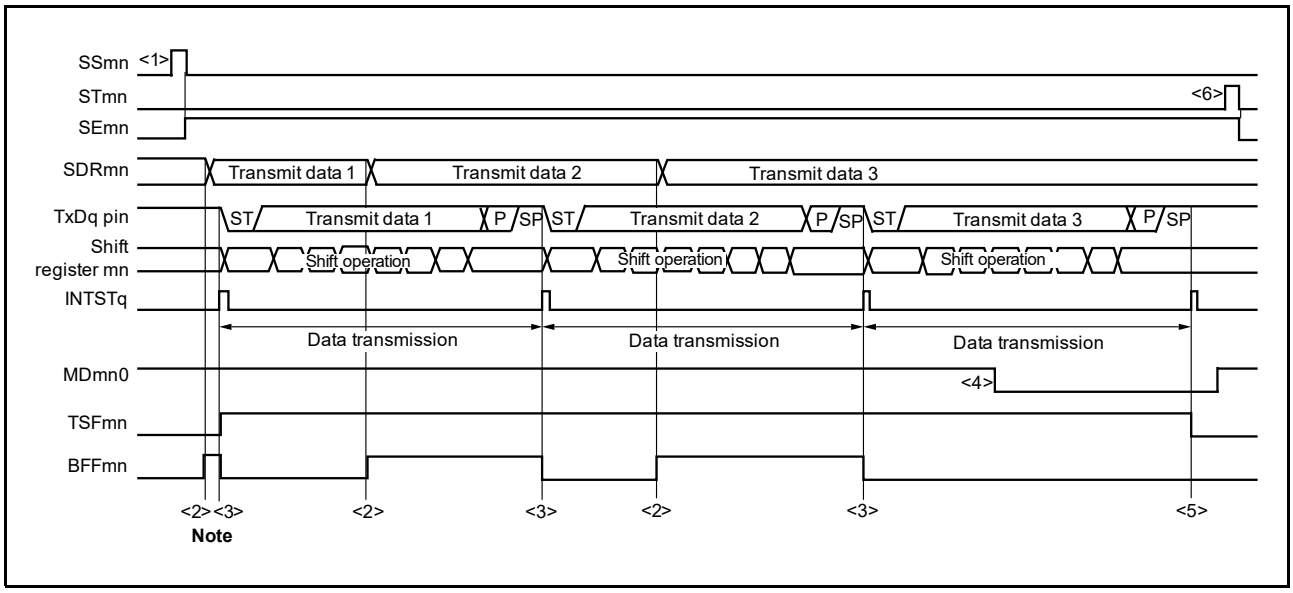
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 2), mn = 00, 02, 10

Figure 24 - 81 Flowchart of UART Transmission (in Single-transmission Mode)



4. Processing flow (in continuous transmission mode)

Figure 24 - 82 Timing Chart of UART Transmission (in Continuous Transmission Mode)

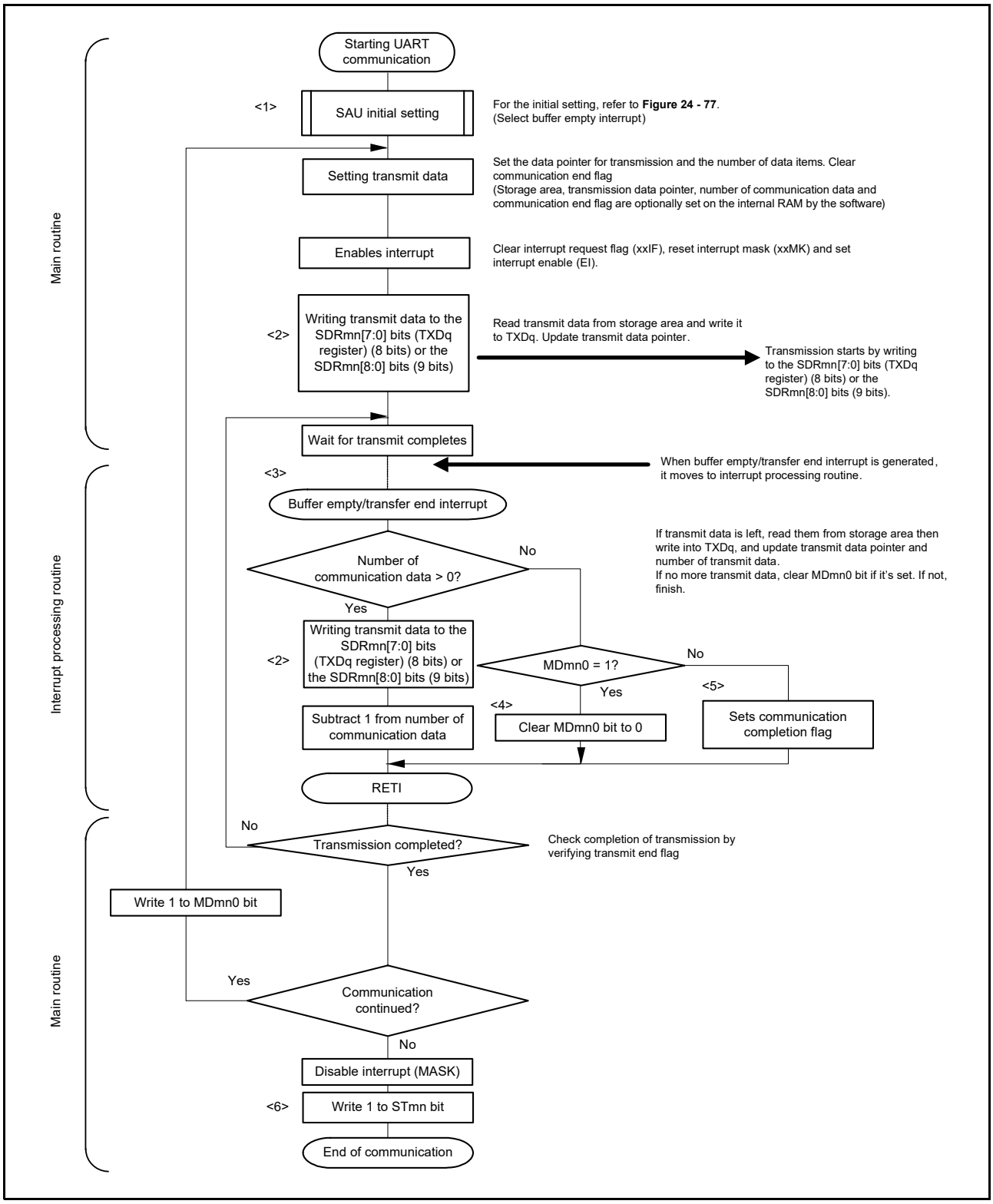


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution **The MDmn0 bit of serial mode register mn (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 m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 2), mn = 00, 02, 10

Figure 24 - 83 Flowchart of UART Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 24 - 82 Timing Chart of UART Transmission (in Continuous Transmission Mode)**.

24.6.2 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	UART2
Target channel	Channel 1 of SAU0	Channel 3 of SAU0	Channel 1 of SAU1
Pins used	RxD0	RxD1	RxD2
Interrupt	INTSR0	INTSR1	INTSR2
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)		
Error interrupt	INTSRE0	INTSRE1	INTSRE2
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**Note 2**	Max. f_{MCK}/6 [bps] (SDRmn[15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]		
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)		
Parity bit	The following selectable • No parity bit (no parity check) • No parity judgment (0 parity) • Even parity check • Odd parity check		
Stop bit	Appending 1 bit		
Data direction	MSB or LSB first		

Note 1. Only UART0 supports the 9-bit data length.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** or **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.

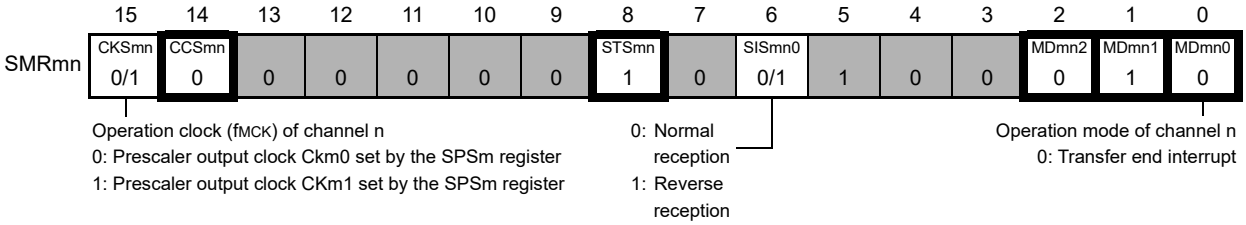
Remark 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11

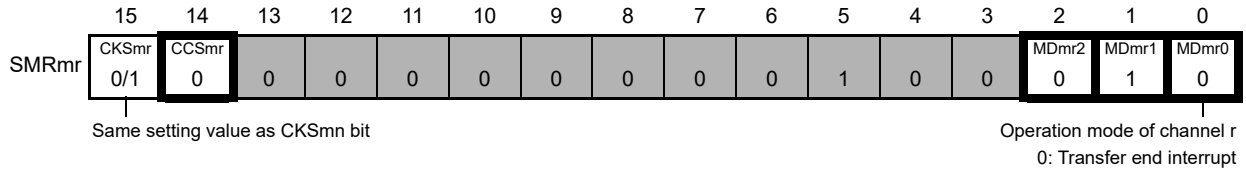
1. Register setting

Figure 24 - 84 Example of Contents of Registers for UART Reception of UART (UART0 to UART2)

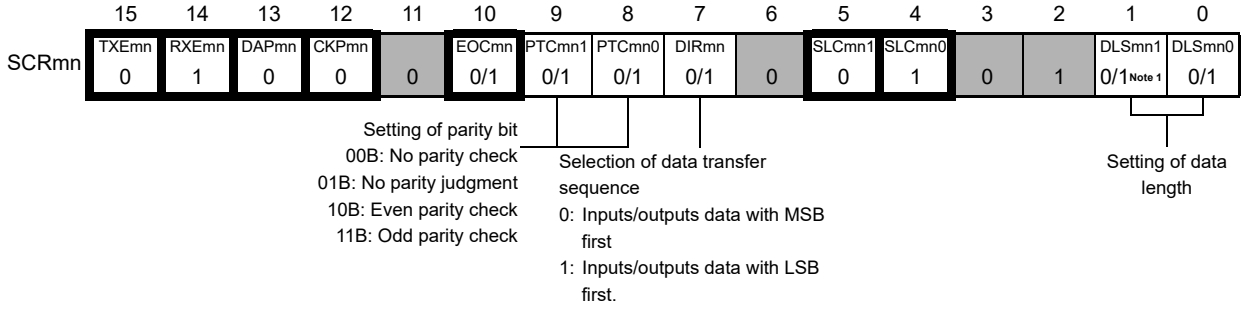
a) Serial mode register mn (SMRmn)



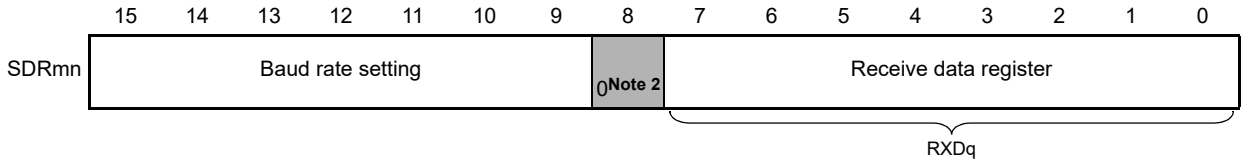
b) Serial mode register mr (SMRmr)



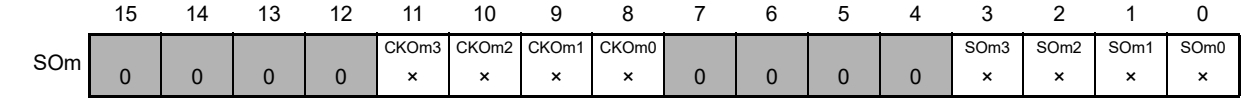
c) Serial communication operation setting register mn (SCRmn)



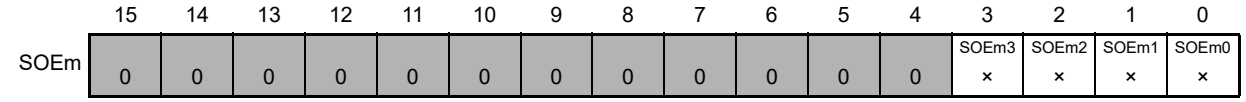
d) Serial data register mn (SDRmn) (lower 8 bits: RXDq)



e) Serial output register m (SOm): This register is not used in this mode.



f) Serial output enable register m (SOEm): This register is not used in this mode.



g) Serial channel start register m (SSm): Set only the bit 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 0/1	SSm2 ×	SSm1 0/1	SSm0 ×

Note 1. Only provided for the SCR01 register. This bit is fixed to 1 for the other registers.

Note 2. When UART performs 9-bit communication, bits 0 to 8 of the SDRm1 register are used as the reception data specification area. Only UART0 supports the 9-bit data length.

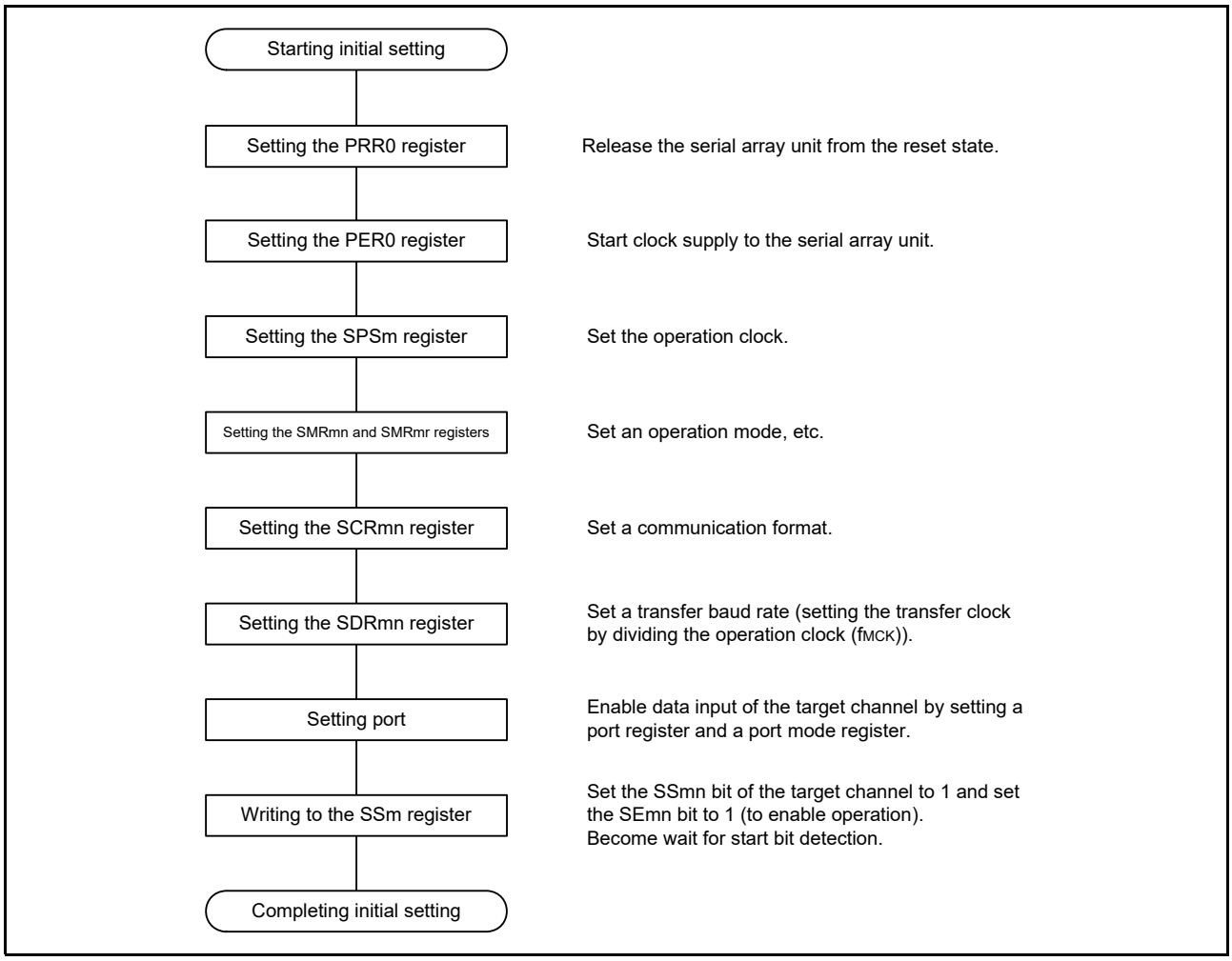
Caution **For the UART reception, set the SMRmr register of channel r to UART transmission mode that is to be paired with channel n.**

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), q: UART number (q = 0 to 2),
r: Channel number (r = n - 1), mn = 01, 03, 11

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

2. Operation procedure

Figure 24 - 85 Initial Setting Procedure for UART Reception



Caution Set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after at least 4 fMCK clock cycles have elapsed.

Figure 24 - 86 Procedure for Stopping UART Reception

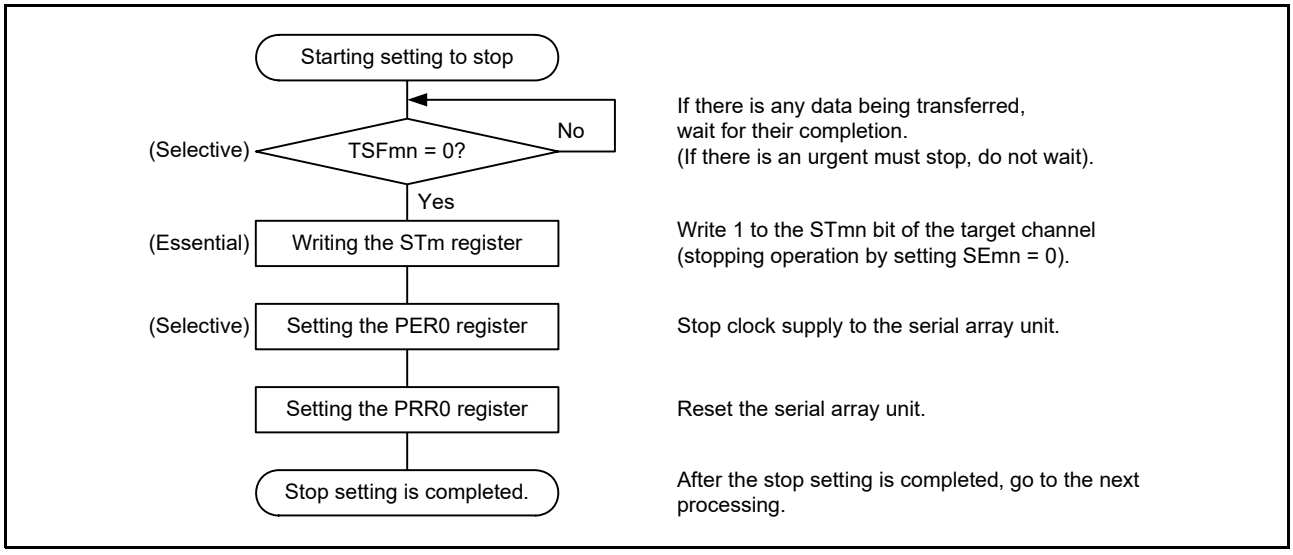
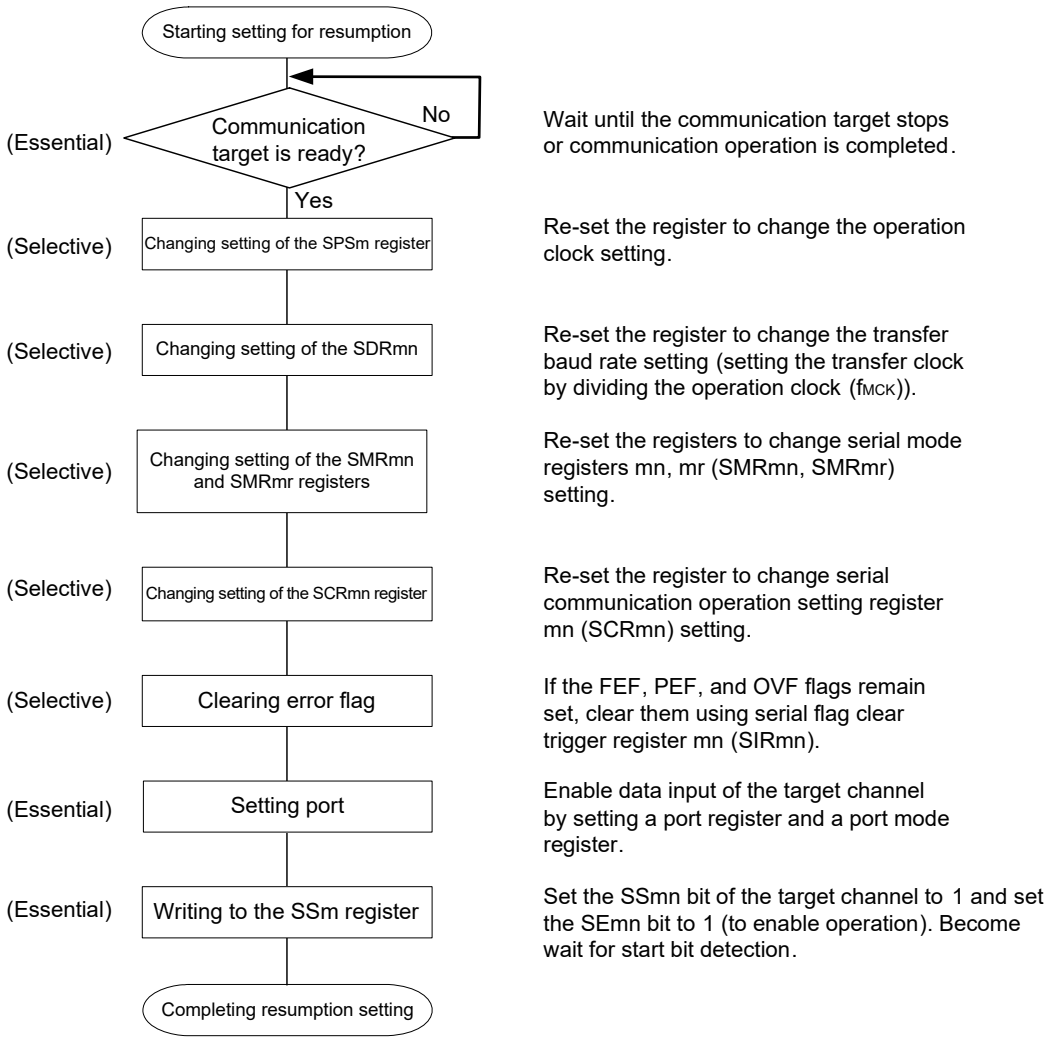


Figure 24 - 87 Procedure for Resuming UART Reception

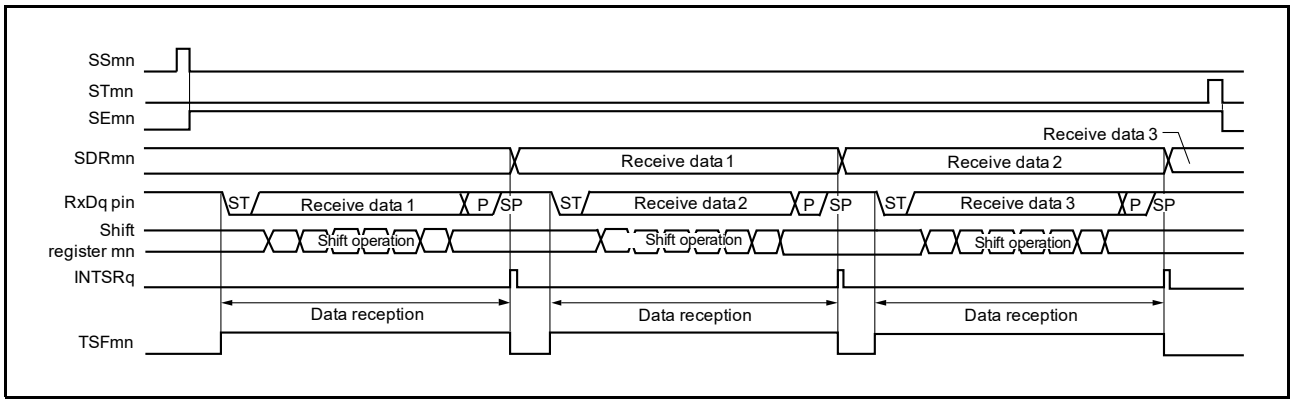


Caution **Set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after at least 4 fMCK clocks have elapsed.**

Remark If PRR0 is rewritten while stopping the communication to reset the serial array unit, wait until the communication target stops or communication finishes, and then perform initialization instead of restarting the communication.

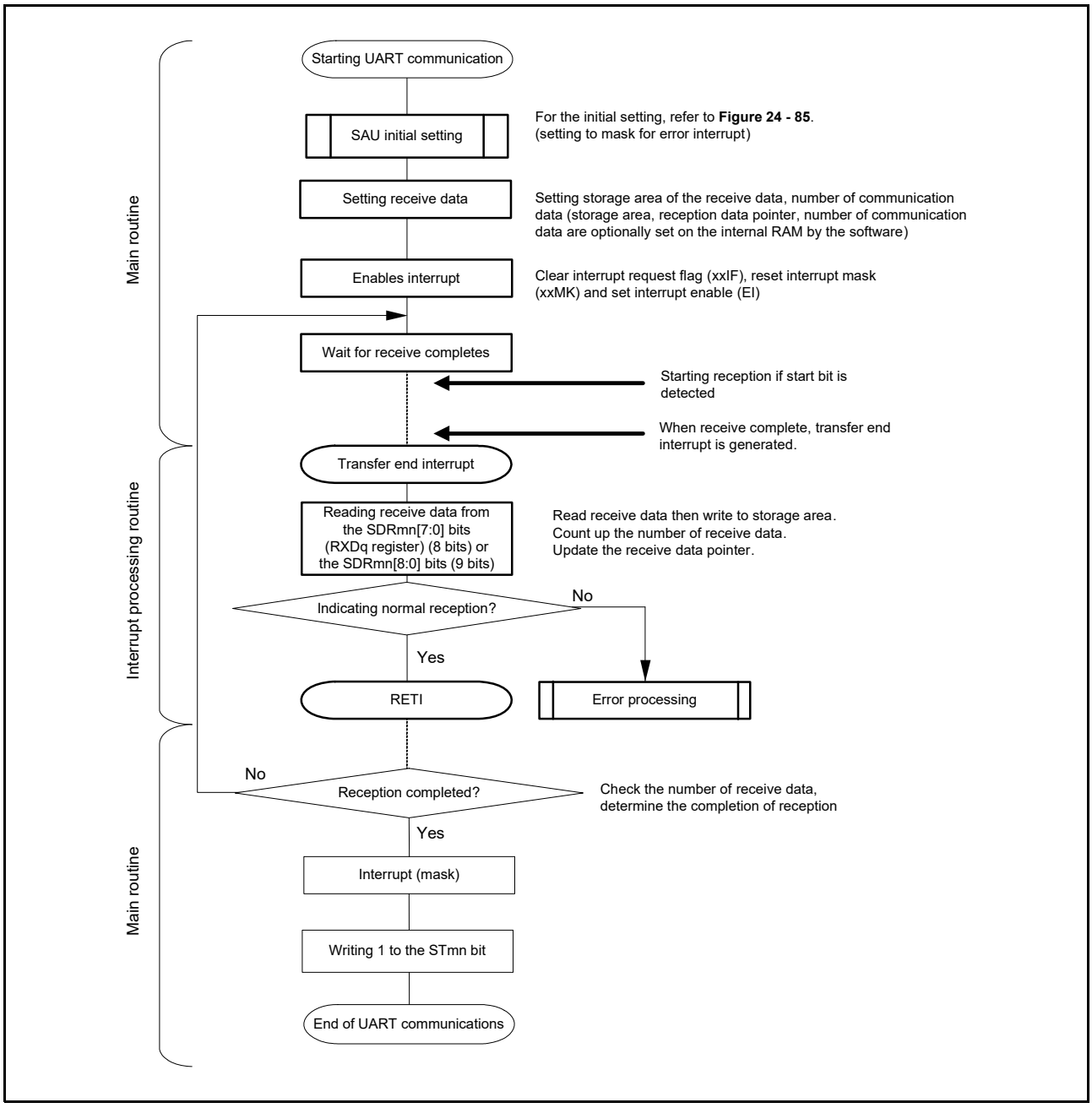
3. Processing flow

Figure 24 - 88 Timing Chart of UART Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), q: UART number (q = 0 to 2), r: Channel number (r = n - 1), mn = 01, 03, 11

Figure 24 - 89 Flowchart of UART Reception



24.6.3 SNOOZE mode

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 UART0 can be set to the SNOOZE mode.

When using UARTq in the SNOOZE mode, make the following settings before entering the STOP mode. (See **Figure 24 - 92 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)** and **Figure 24 - 94 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 24 - 5**.
- Set the EOCmn and SSECm bits. This is for enabling or stopping generation of an error interrupt (INTSREq) when a communication error occurs.
- When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 immediately before switching to the STOP mode. After the initial setting has been completed, set the SSm1 bit of serial channel start register m (SSm) to 1.
- A UARTq starts reception in SNOOZE mode 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 or medium-speed on-chip oscillator clock is selected for f_{CLK}.
When the medium-speed on-chip oscillator clock is selected, use the medium-speed on-chip oscillator trimming register (MIOTRM) to correct the accuracy of the oscillation frequency.

Caution 2. The maximum transfer rate in the SNOOZE mode is 115.2 kbps (when setting FWKUP = 1, $f_{CLK} = f_{IH}$ (32 MHz)).
When FWKUP is set to 1, f_{CLK} cannot be set to a value other than f_{IH} = 32 MHz.

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, and OVFmn flags before setting the SWCm 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 RxD0 signal.
Note, however, that transfer through the UART channel may not start and the CPU may remain in the SNOOZE mode if an input pulse on the RxDq pin 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 transfer.

Table 24 - 5 Setting of the Baud Rate for UART Reception in SNOOZE Mode When Starting of the High-speed On-chip Oscillator at Normal Speed Is Specified (FWKUP = 0)

Baud Rate	High-speed On-chip Oscillator (f_{IH})	Operation Clock (f_{MCK})	SDRmn[15:9]	Maximum Permissible Value	Minimum Permissible Value
4800 bps	32 MHz ± 1.0%Note	$f_{CLK}/2^5$	106	1.45%	-1.67%
	24 MHz ± 1.0%Note	$f_{CLK}/2^5$	79	1.77%	-1.37%
9600 bps	32 MHz ± 1.0%Note	$f_{CLK}/2^4$	106	1.45%	-1.67%
	24 MHz ± 1.0%Note	$f_{CLK}/2^4$	79	1.77%	-1.37%

Table 24 - 6 Setting of the Baud Rate for UART Reception in SNOOZE Mode When Starting of the High-speed On-chip Oscillator at High Speed Is Specified (FWKUP = 1)

Baud Rate	High-speed On-chip Oscillator (f_{IH})	Operation Clock (f_{MCK})	SDRmn [15:9]	Maximum Permissible Value	Minimum Permissible Value
4800 bps	32 MHz ± 1.0%Note	$f_{CLK}/2^5$	106	1.45%	-1.67%
9600 bps		$f_{CLK}/2^4$	106	1.45%	-1.67%
19200 bps		$f_{CLK}/2^3$	106	1.45%	-1.67%
31250 bps		$f_{CLK}/2^3$	65	1.05%	-2.06%
38400 bps		$f_{CLK}/2^2$	106	1.45%	-1.67%
76800 bps		$f_{CLK}/2$	106	1.45%	-1.67%
115200 bps		$f_{CLK}/2$	70	1.93%	-1.21%

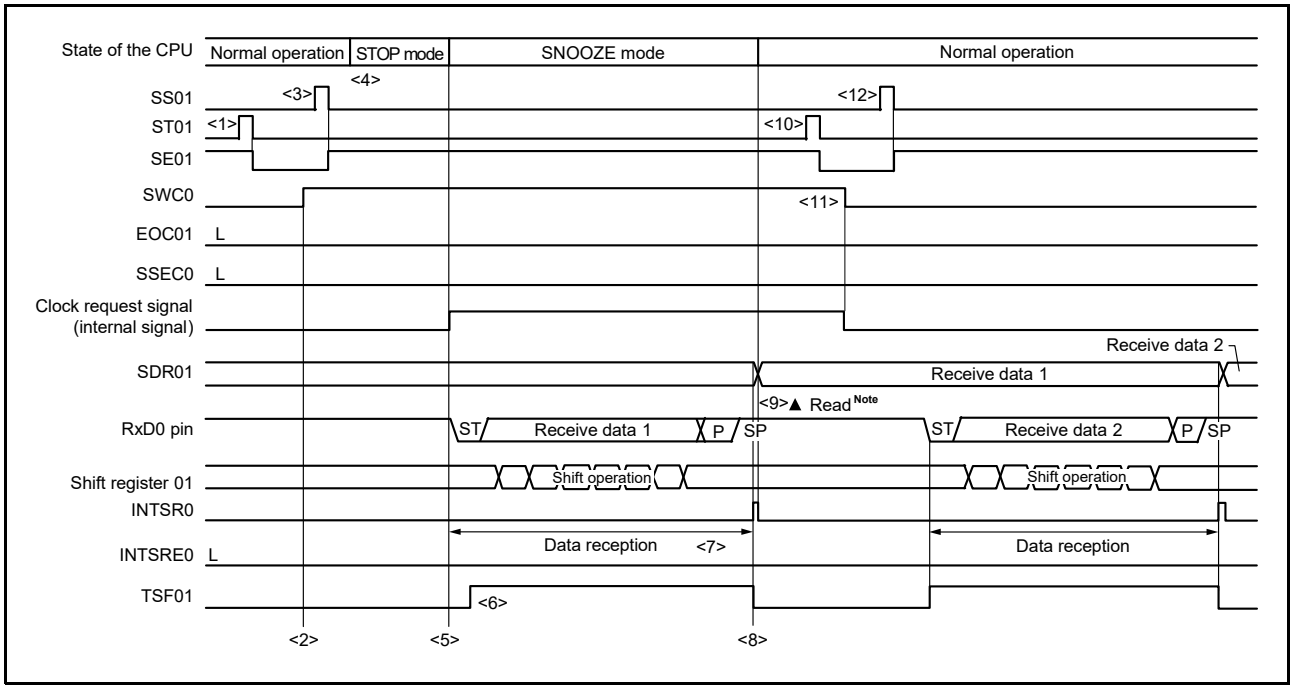
Note When the accuracy of the clock frequency of the high-speed on-chip oscillator is ±1.5% or ±2.0%, the permissible range becomes smaller as shown below.

- In the case of f_{IH} ±1.5%, perform (Maximum permissible value – 0.5%) and (Minimum permissible value + 0.5%) to the values in the above table.
- In the case of f_{IH} ±2.0%, perform (Maximum permissible value – 1.0%) and (Minimum permissible value + 1.0%) to the values in the above table.

Remark The maximum permissible value and minimum permissible value are permissible values for the baud rate in UART reception.

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 24 - 90 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)



Note Read the received data when SWCm = 1.

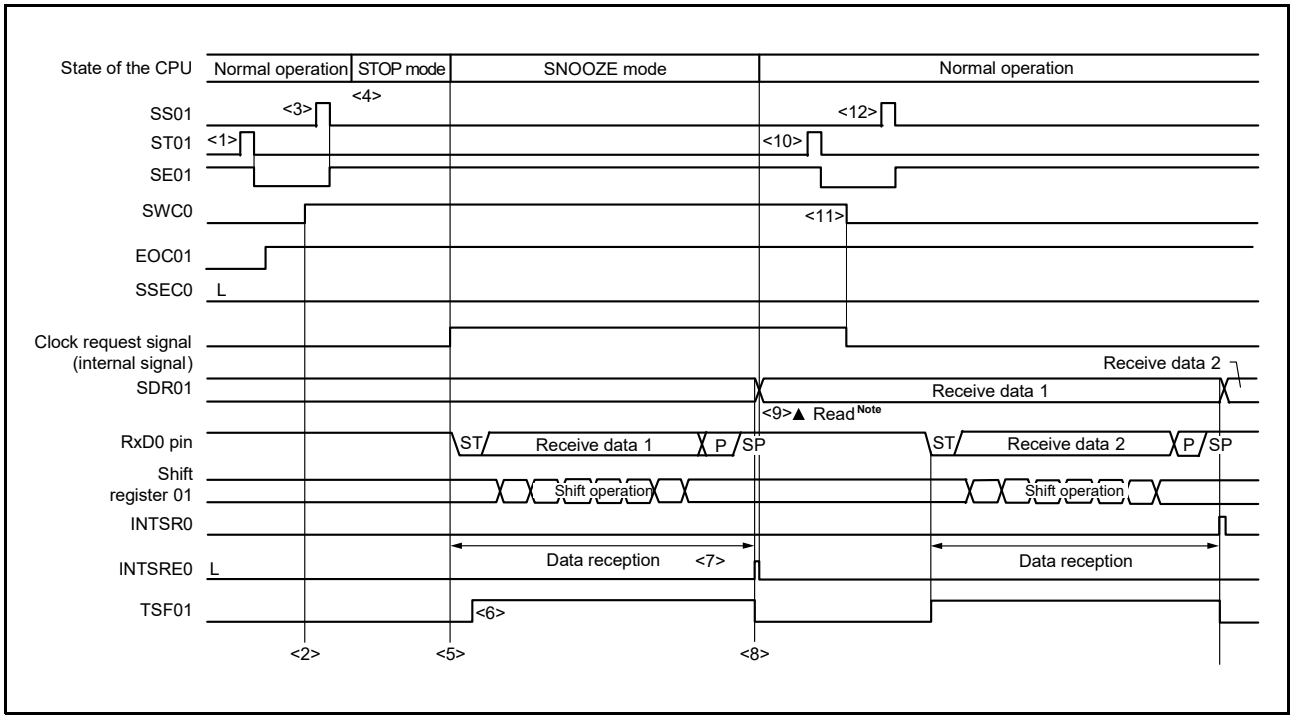
Caution **Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (the SEm1 bit is cleared and the operation stops).**
After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).

Remark 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 24 - 92 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 24 - 91 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)



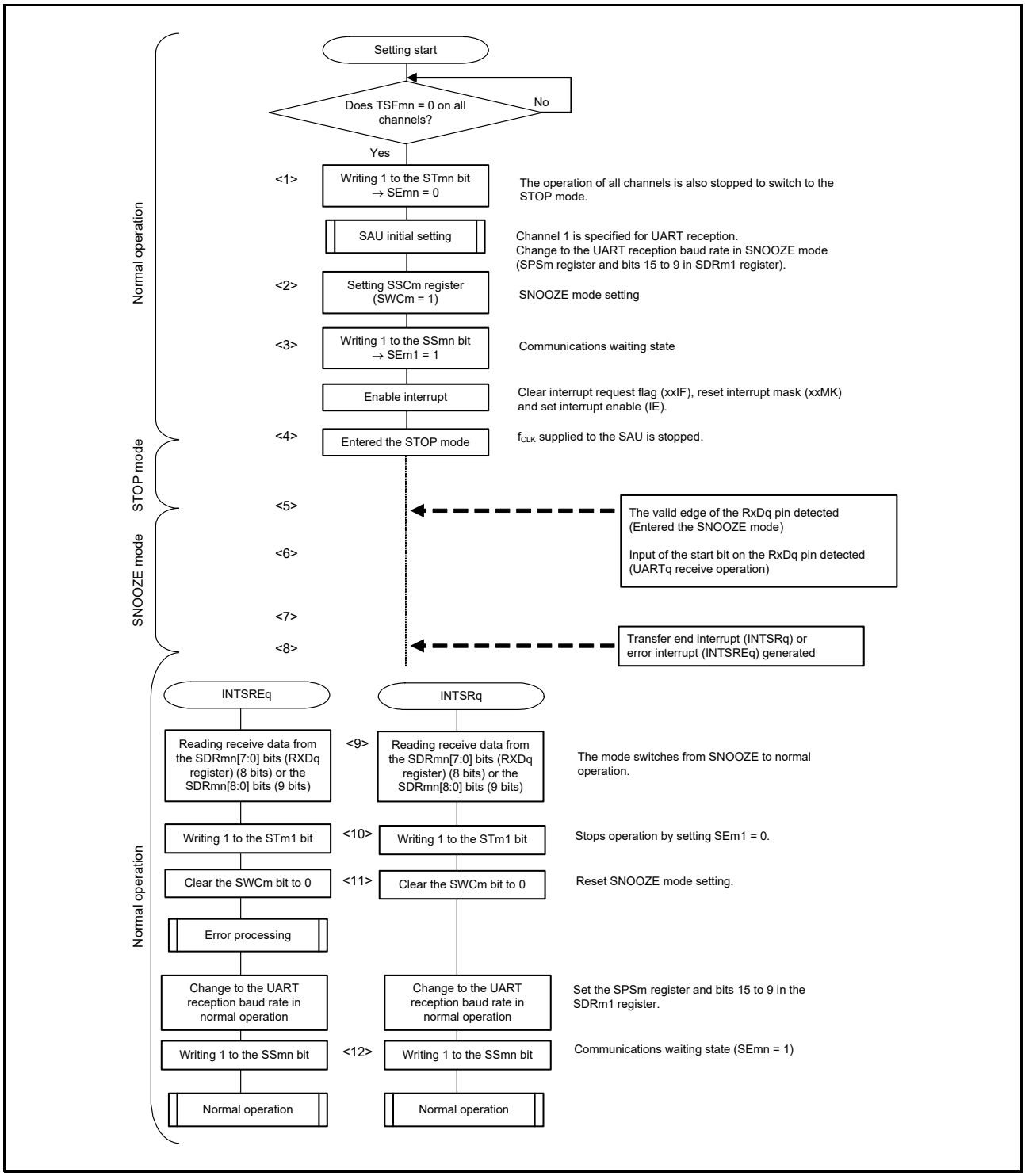
Note Read the received data when SWCm = 1.

Caution **Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (the SEm1 bit is cleared and the operation stops).**
After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).

Remark 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 24 - 92 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)**.

Remark 2. m = 0; q = 0

Figure 24 - 92 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)

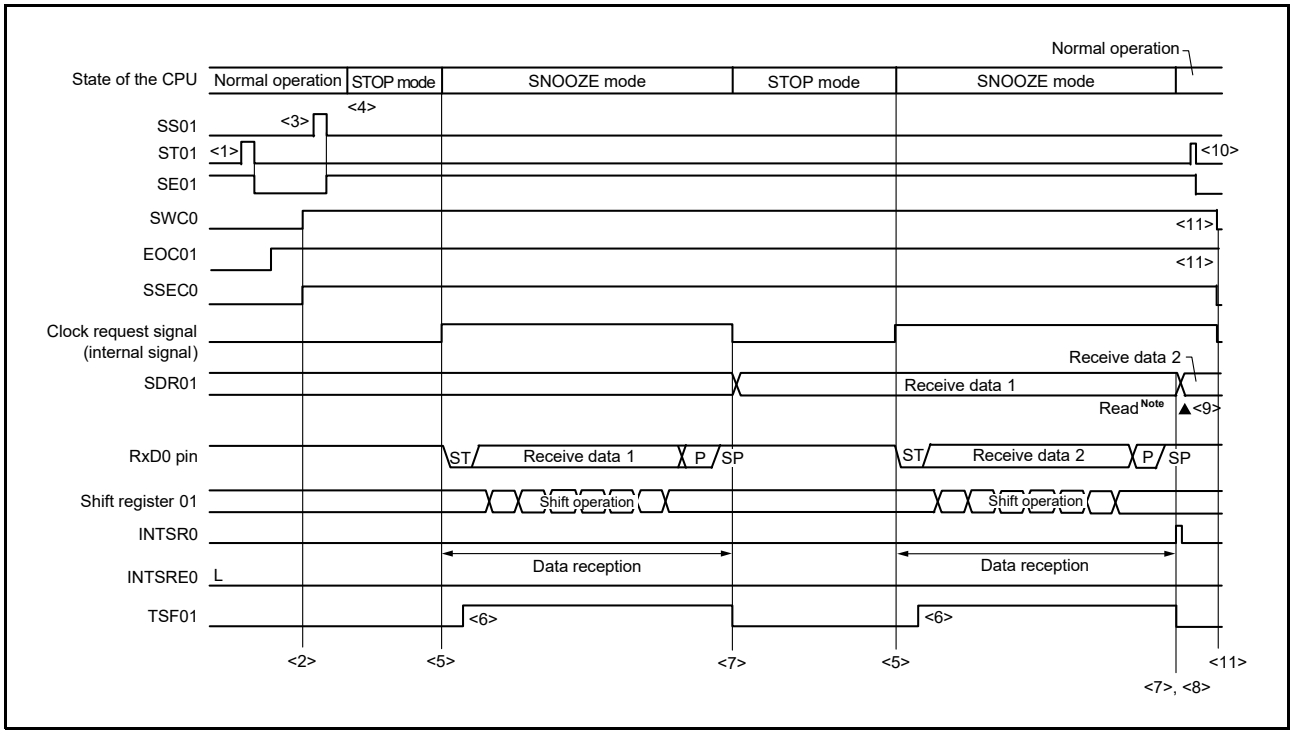


Remark 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 24 - 90 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)** and **Figure 24 - 91 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)**.

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 24 - 93 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



Note Read the received data when SWCm = 1.

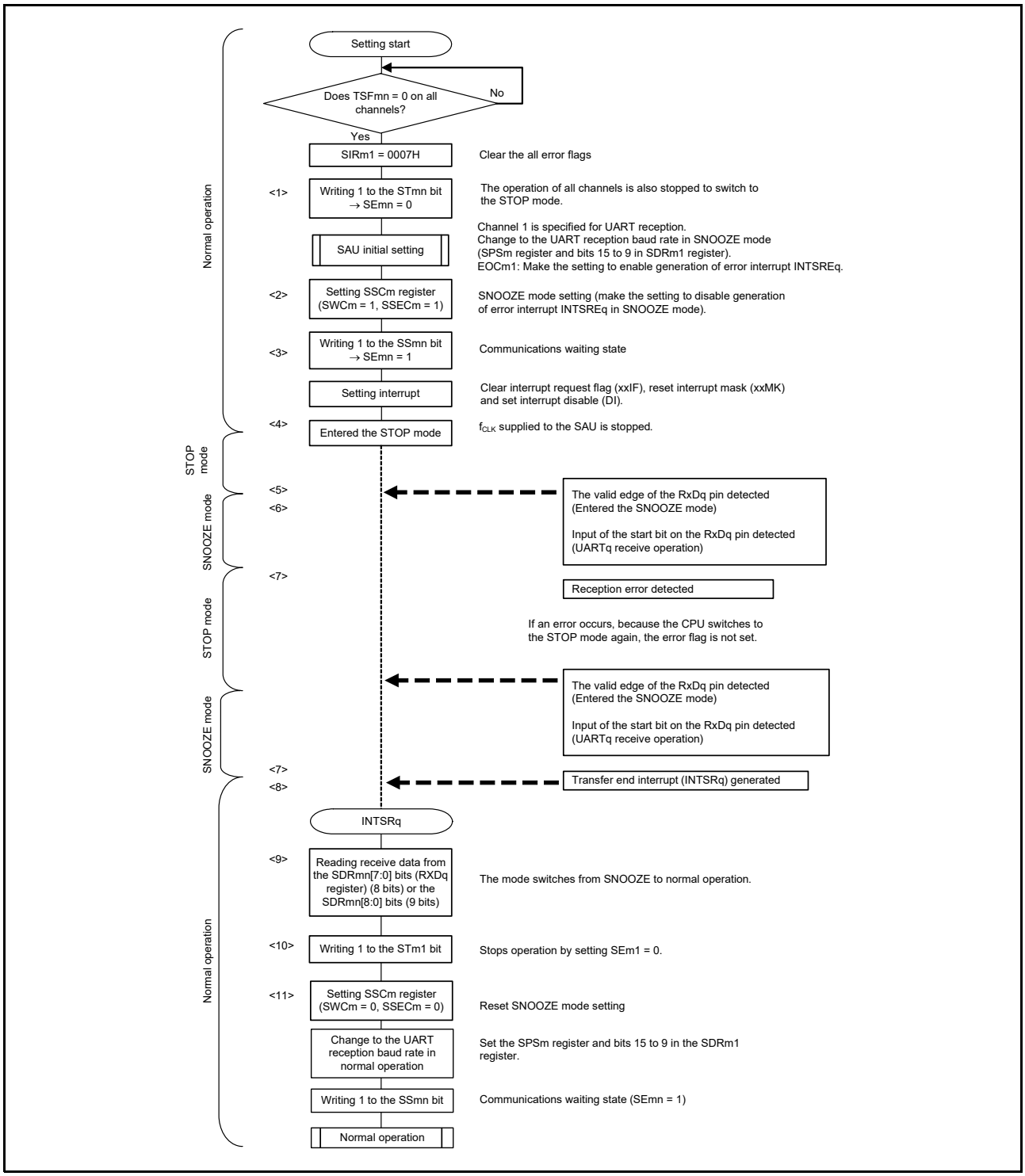
Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (the SEm1 bit is cleared and the operation stops).
After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).

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, and OVFm1 flags 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 24 - 94 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)**.

Remark 2. m = 0; q = 0

Figure 24 - 94 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



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, and OVFm1 flags 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 24 - 93 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)**.

Remark 2. m = 0, q = 0, n = 0 to 3

24.6.4 Calculating baud rate

1. Baud rate calculation expression
   The baud rate for UART (UART0 to UART2) 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, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

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 24 - 7 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} = 32 MHz	f_{CLK} = 48 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	32 MHz	Setting prohibited
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	16 MHz	24 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	8 MHz	12 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	4 MHz	6 MH
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	2 MHz	3 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	1 MHz	1.5 MHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	500 kHz	750 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	250 kHz	375 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	125 kHz	188 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz	93.8 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz	46.9 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz	23.4 kHz
	×	×	×	×	1	1	0	0	$f_{CLK}/2^{12}$	7.81 kHz	11.7 kHz
	×	×	×	×	1	1	0	1	$f_{CLK}/2^{13}$	3.91 kHz	5.86 kHz
	×	×	×	×	1	1	1	0	$f_{CLK}/2^{14}$	1.95 kHz	2.93 kHz
	×	×	×	×	1	1	1	1	$f_{CLK}/2^{15}$	977 Hz	1.46 kHz
1	0	0	0	0	×	×	×	×	f_{CLK}	32 MHz	Setting prohibited
	0	0	0	1	×	×	×	×	$f_{CLK}/2$	16 MHz	24 MHz
	0	0	1	0	×	×	×	×	$f_{CLK}/2^2$	8 MHz	12 MHz
	0	0	1	1	×	×	×	×	$f_{CLK}/2^3$	4 MHz	6 MH
	0	1	0	0	×	×	×	×	$f_{CLK}/2^4$	2 MHz	3 MHz
	0	1	0	1	×	×	×	×	$f_{CLK}/2^5$	1 MHz	1.5 MHz
	0	1	1	0	×	×	×	×	$f_{CLK}/2^6$	500 kHz	750 kHz
	0	1	1	1	×	×	×	×	$f_{CLK}/2^7$	250 kHz	375 kHz
	1	0	0	0	×	×	×	×	$f_{CLK}/2^8$	125 kHz	188 kHz
	1	0	0	1	×	×	×	×	$f_{CLK}/2^9$	62.5 kHz	93.8 kHz
	1	0	1	0	×	×	×	×	$f_{CLK}/2^{10}$	31.25 kHz	46.9 kHz
	1	0	1	1	×	×	×	×	$f_{CLK}/2^{11}$	15.63 kHz	23.4 kHz
	1	1	0	0	×	×	×	×	$f_{CLK}/2^{12}$	7.81 kHz	11.7 kHz
	1	1	0	1	×	×	×	×	$f_{CLK}/2^{13}$	3.91 kHz	5.86 kHz
	1	1	1	0	×	×	×	×	$f_{CLK}/2^{14}$	1.95 kHz	2.93 kHz
	1	1	1	1	×	×	×	×	$f_{CLK}/2^{15}$	977 Hz	1.46 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, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

2. Baud rate error during transmission

The baud rate error of UART (UART0 to UART3) 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} = 32 MHz.

UART Baud Rate (Target Baud Rate)	f_{CLK} = 32 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{CLK}/2^9$	103	300.48 bps	+0.16%
600 bps	$f_{CLK}/2^8$	103	600.96 bps	+0.16%
1200 bps	$f_{CLK}/2^7$	103	1201.92 bps	+0.16%
2400 bps	$f_{CLK}/2^6$	103	2403.85 bps	+0.16%
4800 bps	$f_{CLK}/2^5$	103	4807.69 bps	+0.16%
9600 bps	$f_{CLK}/2^4$	103	9615.38 bps	+0.16%
19200 bps	$f_{CLK}/2^3$	103	19230.8 bps	+0.16%
31250 bps	$f_{CLK}/2^3$	63	31250.0 bps	±0.0%
38400 bps	$f_{CLK}/2^2$	103	38461.5 bps	+0.16%
76800 bps	$f_{CLK}/2$	103	76923.1 bps	+0.16%
153600 bps	f_{CLK}	103	153846 bps	+0.16%
312500 bps	f_{CLK}	50	313725.5 bps	+0.39%

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10

3. Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0 to UART2) 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 **24.6.4 Calculating baud rate**, **1. Baud rate calculation expression**.)

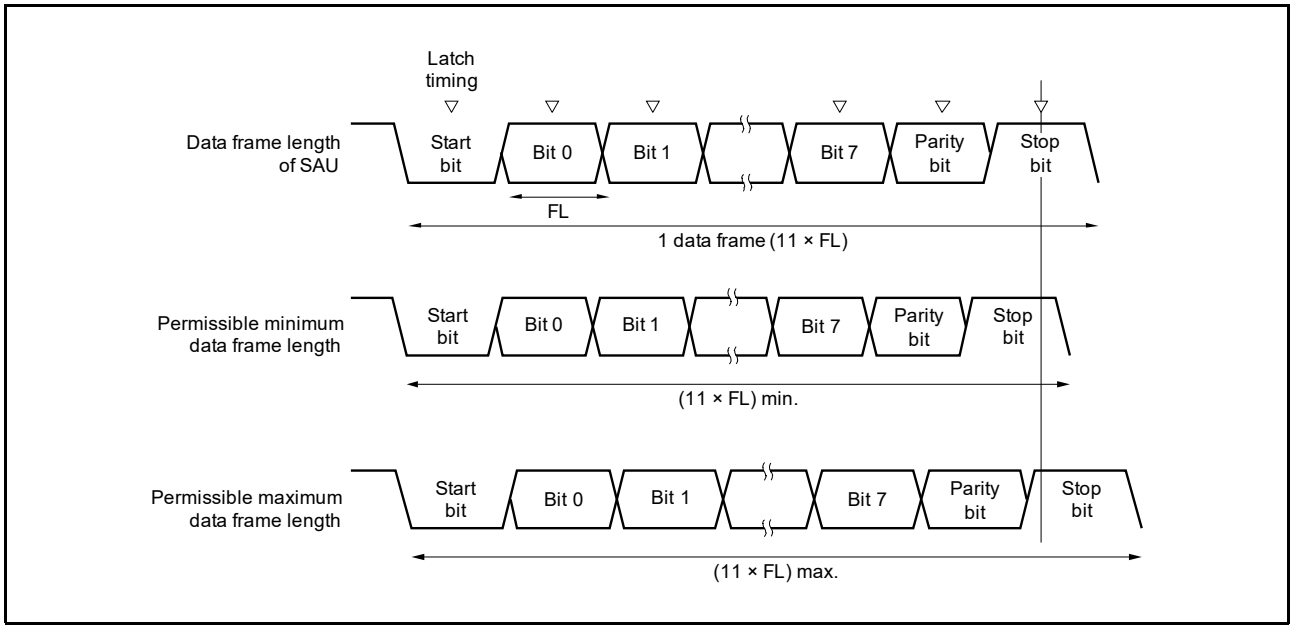
k: SDRmn[15:9] + 1

Nfr: 1 data frame length [bits]

= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11

Figure 24 - 95 Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in **Figure 24 - 95**, 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.

24.6.5 Procedure for processing errors that occurred during UART (UART0 to UART2) communication

The procedure for processing errors that occurred during UART (UART0 to UART3) communication is described in **Tables 24 - 8** and **24 - 9**.

Table 24 - 8 Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation	State of the Hardware	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.	Only the error generated during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

Table 24 - 9 Processing Procedure in Case of Framing Error

Software Manipulation	State of the Hardware	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 serial flag clear trigger register mn (SIRmn). →	The error flag is cleared.	Only the error generated during reading can be cleared, 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.	
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, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

24.7 LIN Communication Operation

24.7.1 LIN transmission

Of UART transmission, UART0 supports LIN communication.

For LIN transmission, channel 0 of unit 0 is used.

UART	UART0	UART1	UART2
Support of LIN communication	Supported	Not supported	Not supported
Target channel	Channel 0 of SAU0	—	—
Pins used	TxD0	—	—
Interrupt	INTST0	—	—
	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	8 bits		
Transfer rate**Note**	Max. f_{MCK}/6 [bps] (SDR00[15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]		
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)		
Parity bit	No parity bit		
Stop bit	Appending 1 bit		
Data direction	LSB first		

Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** or **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**. In general, 2.4, 9.6, or 19.2 kbps is often used in LIN communication.

Remark f_{MCK}: Operation clock frequency of target channel

f_{CLK}: System clock frequency

LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol designed to reduce the cost of an automobile network.

Communication of LIN is single-master communication and up to 15 slaves can be connected to one master.

The slaves are used to control switches, actuators, and sensors, which are connected to the master via LIN.

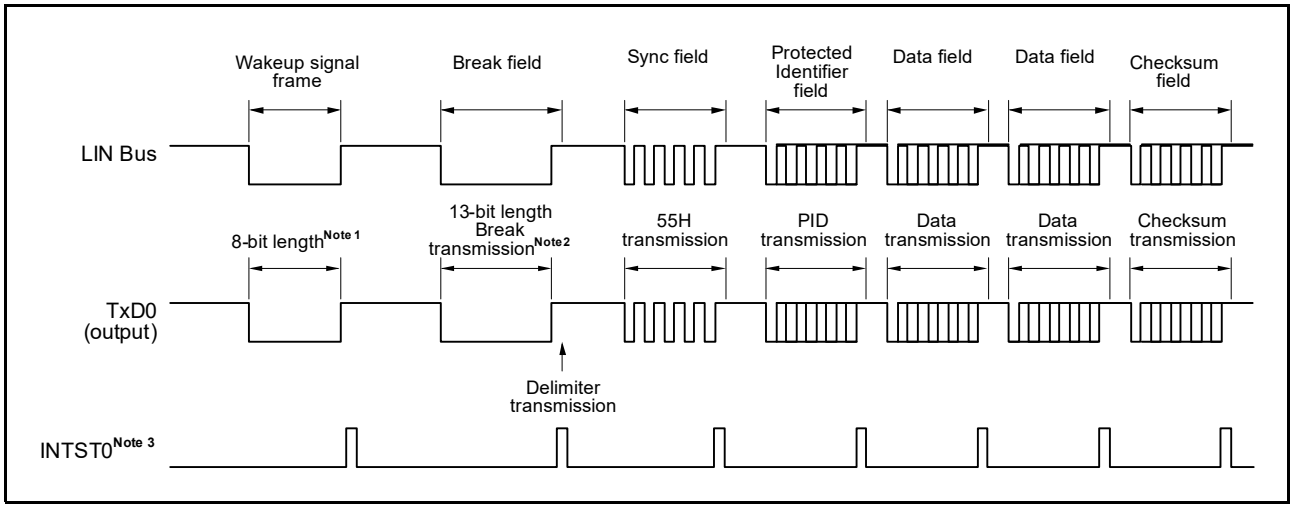
Usually, the master is connected to a network such as CAN (Controller Area Network).

A LIN bus is a single-wire bus to which nodes are connected via transceiver conforming to ISO9141.

According to the protocol of LIN, the master transmits a frame by attaching baud rate information to it. A slave receives this frame and corrects a baud rate error from the master. If the baud rate error of a slave is within ±15%, communication can be established.

Figure 24 - 96 outlines a transmission operation of LIN.

Figure 24 - 96 Transmission Operation of LIN



Note 1. Set the baud rate in accordance with the wakeup signal regulations and transmit data of 80H.

Note 2. A break field is defined to have a width of 13 bits and output a low level. Where the baud rate for main transfer is N [bps], therefore, the baud rate of the break field is calculated as follows.

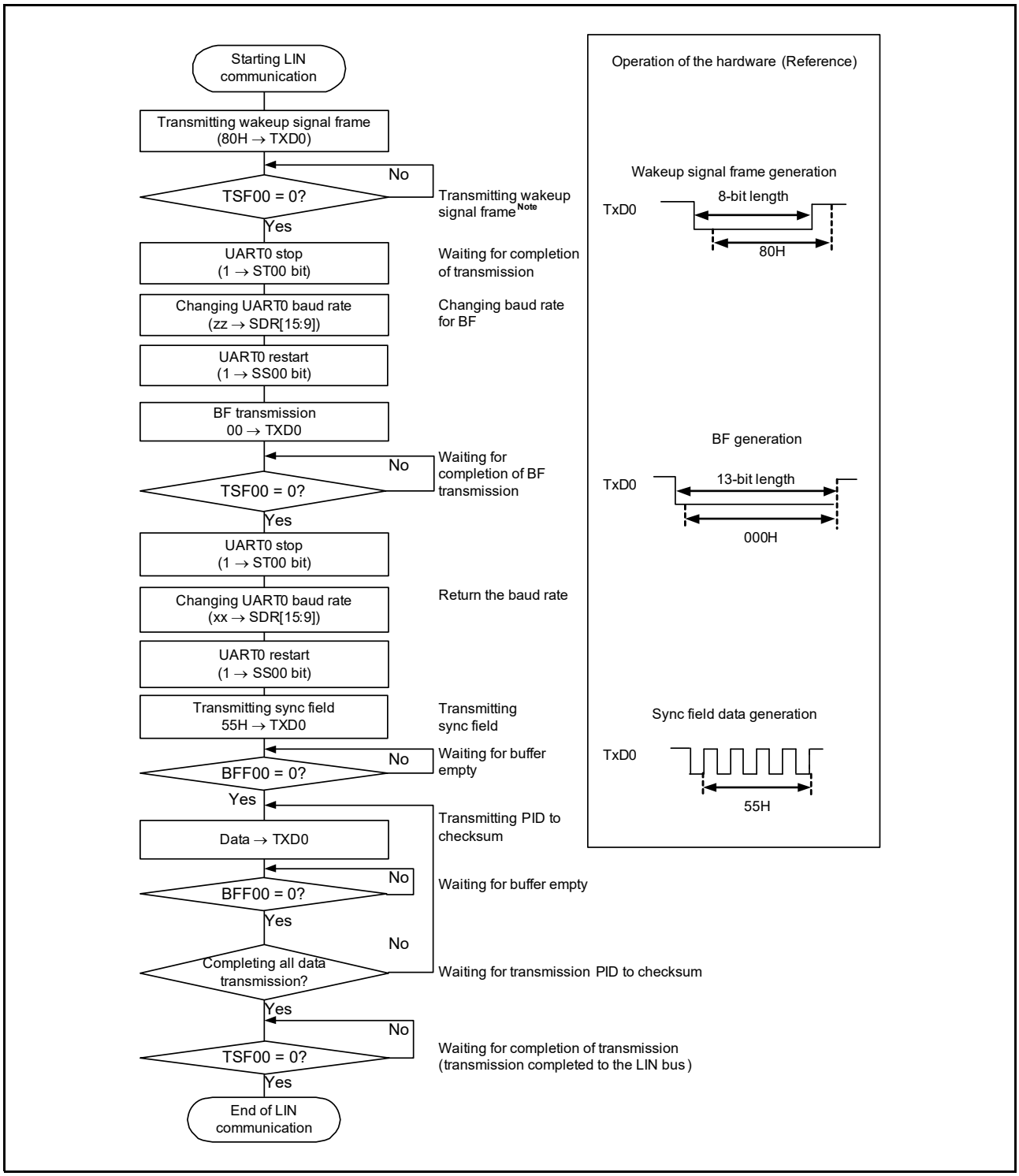
(Baud rate of break field) = 9/13 × N

By transmitting data of 00H at this baud rate, a break field is generated.

Note 3. INTST0 is output upon completion of transmission. INTST0 is also output at BF transmission.

Remark The interval between fields is controlled by software.

Figure 24 - 97 Flowchart for LIN Transmission



Note This is only required if the LIN-bus is being started from sleep mode.

Remark This flow assumes that the initial setting of the UART is completed and transmission is enabled.

24.7.2 LIN reception

Of UART reception, UART0 supports LIN communication.
For LIN reception, channel 1 of unit 1 is used.

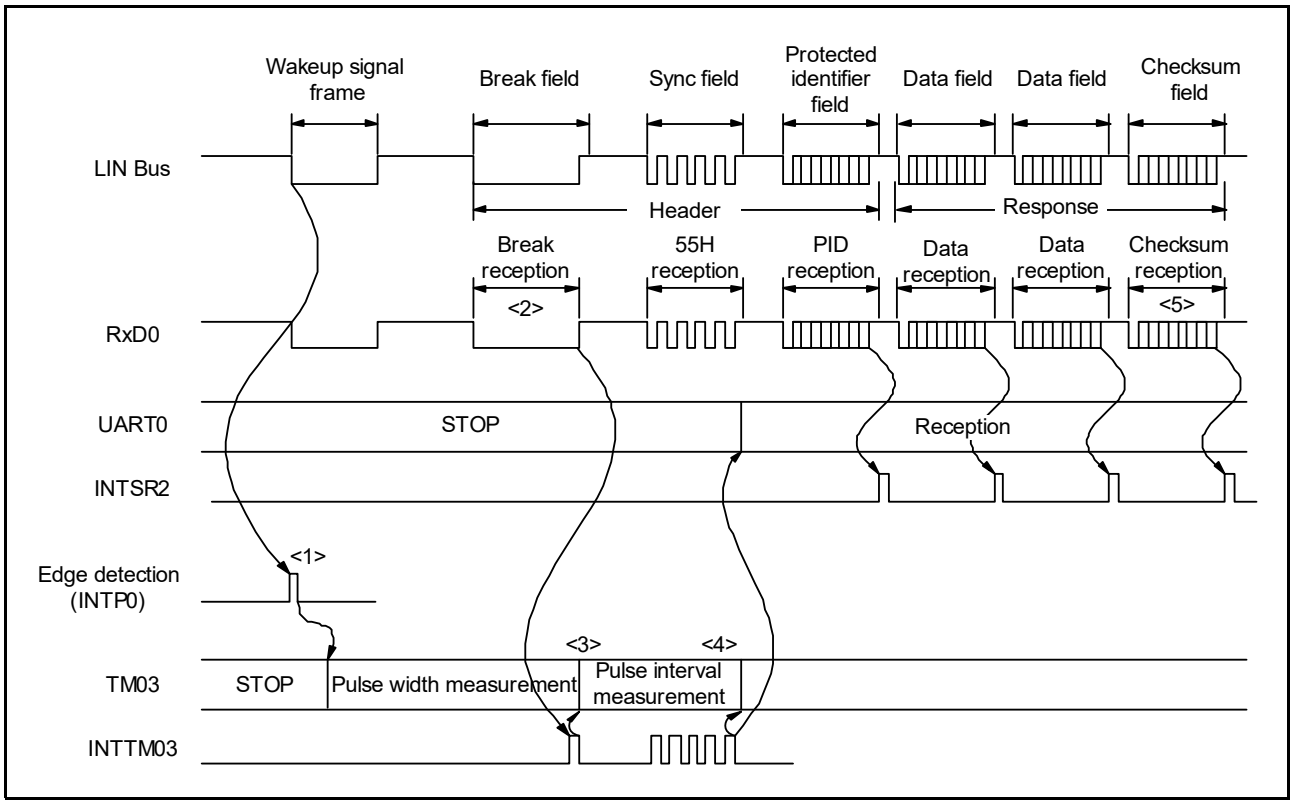
UART	UART0	UART1	UART2
Support of LIN communication	Supported	Not supported	Not supported
Target channel	Channel 1 of SAU0	—	—
Pins used	RxD0	—	—
Interrupt	INTSR0	—	—
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)		
Error interrupt	INTSRE0	—	—
Error detection flag	• Framing Error detection flag (FEF01) • Overrun Error detection flag (OVF01)		
Transfer data length	8 bits		
Transfer rate**Note**	Max. f_{MCK}/6 [bps] (SDR01[15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]		
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)		
Parity bit	No parity bit (The parity bit is not checked.)		
Stop bit	Check the first bit		
Data direction	LSB first		

Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** or **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.

Remark f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

Figure 24 - 98 outlines a reception operation of LIN.

Figure 24 - 98 Reception Operation of LIN



Here is the flow of reception processing.

<1> The wakeup signal is detected by detecting an edge on the INTP0 interrupt pin. When the wakeup signal is detected, set TM03 to the pulse width measurement function to measure the low-level width of the BF signal. Then wait for BF signal reception.

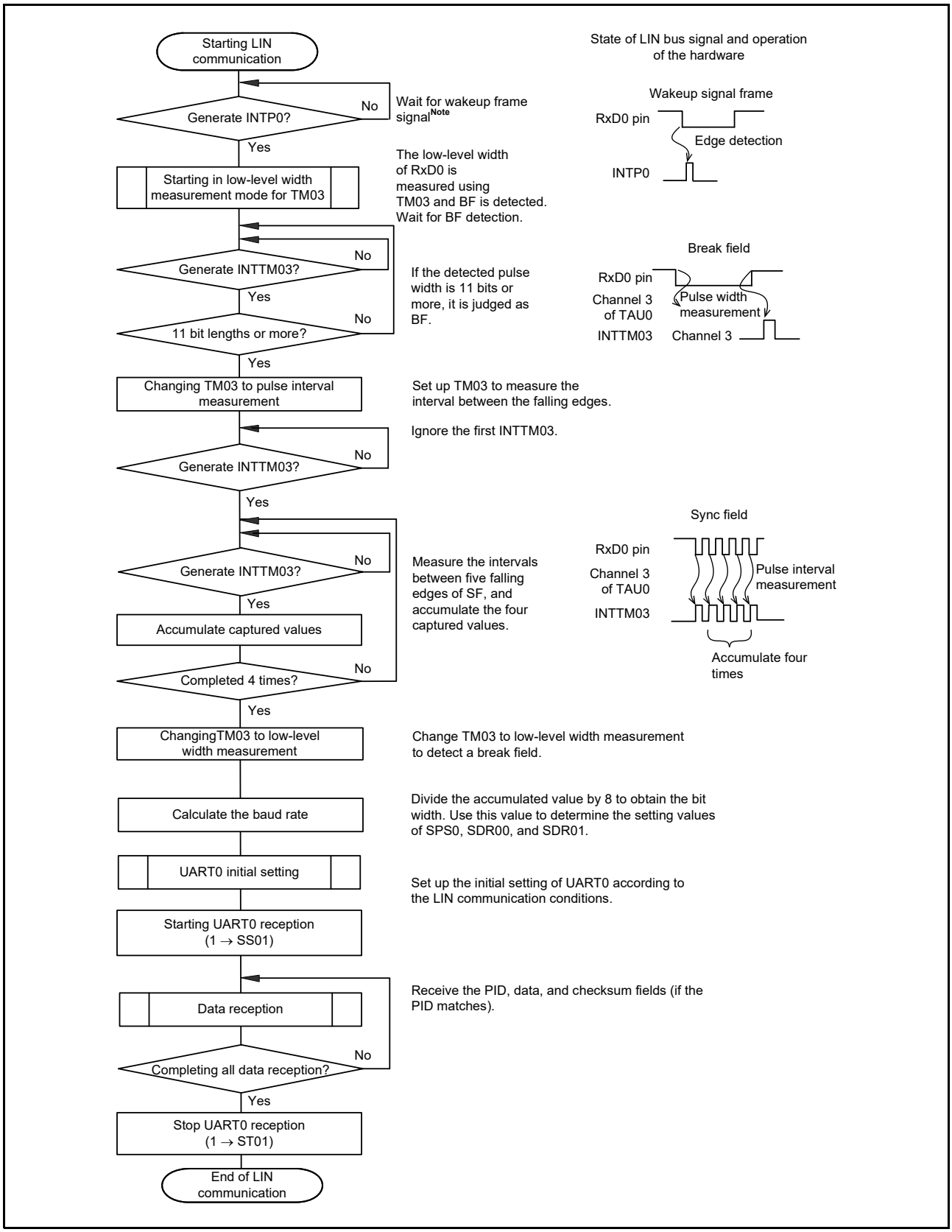
<2> TM03 starts measuring the low-level width upon detection of the falling edge of the BF signal, and then captures the data upon detection of the rising edge of the BF signal. The captured data is used to judge whether it is the BF signal.

<3> When the BF signal has been received normally, change TM03 to pulse interval measurement and measure the interval between the falling edges of the RxD0 signal in the Sync field four times. (see **10.8.4 Operation for input pulse interval measurement**.)

<4> Calculate a baud rate error from the bit interval of sync field (SF). Stop UART0 once and adjust (re-set) the baud rate.

<5> The checksum field should be distinguished by software. In addition, processing to initialize UART0 after the checksum field is received and to wait for reception of BF should also be performed by software.

Figure 24 - 99 Flowchart of LIN Reception



Note This is only required if the LIN bus is in sleep mode.

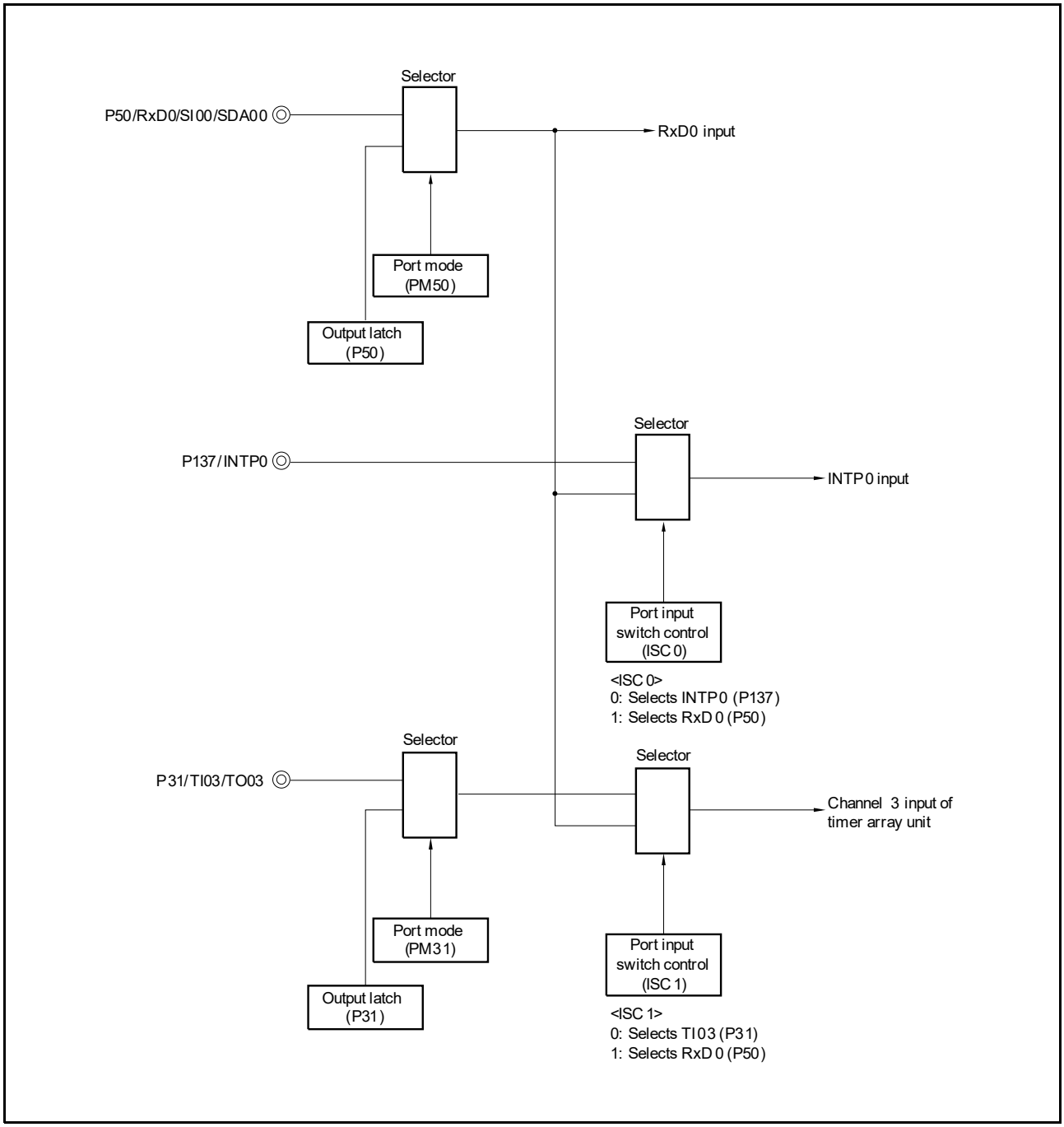
Figure 24 - 100 shows the configuration of ports used for LIN reception.

The wakeup signal transmitted from the master of LIN is received by detecting an edge of an external interrupt (INTP0).

The length of the sync field transmitted from the master can be measured by using the external event capture operation of the timer array unit 0 to calculate a baud-rate error.

By using the port input switching control (the ISC0 and ISC1 bits), the signal input to the reception port (RxD0) can be used as an external interrupt (INTP0) or sent to the timer array unit without additional external connections.

Figure 24 - 100 Port Configuration for Reception of LIN



Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (See **Figure 24 - 21**.)

The peripheral functions used for the LIN communication operation are as follows.

<Peripheral functions used>

- External interrupt (INTP0); Wakeup signal detection
  Usage: To detect an edge of the wakeup signal and the start of communication
- Channel 3 of timer array unit; Baud rate error detection, break field detection.
  Usage: To detect the length of the sync field (SF) and divide it by the number of bits in order to detect a baud rate error. (The interval of the edge input to RxD0 is measured in the capture mode.)
  To measure the low-level width to detect the break field (BF).
- Channels 0 and 1 (UART0) of serial array unit 0 (SAU0)

24.8 Operation of Simplified I2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) Communication

Simplified I2C is a clock synchronous communications function for use in communicating with two or more devices over two lines: serial clock (SCL) and serial data (SDA). This simplified I2C is designed for single communications with devices such as an EEPROM, flash memory, or an A/D converter, and therefore, it only functions as a master.
Operate the control registers by software for setting the start and stop conditions while observing the specifications of the I2C bus line.

[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 I2C]

- Slave transmission, slave reception
- Multi-master function (arbitration loss detection)
- 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 **24.8.3**, **2. Processing flow** for details.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

The channel supporting simplified I2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) is channels 0 to 3 of SAU0 and channel 0 and 1 of SAU1.

<20-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I2C
0	0	—	—	—
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

<24- and 25-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I2C
0	0	—	UART0 (supporting LIN-bus)Note	—
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

<30- and 32-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I2C
0	0	CSI00 (supporting slave select input)	UART0 (supporting LIN-bus)	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

<40- and 44-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I2C
0	0	CSI00 (supporting slave select input)	UART0 (supporting LIN-bus)	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	CSI21		IIC21

<48- and 52-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00 (supporting slave select input)	UART0 (supporting LIN-bus)	IIC00
	1	CSI01		IIC01
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	CSI21		IIC21

<64-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00 (supporting slave select input)	UART0 (supporting LIN-bus)	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	CSI21		IIC21

Note This function can be used when the setting of bit 1 (PIOR01) of peripheral I/O redirection register 0 (PIOR0) is 1.

Simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) performs the following four types of communication operations.

- Address field transmission (See **24.8.1**.)
- Data transmission (See **24.8.2**.)
- Data reception (See **24.8.3**.)
- Stop condition generation (See **24.8.4**.)

24.8.1 Address field transmission

Address field transmission is a transmission operation that first executes in I^2C 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^2C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCL00, SDA00**Note 1**	SCL01, SDA01**Note 1**	SCL10, SDA10**Note 1**	SCL11, SDA11**Note 1**	SCL20, SDA20**Note 1**	SCL21, SDA21**Note 1**
Interrupt	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21
	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^2C. • Max. 1 MHz (fast mode plus) • Max. 400 kHz (fast 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 transmission/reception timing)					
Data direction	MSB first					

Note 1. To perform communication via simplified I^2C, set the N-ch open-drain output (withstand voltage of V_{DD} (20- to 52-pin products)/withstand voltage of EV_{DD} (64-pin products)) mode (POMxx = 1) with the port output mode register xx (POMxx). See **7.3 Registers to Control the Port Function** and **7.5 Register Settings When Using Alternate Function** for details.
To communicate with an external device operating at a different voltage through IIC00, IIC10, and IIC20, set the N-ch open-drain output (withstand voltage of V_{DD} (20- to 52-pin products)/withstand voltage of EV_{DD} (64-pin products)) mode (POMxx = 1) also for the clock input/output pins (SCL00, SCL10, SCL20).
See **7.4.5 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers** for details.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** or **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.

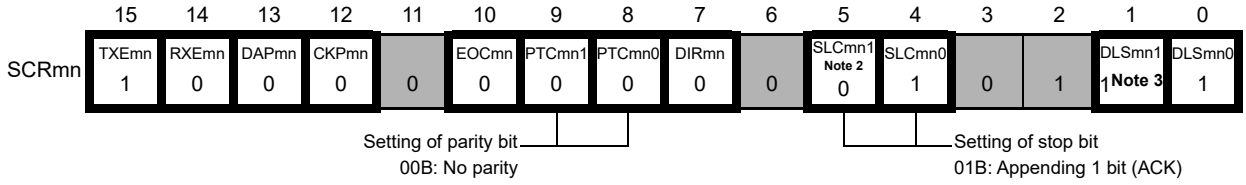
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

1. Register setting

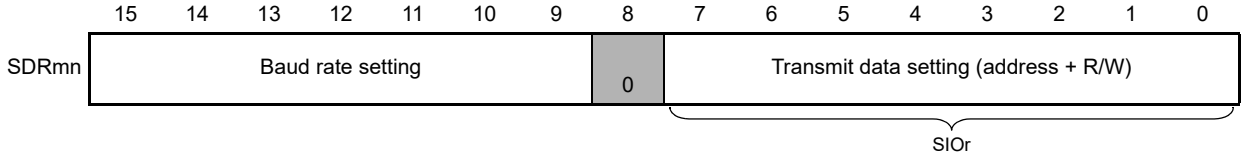
Figure 24 - 101 Example of Contents of Registers for Address Field Transmission of Simplified I2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21)

a) Serial mode register mn (SMRmn)

b) Serial communication operation setting register mn (SCRmn)

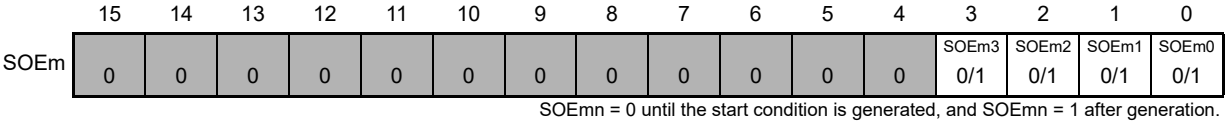


c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)

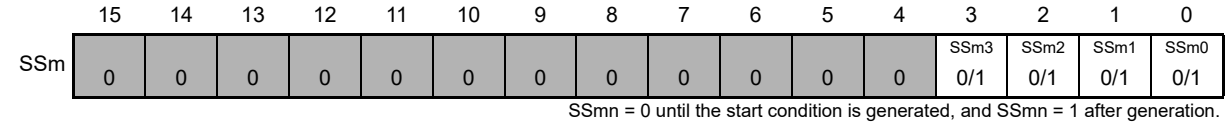


d) Serial output register m (SOm)

e) Serial output enable register m (SOEm)



f) Serial channel start register m (SSm): Set only the bit of the target channel to 1.



Note 1. Only provided for the SMR00, SMR03, and SMR11 registers.

Note 2. Only provided for the SCR00, SCR02, and SCR10 registers.

Note 3. Only provided for the SCR00 and SCR01 registers. This bit is fixed to 1 for the other registers.

(Remarks are listed on the next page.)

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

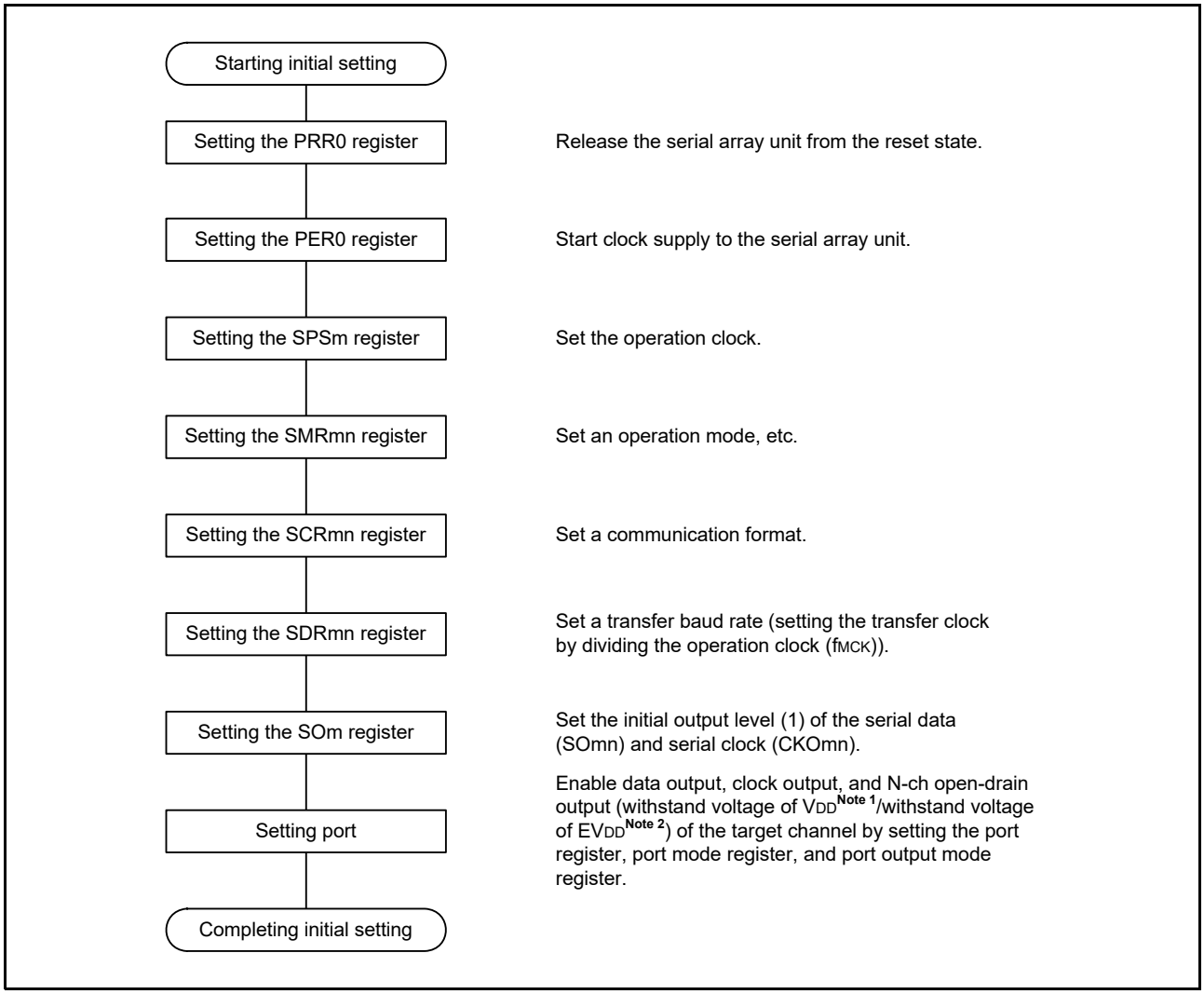
Remark 2. ■: Setting is fixed in the IIC mode

■: Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

2. Operation procedure

Figure 24 - 102 Initial Setting Procedure for Simplified I2C Address Field Transmission

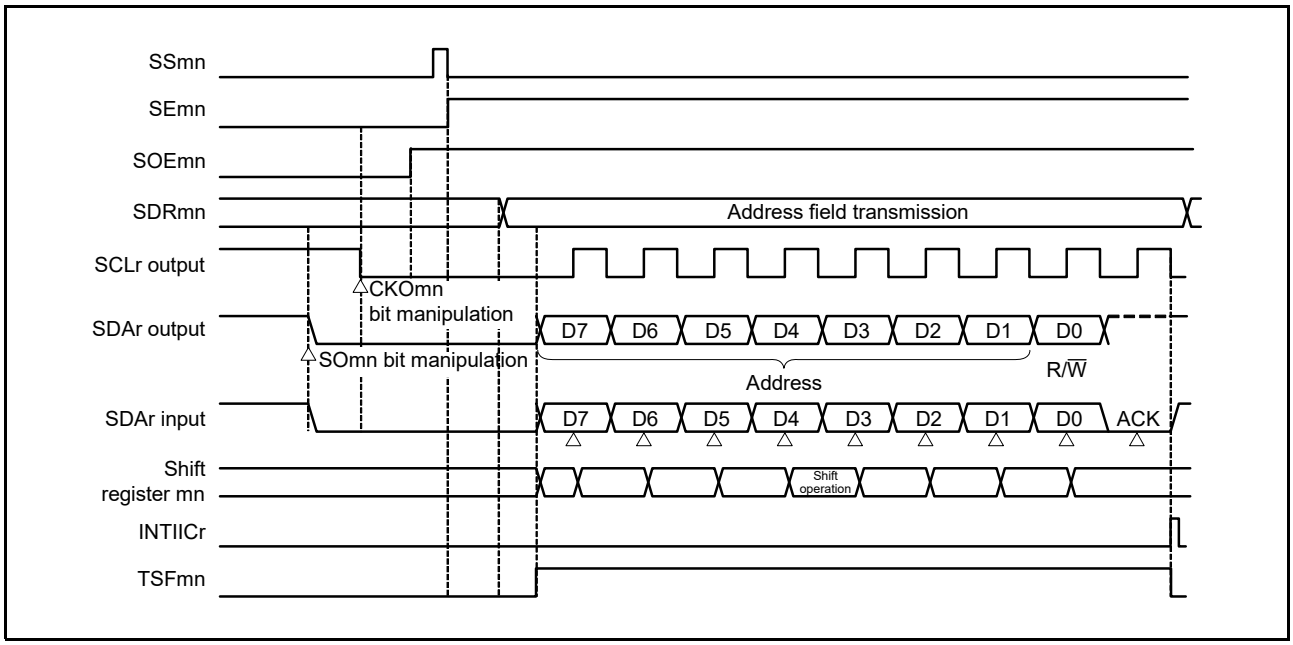


Note 1. This applies to the 20- to 52-pin products.

Note 2. This applies to the 64-pin products.

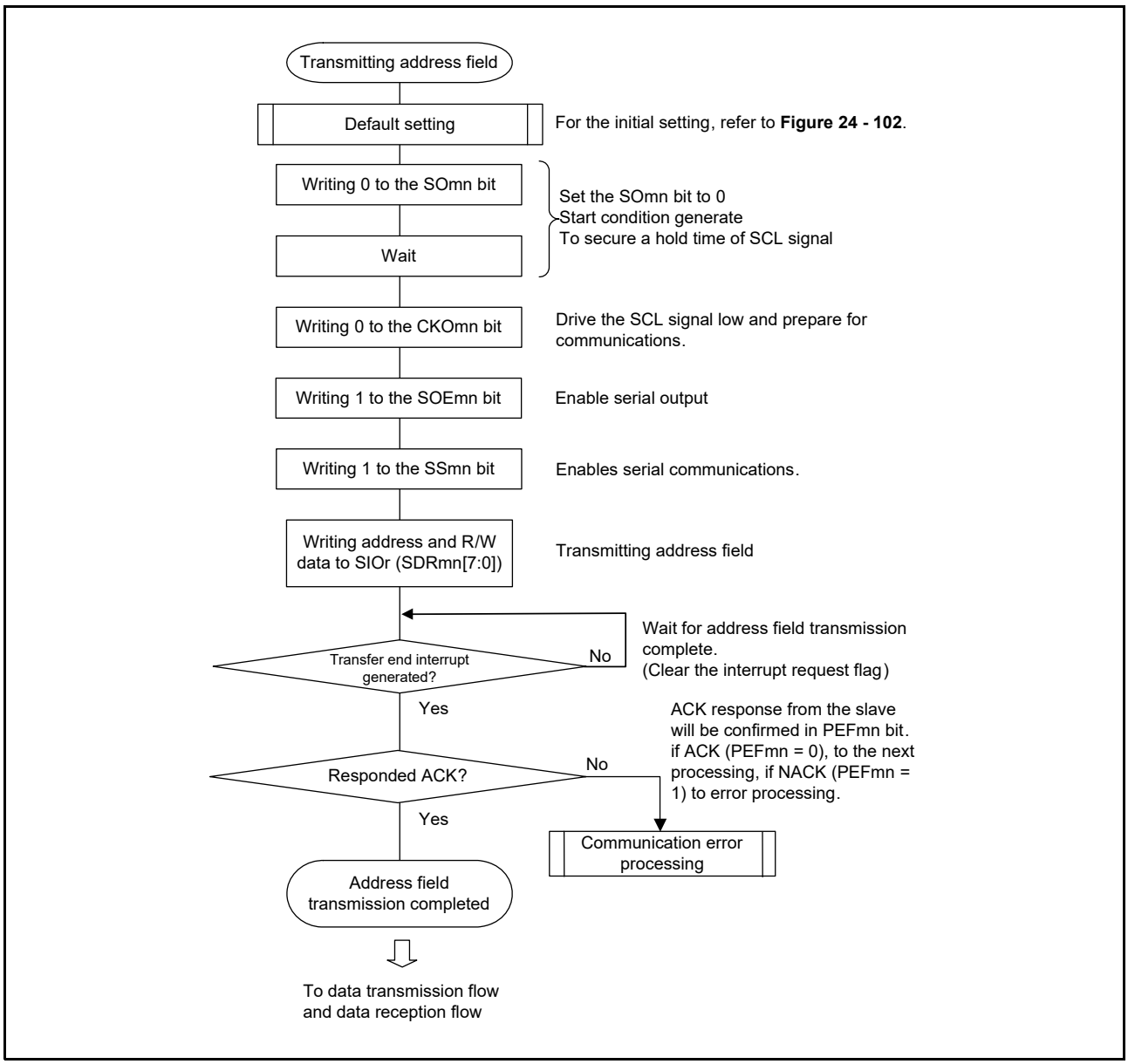
3. Processing flow

Figure 24 - 103 Timing Chart of Address Field Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

Figure 24 - 104 Flowchart of Simplified I2C Address Field Transmission



24.8.2 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	IIC01	IIC10	IIC11	IIC20	IIC21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCL00, SDA00Note 1	SCL01, SDA01Note 1	SCL10, SDA10Note 1	SCL11, SDA11Note 1	SCL20, SDA20Note 1	SCL21, SDA21Note 1
Interrupt	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)					
Error detection flag	ACK error flag (PEFmn)					
Transfer data length	8 bits					
Transfer rateNote 2	Max. fMCK/4 [Hz] (SDRmn[15:9] = 1 or more) fMCK: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I²C. • Max. 1 MHz (fast mode plus) • Max. 400 kHz (fast 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 (withstand voltage of VDD (20- to 52-pin products)/withstand voltage of EVDD (64-pin products)) mode (POMxx = 1) with the port output mode register xx (POMxx). For details, see **7.3 Registers to Control the Port Function** and **7.5 Register Settings When Using Alternate Function**.
To communicate with an external device operating at a different voltage through IIC00, IIC10, and IIC20, set the N-ch open-drain output (withstand voltage of VDD (20- to 52-pin products)/withstand voltage of EVDD (64-pin products)) mode (POMxx = 1) also for the clock input/output pins (SCL00, SCL10, SCL20, SCL30, SCL31).
For details, see **7.4.5 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers**.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 43 Electrical Characteristics (TA = –40 to +105°C)** or **Section 44 Electrical Characteristics (TA = –40 to +125°C)**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

1. Register setting

Figure 24 - 105 Example of Contents of Registers for Data Transmission of Simplified I2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21)

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	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0 Note 1	0	0 Note 1	1	0	0	1	0	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	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	1	0	0	0	0	0	0	0	0	0	0 Note 2	1	0	1	1 Note 3	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 4								Transmit data setting							
								0								
									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					CKOm3	CKOm2	CKOm1	CKOm0					SOm3	SOm2	SOm1	SOm0
	0	0	0	0	0/1 Note 5	0/1 Note 5	0/1 Note 5	0/1 Note 5	0	0	0	0	0/1 Note 5	0/1 Note 5	0/1 Note 5	0/1 Note 5

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													SOEm3	SOEm2	SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	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													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1	0/1

Note 1. Only provided for the SMR01, SMR03, and SMR11 registers.

Note 2. Only provided for the SCR00, SCR02, and SCR10 registers.

Note 3. Only provided for the SCR00 and SCR01 registers. This bit is fixed to 1 for the other registers.

Note 4. Because the setting is completed by address field transmission, setting is not required.

Note 5. The value varies depending on the communication data during communication operation.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

Remark 2. ■: Setting is fixed in the IIC mode

▒: Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

2. Processing flow

Figure 24 - 106 Timing Chart of Data Transmission

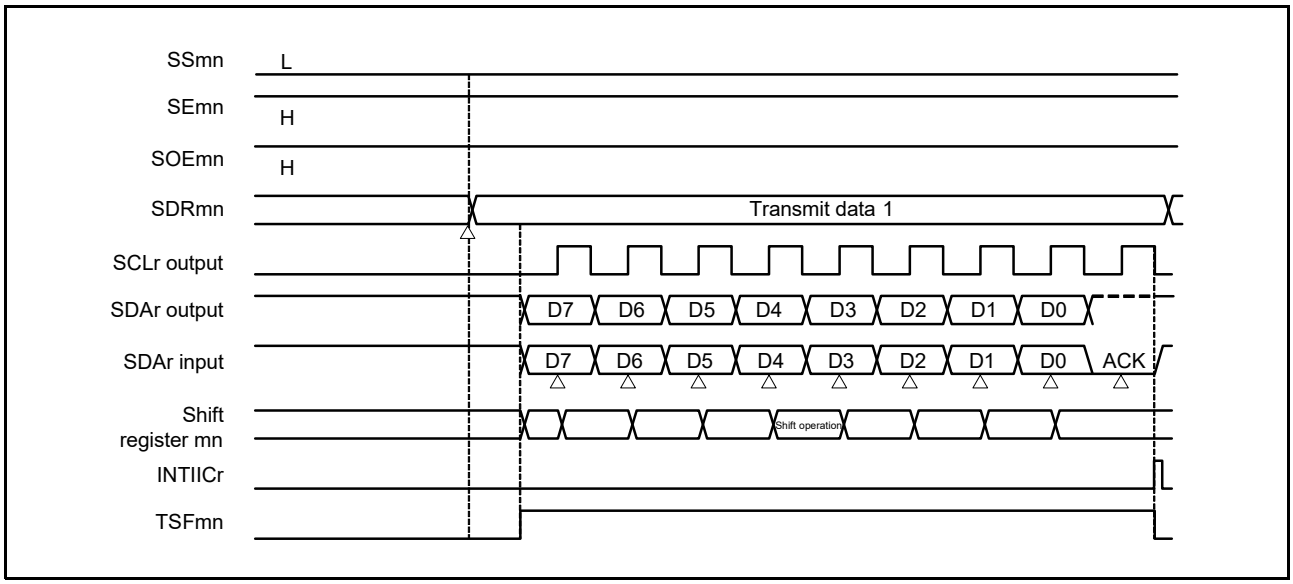
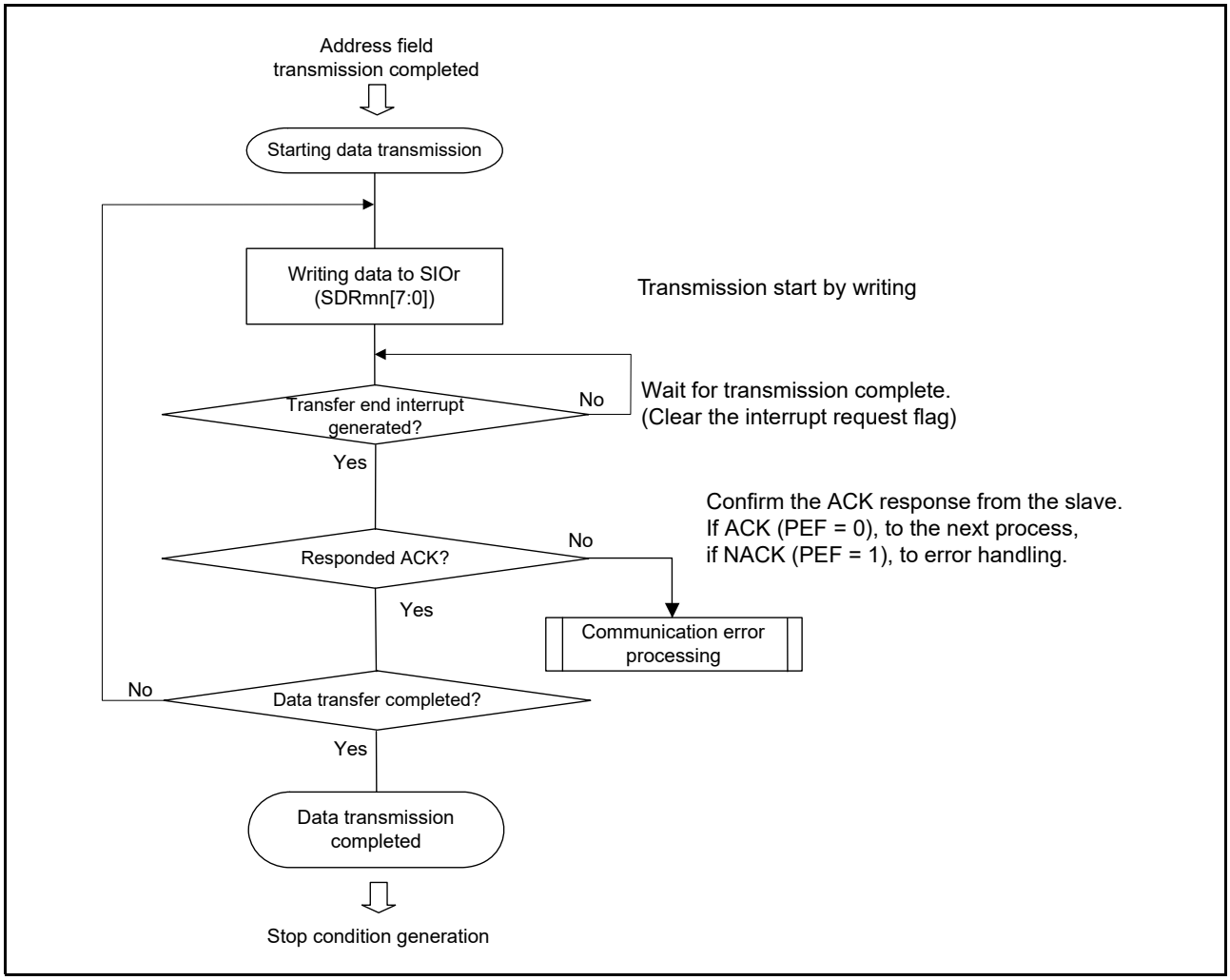


Figure 24 - 107 Flowchart of Simplified I2C Data Transmission



24.8.3 Data reception

Data reception is an operation to receive data from the target for transfer (slave) after transmission of an address field. After all data are received from the slave, a stop condition is generated and the bus is released.

Simplified I²C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCL00, SDA00Note 1	SCL01, SDA01Note 1	SCL10, SDA10Note 1	SCL11, SDA11Note 1	SCL20, SDA20Note 1	SCL21, SDA21Note 1
Interrupt	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21
	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 rateNote 2	Max. fMCK/4 [Hz] (SDRmn[15:9] = 1 or more) fMCK: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I²C. • Max. 1 MHz (fast mode plus) • Max. 400 kHz (fast 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 I²C, set the N-ch open-drain output (withstand voltage of VDD (20- to 52-pin products)/withstand voltage of EVDD (64-pin products)) mode (POMxx = 1) with the port output mode register xx (POMxx). For details, see **7.3 Registers to Control the Port Function** and **7.5 Register Settings When Using Alternate Function**.
To communicate with an external device operating at a different voltage through IIC00, IIC10, and IIC20, set the N-ch open-drain output (withstand voltage of VDD (20- to 52-pin products)/withstand voltage of EVDD (64-pin products)) mode (POMxx = 1) also for the clock input/output pins (SCL00, SCL10, SCL20).
For details, see **7.4.5 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers**.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 43 Electrical Characteristics (TA = –40 to +105°C)** or **Section 44 Electrical Characteristics (TA = –40 to +125°C)**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

1. Register setting

Figure 24 - 108 Example of Contents of Registers for Data Reception of Simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21)

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	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0 Note 1	0	0 Note 1	1	0	0	1	0	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	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	0	1	0	0	0	0	0	0	0	0	0 Note 2	1	0	1	1 Note 3	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 4								Dummy transmit data setting (FFH)							
								0	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					CKOm3	CKOm2	CKOm1	CKOm0					SOm3	SOm2	SOm1	SOm0
	0	0	0	0	0/1 Note 5	0/1 Note 5	0/1 Note 5	0/1 Note 5	0	0	0	0	0/1 Note 5	0/1 Note 5	0/1 Note 5	0/1 Note 5

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													SOEm3	SOEm2	SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1	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													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1	0/1

Note 1. Only provided for the SMR01, SMR03, and SMR11 registers.
Note 2. Only provided for the SCR00, SCR02, and SCR10 registers.
Note 3. Only provided for the SCR00 and SCR01 registers. This bit is fixed to 1 for the other registers.
Note 4. Because the setting is completed by address field transmission, setting is not required.
Note 5. The value varies depending on the communication data during communication operation.

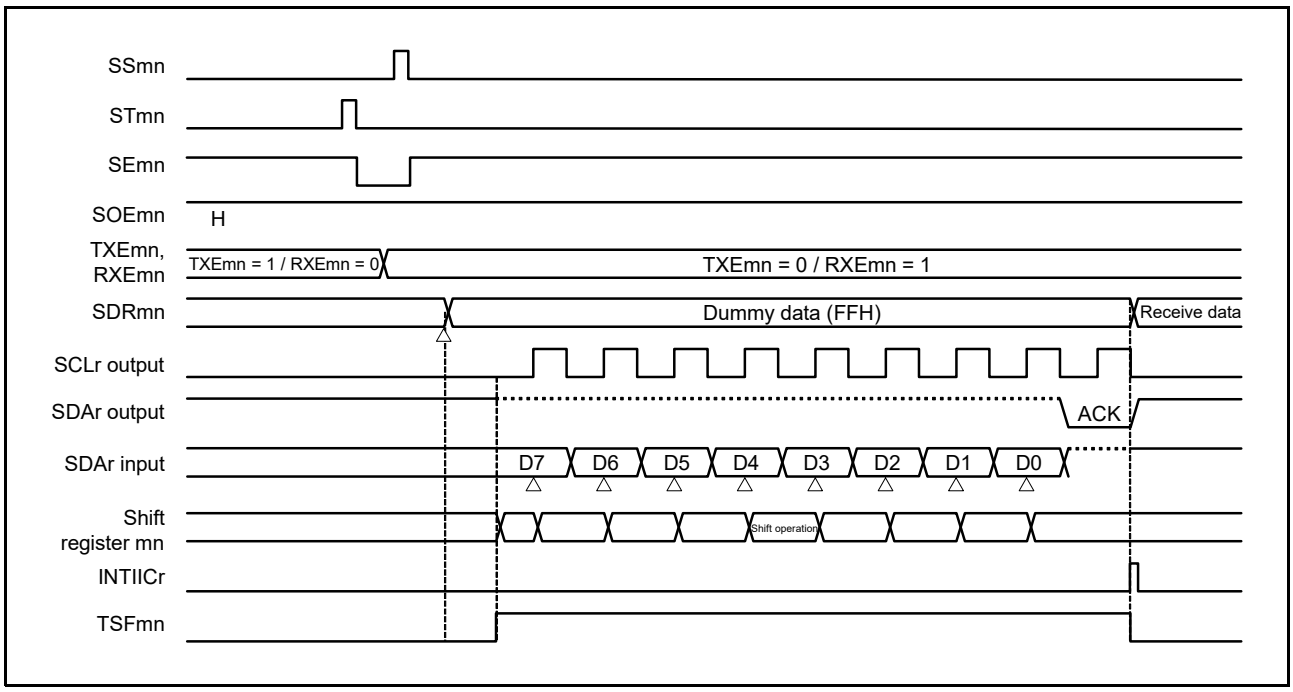
Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

Remark 2. ■: Setting is fixed in the IIC mode
▒: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

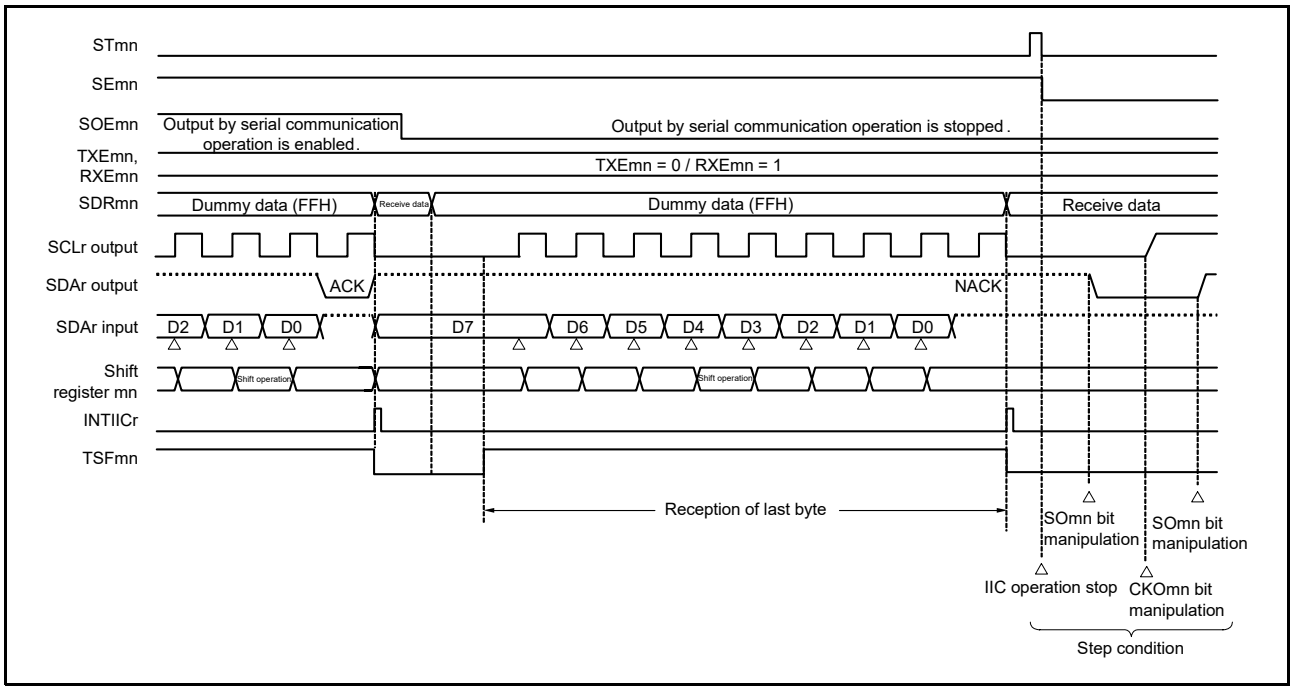
2. Processing flow

Figure 24 - 109 Timing Chart of Data Reception

a) When starting data reception

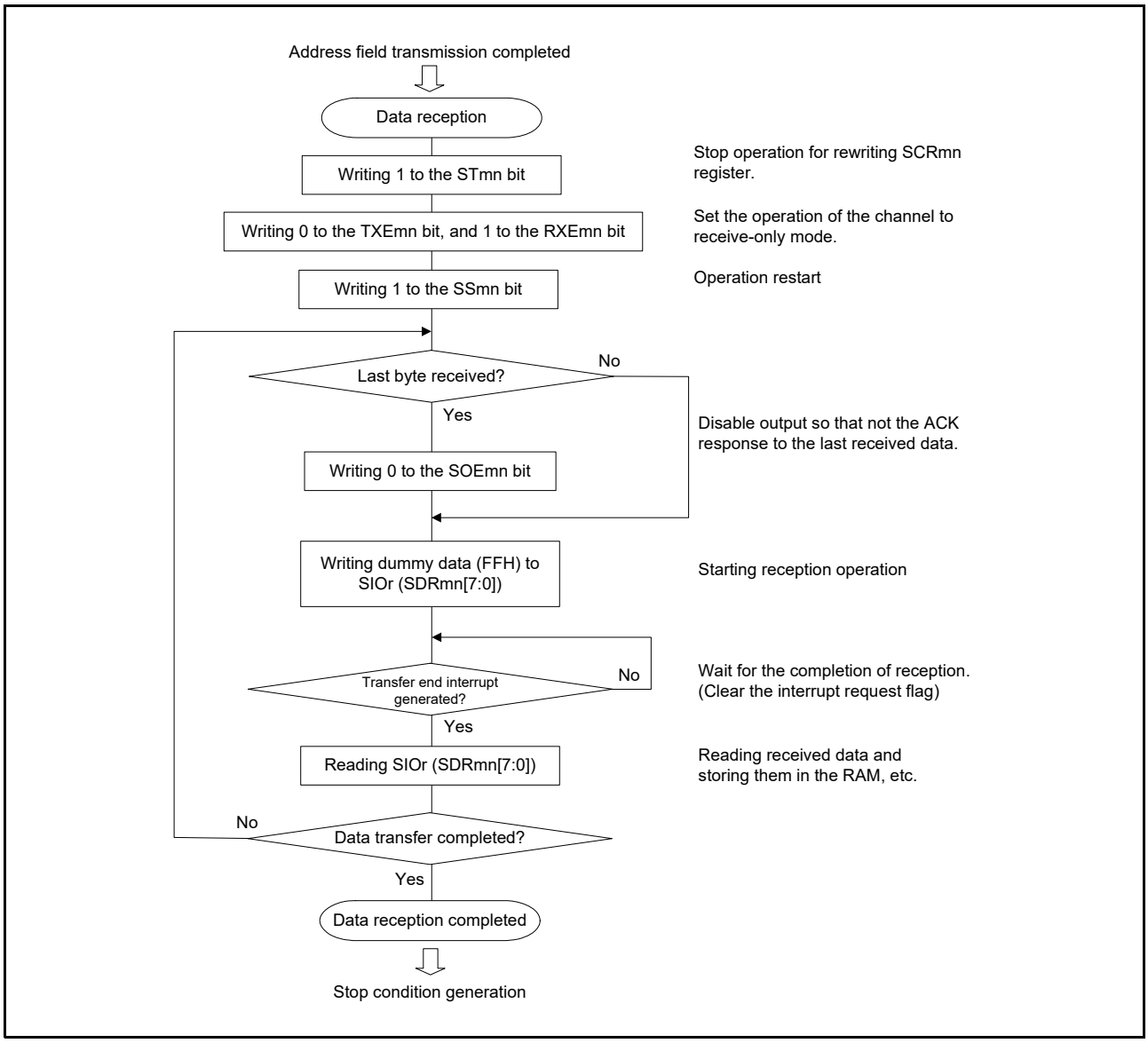


b) When receiving last data



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

Figure 24 - 110 Flowchart of Data Reception



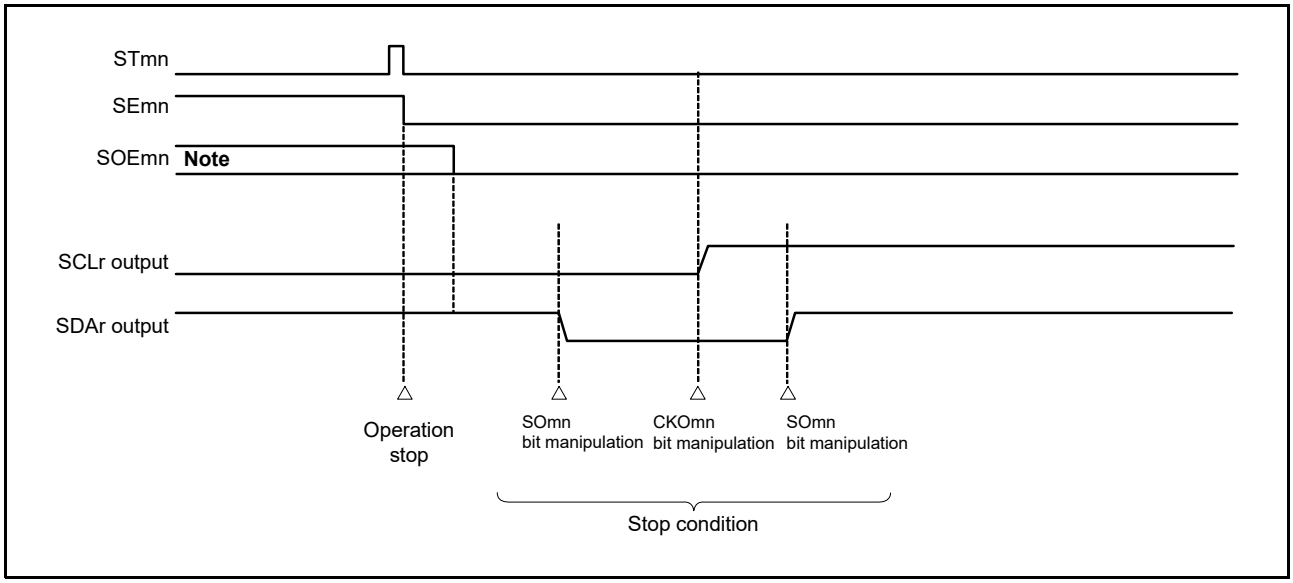
Caution ACK is not output when the last data is received (NACK). Communication is then completed by setting 1 in the STmn bit of serial channel stop register m (STm) to stop operation and generating a stop condition.

24.8.4 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.

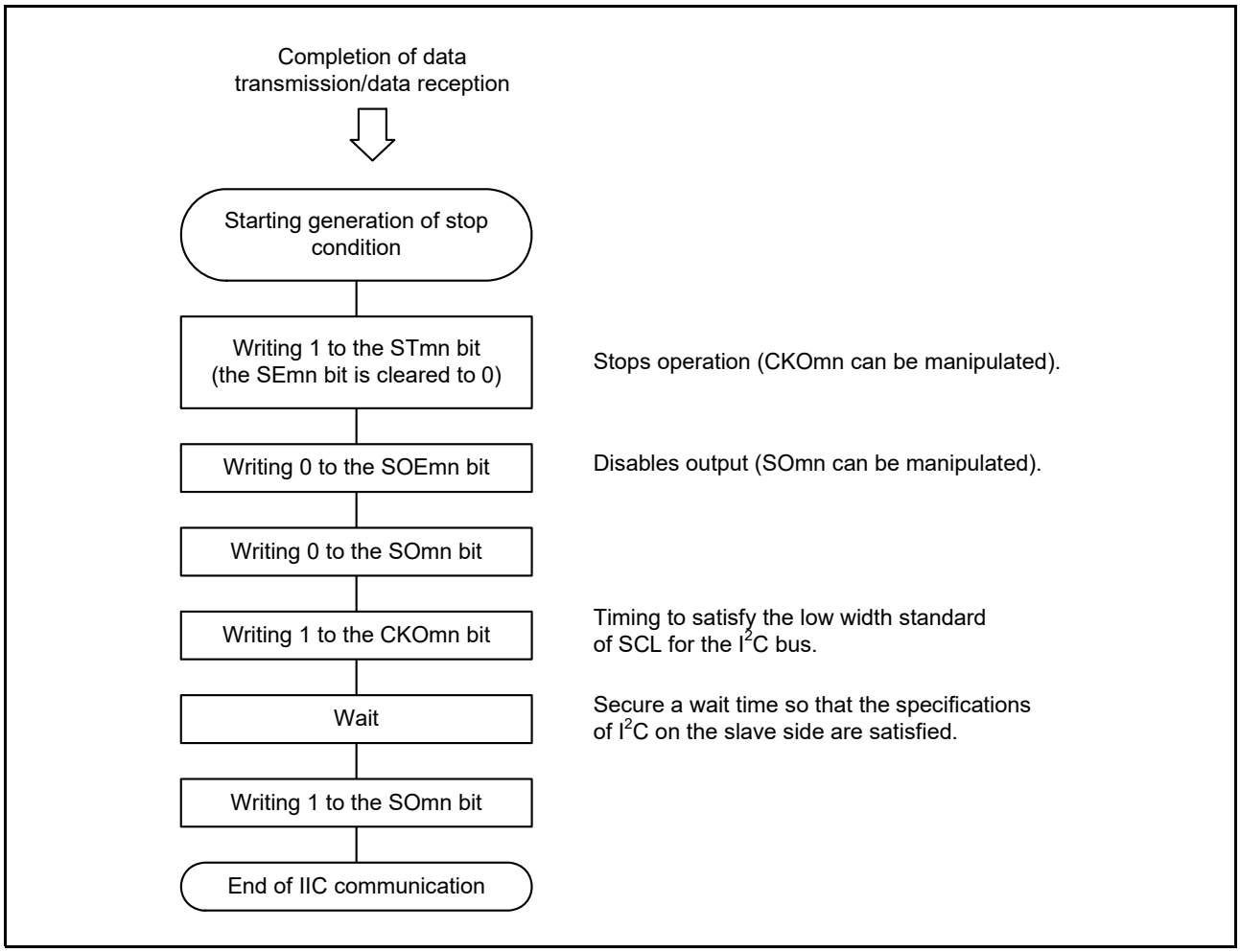
1. Processing flow

Figure 24 - 111 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 24 - 112 Flowchart of Stop Condition Generation



24.8.5 Calculating transfer rate

The transfer rate for simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) 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. Set SDRmn[15:9] to 0000001B or greater.
The duty ratio of the SCL signal output by the simplified I²C is 50%. The I²C bus specifications define that the low-level width of the SCL signal is longer than the high-level width. If 400 kbps (fast mode) or 1 Mbps (fast mode plus) is specified, therefore, the low-level width of the SCL output signal becomes shorter than the value specified in the I²C bus specifications. Make sure that the SDRmn[15:9] value satisfies the I²C 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, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

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 24 - 10 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} = 32 MHz	f_{CLK} = 48 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	32 MHz	Setting prohibited
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	16 MHz	24 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	8 MHz	12 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	4 MHz	6 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	2 MHz	3 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	1 MHz	1.5 MHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	500 kHz	750 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	250 kHz	375 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	125 kHz	188 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz	93.8 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz	46.9 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz	23.4 kHz
1	0	0	0	0	×	×	×	×	f_{CLK}	32 MHz	Setting prohibited
	0	0	0	1	×	×	×	×	$f_{CLK}/2$	16 MHz	24 MHz
	0	0	1	0	×	×	×	×	$f_{CLK}/2^2$	8 MHz	12 MHz
	0	0	1	1	×	×	×	×	$f_{CLK}/2^3$	4 MHz	6 MHz
	0	1	0	0	×	×	×	×	$f_{CLK}/2^4$	2 MHz	3 MHz
	0	1	0	1	×	×	×	×	$f_{CLK}/2^5$	1 MHz	1.5 MHz
	0	1	1	0	×	×	×	×	$f_{CLK}/2^6$	500 kHz	750 kHz
	0	1	1	1	×	×	×	×	$f_{CLK}/2^7$	250 kHz	375 kHz
	1	0	0	0	×	×	×	×	$f_{CLK}/2^8$	125 kHz	188 kHz
	1	0	0	1	×	×	×	×	$f_{CLK}/2^9$	62.5 kHz	93.8 kHz
	1	0	1	0	×	×	×	×	$f_{CLK}/2^{10}$	31.25 kHz	46.9 kHz
	1	0	1	1	×	×	×	×	$f_{CLK}/2^{11}$	15.63 kHz	23.4 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, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

The table below shows an example of setting an I²C transfer rate where f_{MCK} = f_{CLK} = 32 MHz.

I²C Transfer Mode (Desired Transfer Rate)	f_{CLK} = 32 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Transfer Rate	Error from Desired Transfer Rate
100 kHz	f_{CLK}/2	79	100 kHz	0.0%
400 kHz	f_{CLK}	41	380 kHz	5.0%**Note**
1 MHz	f_{CLK}	18	0.84 MHz	16.0%**Note**

Note The error cannot be set to about 0% because the duty ratio of the SCL signal is 50%.

24.8.6 Procedure for processing errors that occurred during simplified I2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) communication

The procedure for processing errors that occurred during simplified I2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) communication is described in **Tables 24 - 11** and **24 - 12**.

Table 24 - 11 Processing Procedure in Case of Overrun Error

Software Manipulation	State of the Hardware	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.	Only the error during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

Table 24 - 12 Processing Procedure in Case of ACK Error in Simplified I2C Mode

Software Manipulation	State of the Hardware	Remark
Reads serial status register mn (SSRmn).		The error type is identified and the read value is used to clear the error flag.
Writes serial flag clear trigger register mn (SIRmn).	The error flag is cleared.	Only the error during reading can be cleared, 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.	The 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 a stop condition.		
Creates a 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, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10, 11

Section 25 Serial Interface IICA (IICA)

This LSI chip has a single serial interface IICA channel that supports the low-power modes of standard mode (100-kHz) and fast mode (400-kHz) operation in compliance with the SMBus and PMBus™ standards.

Caution 1. The I/O pin configuration of the IICAn interface depends on the product. For details, see Section 2 Pin Functions.

Caution 2. Most of the following descriptions in this section use the 64-pin products as an example.

Caution 3. The serial interface IICA only supports operation in compliance with the SMBus and PMBus™ standards in products where the SCLA0 and SDAA0 pin functions are multiplexed with the P60 and P61 pin functions, respectively.

25.1 Functions of Serial Interface IICA

The serial interface IICA has the following three modes.

1. Operation stop mode
   This mode is used when serial transfers are not performed. The operating power can be reduced in this mode.

2. I^2C bus mode (multi-master supported)
   This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCLAn) line and a serial data bus (SDAAn) line. This mode complies with the I^2C bus format and the master device can send start conditions, addresses, transfer directions, acknowledges (ACK), data, and stop conditions to the slave devices, via the serial data bus. The slave device automatically detects these states and data by hardware. This function can simplify the part of application program that controls the I^2C bus. Since the SCLAn and SDAAn pins are used for open drain outputs, serial interface IICA requires pull-up resistors for the serial clock line and the serial data bus line.

3. Wakeup mode
   The STOP mode can be released by generating an interrupt request signal (INTIICAn) when an extension code from the master device or the local address has been received while in STOP mode. This can be set by using the WUPn bit of IICA control register n1 (IICCTLn1). The all address match function is enabled by setting the SVADISn bit of the IICCTLn1 register to 1, allowing any received address is to be determined as a matched address. **Figure 25 - 1** shows a block diagram of serial interface IICA.

Remark n = 0

Figure 25 - 1 Block Diagram of Serial Interface IICA

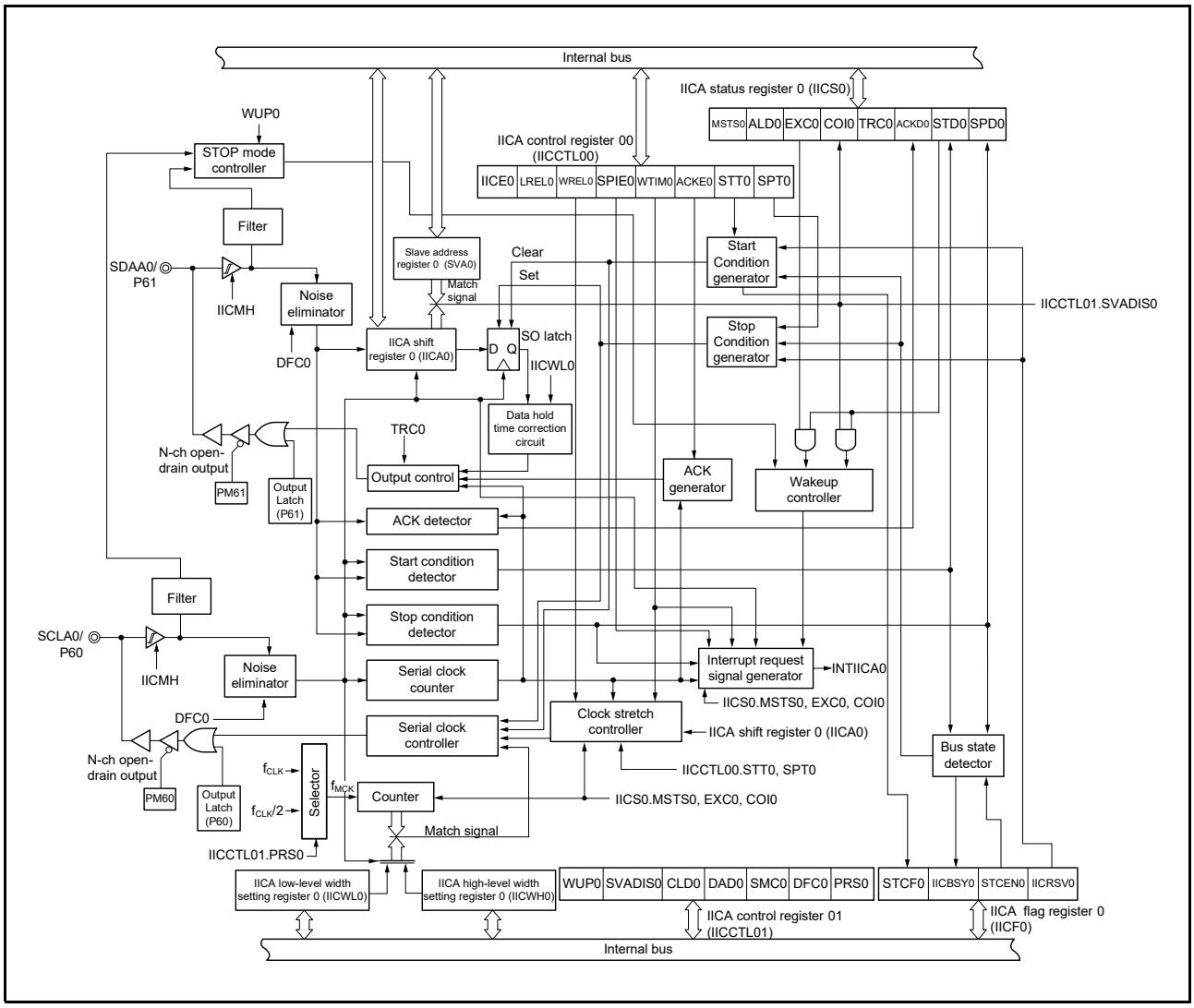
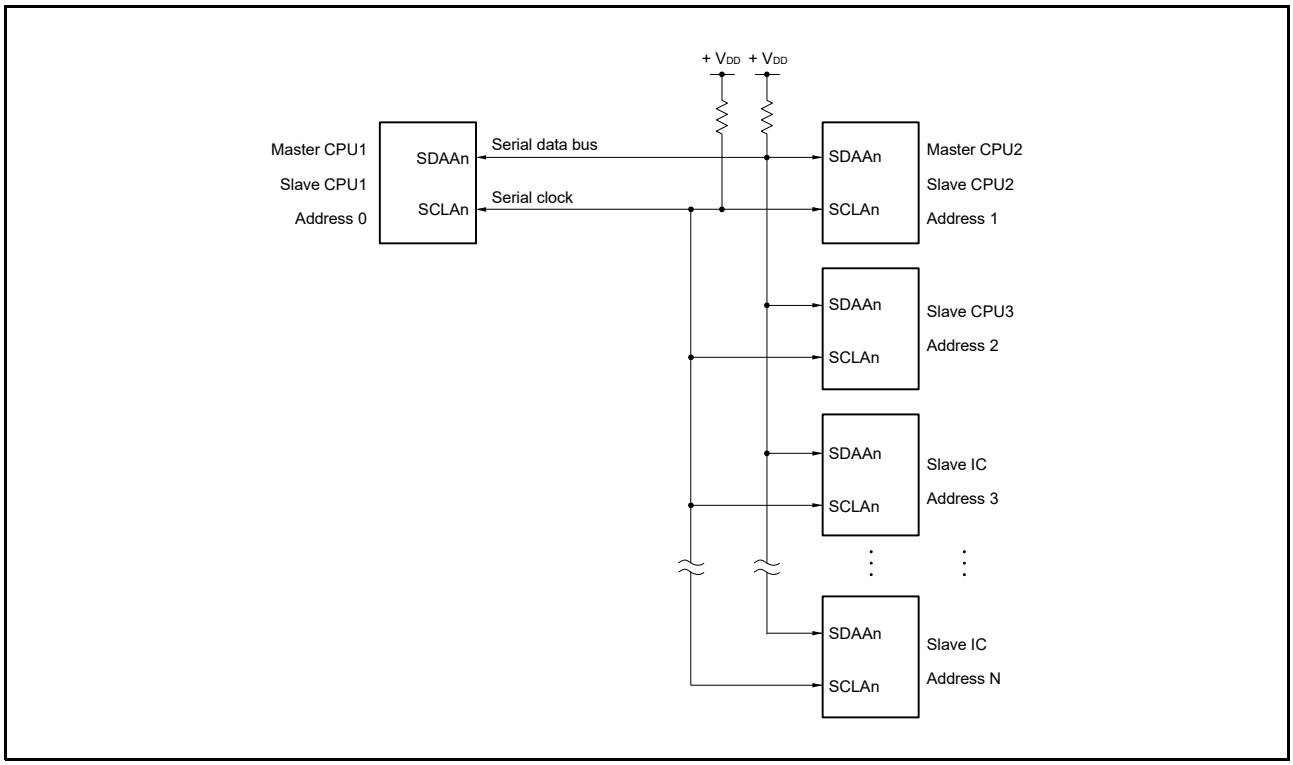


Figure 25 - 2 shows an example of the serial bus configuration using the I2C bus.

Figure 25 - 2 Example of the Serial Bus Configuration Using the I2C Bus



Remark n = 0

25.2 Configuration of Serial Interface IICA

Serial interface IICA includes the following hardware.

Table 25 - 1 Configuration of Serial Interface IICA

Item	Configuration
Registers	• IICA shift register n (IICAn) (n = 0) • Slave address register n (SVAn) (n = 0)
Control registers	• Peripheral enable register 0 (PER0) • Peripheral reset control register 0 (PRR0) • IICA control register n0 (IICCTLn0) (n = 0) • IICA status register n (IICSn) (n = 0) • IICA flag register n (IICFn) (n = 0) • IICA control register n1 (IICCTLn1) (n = 0) • IICA low-level width setting register n (IICWLn) (n = 0) • IICA high-level width setting register n (IICWHn) (n = 0) • IICA input mode selection register (IICM) • Port mode registers xx (PMxx) (xx = 1, 6) • Port registers xx (Pxx) (xx = 1, 6) • Port output mode registers xx (POMxx) (xx = 1) • Port mode control A registers xx (PMCAxx) (xx = 1)

The pins available for use with the SCLA0 and SDAA0 pin functions depend on the product. Set the PIOR02 bit of the PIOR0 register as listed in the following table.

Products	Function	Setting of PIOR02	
		0	1
20-, 24-, and 25-pin products	SCLA0	—	P14
	SDAA0	—	P15
30-, 32-, 40-, 44-, 48-, 52-, and 64-pin products	SCLA0	P60	P14
	SDAA0	P61	P15

For details of the setting of the PIOR02 bit, see **7.3.8 Peripheral I/O redirection registers x (PIORx)**.

1. IICA shift register n (IICAn)
   The IICAn register is used to convert 8-bit serial data to 8-bit parallel data and vice versa in synchronization with the serial clock. The IICAn register can be used for both transmission and reception. The actual transmit and receive operations can be controlled by writing to and reading from the IICAn register. Release serial interface IICA from the clock stretch state and start data transfer by writing data to the IICAn register during the clock stretch period. The IICAn register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 25 - 3 Format of IICA Shift Register n (IICAn)

Address: FFF50H (IICA0)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
IICAn								

Caution 1. Do not write data to the IICAn register during data transfer.

Caution 2. Write to or read from the IICAn register only during the clock stretch period. Accessing the IICAn register in a communication state other than during the clock stretch period is prohibited. When the device serves as the master, however, the IICAn register can be written only once after the communication trigger bit (IICCTLn0.STTn) is set to 1.

Caution 3. When communication is reserved, write data to the IICAn register after the interrupt triggered by a stop condition is detected.

Remark n = 0

2. Slave address register n (SVAn)
   This register holds seven bits [A6, A5, A4, A3, A2, A1, and A0] of the local address when in slave mode. The SVAn register can be set by an 8-bit memory manipulation instruction. However, rewriting to this register is prohibited while STDn = 1 (while the start condition is detected). The value of this register following a reset is 00H.

Figure 25 - 4 Format of Slave Address Register n (SVAn)

Address: F0234H (SVA0)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
SVAn	A6	A5	A4	A3	A2	A1	A0	0Note

Note Bit 0 is fixed to 0.

Remark n = 0

3. SO latch
   The SO latch is used to retain the SDAAn pin's output level.

4. Wakeup controller
   This circuit generates an interrupt request signal (INTIICAn) when the received address matches the address value set to the slave address register n (SVAn), when any address is received while the all address match function is enabled, or when an extension code is received.

5. Serial clock counter
   This counter counts the serial clock cycles that are output or input during transmit/receive operations and is used to verify that 8-bit data was transmitted or received.

6. Interrupt request signal generator
   This circuit controls the generation of interrupt request signals (INTIICAn). An I^2C interrupt request is generated by the following two triggers.
   - Falling edge of eighth or ninth clock of the serial clock (set by the IICCTLn0.WTIMn bit)
   - Interrupt request generated when a stop condition is detected (set by the IICCTLn0.SPIEn bit)

7. Serial clock controller
   In master mode, this circuit generates the serial clock, which is output via the SCLAn pin.

8. Clock stretch controller
   This circuit controls the timing of clock stretching.

9. ACK generator, stop condition detector, start condition detector, and ACK detector
   These circuits generate or detect each state.

10. Data hold time correction circuit
    This circuit generates the hold time for data after the falling edge of the serial clock.

11. Start condition generator
    This circuit generates a start condition when the IICCTLn0.STTn bit is set to 1. However, while communication reservations are disabled (IICFn.IICRSVn = 1) and the bus is busy (IICFn.IICBSYn = 1), start condition requests are ignored and the IICFn.STCFn flag is set to 1.

12. Stop condition generator
    This circuit generates a stop condition when the SPTn bit is set to 1.

13. Bus state detector
    This circuit detects whether or not the bus is released by detecting start conditions and stop conditions. However, the bus state cannot be detected immediately after the IICA operation is enabled. Use the IICFn.STCENn bit to specify the initial state.

Remark n = 0

25.3 Registers to Control Serial Interface IICA

The following registers are used to control serial interface IICA.

- Peripheral enable register 0 (PER0)
- Peripheral reset control register 0 (PRR0)
- IICA control register n0 (IICCTLn0) (n = 0)
- IICA status register n (IICSn) (n = 0)
- IICA flag register n (IICFn) (n = 0)
- IICA control register n1 (IICCTLn1) (n = 0)
- IICA low-level width setting register n (IICWLn) (n = 0)
- IICA high-level width setting register n (IICWHn) (n = 0)
- IICA input mode selection register (IICM)
- Port mode registers xx (PMxx) (xx = 1, 6)
- Port registers xx (Pxx) (xx = 1, 6)
- Port output mode registers xx (POMxx) (xx = 1)
- Port mode control A registers xx (PMCAxx) (xx = 1)

25.3.1 Peripheral enable register 0 (PER0)

The PER0 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. To use the serial interface IICA, be sure to set bit 4 (IICA0EN) of this register to 1. The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 25 - 5 Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

IICAnEN	Control of supply of an input clock to the serial interface IICA
0	Stops supply of an input clock. • The SFRs used by the serial interface IICA cannot be written. • When an SFR used by the serial interface IICA is read, the value returned is 00H or 0000H.
1	Enables supply of an input clock. • The SFRs used by the serial interface IICA can be read and written.

Caution 1. When setting the serial interface IICA, make sure that the setting of the IICAnEN bit is 1 before setting the following registers. If IICAnEN = 0, the values of the registers in and related to the serial interface IICA are returned to their initial values, and writing to them is ignored (except for port mode register 6 (PM6) and port register 6 (P6)).
- **IICA control register n0 (IICCTLn0)**
- **IICA flag register n (IICFn)**
- **IICA status register n (IICSn)**
- **IICA control register n1 (IICCTLn1)**
- **IICA low-level width setting register n (IICWLn)**
- **IICA high-level width setting register n (IICWHn)**
- **IICA shift register n (IICAn)**
- **Slave address register n (SVAn)**

Caution 2. For a precautionary note on the number of pins of RL78/G24 products, see Section 9 Clock Generator.

Caution 3. Do not change the value of a bit of the PER0 register while operation of the corresponding on-chip peripheral module is enabled. Only change a value while the corresponding on-chip peripheral module is stopped.

Remark n = 0

25.3.2 Peripheral reset control register 0 (PRR0)

The PRR0 register is used to control resetting of the on-chip peripheral modules. Bit 4 (IICA0RES) in this register controls resetting and release from the reset state of the serial interface IICA. To place the serial interface IICA in the reset state, set bit 4 (IICA0RES) of this register to 1. The PRR0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 25 - 6 Format of Peripheral Reset Control Register 0 (PRR0)

Address: F00F1H
After reset: 00H
R/W: R/W

Symbol	7	6	<5>	<4>	<3>	<2>	1	<0>
PRR0	0	0	ADCRES	IICA0RES	SAU1RES	SAU0RES	0	TAU0RES

IICAnRES	Control resetting of the serial interface IICA
0	The serial interface IICA is released from the reset state.
1	The serial interface IICA is in the reset state. • The SFRs for use with the serial interface IICA are initialized.

Remark n = 0

25.3.3 IICA control register n0 (IICCTLn0) (n = 0)

The IICCTLn0 register is used to enable or disable the I2C operations, set the timing of clock stretching, and set other I2C operations. This register can be set by a 1-bit or 8-bit memory manipulation instruction. Note that bits SPIEn, WTIMn, and ACKEn must be set while the setting of IICEn is 0 or this module is in the clock stretch state. These bits can be set at the same time as setting the IICEn bit 1. The value of this register following a reset is 00H.

Figure 25 - 7 Format of IICA Control Register n0 (IICCTLn0) (1/5)

Address: F0230H (IICCTL00)
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICCTLn0	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

IICEn	I2C operation enable
0	Stop operation. Reset the IICA status register n (IICSn)Note 1. Stop internal operation.
1	Enable operation.
Be sure to set this bit to 1 while the SCLAn and SDAAn lines are at high level.	
Condition for clearing (IICEn = 0)	Condition for setting (IICEn = 1)
• Cleared by instruction • Reset	• Set by instruction

Figure 25 - 7 Format of IICA Control Register n0 (IICCTLn0) (2/5)

LRELn **Notes 2, 3**	Exit from communications
0	Normal operation
1	IICA exits from the current communications and sets standby mode. This setting is automatically cleared to 0 after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCLAn and SDAAn lines are set to high impedance. The following flags of IICA control register n0 (IICCTLn0) and the IICA status register n (IICSn) are cleared to 0. • STTn, SPTn, MSTSn, EXCn, COIn, TRCn, ACKDn, and STDn
The standby mode following exit from communications remains in effect until the following communications entry conditions are met. • After a stop condition is detected, restart is in master mode. • An address match, extension code reception, or address reception with the all address match function enabled occurs after the start condition.	
Condition for clearing (LRELn = 0)	Condition for setting (LRELn = 1)
• Automatically cleared after execution • Reset	• Set by instruction

WRELn **Notes 2, 3**	Release from the clock stretch state
0	The interface is not released from the clock stretch state.
1	The interface is released from the clock stretch state. After release from the clock stretch state, this bit is automatically cleared to 0.
When the WRELn bit is set (for release from the clock stretch state) during the clock stretch period at the ninth clock pulse in the transmission state (IICSn.TRCn = 1), the SDAAn line goes into the high impedance state (IICSn.TRCn = 0).	
Condition for clearing (WRELn = 0)	Condition for setting (WRELn = 1)
• Automatically cleared after execution • Reset	• Set by instruction

SPIEn**Note 2**	Enable/disable generation of interrupt request when stop condition is detected
0	Disable
1	Enable
If the IICCTLn1.WUPn bit is 1, no stop condition interrupt will be generated even if SPIEn = 1.	
Condition for clearing (SPIEn = 0)	Condition for setting (SPIEn = 1)
• Cleared by instruction • Reset	• Set by instruction

Figure 25 - 7 Format of IICA Control Register n0 (IICCTLn0) (3/5)

WTIMn **Note 4**	Control of clock stretching and interrupt request generation
0	An interrupt request is generated on the falling edge of the eighth clock cycle. • Master mode: After the output of eight clock pulses, the clock output is set to the low level and clock stretching is set. • Slave mode: After the input of eight clock pulses, the clock is set to the low level and clock stretching is set for the master device.
1	An interrupt request is generated on the falling edge of the ninth clock cycle. • Master mode: After the output of nine clock pulses, the clock output is set to the low level and clock stretching is set. • Slave mode: After the input of nine clock pulses, the clock is set to the low level and clock stretching is set for the master device.

An interrupt is generated on the falling edge of the ninth clock cycle during address transfer independently of the setting of this bit. The setting of this bit is valid when the address transfer is completed. In master mode, clock stretching is inserted at the falling edge of the ninth clock cycle during address transfer. For a slave device that has received a local address, clock stretching is inserted at the falling edge of the ninth clock cycle after an acknowledge (ACK) is issued. However, if the slave device has received an extension code, clock stretching is inserted at the falling edge of the eighth clock cycle. When an address is received while the all address match function is enabled, clock stretching is inserted at the falling edge of the eighth clock cycle.

Condition for clearing (WTIMn = 0)	Condition for setting (WTIMn = 1)
• Cleared by instruction • Reset	• Set by instruction

ACKEn **Notes 4, 5**	Acknowledgment control
0	Disable acknowledgment.
1	Enable acknowledgment. During the ninth clock period, the SDAAn line is set to low level.

Condition for clearing (ACKEn = 0)	Condition for setting (ACKEn = 1)
• Cleared by instruction • Reset	• Set by instruction

Figure 25 - 7 Format of IICA Control Register n0 (IICCTLn0) (4/5)

STTn **Notes 2, 6**	Start condition trigger
0	Do not generate a start condition.
1	When bus is released (in standby state, when IICFn.IICBSYn = 0): If this bit is set to 1, a start condition is generated (startup as the master). When a third party is communicating: • When communication reservation function is enabled (IICFn.IICRSVn = 0) Functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus is released. • When communication reservation function is disabled (IICFn.IICRSVn = 1) Even if this bit is set to 1, the STTn bit is cleared and the STTn clear flag (IICFn.STCFn) is set to 1. No start condition is generated. In the clock stretch state (for a master device): Generates a restart condition after release from the clock stretch state.

Cautions concerning set timing
- For master reception: Cannot be set to 1 during transfer. Can be set to 1 only during the clock stretch period when the ACKEn bit has been cleared to 0 and slave has been notified of final reception.
- For master transmission: A start condition cannot be generated normally during the acknowledge period. Set to 1 during the clock stretch period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as stop condition trigger (SPTn).
- Once STTn is set to 1, setting it to 1 again before the clear condition is met is not allowed.

Condition for clearing (STTn = 0)	Condition for setting (STTn = 1)
• Cleared by setting the STTn bit to 1 while communication reservation is prohibited. • Cleared by loss in arbitration • Cleared after start condition is generated by master device • Cleared by LRELn = 1 (exit from communications) • When IICEn = 0 (operation stop) • Reset	• Set by instruction

Figure 25 - 7 Format of IICA Control Register n0 (IICCTLn0) (5/5)

SPTn[Note 7]	Stop condition trigger
0	Stop condition is not generated.
1	Stop condition is generated (termination of master device's transfer).

Cautions concerning set timing
- For master reception: Cannot be set to 1 during transfer. Can be set to 1 only during the clock stretch period when the ACKEn bit has been cleared to 0 and slave has been notified of final reception.
- For master transmission: A stop condition cannot be generated normally during the acknowledge period. Therefore, set it during the clock stretch period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as start condition trigger (STTn).
- This bit can be set to 1 only when in master mode.
- When the WTIMn bit has been cleared to 0, if the SPTn bit is set to 1 during the clock stretch period that follows output of eight clock pulses, note that a stop condition will be generated during the high-level period of the ninth clock after release from the clock stretch state. The WTIMn bit should be changed from 0 to 1 during the clock stretch period following the output of eight clock pulses, and the SPTn bit should be set to 1 during the clock stretch period that follows the output of the ninth clock.
- Once SPTn is set to 1, setting it to 1 again before the clear condition is met is not allowed.

Condition for clearing (SPTn = 0)	Condition for setting (SPTn = 1)
• Cleared by loss in arbitration • Automatically cleared after stop condition is detected • Cleared by LRELn = 1 (exit from communications) • When IICEn = 0 (operation stop) • Reset	• Set by instruction

Note 1. The IICA status register n (IICSn), the STCFn and IICBSYn flags of the IICA flag register n (IICFn), and the CLDn and DADn bits of IICA control register n1 (IICCTLn1) are reset.

Note 2. The signal of this bit is invalid while IICEn is 0.

Note 3. Reading the LRELn and WRELn bits always returns 0.

Note 4. The signal of this bit is invalid while IICEn is 0. Set this bit during that period.

Note 5. The set value is invalid during address transfer and if the code is not an extension code, and the all address match function is disabled. When the device serves as a slave and the addresses match, an acknowledgment is generated regardless of the set value.

Note 6. The STTn bit is always read as 0.

Note 7. The SPTn bit is always read as 0.

Caution 1. If the operation of I^2C is enabled (IICEn = 1) when the SCLAn line is high level, the SDAAn line is low level, and the digital filter is turned on (IICCTLn1.DFCn = 1), a start condition will be inadvertently detected immediately. In this case, set the LRELn bit to 1 by using a 1-bit memory manipulation instruction immediately after enabling operation of I^2C (IICEn = 1).

Caution 2. When bit 3 (TRCn) of the IICA status register n (IICSn) is set to 1 (transmission state), bit 5 (WRELn) of IICA control register n0 (IICCTLn0) is set to 1 during the ninth clock and the interface is released from the clock stretch state, after which the TRCn flag is cleared (reception state) and the SDAAn line is set to the high impedance state. Release the interface from the clock stretch state while the TRCn flag is 1 (transmission state) by writing to the IICA shift register n.

Remark 1. Bit 0 (SPTn) becomes 0 when it is read after data setting.

Remark 2. n = 0

25.3.4 IICA status register n (IICSn) (n = 0)

The IICSn register indicates the state of the I^2C. This register can only be read by a 1-bit or 8-bit memory manipulation instruction while the setting of IICCTLn0.STTn is 1 or this module is in the clock stretch state. The value of this register following a reset is 00H.

Caution Reading the IICSn register while the address match wakeup function is enabled (IICCTLn1.WUPn = 1) in STOP mode is prohibited. When the IICCTLn1.WUPn bit is changed from 1 to 0 (wakeup function is stopped), regardless of the INTIICAn interrupt request signal, a change in the state is not reflected until the next start condition or stop condition is detected. To use the wakeup function, therefore, enable (IICCTLn0.SPIEn = 1) the interrupt generated by detecting a stop condition and read the IICSn register after the interrupt has been detected.

Figure 25 - 8 Format of IICA Status Register n (IICSn) (1/3)

Address: FFF51H (IICS0)
After reset: 00H
R/W: R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICSn	MSTSn	ALDn	EXCn	COIn	TRCn	ACKDn	STDn	SPDn

MSTSn	Master status check flag
0	Slave state or communications standby state
1	Master communications state

Condition for clearing (MSTSn = 0)	Condition for setting (MSTSn = 1)
• When a stop condition is detected • When ALDn = 1 (arbitration loss) • Cleared by IICCTLn0.LRELn = 1 (exit from communications) • When the IICCTLn0.IICEn bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is generated

ALDn	Detection of arbitration loss
0	This value indicates either that arbitration is not in progress or that the result of arbitration was a win.
1	This value indicates that the result of arbitration was a loss. The MSTSn flag is cleared when the ALDn flag has this value.

Condition for clearing (ALDn = 0)	Condition for setting (ALDn = 1)
• Automatically cleared after the IICCTLn0.LRELn bit is read**Note 1** • When the IICCTLn0.IICEn bit changes from 1 to 0 (operation stop) • Reset	• When the arbitration result is a “loss”.

Figure 25 - 8 Format of IICA Status Register n (IICSn) (2/3)

EXCn	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received. Or, the all address match function is enabled.

Condition for clearing (EXCn = 0)	Condition for setting (EXCn = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by IICCTLn0.LRELn = 1 (exit from communications) • When the IICCTLn0.IICEn bit changes from 1 to 0 (operation stop) • Reset	• When the higher four bits of the received address data is either 0000 or 1111 (set at the rising edge of the eighth clock). • When an address is received while the all address match function is enabled (IICCTLn1.SVADISn = 1) (set at the rising edge of the eighth clock).

COIn	Detection of matching addresses
0	Addresses do not match.
1	Addresses match. Or, the all address match function is enabled.

Condition for clearing (COIn = 0)	Condition for setting (COIn = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by IICCTLn0.LRELn = 1 (exit from communications) • When the IICCTLn0.IICEn bit changes from 1 to 0 (operation stop) • Reset	• When the received address matches the local address (slave address register n (SVAn)) (set at the rising edge of the eighth clock). • When an address is received while the all address match function is enabled (IICCTLn1.SVADISn = 1) (set at the rising edge of the eighth clock).

TRCn	Detection of transmission/reception state
0	Reception state (non-transmission state). The SDAAn line is set for high impedance.
1	Transmission state. The value in the SOn latch is enabled for output to the SDAAn line (valid starting at the falling edge of the first byte's ninth clock).

Condition for clearing (TRCn = 0)	Condition for setting (TRCn = 1)
<Both master and slave> • When a stop condition is detected • Cleared by IICCTLn0.LRELn = 1 (exit from communications) • When the IICCTLn0.IICEn bit changes from 1 to 0 (operation stop) • Cleared by IICCTLn0.WRELn = 1**Note 2** (release from the clock stretch state) • When the ALDn flag changes from 0 to 1 (arbitration loss) • Reset • When not used for communication (MSTSn, EXCn, COIn = 0) <Master> • When 1 is output to the first byte's LSB (transfer direction specification bit) <Slave> • When a start condition is detected • When 0 is input to the first byte's LSB (transfer direction specification bit)	<Master> • When a start condition is generated • When 0 (master transmission) is output to the LSB (transfer direction specification bit) of the first byte (during address transfer) <Slave> • When 1 (slave transmission) is input to the LSB (transfer direction specification bit) of the first byte from the master (during address transfer)

Figure 25 - 8 Format of IICA Status Register n (IICSn) (3/3)

ACKDn	Detection of acknowledge (ACK)
0	Acknowledge was not detected.
1	Acknowledge was detected.

Condition for clearing (ACKDn = 0)	Condition for setting (ACKDn = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock • Cleared by IICCTLn0.LRELn = 1 (exit from communications) • When the IICCTLn0.IICEn bit changes from 1 to 0 (operation stop) • Reset	• After the SDAAn line is set to low level at the rising edge of SCLAn line's ninth clock

STDn	Detection of start condition
0	Start condition was not detected.
1	Start condition was detected. This indicates that the address transfer period is in effect.

Condition for clearing (STDn = 0)	Condition for setting (STDn = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by IICCTLn0.LRELn = 1 (exit from communications) • When the IICCTLn0.IICEn bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is detected

SPDn	Detection of stop condition
0	Stop condition was not detected.
1	Stop condition was detected. The master device's communication is terminated and the bus is released.

Condition for clearing (SPDn = 0)	Condition for setting (SPDn = 1)
• At the rising edge of the address transfer byte's first clock following setting of this bit and detection of a start condition • When the IICCTLn1.WUPn bit changes from 1 to 0 • When the IICCTLn0.IICEn bit changes from 1 to 0 (operation stop) • Reset	• When a stop condition is detected

Note 1. The ALDn flag is also cleared when a 1-bit memory manipulation instruction is executed for another bit in the IICSn register. Therefore, when using the ALDn flag, read the data of this flag before the data of the other bits.

Note 2. When the TRCn flag is set to 1 (transmission state), bit 5 (WRELn) of IICA control register n0 (IICCTLn0) is set to 1 during the ninth clock and the interface is released from the clock stretch state, after which the TRCn flag is cleared (reception state) and the SDAAn line is set to the high impedance state. Release the interface from the clock stretch state while the TRCn flag is 1 (transmission state) by writing to the IICA shift register n.

Remark n = 0

25.3.5 IICA flag register n (IICFn) (n = 0)

The IICFn register sets the operation mode of I^2C and indicates the state of the I^2C bus. This register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the STTn clear flag (STCFn) and I^2C bus status flag (IICBSYn) bits are read-only. The IICRSVn bit can be used to enable or disable the communication reservation. The STCENn bit can be used to set the initial value of the IICBSYn flag. The IICRSVn and STCENn bits can be written only when the operation of I^2C is disabled (IICCTLn0.IICEn = 0). The IICFn register is read-only while the operation of the I^2C is enabled. The value of this register following a reset is 00H.

Figure 25 - 9 Format of IICA Flag Register n (IICFn) (1/2)

Address: FFF52H (IICF0)
After reset: 00H
R/W: R/W**Note**

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
IICFn	STCFn	IICBSYn	0	0	0	0	STCENn	IICRSVn

STCFn	STTn clear flag
0	Generate start condition
1	Start condition generation unsuccessful: clear the STTn flag

Condition for clearing (STCFn = 0)	Condition for setting (STCFn = 1)
• Cleared by IICCTLn0.STTn = 1 • When IICCTLn0.IICEn = 0 (operation stop) • Reset	• Generating start condition unsuccessful and the IICCTLn0.STTn bit cleared to 0 when communication reservation is disabled (IICRSVn = 1).

IICBSYn	I^2C bus status flag
0	Bus released state (initial communications state when STCENn = 1)
1	Bus communications state (initial communications state when STCENn = 0)

Condition for clearing (IICBSYn = 0)	Condition for setting (IICBSYn = 1)
• When a stop condition is detected • When IICCTLn0.IICEn = 0 (operation stop) • Reset	• When a start condition is detected • Setting of the IICCTLn0.IICEn bit when STCENn = 0

Figure 25 - 9 Format of IICA Flag Register n (IICFn) (2/2)

STCENn	Initial start enable trigger
0	After operation is enabled (IICCTLn0.IICEn = 1), enable generation of a start condition upon detection of a stop condition.
1	After operation is enabled (IICCTLn0.IICEn = 1), enable generation of a start condition without detecting a stop condition.

Condition for clearing (STCENn = 0)	Condition for setting (STCENn = 1)
• Cleared by instruction • When a start condition is detected • Reset	• Set by instruction

IICRSVn	Communication reservation function disable bit
0	Enable communication reservation
1	Disable communication reservation

Condition for clearing (IICRSVn = 0)	Condition for setting (IICRSVn = 1)
• Cleared by instruction • Reset	• Set by instruction

Note Bits 7 and 6 are read-only.

Caution 1. Write to the STCENn bit only when the operation is stopped (IICCTLn0.IICEn = 0).

Caution 2. As the bus is recognized as being in the released state (IICBSYn = 0) regardless of its actual state when STCENn = 1. When generating the first start condition (IICCTLn0.STTn = 1), it is necessary to verify that no third party communications are in progress in order to prevent such communications from being destroyed.

Caution 3. Write to the IICRSVn bit only when the operation is stopped (IICCTLn0.IICEn = 0).

Remark n = 0

25.3.6 IICA control register n1 (IICCTLn1) (n = 0)

The IICCTLn1 register is used to set the operation mode of I^2C and detect the states of the SCLAn and SDAAn pins. This register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the CLDn and DADn bits are read-only. Set the IICCTLn1 register, except the WUPn bit, while operation of I^2C is disabled (the IICCTLn0.IICEn bit is 0). The value of this register following a reset is 00H.

Figure 25 - 10 Format of IICA Control Register n1 (IICCTLn1) (1/2)

Address: F0231H (IICCTL01)
After reset: 00H
R/W: R/W**Note 1**

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
IICCTLn1	WUPn	SVADISn	CLDn	DADn	SMCn	DFCn	0	PRSn

WUPn	Control of address match wakeup
0	Stops operation of address match wakeup function in STOP mode.
1	Enables operation of address match wakeup function in STOP mode.

To shift to STOP mode when WUPn = 1, execute the STOP instruction at least three cycles of f_{MCK} after setting the WUPn bit to 1 (see **Figure 25 - 23 Flow When Setting WUPn = 1**). Clear the WUPn bit to 0 after the address has matched, an address has been received while the all address match function is enabled, or an extension code has been received. The subsequent communication can be entered by clearing the WUPn bit to 0. (The interface must be released from the clock stretch state and transmit data must be written after the WUPn bit has been cleared to 0.) The interrupt timing when the address has matched, when an address has been received while the all address match function is enabled, or when an extension code has been received, while WUPn = 1, is identical to the interrupt timing when WUPn = 0. (A delay of the difference of sampling by the clock will occur.) Furthermore, when WUPn = 1, a stop condition interrupt is not generated even if the IICCTLn0.SPIEn bit is set to 1.

Condition for clearing (WUPn = 0)	Condition for setting (WUPn = 1)
• Cleared by instruction (after address match, address reception with the all address match function enabled, or extension code reception)	• Set by instruction (when the MSTSn, EXCn, and COIn flags are 0, and the STDn flag also 0 (communication not entered))**Note 2**

Figure 25 - 10 Format of IICA Control Register n1 (IICCTLn1) (2/2)

SVADISn	Address match disabling flag
0	Disables the all address match function.
1	Enables the all address match function.
When SVADISn = 1, IICA considers any address as address match, and performs the same operation as that when an extension code is received. Therefore, IICSn.COIn is set to 1, and IICSn.EXCn is set to 1. For details about extension code reception, see **25.5.11 Extension code**.	

CLDn	Detection of SCLAn pin level (valid only when IICCTLn0.IICEn = 1)
0	The SCLAn pin was detected at low level.
1	The SCLAn pin was detected at high level.

Condition for clearing (CLDn = 0)	Condition for setting (CLDn = 1)
• When the SCLAn pin is at low level • When IICCTLn0.IICEn = 0 (operation stop) • Reset	• When the SCLAn pin is at high level

DADn	Detection of SDAAn pin level (valid only when IICCTLn0.IICEn = 1)
0	The SDAAn pin was detected at low level.
1	The SDAAn pin was detected at high level.

Condition for clearing (DADn = 0)	Condition for setting (DADn = 1)
• When the SDAAn pin is at low level • When IICCTLn0.IICEn = 0 (operation stop) • Reset	• When the SDAAn pin is at high level

SMCn	Operation mode switching
0	Operates in standard mode (fastest transfer rate: 100 kbps).
1	Operates in fast mode (fastest transfer rate: 400 kbps) or fast mode plus (fastest transfer rate: 1 Mbps).

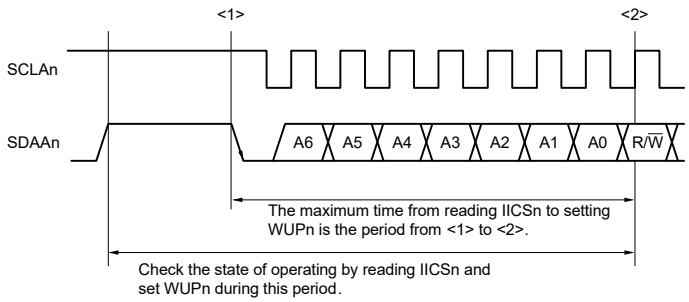
DFCn	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
Use the digital filter only in fast mode and fast mode plus. The digital filter is used for noise elimination. The transfer clock does not vary, regardless of the DFCn bit being set to 1 or cleared to 0.	

PRSn	Control of IICA operation clock (f_{MCK})
0	Selects f_{CLK} (1 MHz ≤ f_{CLK} ≤ 20 MHz).
1	Selects $f_{CLK}/2$ (20 MHz < f_{CLK}).

Note 1. Bits 5 and 4 are read-only.

(Note, Cautions, and Remark are listed on the next page.)

Note 2. The state of the IICA status register n (IICSn) must be checked and the WUPn bit must be set during the period shown below.



Caution 1. Set the PRSn bit to 1 only when the frequency of f_{CLK} exceeds 20 MHz.
When the frequency of f_{CLK} is 48 MHz, the filter width of the digital filter is set to 41.67 ns. To ensure 50 ns or a greater filter width, set f_{CLK} to 32 MHz or a lower frequency or use an external noise filter.

Caution 2. Note the minimum f_{CLK} operation frequency when setting the transfer clock. The minimum f_{CLK} operation frequency for serial interface IICA is determined according to the mode.
Fast mode: f_{CLK} = 3.5 MHz (min.)
Fast mode plus: f_{CLK} = 10 MHz (min.)
Normal mode: f_{CLK} = 1 MHz (min.)

Remark n = 0

25.3.7 IICA low-level width setting register n (IICWLn) (n = 0)

The IICWLn register is used to set the low-level width (t_{LOW}) of the SCLAn pin signal that is output by serial interface IICA and to control the SDAAn pin signal. This register can be set by an 8-bit memory manipulation instruction. Set the IICWLn register while operation of I^2C is disabled (the IICCTLn0.IICEn bit is 0). The value of this register following a reset is FFH. For details about setting the IICWLn register, see **25.4.2 Setting transfer clock by using IICWLn and IICWHn registers**. The data hold time is one-quarter of the time set by the IICWLn register.

Figure 25 - 11 Format of IICA Low-level Width Setting Register n (IICWLn)

Address: F0232H (IICWL0)
After reset: FFH
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
IICWLn								

Remark n = 0

25.3.8 IICA high-level width setting register n (IICWHn) (n = 0)

The IICWHn register is used to set the high-level width of the SCLAn pin signal that is output by serial interface IICA and to control the SDAAn pin signal. This register can be set by an 8-bit memory manipulation instruction. Set the IICWHn register while operation of I^2C is disabled (the IICCTLn0.IICEn bit is 0). The value of this register following a reset is FFH.

Figure 25 - 12 Format of IICA High-level Width Setting Register n (IICWHn)

Address: F0233H (IICWH0)
After reset: FFH
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
IICWHn								

Remark 1. See the following sections for the procedures for setting the transfer clock on the master side and setting the IICWLn and IICWHn registers on the slave side.
- Transfer clock on the master side:
  25.4.2 Setting transfer clock by using IICWLn and IICWHn registers, **1. Setting transfer clock on master side**
- IICWLn and IICWHn registers on the salve side:
  25.4.2 Setting transfer clock by using IICWLn and IICWHn registers, **2. Setting IICWLn and IICWHn registers on slave side (The fractional parts of all setting values are rounded up.)**

Remark 2. n = 0

25.3.9 IICA input mode selection register (IICM)

The IICM register is used to select the input threshold for the IICA interface. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 25 - 13 Format of IICA Input Mode Selection Register (IICM)

Address: F007AH (IICM)
After reset: 00H
R/W: R/W**Note**

Symbol	7	6	5	4	3	2	1	0
IICM	0	0	0	0	0	0	0	IICSH

IICSH	Input threshold selection for the IICA interface
0	I^2C specification input threshold is selected.
1	SMBus 3.0 (1.35 V) input threshold is selected.

Note Be sure to clear bits 7 to 1 to 0.

Caution **Do not manipulate the IICSH bit during self-programming (while the flash memory sequencer is not in non-programmable mode).**

25.3.10 Registers for controlling the port functions multiplexed with the inputs and outputs of the serial interface IICA

Set the following registers to control the port functions multiplexed with the inputs and outputs of the serial interface IICA.

- Port mode registers xx (PMxx)
- Port registers xx (Pxx)
- Port output mode registers xx (POMxx)
- Port output mode registers xx (POMxx)

For details, see the following sections.

- **7.3.1 Port mode registers xx (PMxx)**
- **7.3.2 Port registers xx (Pxx)**
- **7.3.5 Port output mode registers xx (POMxx)**
- **7.3.7 Port mode control A registers xx (PMCAxx)**

When the pin multiplexed with SCLA0 is to be used for input/output of the clock and the pin multiplexed with SDAA0 is to be used for input/output of serial data, set the following register bits corresponding to each port to 0.

- Port mode register xx (PMxx)
- Port register xx (Pxx)
- Port mode control A register xx (PMCAxx)

Furthermore, set the corresponding bit in port output mode register xx (POMxx) to 1.

Set the IICEn bit (bit 7 of IICA control register n0 (IICCTLn0)) to 1 before setting the output mode because the P60/SCLA0 and P61/SDAA0 pin outputs are fixed to the low level when the IICEn bit is 0.

Remark xx = 1, 6
Note that POM6 and PMCA6 are not present in the RL78/G24 products.

25.4 I2C Bus Mode Functions

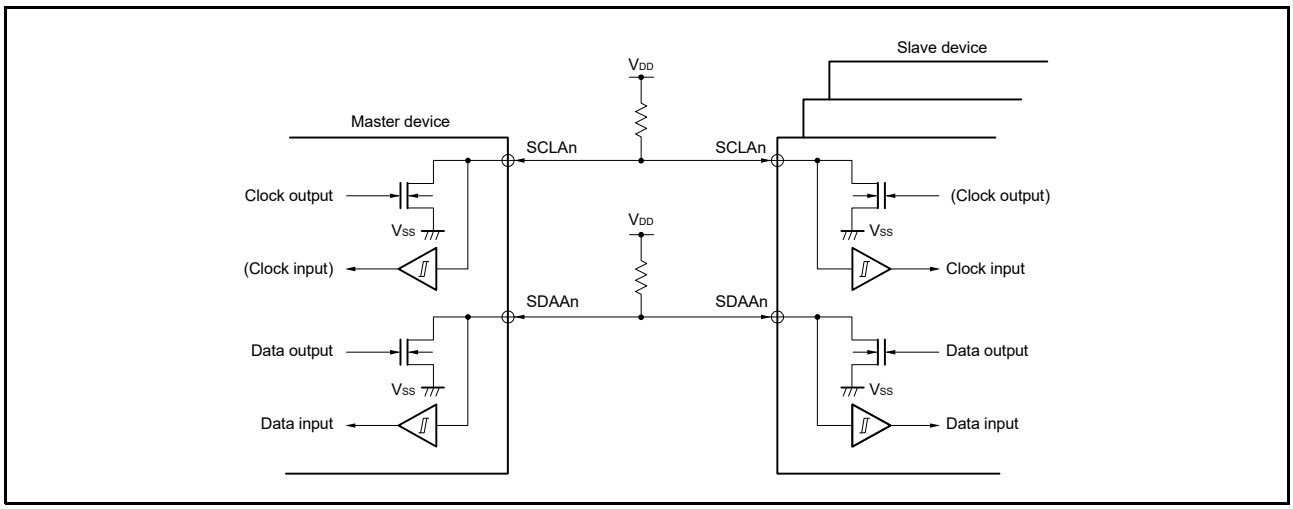
25.4.1 Pin configuration

The serial clock pin (SCLAn) and the serial data bus pin (SDAAn) are configured as follows.

1. SCLAn: This pin is used for serial clock input and output. This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.
2. SDAAn: This pin is used for serial data input and output. This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

Figure 25 - 14 Pin Configuration Diagram



Remark n = 0

25.4.2 Setting transfer clock by using IICWLn and IICWHn registers

1. Setting transfer clock on master side

$$\text{Transfer clock} = \frac{f_{MCK}}{IICWL + IICWH + f_{MCK}(t_R + t_F)}$$

At this time, the optimal setting values of the IICWLn and IICWHn registers are as follows.
(The fractional parts of all setting values are rounded up.)

- When the fast mode

$$IICWLn = \frac{0.52}{\text{Transfer clock}} \times f_{MCK}$$

$$IICWHn = \left(\frac{0.48}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

- When the normal mode

$$IICWLn = \frac{0.47}{\text{Transfer clock}} \times f_{MCK}$$

$$IICWHn = \left(\frac{0.53}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

- When the fast mode plus

$$IICWLn = \frac{0.50}{\text{Transfer clock}} \times f_{MCK}$$

$$IICWHn = \left(\frac{0.50}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

2. Setting IICWLn and IICWHn registers on slave side
(The fractional parts of all setting values are rounded up.)
   - When the fast mode
     $IICWLn = 1.3\ \mu s \times f_{MCK}$
     $IICWHn = (1.2\ \mu s - t_R - t_F) \times f_{MCK}$
   - When the normal mode
     $IICWLn = 4.7\ \mu s \times f_{MCK}$
     $IICWHn = (5.3\ \mu s - t_R - t_F) \times f_{MCK}$
   - When the fast mode plus
     $IICWLn = 0.50\ \mu s \times f_{MCK}$
     $IICWHn = (0.50\ \mu s - t_R - t_F) \times f_{MCK}$

Caution 1. Set bit 0 (PRSn) of IICA control register n1 (IICCTLn1) to 1 only when the frequency of f_{CLK} exceeds 20 MHz.
When the frequency of f_{CLK} is 48 MHz, the filter width of the digital filter is set to 41.67 ns. To ensure 50 ns or a greater filter width, set f_{CLK} to 32 MHz or a lower frequency or use an external noise filter.

Caution 2. Note the minimum f_{CLK} operation frequency when setting the transfer clock. The minimum f_{CLK} operation frequency for serial interface IICA is determined according to the mode.
Fast mode: f_{CLK} = 3.5 MHz (min.)
Fast mode plus: f_{CLK} = 10 MHz (min.)
Normal mode: f_{CLK} = 1 MHz (min.)

(Remarks are listed on the next page.)

Remark 1. Calculate the rise time (t_R) and fall time (t_F) of the SDAAn and SCLAn signals separately, because they differ depending on the pull-up resistor and wire load.

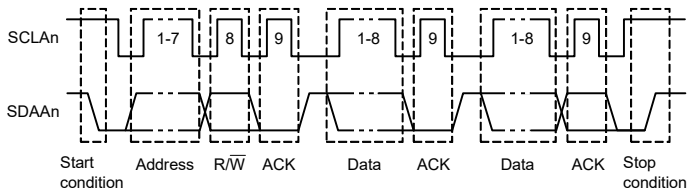
Remark 2. IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
t_F: SDAAn and SCLAn signal falling times
t_R: SDAAn and SCLAn signal rising times
f_{MCK}: IICA operation clock frequency

Remark 3. n = 0

25.5 I2C Bus Definitions and Control Methods

The following section describes the I2C bus's serial data communication format and the signals used by the I2C bus. **Figure 25 - 15** shows the transfer timing for the “start condition”, “address”, “data”, and “stop condition” output via the I2C bus's serial data bus.

Figure 25 - 15 I2C Bus Serial Data Transfer Timing

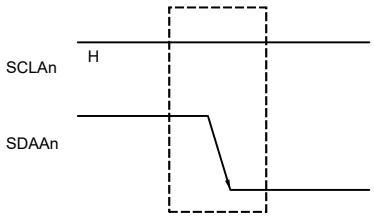


The master device generates the start condition, slave address, and stop condition. The acknowledge (ACK) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data). The serial clock (SCLAn) is continuously output by the master device. However, for the slave device, the period over which the SCLAn pin is at the low level can be extended and clock stretching can be inserted.

25.5.1 Start conditions

When the SCLAn pin is at high level, changing the SDAAn pin from the high level to the low level generates a stop condition. A stop condition is a signal that the master device generates to the slave device when starting a serial transfer. When the device is used as a slave, start conditions can be detected.

Figure 25 - 16 Start Conditions



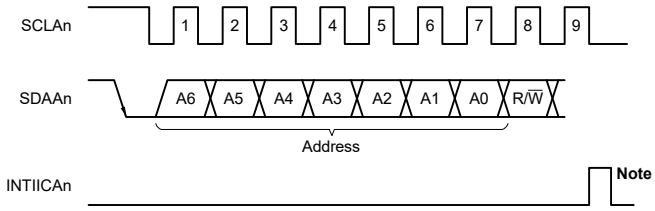
A start condition is output when bit 1 (STTn) of IICA control register n0 (IICCTLn0) is set to 1 after a stop condition has been detected (IICSn.SPDn = 1). When a start condition is detected, bit 1 (STDn) of the IICSn register is set to 1.

Remark n = 0

25.5.2 Address

The address is defined by the 7 bits of data that follow the start condition. An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via the bus lines. Therefore, each slave device connected via the bus lines must have a unique address. The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in the slave address register n (SVAn). If the address data matches the SVAn register values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

Figure 25 - 17 Addresses



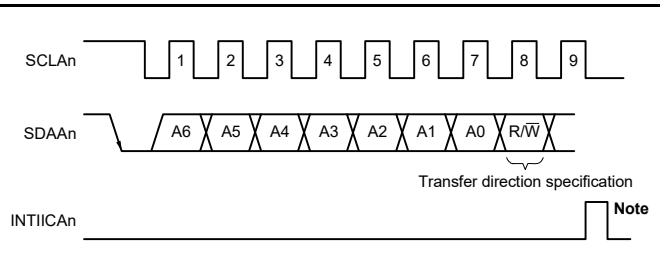
Note INTIICAn is not issued if data other than a local address or extension code is received while the all address match function is disabled during slave device operation.

Addresses are output when a total of 8 bits consisting of the slave address and the transfer direction described in **25.5.3 Transfer direction specification** are written to the IICA shift register n (IICAn). The received addresses are written to the IICAn register. The slave address is assigned to the higher 7 bits of the IICAn register.

25.5.3 Transfer direction specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction. When this transfer direction specification bit has a value of 0, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of 1, it indicates that the master device is receiving data from a slave device.

Figure 25 - 18 Transfer Direction Specification



Note INTIICAn is not issued if data other than a local address or extension code is received while the all address match function is disabled during slave device operation.

Remark n = 0

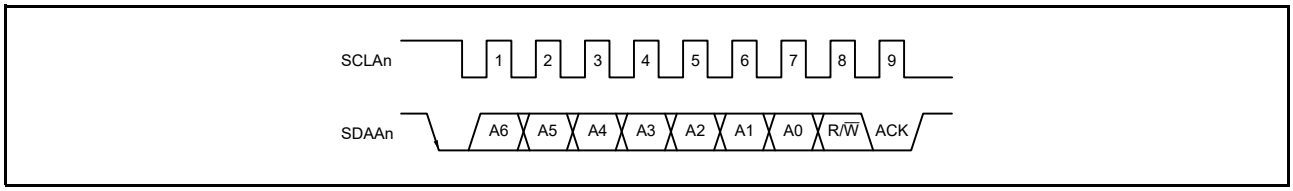
25.5.4 Acknowledge (ACK)

ACK is used to check the state of serial data at the transmission and reception sides. The reception side returns ACK each time it has received 8-bit data. The transmission side usually receives ACK after transmitting 8-bit data. When ACK is returned from the reception side, it is assumed that reception has been correctly performed and processing is continued. Whether ACK has been detected can be checked by using bit 2 (ACKDn) of the IICA status register n (IICSn). When the master receives the last data item, it does not return ACK and instead generates a stop condition. If a slave does not return ACK after receiving data, the master outputs a stop condition or restart condition and stops transmission. If ACK is not returned, the possible causes are as follows.

<1> Reception was not performed normally.
<2> The final data item was received.
<3> The reception side specified by the address does not exist.

To generate ACK, the reception side makes the SDAAn line low at the ninth clock (indicating normal reception). Automatic generation of ACK is enabled by setting bit 2 (ACKEn) of IICA control register n0 (IICCTLn0) to 1. Bit 3 (TRCn) of the IICSn register is set to the value of the eighth bit that follows 7-bit address information. Usually, set the ACKEn bit to 1 for reception (TRCn = 0). If a slave can receive no more data during reception (TRCn = 0) or does not require the next data item, then the slave must inform the master, by clearing the ACKEn bit to 0, that it will not receive any more data. When the master does not require the next data item during reception (TRCn = 0), it must clear the ACKEn bit to 0 so that ACK is not generated. In this way, the master informs a slave at the transmission side that it does not require any more data (transmission will be stopped).

Figure 25 - 19 ACK



When the local address is received, ACK is automatically generated, regardless of the value of the ACKEn bit. When an address other than that of the local address is received, ACK is not generated (NACK). When an extension code is received, or when an address is received while the all address match function is enabled, ACK is generated if the ACKEn bit is set to 1 in advance. How ACK is generated when data is received depends on the setting of the timing of clock stretching as follows.

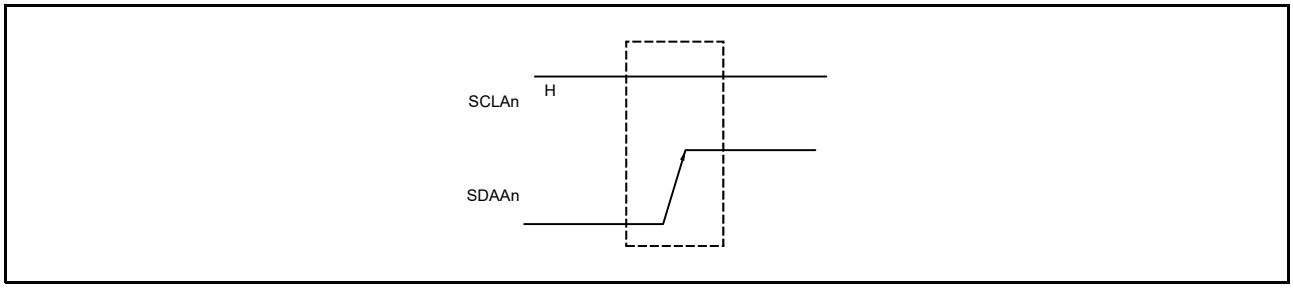
- When the falling edge of the eighth clock pulse is selected (bit 3 (WTIMn) of the IICCTLn0 register = 0):
  By setting the ACKEn bit to 1 before release from the clock stretch state, ACK is generated at the falling edge of the eighth clock cycle of the SCLAn pin.
- When the falling edge of the ninth clock pulse is selected (bit 3 (WTIMn) of the IICCTLn0 register = 1):
  ACK is generated if the ACKEn bit is set to 1 in advance.

Remark n = 0

25.5.5 Stop condition

When the SCLAn pin is at high level, changing the SDAAn pin from low level to high level generates a stop condition. A stop condition is a signal that the master device generates to the slave device when serial transfer has been completed. When the device is used as a slave, stop conditions can be detected.

Figure 25 - 20 Stop Condition



A stop condition is generated when bit 0 (SPTn) of IICA control register n0 (IICCTLn0) is set to 1. When the stop condition is detected, bit 0 (SPDn) of the IICA status register n (IICSn) is set to 1 and INTIICAn is generated when bit 4 (SPIEn) of the IICCTLn0 register is set to 1.

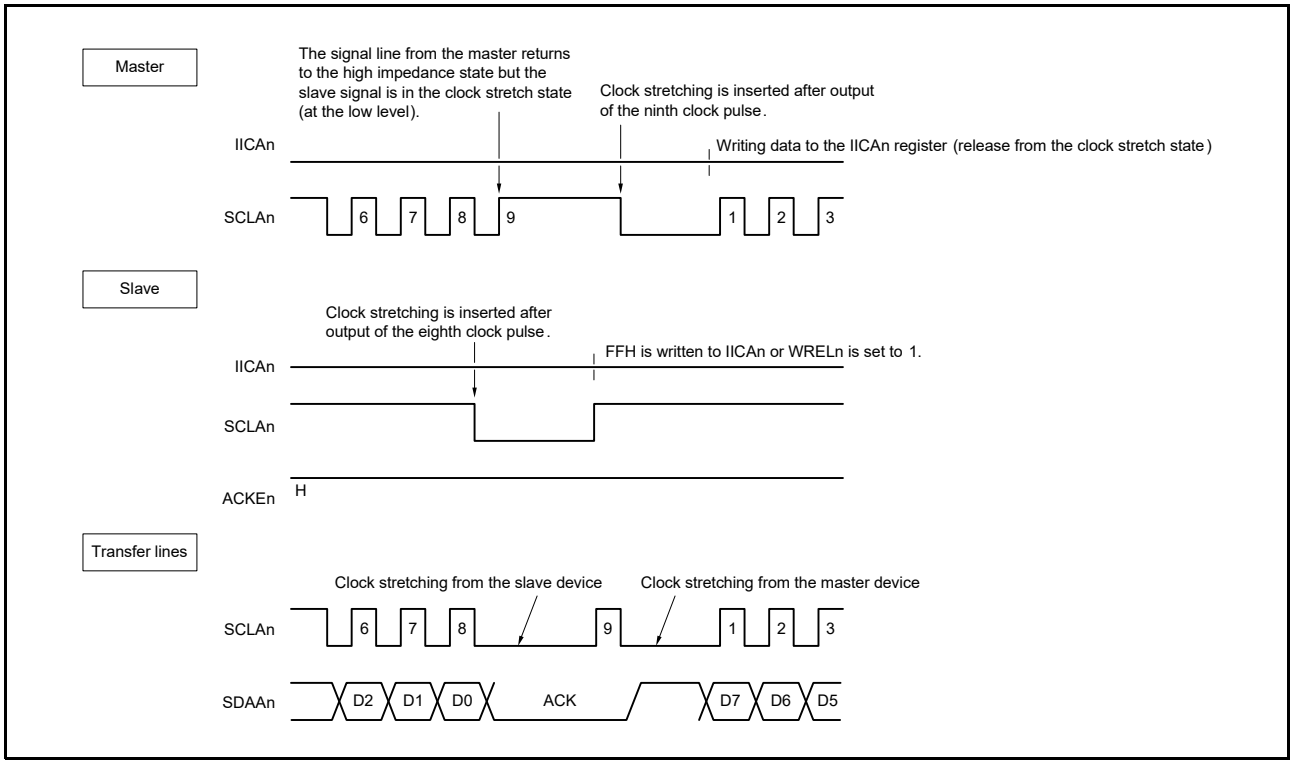
Remark n = 0

25.5.6 Clock stretching

Clock stretching is used to notify the other party in communications that a device (master or slave) is preparing to transmit or receive data (i.e., the interface is in the clock stretch state). Setting the SCLAn pin to the low level indicates the clock stretch state to the other party. When clock stretching is released for both the master and slave devices, the next data transfer can start.

Figure 25 - 21 Clock Stretching (1/2)

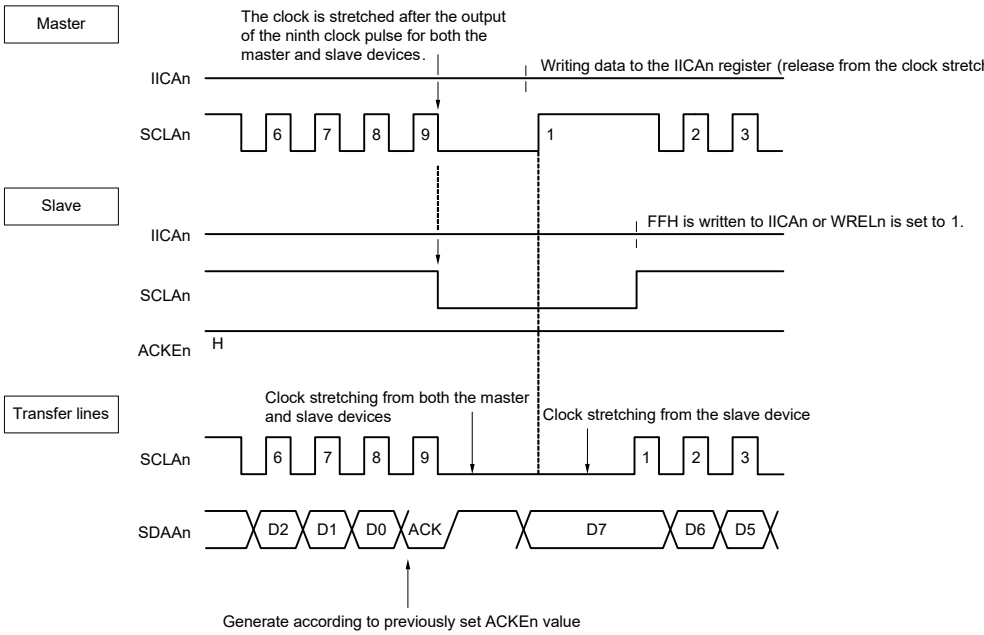
1. When clock stretching is inserted after the falling edge of the ninth clock pulse in the master device and the falling edge of the eighth clock pulse in the salve device
(master: transmission, slave: reception, and ACKEn = 1)



Remark n = 0

Figure 25 - 21 Clock Stretching (2/2)

2. When clock stretching is inserted after the falling edge of the ninth clock pulse in both the master and slave devices (master: transmission, slave: reception, and ACKEn = 1)



Remark ACKEn: Bit 2 of IICA control register n0 (IICCTLn0)
WRELn: Bit 5 of IICA control register n0 (IICCTLn0)

Clock stretching is automatically generated depending on the setting of bit 3 (WTIMn) of IICA control register n0 (IICCTLn0). Normally, the receiving side releases the clock stretch state when the IICCTLn0.WRELn bit is set to 1 or when FFH is written to the IICA shift register n (IICAn), and the transmitting side releases the clock stretch state when data is written to the IICAn register. The master device can also releases the clock stretch state via either of the following methods.

- By setting the IICCTLn0.STTn bit to 1
- By setting the IICCTLn0.SPTn bit to 1

Remark n = 0

25.5.7 Release from clock stretching

The I^2C interface usually releases the clock stretch state by the following processing.

- Writing data to the IICA shift register n (IICAn)
- Setting bit 5 (WRELn) of IICA control register n0 (IICCTLn0) (release from the clock stretch state)
- Setting bit 1 (STTn) of the IICCTLn0 register (generating start condition)**Note**
- Setting bit 0 (SPTn) of the IICCTLn0 register (generating stop condition)**Note**

Note Master only

Executing the above processing for release from clock stretching leads to IICA releasing the clock stretch state after which communications are resumed. To release the clock stretch state and transmit data (including addresses), write the data to the IICAn register. To receive data after release from the clock stretch state, or to complete data transmission, set bit 5 (WRELn) of the IICCTLn0 register to 1. To generate a restart condition after release from the clock stretch state, set bit 1 (STTn) of the IICCTLn0 register to 1. To generate a stop condition after release from the clock stretch state, set bit 0 (SPTn) of the IICCTLn0 register to 1. Execute the processing for release only once for each period in the clock stretch state. If, for example, data is written to the IICAn register after release from the clock stretch state by setting the WRELn bit to 1, an incorrect value may be output to SDAAn line because the timing for changing the SDAAn line conflicts with the timing for writing the IICAn register. In addition to the above, communications are stopped if the IICEn bit is cleared to 0 when communications have been aborted, so that the clock stretch state can be released. If the I^2C bus has deadlocked due to noise, the device can exit from communications by setting bit 6 (LRELn) of the IICCTLn0 register to 1, so that the clock stretch state can be released.

Caution **If the processing for release from clock stretching is executed when WUPn = 1, the clock stretch state will not be released.**

Remark n = 0

25.5.8 Timing of generation of the interrupt request signal (INTIICAn) and control of clock stretching

The setting of bit 3 (WTIMn) of IICA control register n0 (IICCTLn0) determines the timing by which INTIICAn is generated and controls clock stretching, as shown in **Table 25 - 2**.

Table 25 - 2 INTIICAn Generation Timing and Control of Clock Stretching

WTIMn	During Slave Device Operation			During Master Device Operation		
	Address	Data Reception	Data Transmission	Address	Data Reception	Data Transmission
0	9**Notes 1, 2**	8**Note 2**	8**Note 2**	9	8	8
1	9**Notes 1, 2**	9**Note 2**	9**Note 2**	9	9	9

Note 1. The slave device's INTIICAn signal and clock stretching occur at the falling edge of the ninth clock cycle only when there is a match with the address set to the slave address register n (SVAn). At this point, ACK is generated regardless of the value set to the IICCTLn0 register's bit 2 (ACKEn). For a slave device that has received an extension code, or has received an address while the all address match function is enabled, INTIICAn occurs at the falling edge of the eighth clock. However, if the address does not match after restart, INTIICAn is generated at the falling edge of the ninth clock cycle, but clock stretching does not occur.

Note 2. If the received address does not match the contents of the slave address register n (SVAn), the all address match function is disabled, and extension code is not received, neither INTIICAn nor clock stretching occurs.

Remark The numbers in the table indicate the pulses of the serial clock signal. Interrupt requests and control of clock stretching are both synchronized with the falling edge of these clock pulses.

1. During address transmission/reception
   Slave device operation: The timing of the interrupt and clock stretching depends on the conditions described in **Note 1** and **Note 2** above, regardless of the setting of the WTIMn bit.
   Master device operation: The interrupt and clock stretching occur at the falling edge of the ninth clock cycle, regardless of the setting of the WTIMn bit.

2. During data reception
   Master/slave device operation: The timing of the interrupt and clock stretching depends on the setting of the WTIMn bit.

3. During data transmission
   Master/slave device operation: The timing of the interrupt and clock stretching depends on the setting of the WTIMn bit.

Remark n = 0

4. Release from clock stretching
   The four types of processing for release from clock stretching are as follows.
   - Writing data to the IICA shift register n (IICAn)
   - Setting bit 5 (WRELn) of IICA control register n0 (IICCTLn0) (release from the clock stretch state)
   - Setting bit 1 (STTn) of the IICCTLn0 register (generating start condition)**Note**
   - Setting bit 0 (SPTn) of the IICCTLn0 register (generating stop condition)**Note**

   Note Master only

   When the falling edge of the eighth clock pulse is selected as the clock stretching timing (WTIMn = 0), the presence/absence of ACK generation must be determined before release from the clock stretch state.

5. Detection of stop condition
   INTIICAn is generated when a stop condition is detected (only when SPIEn = 1).

25.5.9 Address match detection method

In I^2C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address. Address match can be detected automatically by hardware. An interrupt request signal (INTIICAn) occurs only when the address set to the slave address register n (SVAn) matches the slave address sent by the master device, when an address is received while the all address match function is enabled (IICCTLn1.SVADISn = 1), or when an extension code has been received.

25.5.10 Error detection

In I^2C bus mode, the state of the serial data bus (SDAAn) during data transmission is captured by the IICA shift register n (IICAn) of the transmitting device, so the IICA data prior to transmission can be compared with the transmitted IICA data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

Remark n = 0

25.5.11 Extension code

1. When the higher 4 bits of the receive address are either 0000 or 1111, the detection of extension code reception flag (EXCn) is set to 1 for extension code reception and an interrupt request signal (INTIICAn) is issued at the falling edge of the eighth clock. When an address is received while the all address match function is enabled, it is also determined that an extension code has been received. The local address stored in the slave address register n (SVAn) is not affected.
2. The settings below are specified if 11110xx0 is transferred from the master by using a 10-bit address transfer while the SVAn register is set to 11110xx0 or if an address is received while the all address match function is enabled. Note that INTIICAn occurs at the falling edge of the eighth clock.
   - Higher four bits of data match or the all address match function is enabled: IICSn.EXCn = 1
   - Seven bits of data match or the all address match function is enabled: IICSn.COIn = 1
3. Since the processing after the interrupt request occurs differs according to the data that follows the extension code, such processing is performed by software. If the extension code is received or an address is received with the all address match function enabled during operation as a slave, then the slave device is participating in communication even if its address does not match. For example, after the extension code is received, if you do not wish to operate the target device as a slave device, set the IICCTLn0.LRELn bit to 1 to set the standby mode for the next communication operation.

Table 25 - 3 Bit Definitions of Major Extension Codes

Slave Address	R/$\overline{W}$ Bit	Description
0 0 0 0 0 0 0	0	General call address
1 1 1 1 0 x x	0	10-bit slave address specification (during address authentication)
1 1 1 1 0 x x	1	10-bit slave address specification (after address match, when read command is issued)

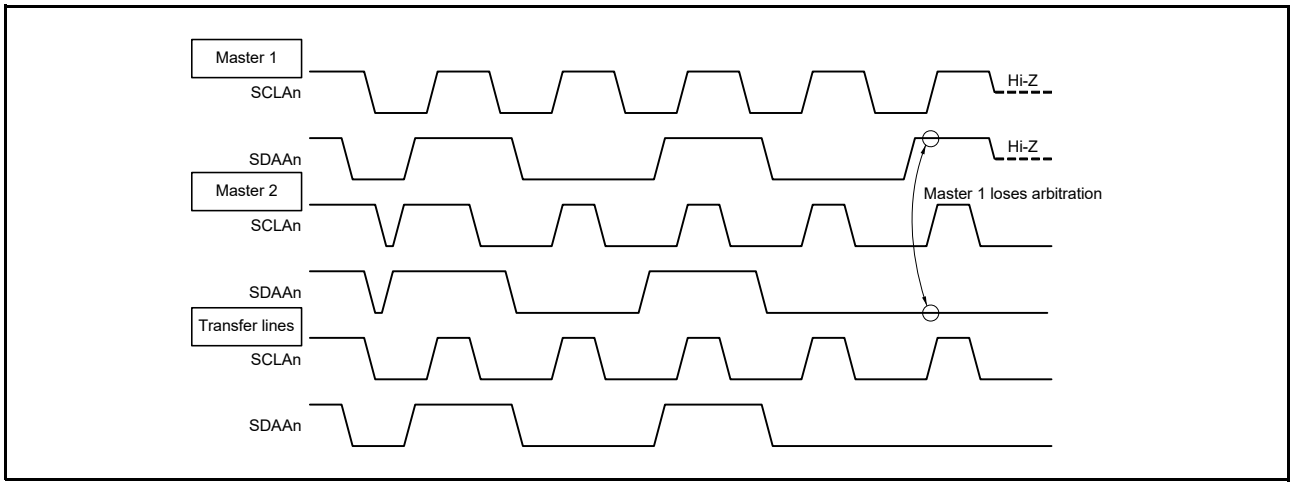
Remark 1. See the I^2C bus specifications issued by NXP Semiconductors for details of extension codes other than those described above.

Remark 2. n = 0

25.5.12 Arbitration

When several master devices simultaneously generate a start condition (when the IICCTLn0.STTn bit is set to 1 before the IICSn.STDn flag is set to 1), communication among the master devices is performed as the number of clocks are adjusted until the data differs. This kind of operation is called arbitration. When one of the master devices loses in arbitration, a detection of arbitration loss flag (ALDn) in the IICA status register n (IICSn) is set to 1 via the timing by which the arbitration loss occurred, and the SCLAn and SDAAn lines are both set to high impedance, which releases the bus. The arbitration loss is detected by checking ALDn = 1 by software at the timing of the next interrupt request (the eighth or ninth clock cycle, when a stop condition is detected, etc.). For details of interrupt request timing, see **25.5.8 Timing of generation of the interrupt request signal (INTIICAn) and control of clock stretching**.

Figure 25 - 22 Arbitration Timing Example



Remark n = 0

Table 25 - 4 State during Arbitration and Interrupt Request Generation Timing

State during Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer[Note 1]
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During ACK transfer period after data transmission	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is generated (when SPIEn = 1)[Note 2]
When data is at low level while attempting to generate a restart condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When stop condition is detected while attempting to generate a restart condition	When stop condition is generated (when SPIEn = 1)[Note 2]
When data is at low level while attempting to generate a stop condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When SCLAn is at low level while attempting to generate a restart condition	

Note 1. When IICCTLn0.WTIMn = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIMn = 0, the extension code's slave address is received, and an address is received while the all address match function is enabled, an interrupt request occurs at the falling edge of the eighth clock.

Note 2. When there is a chance that arbitration will occur, set SPIEn = 1 for master device operation.

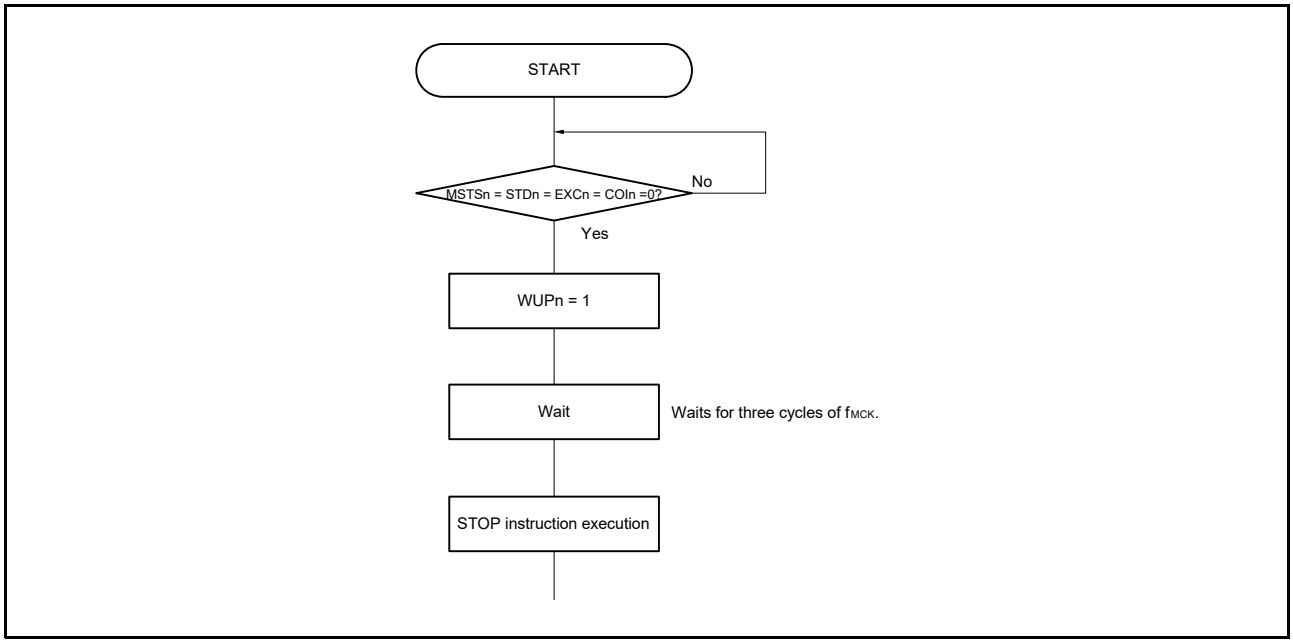
Remark 1. SPIEn: Bit 4 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0

25.5.13 Wakeup function

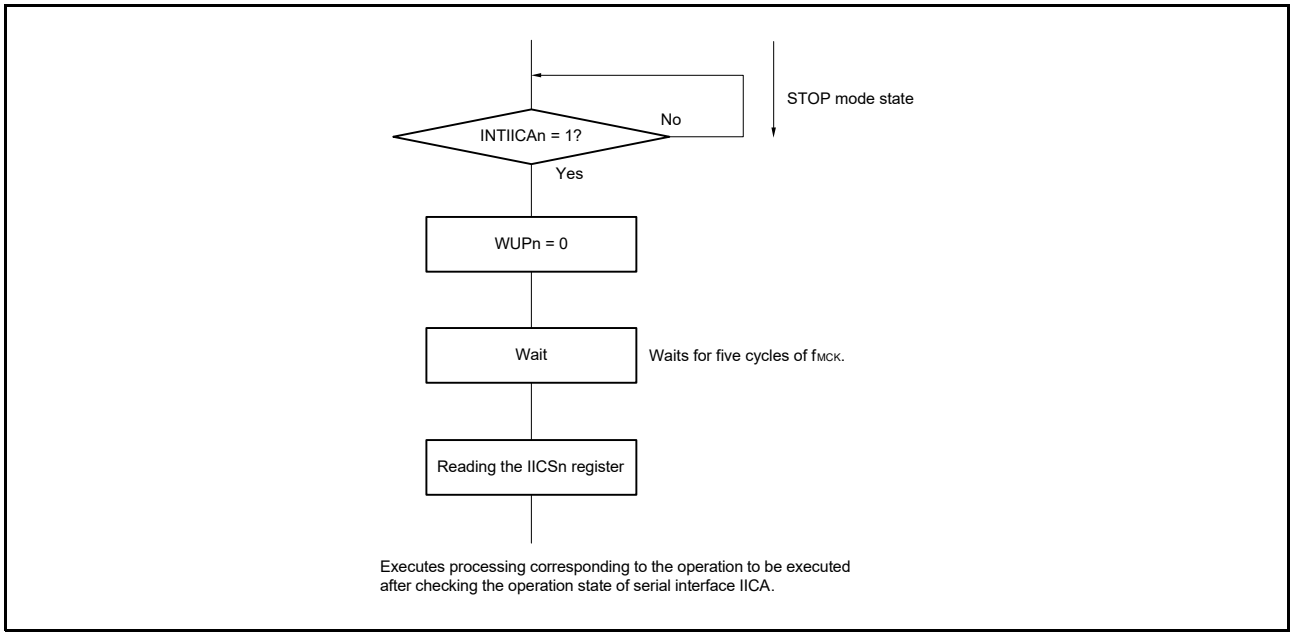
The I^2C bus slave function is a function that generates an interrupt request signal (INTIICAn) when the local address is received, an address is received while the all address match function is enabled, or an extension code is received. This function makes processing more efficient by preventing unnecessary INTIICAn signal from occurring when addresses do not match while the all address match function is disabled. When a start condition is detected, wakeup standby mode is set. Even a master that has generated a start condition enters the wakeup standby state while transmitting an address because the master may become a slave due to an arbitration loss. To use the wakeup function in the STOP mode, set the WUPn bit to 1. Addresses can be received regardless of the operation clock. An interrupt request signal (INTIICAn) is also generated when the local address is received, an address is received while the all address match function is enabled, or an extension code is received. Operation returns to normal operation by using an instruction to clear the WUPn bit to 0 after this interrupt has been generated. **Figure 25 - 23** shows the flow for setting WUPn = 1 and **Figure 25 - 24** shows the flow for setting WUPn = 0 upon an address match (or when the all address match function is enabled).

Figure 25 - 23 Flow When Setting WUPn = 1



Remark n = 0

Figure 25 - 24 Flow When Setting WUPn = 0 upon Address Match (or When the All Address Match Function is Enabled) (Including Extension Code Reception)

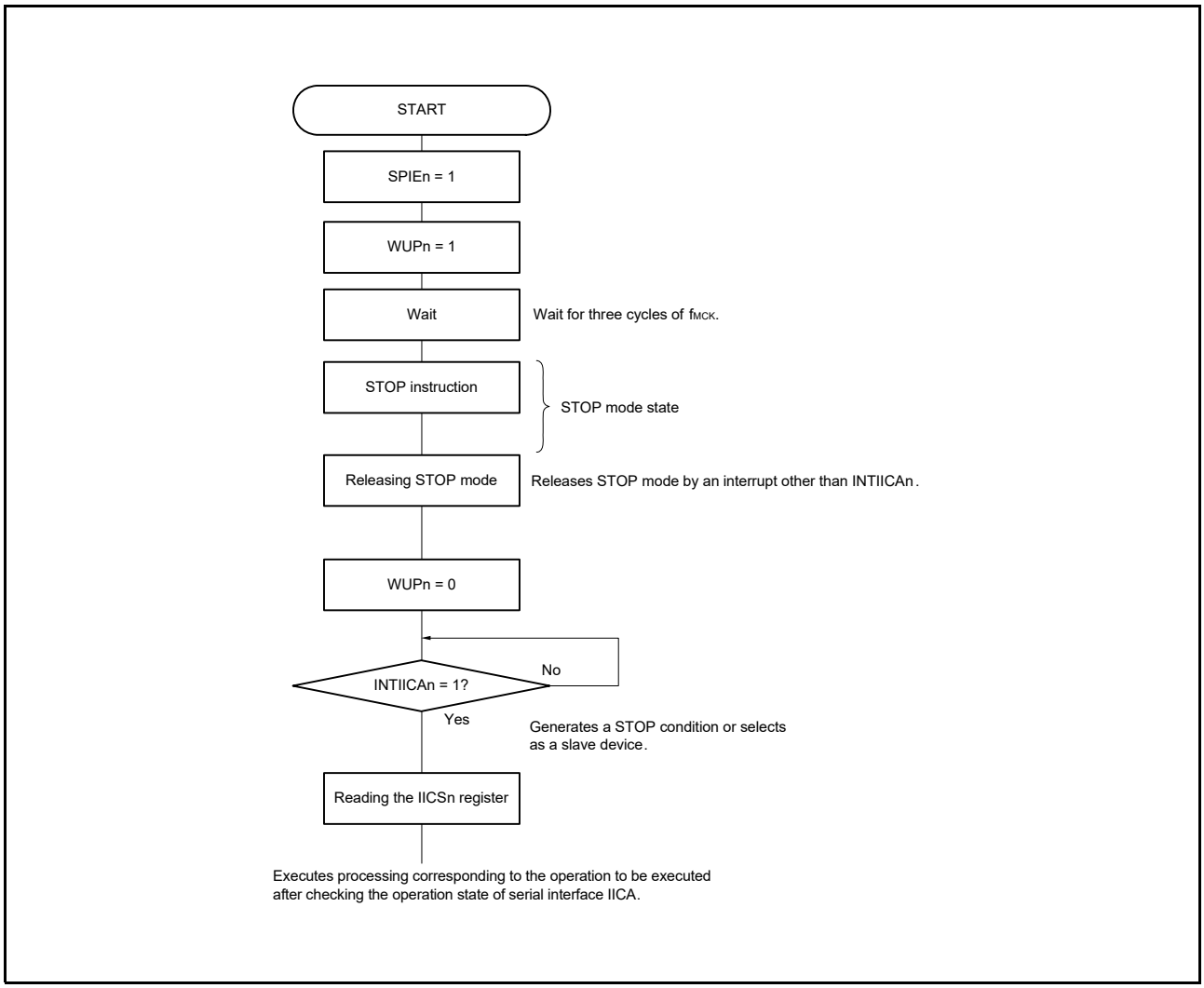


Use the following flows to perform the processing to release the STOP mode other than by an interrupt request signal (INTIICAn) generated from serial interface IICA.

- When operating next IIC communication as master:
  Flow shown in **Figure 25 - 25 When Operating as Master Device after Releasing STOP Mode Other Than by INTIICAn**.
- When operating next IIC communication as slave:
  When released by INTIICAn interrupt: Same as the flow in **Figure 25 - 24 Flow When Setting WUPn = 0 upon Address Match (or When the All Address Match Function is Enabled) (Including Extension Code Reception)**.
  When released by other than INTIICAn interrupt: Wait for INTIICAn interrupt with WUPn left set to 1.

Remark n = 0

Figure 25 - 25 When Operating as Master Device after Releasing STOP Mode Other Than by INTIICAn



Remark n = 0

25.5.14 Communication reservation

1. When communication reservation function is enabled (IICFn.IICRSVn = 0)
   To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

   - When arbitration results in neither master nor slave operation
   - While the all address match function is disabled, when an extension code is received and slave operation is disabled (ACK is not returned and the bus was released by setting the IICCTLn0.LRELn bit to 1 and exiting from communication)

   If bit 1 (STTn) of the IICCTLn0 register is set to 1 while the bus is not used, a start condition is automatically generated and wait state is entered after the bus is released (after a stop condition is detected). If an address is written to the IICA shift register n (IICAn) after bit 4 (SPIEn) of the IICCTLn0 register was set to 1, and it was detected by generation of an interrupt request signal (INTIICAn) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to the IICAn register before the stop condition is detected is invalid. When the STTn bit has been set to 1, the operation mode (as start condition or as communication reservation) is determined according to the state of the bus.

   - If the bus has been released: A start condition is generated
   - If the bus has not been released (standby mode): Communication reservation

   Check whether the communication reservation operates or not by using the MSTSn flag (bit 7 of the IICA status register n (IICSn)) after the STTn bit is set to 1 and the wait time elapses. Use software to secure the wait time calculated by the following expression.

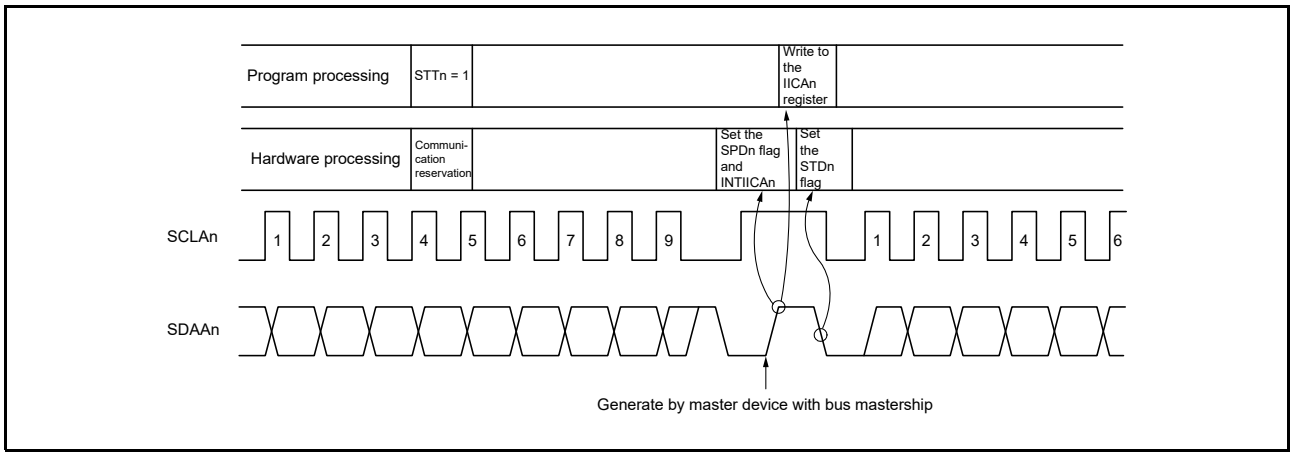
Wait time from setting STTn = 1 to checking the MSTSn flag: (IICWLn setting value + IICWHn setting value + 4)/f_{MCK} + t_F × 2

Remark 1. IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
t_F: SDAAn and SCLAn signal falling times
f_{MCK}: IICA operation clock frequency

Remark 2. n = 0

Figure 25 - 26 shows the communication reservation timing.

Figure 25 - 26 Communication Reservation Timing

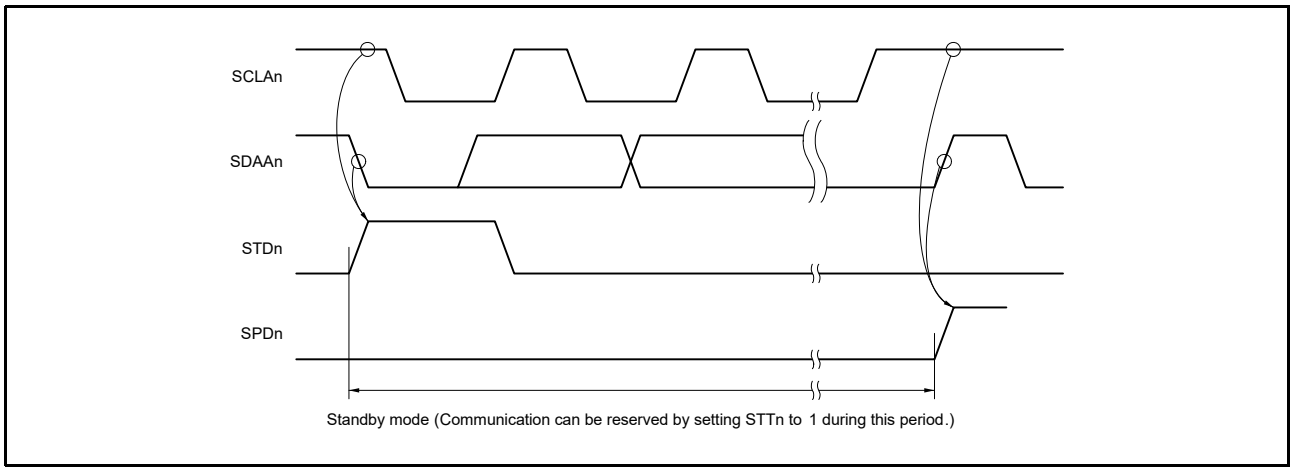


Remark 1. IICAn: IICA shift register n
STTn: Bit 1 of IICA control register n0 (IICCTLn0)
STDn: Bit 1 of IICA status register n (IICSn)
SPDn: Bit 0 of IICA status register n (IICSn)

Remark 2. n = 0

Communication reservations are accepted via the timing shown in **Figure 25 - 27**. After the IICSn.STDn flag is set to 1, a communication reservation can be made by setting the IICCTLn0.STTn bit to 1 before a stop condition is detected.

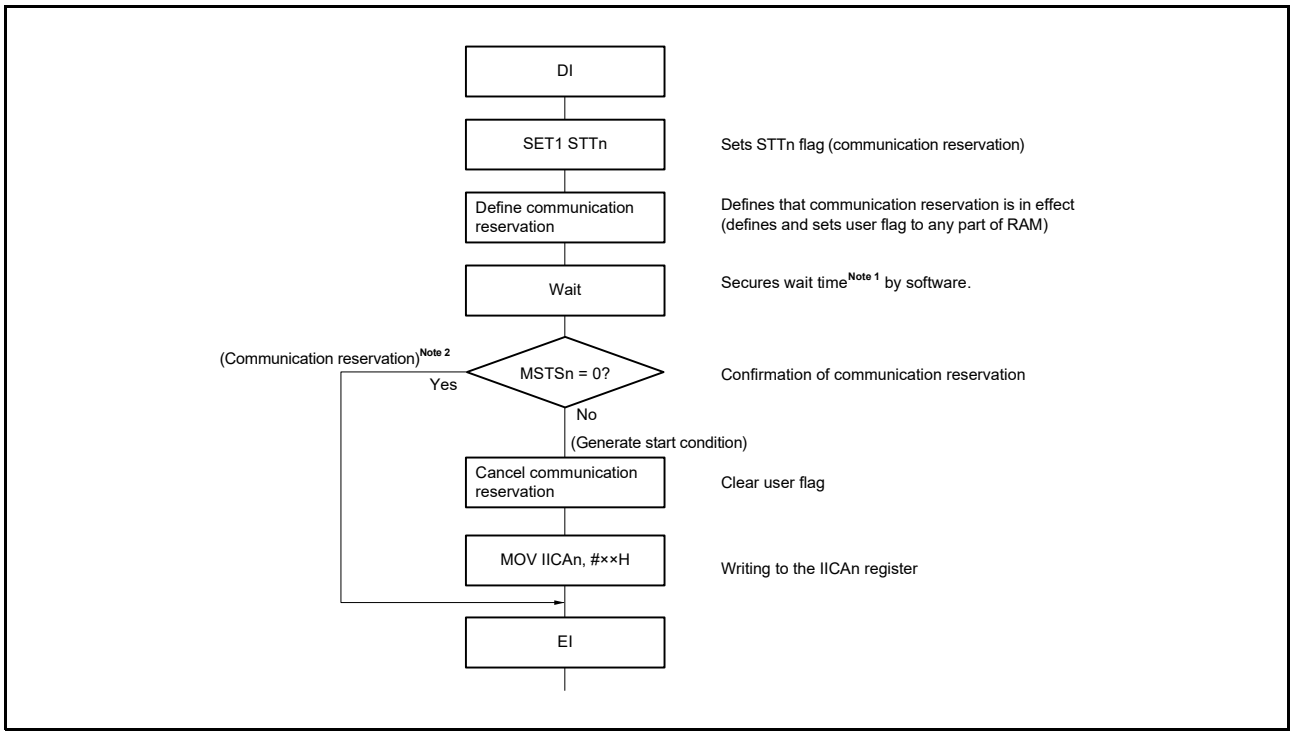
Figure 25 - 27 Timing for Accepting Communication Reservations



Remark n = 0

Figure 25 - 28 shows the communication reservation protocol.

Figure 25 - 28 Communication Reservation Protocol



Note 1. The wait time is calculated as follows.
(IICWLn setting value + IICWHn setting value + 4)/f_{MCK} + t_F × 2

Note 2. The communication reservation operation executes a write to the IICA shift register n (IICAn) when a stop condition interrupt request occurs.

Remark 1. STTn: Bit 1 of IICA control register n0 (IICCTLn0)
MSTSn: Bit 7 of IICA status register n (IICSn)
IICAn: IICA shift register n
IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
t_F: SDAAn and SCLAn signal falling times
f_{MCK}: IICA operation clock frequency

Remark 2. n = 0

2. When communication reservation function is disabled (IICFn.IICRSVn = 1)
When bit 1 (STTn) of IICA control register n0 (IICCTLn0) is set to 1 when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The states where the bus is not in use consist of the following two states.

- When arbitration results in neither master nor slave operation
- While the all address match function is disabled, when an extension code is received and slave operation is disabled (ACK is not returned and the bus was released by setting the IICCTLn0.LRELn bit to 1 and exiting from communication)

To confirm whether the start condition was generated or request was rejected, check the STCFn flag (bit 7 of the IICFn register). It takes up to 5 cycles of f_{MCK} until the STCFn flag is set to 1 after setting STTn = 1. Therefore, secure the time by software.

Remark n = 0

25.5.15 Cautions

1. When STCENn = 0
   Immediately after I^2C operation is enabled (IICEn = 1), the bus is recognized as being in a communications state (IICBSYn = 1) regardless of its actual state. When changing from a mode in which no stop condition has been detected to a master device communication mode, first generate a stop condition to release the bus, then perform master device communication. When using multiple masters, it is not possible to perform master device communication when the bus has not been released (when a stop condition has not been detected). Use the following sequence for generating a stop condition.

   <1> Set IICA control register n1 (IICCTLn1).
   <2> Set bit 7 (IICEn) of IICA control register n0 (IICCTLn0) to 1.
   <3> Set bit 0 (SPTn) of the IICCTLn0 register to 1.

2. When STCENn = 1
   Immediately after I^2C operation is enabled (IICEn = 1), the bus is recognized as being in the released state (IICBSYn = 0) regardless of its actual state. To generate the first start condition (STTn = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

3. If other I^2C communications are already in progress
   If I^2C operation is enabled and the device participates in communication already in progress when the SDAAn pin is low and the SCLAn pin is high, the IICA recognizes that the SDAAn pin has gone low (detects a start condition). If the value on the bus at this time can be recognized as an extension code or the all address match function is enabled, ACK is returned, but this interferes with other I^2C communications. To avoid this, start the IICA in the following sequence.

   <1> Clear bit 4 (SPIEn) of the IICCTLn0 register to 0 to disable generation of an interrupt request signal (INTIICAn) when the stop condition is detected.
   <2> Set bit 7 (IICEn) of the IICCTLn0 register to 1 to enable the operation of the IICA.
   <3> Wait for detection of the start condition.
   <4> Set bit 6 (LRELn) of the IICCTLn0 register to 1 before ACK is returned (4 to 72 cycles of f_{MCK} after setting the IICEn bit to 1), to forcibly disable detection.

4. Setting the STTn and SPTn bits (bits 1 and 0 of the IICCTLn0 register) again after they are set and before they are cleared to 0 is prohibited.

5. When transmission is reserved, set the SPIEn bit (bit 4 of the IICCTLn0 register) to 1 so that an interrupt request is generated when the stop condition is detected. Transfer is started when communication data is written to the IICA shift register n (IICAn) after the interrupt request is generated. Unless the interrupt is generated when the stop condition is detected, the device stops in the wait state because the interrupt request is not generated when communication is started. However, it is not necessary to set the SPIEn bit to 1 when the MSTSn flag (bit 7 of the IICA status register n (IICSn)) is detected by software.

Remark n = 0

25.5.16 Communication operations

The following shows three operation procedures with the flowchart.

1. Master operation in single-master system
   The flowchart when using the RL78/G24 as the master in a single master system is shown below. This flowchart is broadly divided into the initial settings and communication processing. Execute the initial settings at startup. If communication with the slave is required, prepare the communication and then execute communication processing.

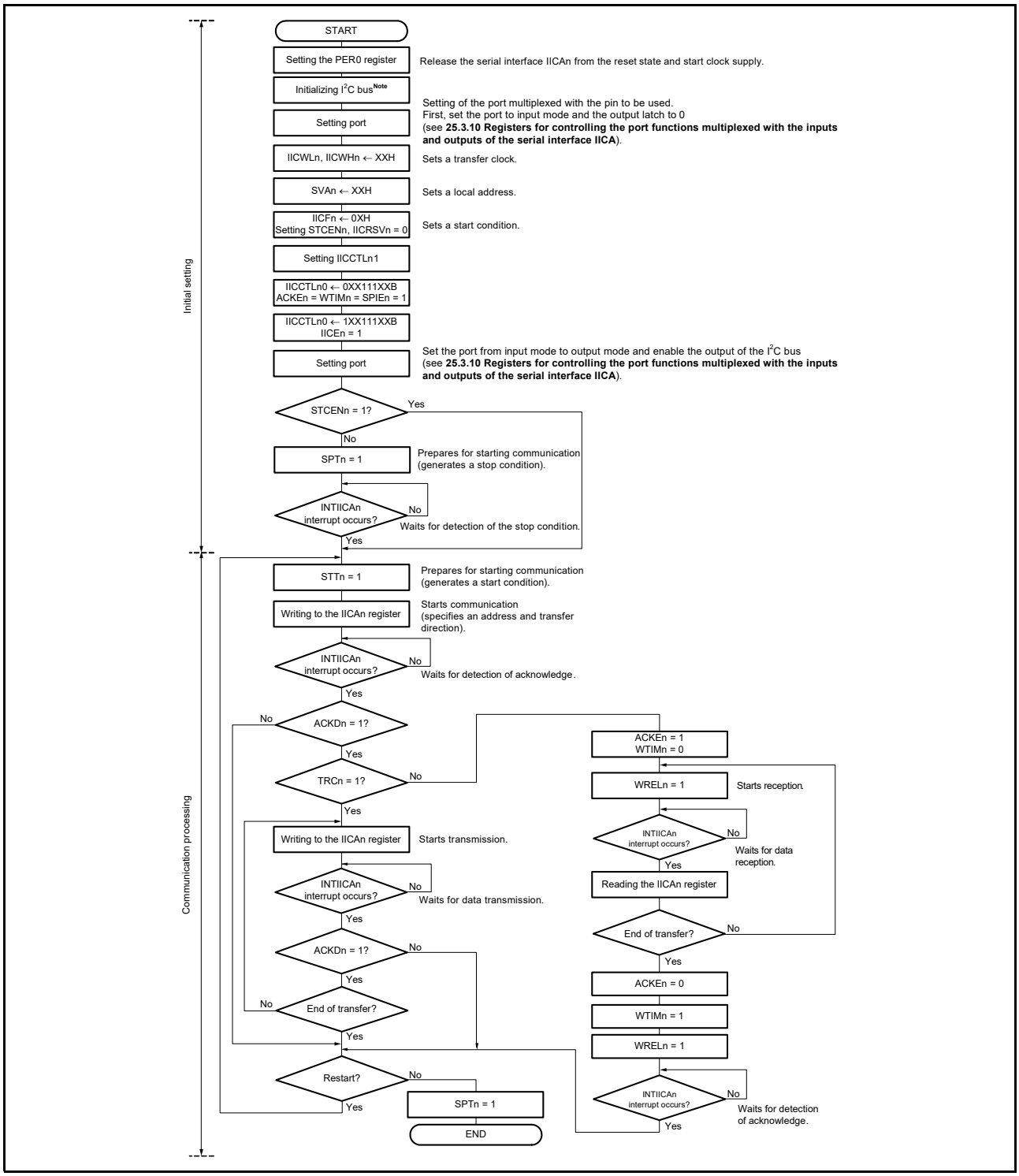
2. Master operation in multi-master system
   In the I^2C bus multi-master system, whether the bus is released or used cannot be judged by the I^2C bus specifications when a device takes part in a communication. Here, when data and clock are at a high level for a certain period (1 frame), the RL78/G24 takes part in a communication with bus released state. This flowchart is broadly divided into the initial settings, communication waiting, and communication processing. The processing when the RL78/G24 looses in arbitration and is specified as the slave is omitted here, and only the processing as the master is shown. Execute the initial settings at startup to take part in a communication. Then, wait for the communication request as the master or wait for the specification as the slave. The actual communication is performed in the communication processing, and it supports the transmission/reception with the slave and the arbitration with other masters.

3. Slave operation
   An example of when the RL78/G24 is used as the I^2C bus slave is shown below. When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIICAn interrupt occurrence (communication waiting). When an INTIICAn interrupt occurs, the communications state is judged and the result is passed as a flag over to the main processing. By checking the flags, necessary communication processing is performed.

Remark n = 0

1. Master operation in single-master system

Figure 25 - 29 Master Operation in Single-master System



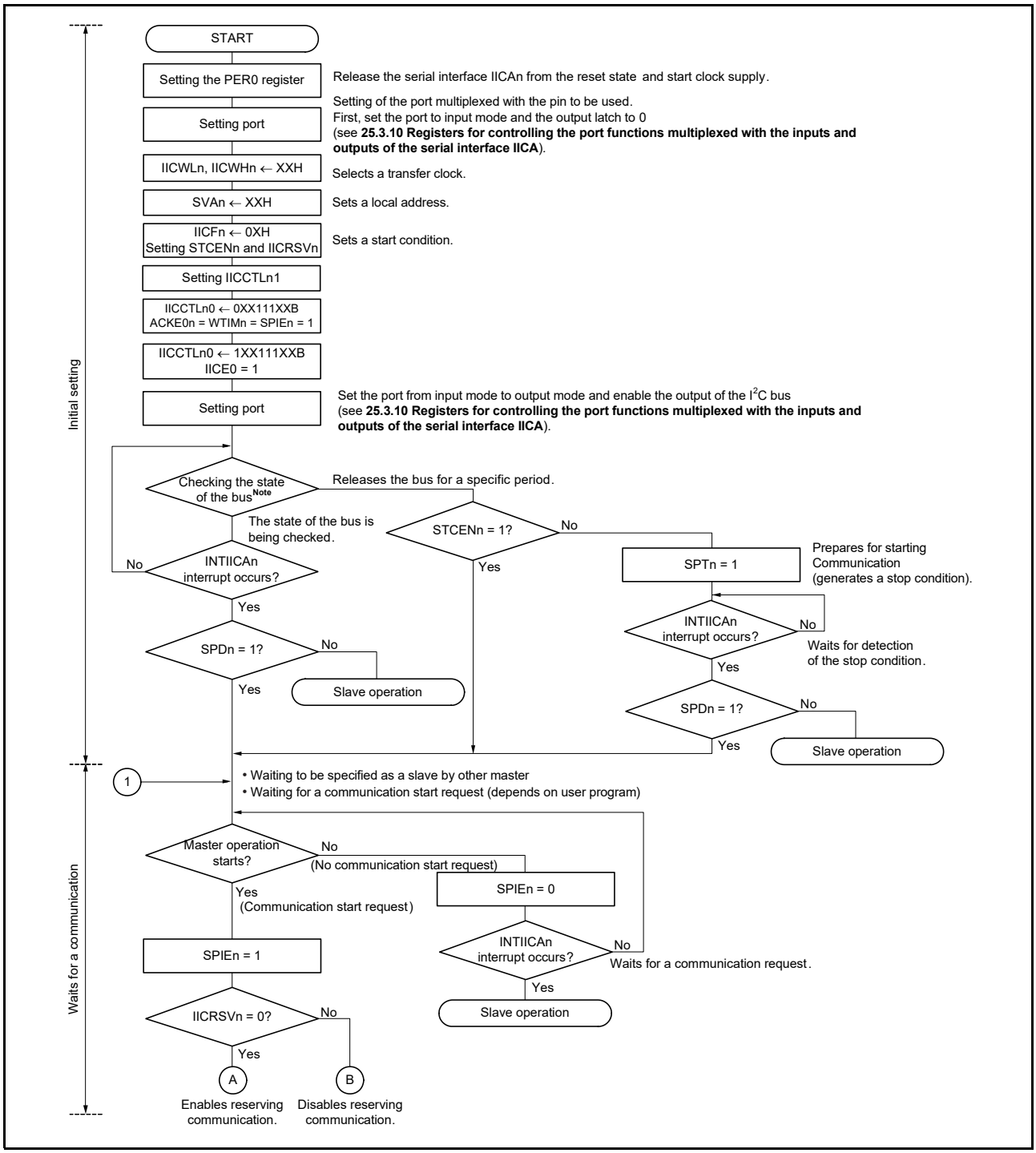
Note Release (SCLAn and SDAAn pins = high level) the I2C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAAn pin, for example, set the SCLAn pin in the output port mode, and output a clock pulse from the output port until the SDAAn pin is constantly at high level.

Remark 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

Remark 2. n = 0

2. Master operation in multi-master system

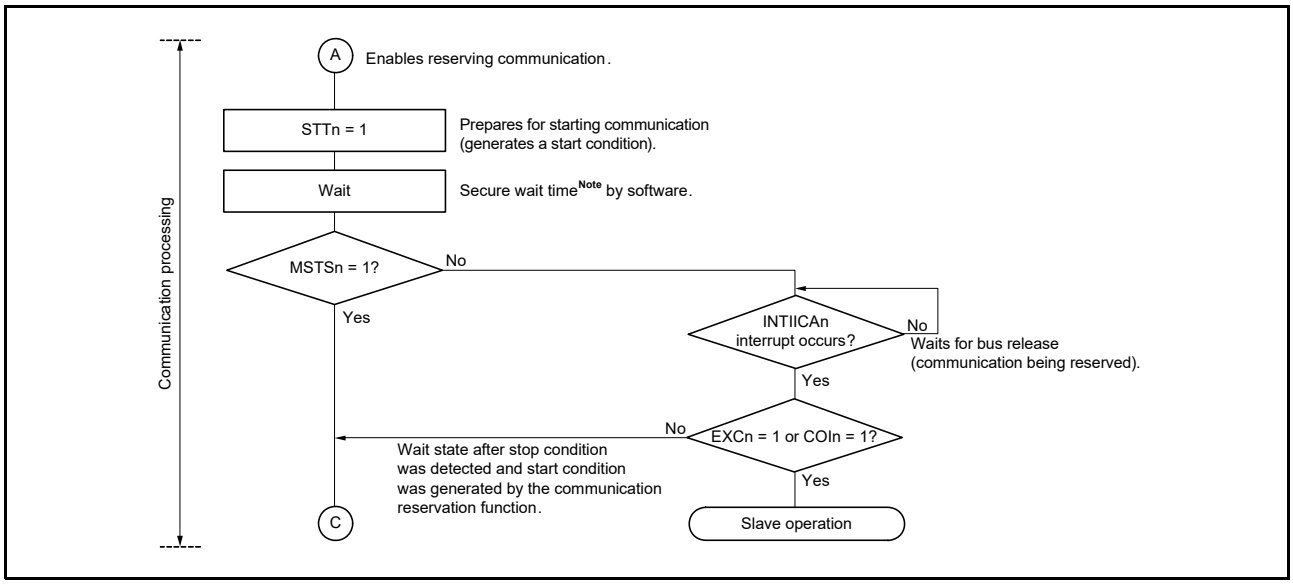
Figure 25 - 30 Master Operation in Multi-master System (1/3)



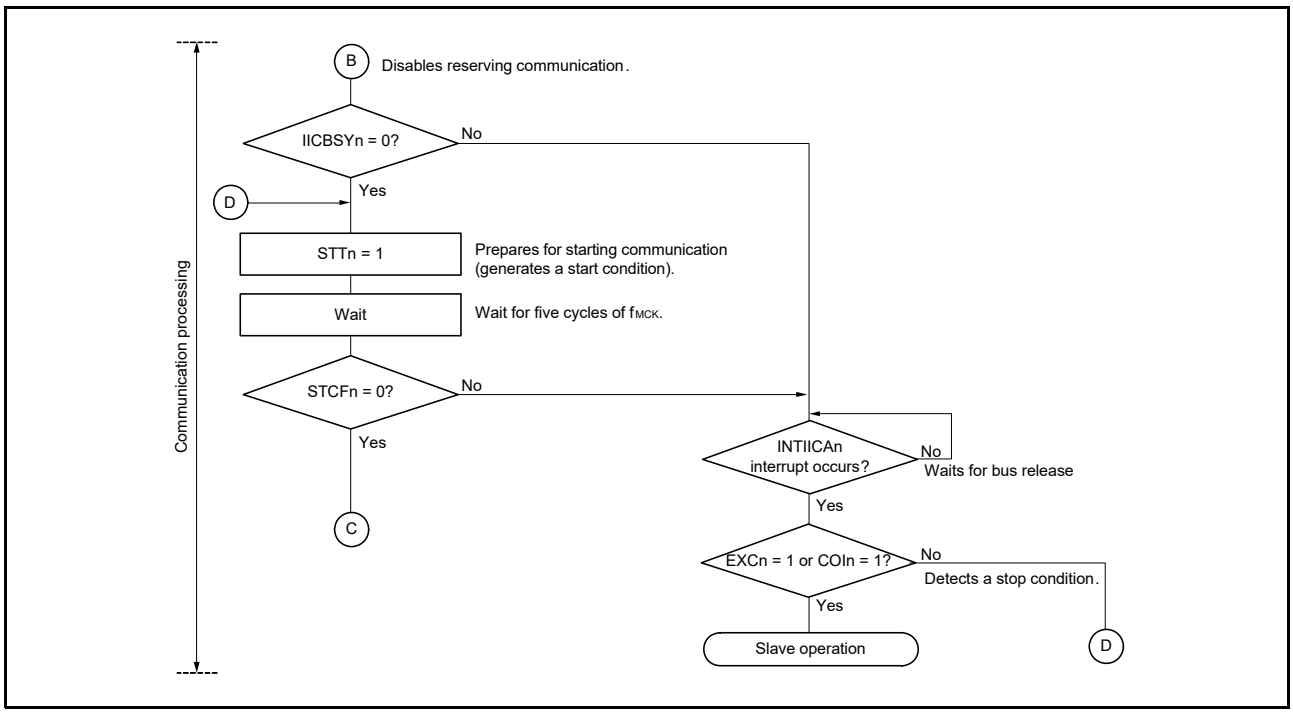
Note Confirm that the bus is released (CLDn = 1, DADn = 1) for a specific period (for example, for a period of one frame). If the SDAAn pin is constantly at low level, decide whether to release the I^2C bus (SCLAn and SDAAn pins = high level) in conformance with the specifications of the product that is communicating.

Remark n = 0

Figure 25 - 30 Master Operation in Multi-master System (2/3)



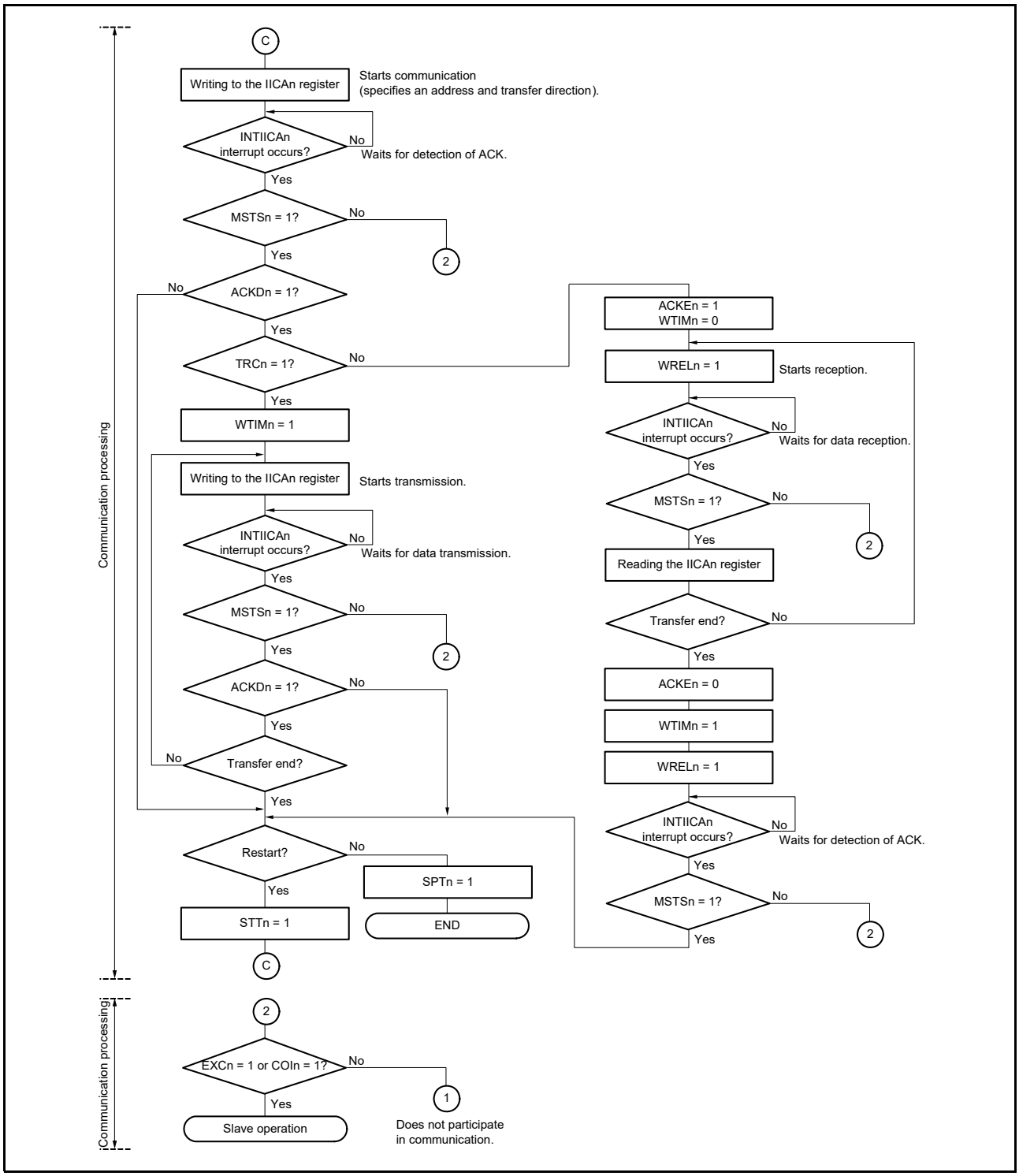
Note The wait time is calculated as follows.
(IICWLn setting value + IICWHn setting value + 4)/f_{MCK} + t_F × 2



Remark 1. IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
t_F: SDAAn and SCLAn signal falling times
f_{MCK}: IICA operation clock frequency

Remark 2. n = 0

Figure 25 - 30 Master Operation in Multi-master System (3/3)



Remark 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

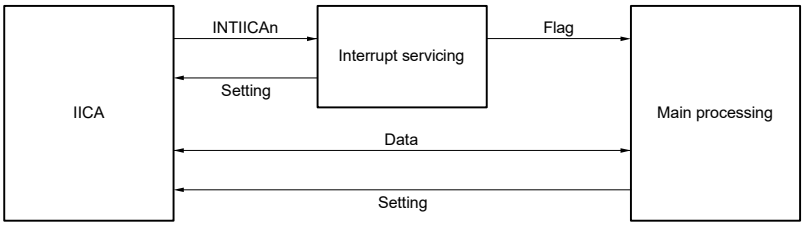
Remark 2. To use the device as a master in a multi-master system, read the MSTSn flag each time interrupt INTIICAn has occurred to check the arbitration result.

Remark 3. To use the device as a slave in a multi-master system, check the state by using the IICA status register n (IICSn) and IICA flag register n (IICFn) each time interrupt INTIICAn has occurred, and determine the processing to be performed next.

Remark 4. n = 0

3. Slave operation

The processing procedure of the slave operation is as follows. Basically, the slave operation is event-driven. Therefore, processing by the INTIICAn interrupt (processing that must substantially change the state of operating such as detection of a stop condition during communication) is necessary. In the following explanation, it is assumed that the all address match function is disabled and the extension code is not supported for data communication. It is also assumed that the INTIICAn interrupt servicing routine only handles state transition processing, and that the main processing handles actual data communications.



Therefore, data communication processing is performed by preparing the following three flags and passing them to the main processing instead of INTIICAn.

<1> Communication mode flag

This flag indicates the following two communications states.

- Clear mode: State in which data communications are not in progress
- Communication mode: State in which data communications are in progress (from valid address detection to stop condition detection, no detection of ACK from master, address mismatch)

<2> Ready flag

This flag indicates that data communication is enabled. Its function is the same as the INTIICAn interrupt for ordinary data communication. This flag is set by interrupt servicing and cleared by the main processing. Clear this flag by interrupt servicing when communication is started. However, the ready flag is not set by interrupt servicing when the first data is transmitted. Therefore, the first data is transmitted without the flag being cleared (an address match is interpreted as a request for the next data).

<3> Communication direction flag

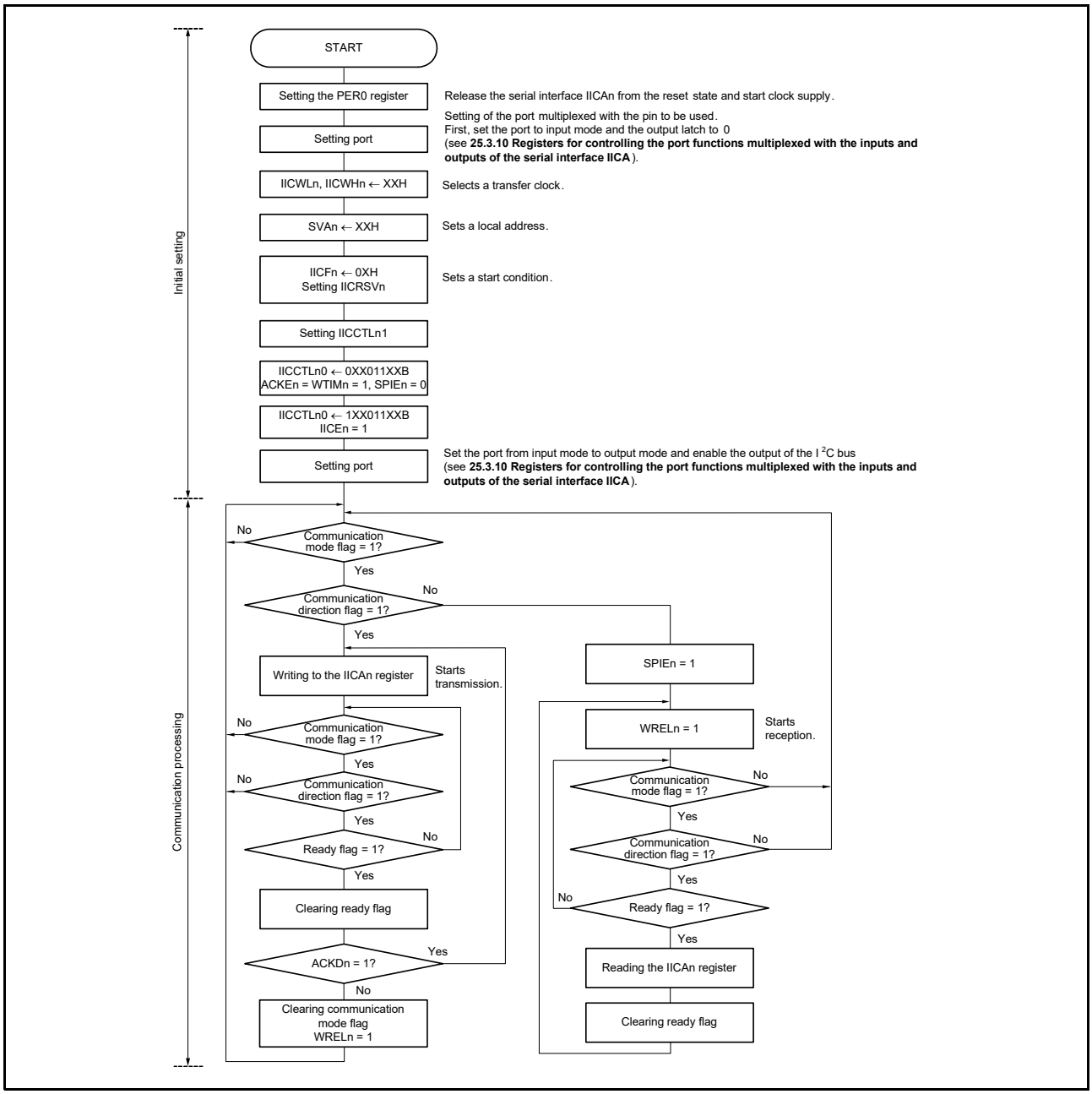
This flag indicates the direction of communication. Its value is the same as the TRCn flag.

Remark n = 0

The main processing of the slave operation is explained next. Start serial interface IICA and wait until communication is enabled. When communication is enabled, execute communication by using the communication mode flag and ready flag (processing of the stop condition and start condition is performed by an interrupt. Here, check the state by using the flags). The transmission operation is repeated until the master no longer returns ACK. If ACK is not returned from the master, communication is completed. For reception, the necessary amount of data is received. When communication is completed, ACK is not returned as the next data. After that, the master generates a stop condition or restart condition. Exit from the communications state occurs in this way.

Figure 25 - 31 Slave Operation Flowchart (1)

<R>



Remark 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

Remark 2. n = 0

An example of the processing procedure of the slave with the INTIICAn interrupt is explained below (processing is performed assuming that the all address match function is disabled and no extension code is used). The INTIICAn interrupt checks the state, and the following operations are performed.

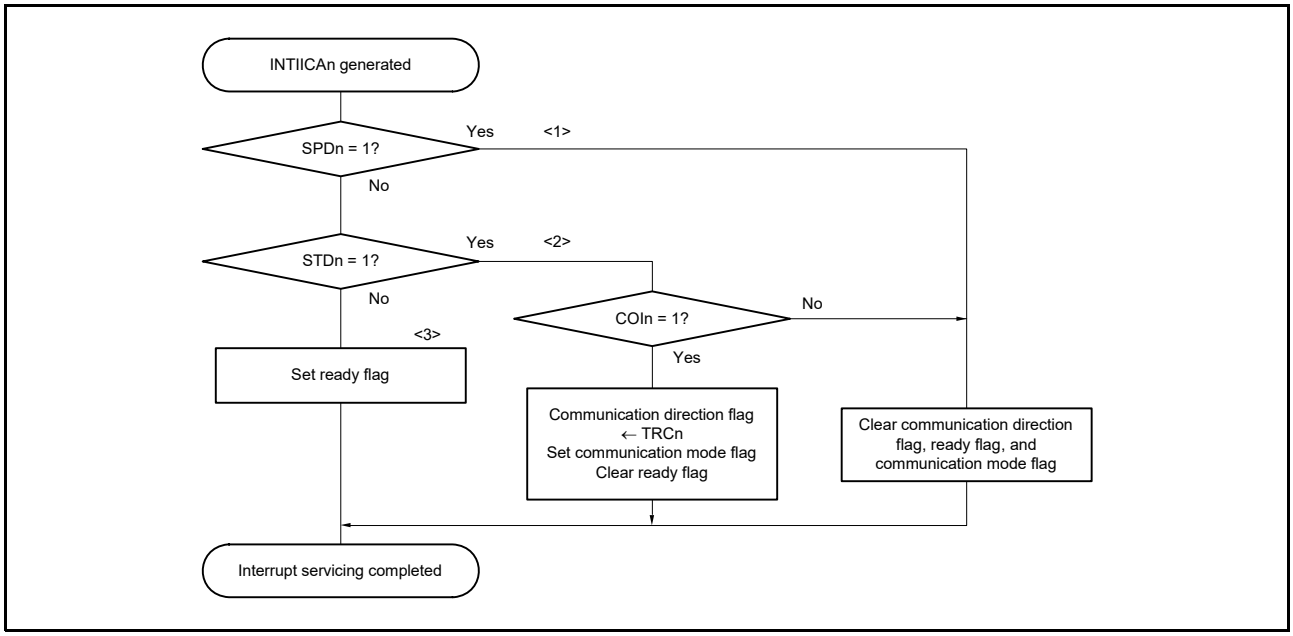
<1> Communication is stopped if the stop condition is issued.

<2> If the start condition is issued, the address is checked and communication is completed if the address does not match. If the address matches, the communication mode is set, wait is canceled, and processing returns from the interrupt (the ready flag is cleared).

<3> For data transmit/receive, only the ready flag is set. Processing returns from the interrupt with the I^2C bus remaining in the wait state.

Remark <1> to <3> above correspond to <1> to <3> in **Figure 25 - 32 Slave Operation Flowchart (2)**.

Figure 25 - 32 Slave Operation Flowchart (2)



Remark n = 0

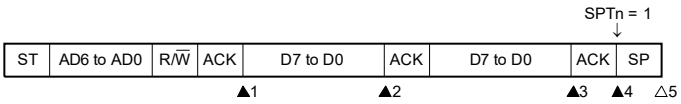
25.5.17 Timing of I2C interrupt request signal (INTIICAn) occurrence

The timing of transmitting or receiving data and generation of interrupt request signal INTIICAn, and the value of the IICA status register n (IICSn) when the INTIICAn signal is generated are shown below.

Remark 1. ST: Start condition
AD6 to AD0: Address
R/$\overline{W}$: Transfer direction specification
ACK: Acknowledge
D7 to D0: Data
SP: Stop condition

Remark 2. n = 0

1. Master device operation
   a) Start ~ Address ~ Data ~ Data ~ Stop (transmission/reception)
      i) When WTIMn = 0

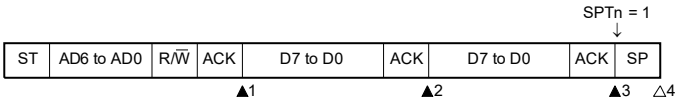


▲1: IICSn = 1000×110B
▲2: IICSn = 1000×000B
▲3: IICSn = 1000×000B (Sets the WTIMn bit to 1)**Note**
▲4: IICSn = 1000××00B (Sets the SPTn bit to 1)
△5: IICSn = 00000001B

Note To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

      ii) When WTIMn = 1



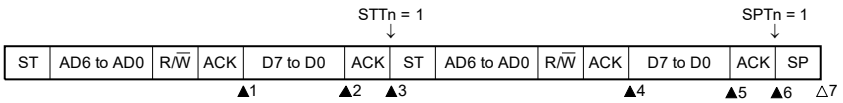
▲1: IICSn = 1000×110B
▲2: IICSn = 1000×100B
▲3: IICSn = 1000××00B (Sets the SPTn bit to 1)
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0

b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

i) When WTIMn = 0



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1)**Note 1**

▲3: IICSn = 1000××00B (Clears the WTIMn bit to 0**Note 2**, sets the STTn bit to 1)

▲4: IICSn = 1000×110B

▲5: IICSn = 1000×000B (Sets the WTIMn bit to 1)**Note 3**

▲6: IICSn = 1000××00B (Sets the SPTn bit to 1)

△7: IICSn = 00000001B

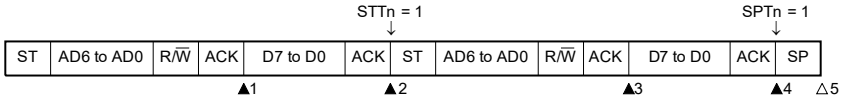
Note 1. To generate a start condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

Note 2. Clear the WTIMn bit to 0 to restore the original setting.

Note 3. To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1



▲1: IICSn = 1000×110B

▲2: IICSn = 1000××00B (Sets the STTn bit to 1)

▲3: IICSn = 1000×110B

▲4: IICSn = 1000××00B (Sets the SPTn bit to 1)

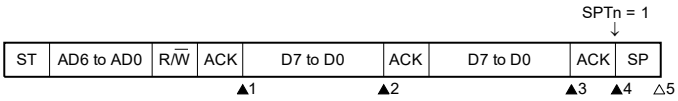
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0

c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

i) When WTIMn = 0



▲1: IICSn = 1010×110B

▲2: IICSn = 1010×000B

▲3: IICSn = 1010×000B (Sets the WTIMn bit to 1)**Note**

▲4: IICSn = 1010××00B (Sets the SPTn bit to 1)

△5: IICSn = 00000001B

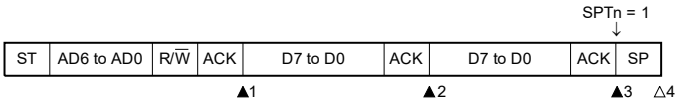
Note To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

ii) When WTIMn = 1



▲1: IICSn = 1010×110B

▲2: IICSn = 1010×100B

▲3: IICSn = 1010××00B (Sets the SPTn bit to 1)

△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

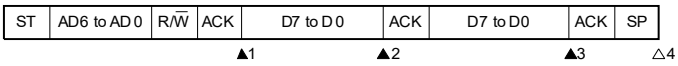
×: Don't care

Remark n = 0

2. Slave device operation (slave address data reception)

a) Start ~ Address ~ Data ~ Data ~ Stop

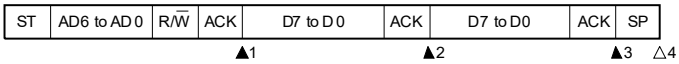
i) When WTIMn = 0



▲1: IICSn = 0001×110B
▲2: IICSn = 0001×000B
▲3: IICSn = 0001×000B
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1



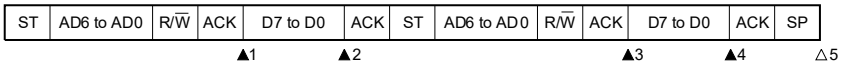
▲1: IICSn = 0001×110B
▲2: IICSn = 0001×100B
▲3: IICSn = 0001××00B
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0

b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

i) When WTIMn = 0 (after restart, matches with SVAn, the all address match function is disabled)



▲1: IICSn = 0001×110B
▲2: IICSn = 0001×000B
▲3: IICSn = 0001×110B
▲4: IICSn = 0001×000B
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1 (after restart, matches with SVAn, the all address match function is disabled)

▲1: IICSn = 0001×110B
▲2: IICSn = 0001××00B
▲3: IICSn = 0001×110B
▲4: IICSn = 0001××00B
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0

c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

i) When WTIMn = 0

(after restart, does not match address (= extension code, the all address match function is disabled))

▲1: IICSn = 0001×110B
▲2: IICSn = 0001×000B
▲3: IICSn = 0010×010B
▲4: IICSn = 0010×000B
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1

(after restart, does not match address (= extension code, the all address match function is disabled))

▲1: IICSn = 0001×110B
▲2: IICSn = 0001××00B
▲3: IICSn = 0010×010B
▲4: IICSn = 0010×110B
▲5: IICSn = 0010××00B
△6: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0

d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

i) When WTIMn = 0

(after restart, does not match address (= not extension code, the all address match function is disabled))

▲1: IICSn = 0001×110B
▲2: IICSn = 0001×000B
▲3: IICSn = 00000×10B
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1

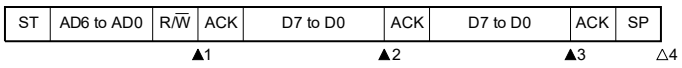
(after restart, does not match address (= not extension code, the all address match function is disabled))

▲1: IICSn = 0001×110B
▲2: IICSn = 0001××00B
▲3: IICSn = 00000×10B
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0

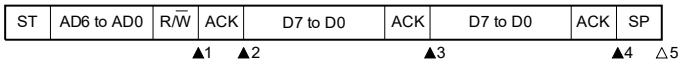
3. Slave device operation (when receiving extension code and the all address match function is disabled)
   The device is always participating in communication when it receives an extension code.
   a) Start ~ Code ~ Data ~ Data ~ Stop
      i) When WTIMn = 0



▲1: IICSn = 0010×010B
▲2: IICSn = 0010×000B
▲3: IICSn = 0010×000B
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

      ii) When WTIMn = 1



▲1: IICSn = 0010×010B
▲2: IICSn = 0010×110B
▲3: IICSn = 0010×100B
▲4: IICSn = 0010××00B
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0

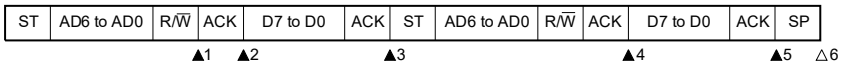
b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

i) When WTIMn = 0 (after restart, matches with SVAn, the all address match function is disabled)

▲1: IICSn = 0010×010B
▲2: IICSn = 0010×000B
▲3: IICSn = 0001×110B
▲4: IICSn = 0001×000B
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1 (after restart, matches with SVAn, the all address match function is disabled)



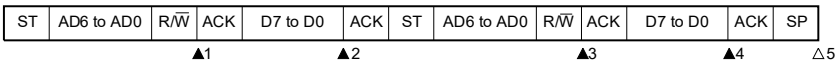
▲1: IICSn = 0010×010B
▲2: IICSn = 0010×110B
▲3: IICSn = 0010××00B
▲4: IICSn = 0001×110B
▲5: IICSn = 0001××00B
△6: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0

c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

i) When WTIMn = 0 (after restart, extension code reception, the all address match function is disabled)



▲1: IICSn = 0010×010B
▲2: IICSn = 0010×000B
▲3: IICSn = 0010×010B
▲4: IICSn = 0010×000B
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1 (after restart, extension code reception, the all address match function is disabled)

▲1: IICSn = 0010×010B
▲2: IICSn = 0010×110B
▲3: IICSn = 0010××00B
▲4: IICSn = 0010×010B
▲5: IICSn = 0010×110B
▲6: IICSn = 0010××00B
△7: IICSn = 00000001B

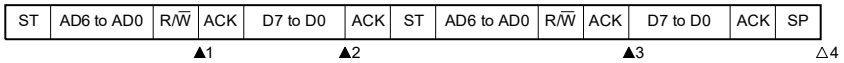
Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0

d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

i) When WTIMn = 0

(after restart, does not match address (= not extension code, the all address match function is disabled))



▲1: IICSn = 0010×010B
▲2: IICSn = 0010×000B
▲3: IICSn = 00000×10B
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1

(after restart, does not match address (= not extension code, the all address match function is disabled))

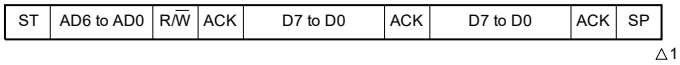
▲1: IICSn = 0010×010B
▲2: IICSn = 0010×110B
▲3: IICSn = 0010××00B
▲4: IICSn = 00000×10B
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0

4. Operation without communication

a) Start ~ Code ~ Data ~ Data ~ Stop

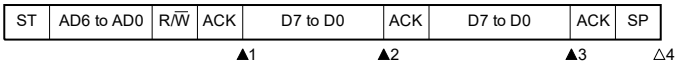


△1: IICSn = 00000001B

Remark △: Generated only when SPIEn = 1

Remark n = 0

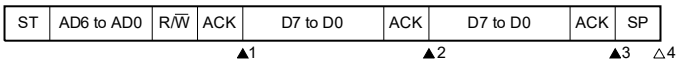
5. Arbitration loss operation (operation as slave after arbitration loss)
   When the device is used as a master in a multi-master system, read the MSTSn flag each time interrupt request signal INTIICAn has occurred to check the arbitration result.
   a) When arbitration loss occurs during transmission of slave address data
      i) When WTIMn = 0



▲1: IICSn = 0101×110B
▲2: IICSn = 0001×000B
▲3: IICSn = 0001×000B
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

      ii) When WTIMn = 1



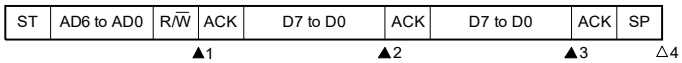
▲1: IICSn = 0101×110B
▲2: IICSn = 0001×100B
▲3: IICSn = 0001××00B
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0

b) When arbitration loss occurs during transmission of extension code (the all address match function is disabled)

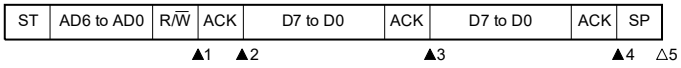
i) When WTIMn = 0



▲1: IICSn = 0110×010B
▲2: IICSn = 0010×000B
▲3: IICSn = 0010×000B
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1

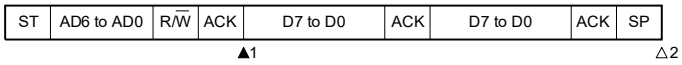


▲1: IICSn = 0110×010B
▲2: IICSn = 0010×110B
▲3: IICSn = 0010×100B
▲4: IICSn = 0010××00B
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0

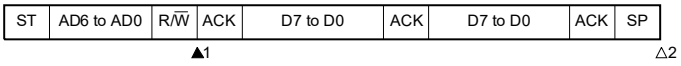
6. Operation when arbitration loss occurs (no communication after arbitration loss)
   When the device is used as a master in a multi-master system, read the MSTSn flag each time interrupt request signal INTIICAn has occurred to check the arbitration result.
   a) When arbitration loss occurs during transmission of slave address data (when WTIMn = 1)



▲1: IICSn = 01000110B
△2: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1

   b) When arbitration loss occurs during transmission of extension code (the all address match function is disabled)



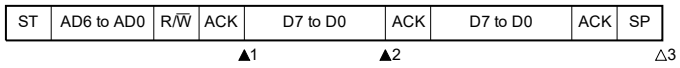
▲1: IICSn = 0110×010B
Sets LRELn = 1 by software
△2: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0

c) When arbitration loss occurs during transmission of data

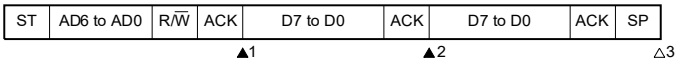
i) When WTIMn = 0



▲1: IICSn = 10001110B
▲2: IICSn = 01000000B
△3: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1

ii) When WTIMn = 1



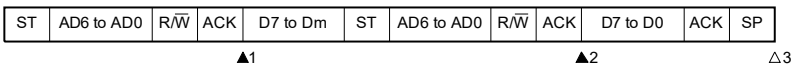
▲1: IICSn = 10001110B
▲2: IICSn = 01000100B
△3: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1

Remark n = 0

d) When loss occurs due to restart condition during data transfer

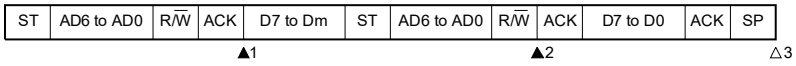
i) Not extension code (Example: unmatches with SVAn, the all address match function is disabled)



▲1: IICSn = 1000×110B
▲2: IICSn = 01000110B
△3: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care
m = 6 to 0

ii) Extension code (the all address match function is disabled)

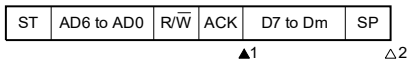


▲1: IICSn = 1000×110B
▲2: IICSn = 01100010B
Sets LRELn = 1 by software
△3: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care
m = 6 to 0

Remark n = 0

e) When loss occurs due to stop condition during data transfer



▲1: IICSn = 10000110B

△2: IICSn = 01000001B

Remark ▲: Always generated

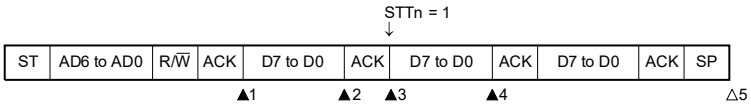
△: Generated only when SPIEn = 1

×: Don't care

m = 6 to 0

f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

i) When WTIMn = 0



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1)

▲3: IICSn = 1000×100B (Clears the WTIMn bit to 0)

▲4: IICSn = 01000000B

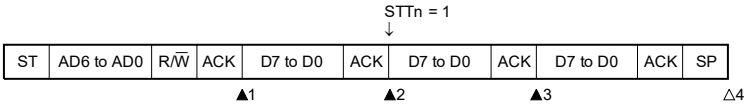
△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

ii) When WTIMn = 1



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×100B (Sets the STTn bit to 1)

▲3: IICSn = 01000100B

△4: IICSn = 00000001B

Remark ▲: Always generated

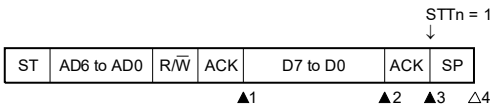
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0

g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

i) When WTIMn = 0



▲1: IICSn = 1000×110B
▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1)
▲3: IICSn = 1000××00B (Sets the STTn bit to 1)
△4: IICSn = 01000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1

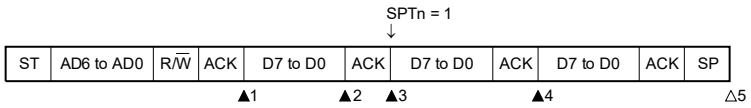
▲1: IICSn = 1000×110B
▲2: IICSn = 1000××00B (Sets the STTn bit to 1)
△3: IICSn = 01000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0

h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

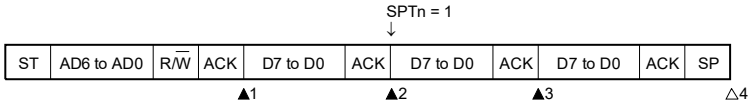
i) When WTIMn = 0



▲1: IICSn = 1000×110B
▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1)
▲3: IICSn = 1000×100B (Clears the WTIMn bit to 0)
▲4: IICSn = 01000100B
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1



▲1: IICSn = 1000×110B
▲2: IICSn = 1000×100B (Sets the SPTn bit to 1)
▲3: IICSn = 01000100B
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0

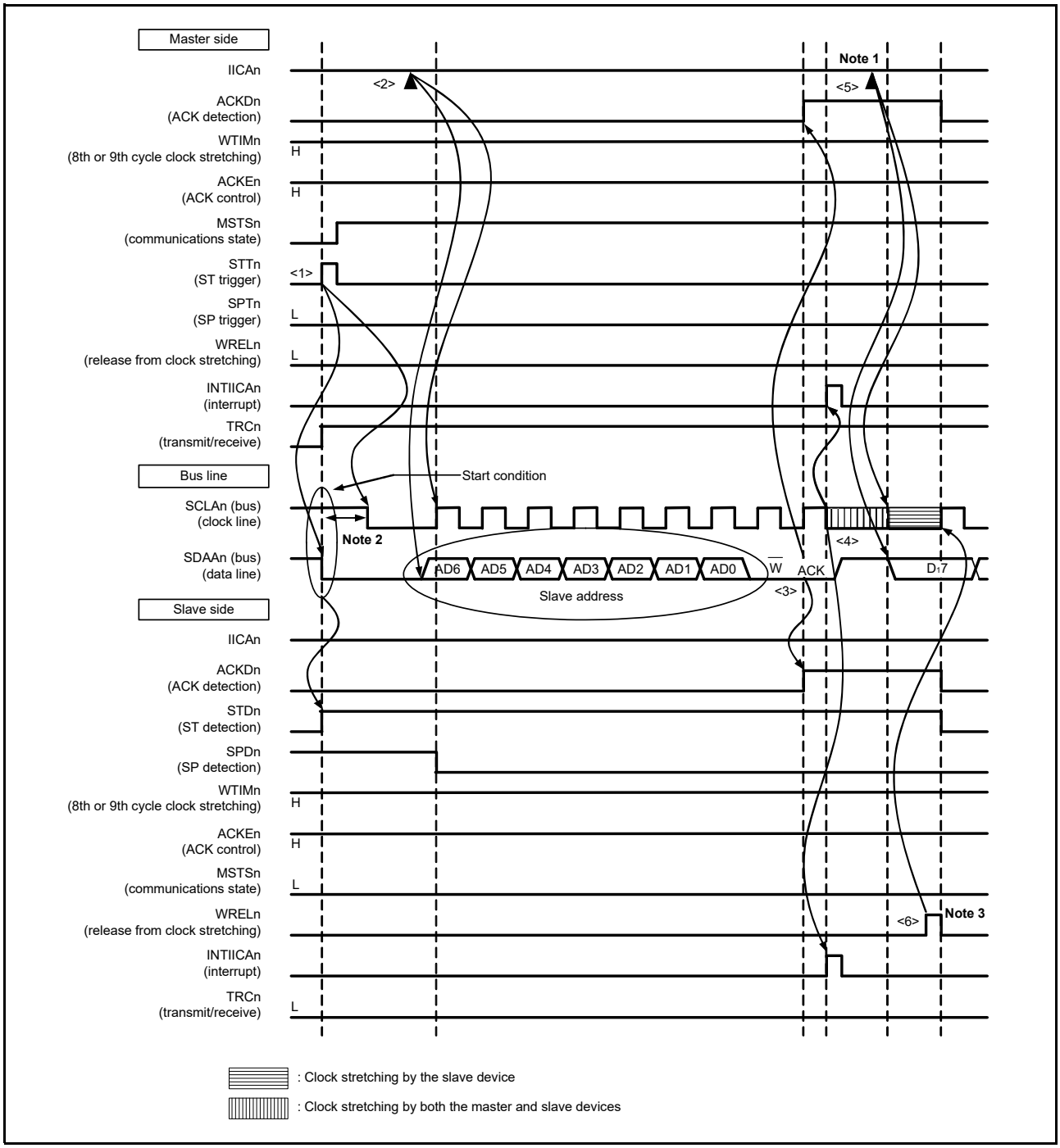
25.6 Timing Charts

When using the I^2C bus mode, the master device outputs an address via the serial bus to select one of several slave devices as its communication partner. After outputting the slave address, the master device transmits the TRCn flag (bit 3 of the IICA status register n (IICSn)), which specifies the data transfer direction, and then starts serial communication with the slave device. **Figures 25 - 33** and **25 - 34** show timing charts of the data communication. The IICA shift register n (IICAn)'s shift operation is synchronized with the falling edge of the serial clock (SCLAn). The transmit data is transferred to the SO latch and is output (MSB first) via the SDAAn pin. Data input via the SDAAn pin is captured into IICAn at the rising edge of SCLAn. In the timing charts described in this section, it is assumed that the all address match function is disabled.

Remark n = 0

Figure 25 - 33 Example of Master to Slave Communications (9th Cycle Clock Stretching Is Selected for Both the Master and Slave) (1/4)

1. Start condition to address to data



Note 1. For release from the clock stretch state during transmission by a master device, write data to the IICAn register instead of setting the WRELn bit.

Note 2. Make sure that the time between the fall of the SDAAn pin signal and the fall of the SCLAn pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

Note 3. For release from the clock stretch state during reception by a slave device, write FFH to IICAn or set the WRELn bit.

Remark n = 0

The meanings of <1> to <6> in **1. Start condition to address to data** in **Figure 25 - 33** are explained below.

<1> The start condition trigger is set by the master device (STTn = 1) and a start condition (i.e. SCLAn = 1 and SDAAn changes from 1 to 0) is generated once the bus data line goes low (SDAAn). When the start condition is subsequently detected, the master device enters the master device communications state (MSTSn = 1). The master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.

<2> The master device writes the address + W (transmission) to the IICA shift register n (IICAn) and transmits the slave address.

<3> In the slave device if the address received matches the address (SVAn value) of a slave device**Note**, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device with the address matching the transmitted slave address sets the clock stretch state (SCLAn = 0) and issues an interrupt (INTIICAn: address match)**Note**.

<5> The master device writes the data to transmit to the IICAn register and releases the clock stretch state set by the master device.

<6> If the slave device releases the clock stretch state (WRELn = 1), the master device starts transferring data to the slave device.

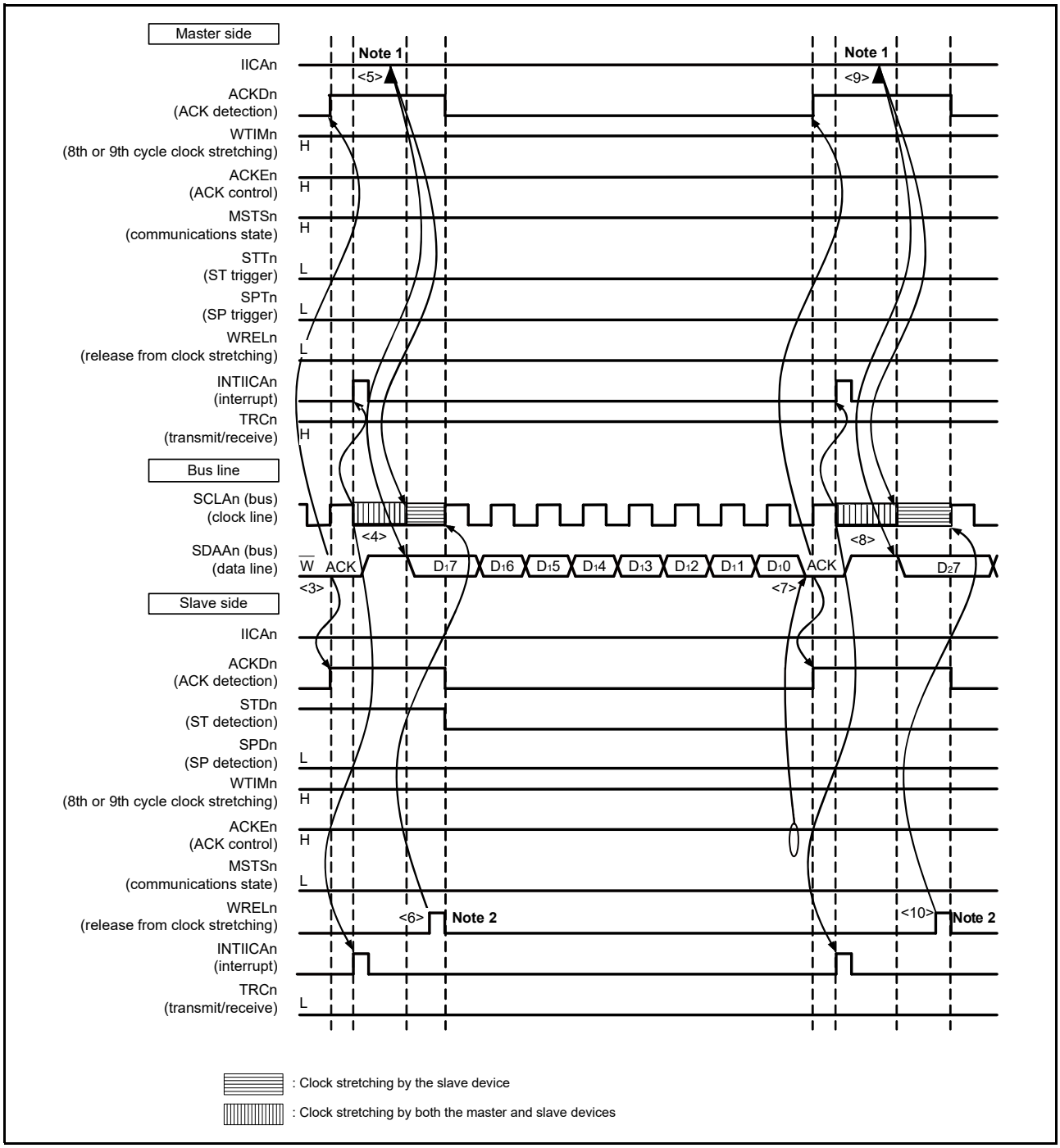
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set the clock stretch state. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <15> in **Figure 25 - 33** represent the entire procedure for communicating data using the I^2C bus.
1. Start condition to address to data in **Figure 25 - 33** shows the processing from <1> to <6>,
2. Address to data to data in **Figure 25 - 33** shows the processing from <3> to <10>, and
3. Data to data to stop condition in **Figure 25 - 33** shows the processing from <7> to <15>.

Remark 2. n = 0

Figure 25 - 33 Example of Master to Slave Communications (9th Cycle Clock Stretching Is Selected for Both the Master and Slave) (2/4)

2. Address to data to data



Note 1. For release from the clock stretch state during transmission by a master device, write data to the IICAn register instead of setting the WRELn bit.

Note 2. For release from the clock stretch state during reception by a slave device, write FFH to IICAn or set the WRELn bit.

Remark n = 0

The meanings of <3> to <10> in **2. Address to data to data** in **Figure 25 - 33** are explained below.

<3> In the slave device if the address received matches the address (SVAn value) of a slave device**Note**, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device with the address matching the transmitted slave address sets the clock stretch state (SCLAn = 0) and issues an interrupt (INTIICAn: address match)**Note**.

<5> The master device writes the data to transmit to the IICA shift register n (IICAn) and releases the clock stretch state set by the master device.

<6> If the slave device releases the clock stretch state (WRELn = 1), the master device starts transferring data to the slave device.

<7> After data transfer is completed, because of ACKEn = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<9> The master device writes the data to transmit to the IICAn register and releases the clock stretch state set by the master device.

<10> The slave device reads the received data and releases the clock stretch state (WRELn = 1). The master device then starts transferring data to the slave device.

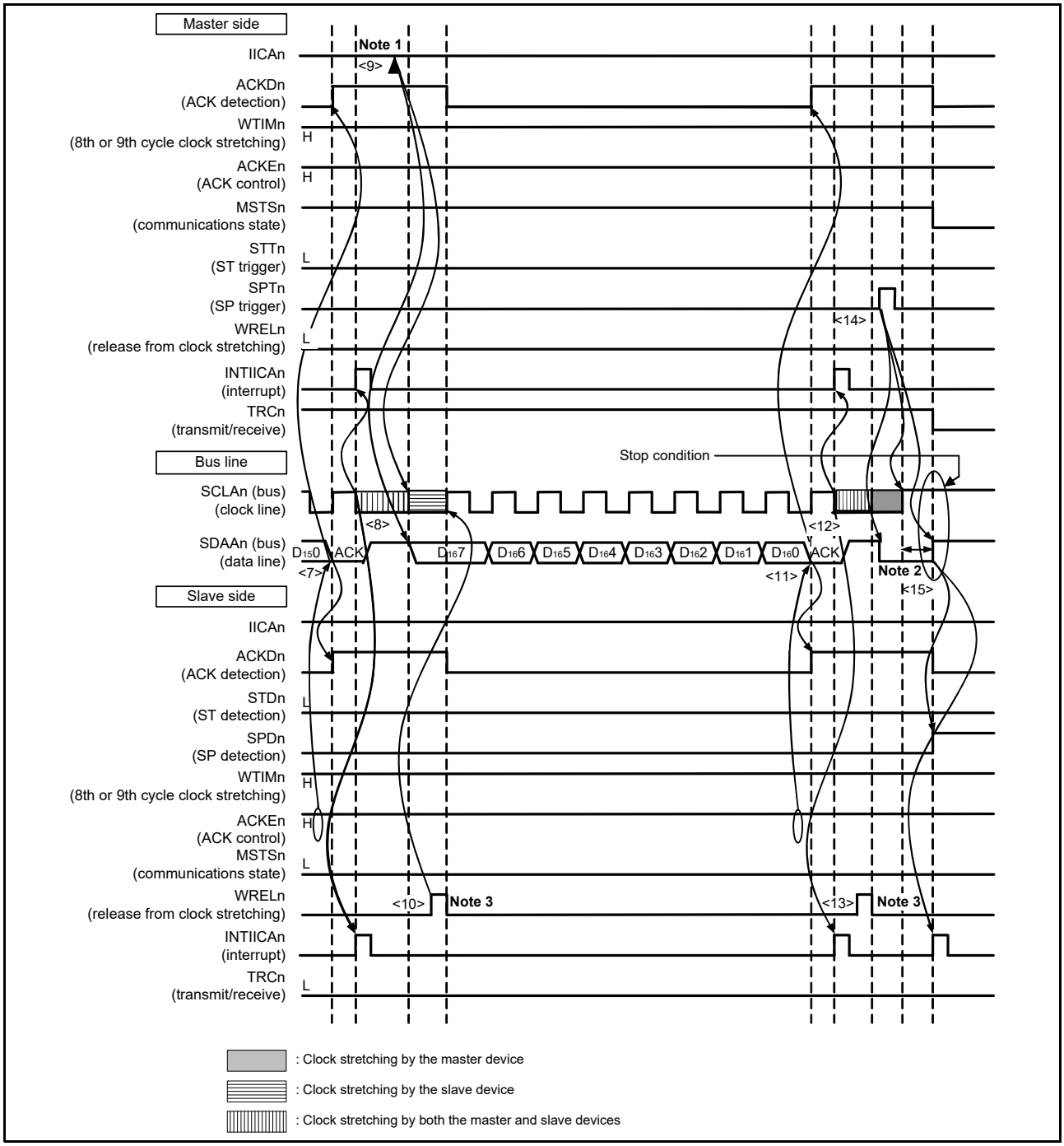
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set the clock stretch state. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <15> in **Figure 25 - 33** represent the entire procedure for communicating data using the I^2C bus.
1. Start condition to address to data in **Figure 25 - 33** shows the processing from <1> to <6>,
2. Address to data to data in **Figure 25 - 33** shows the processing from <3> to <10>, and
3. Data to data to stop condition in **Figure 25 - 33** shows the processing from <7> to <15>.

Remark 2. n = 0

Figure 25 - 33 Example of Master to Slave Communications (9th Cycle Clock Stretching Is Selected for Both the Master and Slave) (3/4)

3. Data to data to stop condition



Note 1. For release from the clock stretch state during transmission by a master device, write data to the IICAn register instead of setting the WRELn bit.

Note 2. Make sure that the time between the rise of the SCLAn pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

Note 3. For release from the clock stretch state during reception by a slave device, write FFH to IICAn or set the WRELn bit.

Remark n = 0

The meanings of <7> to <15> in **3. Data to data to stop condition** in **Figure 25 - 33** are explained below.

<7> After data transfer is completed, because of ACKEn = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<9> The master device writes the data to transmit to the IICA shift register n (IICAn) and releases the clock stretch state set by the master device.

<10> The slave device reads the received data and releases the clock stretch state (WRELn = 1). The master device then starts transferring data to the slave device.

<11> When data transfer is complete, the slave device (ACKEn =1) sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<12> The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<13> The slave device reads the received data and releases the clock stretch state (WRELn = 1).

<14> By the master device setting a stop condition trigger (SPTn = 1), the bus data line is cleared (SDAAn = 0) and the bus clock line is set (SCLAn = 1). After the stop condition setup time has elapsed, by setting the bus data line (SDAAn = 1), the stop condition is then generated (i.e. SCLAn =1 and SDAAn changes from 0 to 1).

<15> When a stop condition is generated, the slave device detects the stop condition and issues an interrupt (INTIICAn: stop condition).

Remark 1. <1> to <15> in **Figure 25 - 33** represent the entire procedure for communicating data using the I^2C bus.

1. Start condition to address to data in **Figure 25 - 33** shows the processing from <1> to <6>,

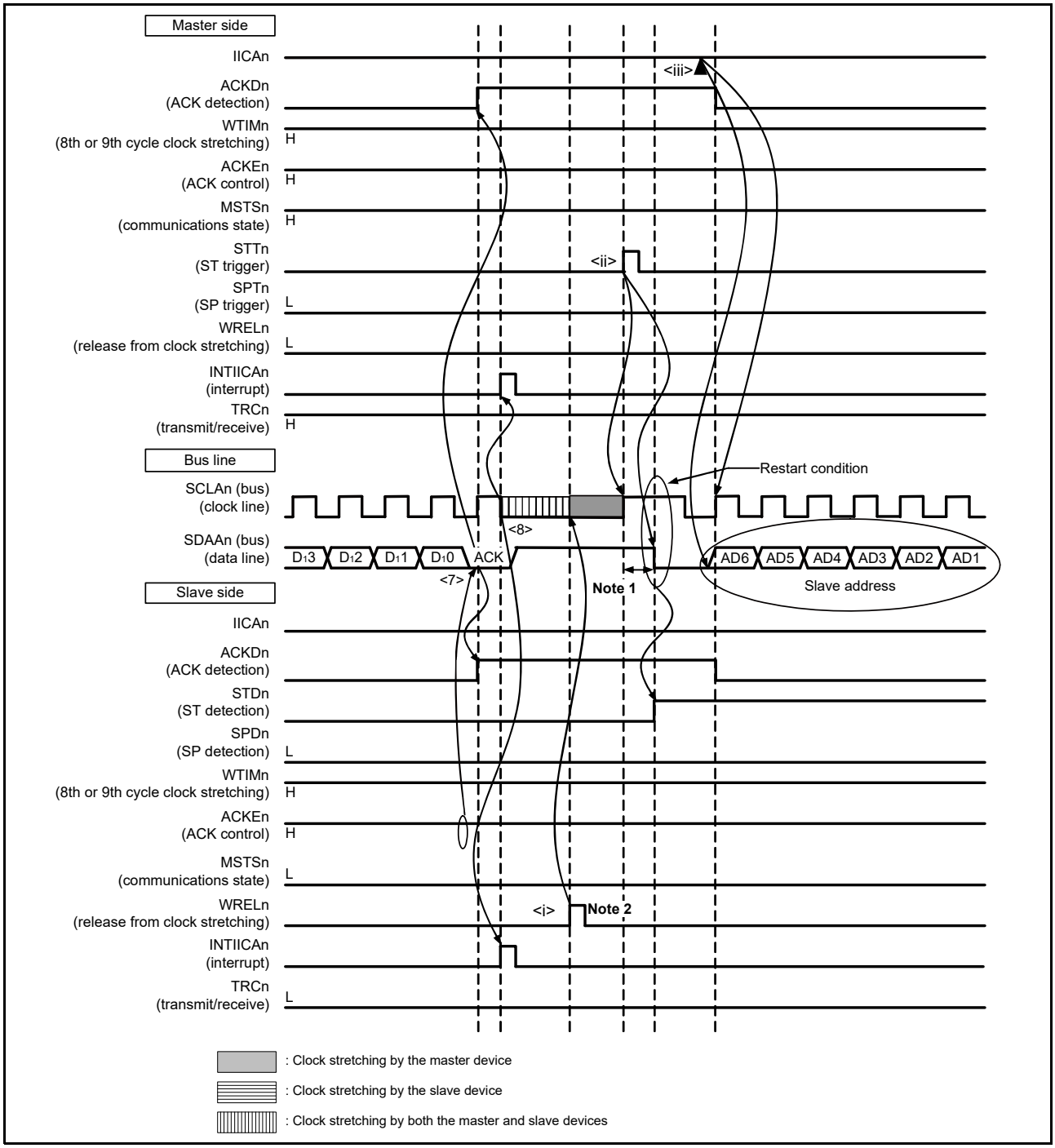
2. Address to data to data in **Figure 25 - 33** shows the processing from <3> to <10>, and

3. Data to data to stop condition in **Figure 25 - 33** shows the processing from <7> to <15>.

Remark 2. n = 0

Figure 25 - 33 Example of Master to Slave Communications (9th Cycle Clock Stretching Is Selected for Both the Master and Slave) (4/4)

4. Data to restart condition to address



Note 1. Make sure that the time between the rise of the SCLAn pin signal and the generation of the start condition after a restart condition has been issued is at least 4.7 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

Note 2. For release from the clock stretch state during reception by a slave device, write FFH to IICAn or set the WRELn bit.

Remark n = 0

The following describes the operations in **4. Data to restart condition to address** in **Figure 25 - 33**. After the operations in steps <7> and <8>, the operations in steps <i> to <iii> are performed. These steps return the processing to step <3>, the data transmission step.

<7> After data transfer is completed, because of ACKEn = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<i> The slave device reads the received data and releases the clock stretch state (WRELn = 1).

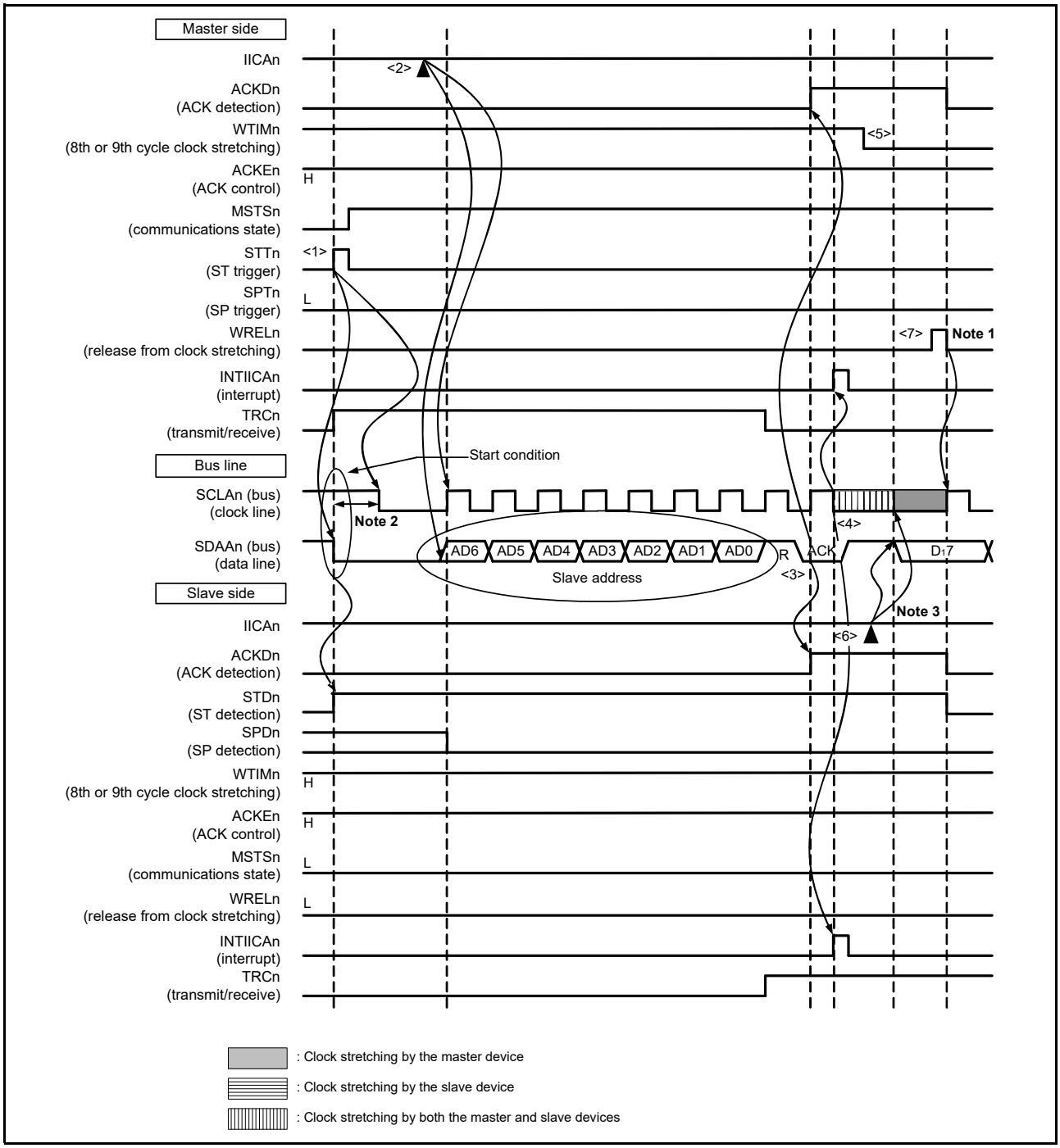
<ii> The start condition trigger is set again by the master device (STTn = 1) and a start condition (i.e. SCLAn =1 and SDAAn changes from 1 to 0) is generated once the bus clock line goes high (SCLAn = 1) and the bus data line goes low (SDAAn = 0) after the restart condition setup time has elapsed. When the start condition is subsequently detected, the master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.

<iii> The master device writing the address + R/W (transmission) to the IICA shift register (IICAn) enables the slave address to be transmitted.

Remark n = 0

Figure 25 - 34 Example of Slave to Master Communications (8th Cycle Clock Stretching Is Selected for the Master and 9th Cycle Clock Stretching Is Selected for the Slave) (1/3)

1. Start condition to address to data



Note 1. For release from the clock stretch state during reception by a master device, write FFH to IICAn or set the WRELn bit.

Note 2. Make sure that the time between the fall of the SDAAn pin signal and the fall of the SCLAn pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

Note 3. For release from the clock stretch state during transmission by a slave device, write data to the IICAn register instead of setting the WRELn bit.

Remark n = 0

The meanings of <1> to <7> in **1. Start condition to address to data** in **Figure 25 - 34** are explained below.

<1> The start condition trigger is set by the master device (STTn = 1) and a start condition (i.e. SCLAn = 1 and SDAAn changes from 1 to 0) is generated once the bus data line goes low (SDAAn). When the start condition is subsequently detected, the master device enters the master device communications state (MSTSn = 1). The master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.

<2> The master device writes the address + R (reception) to the IICA shift register n (IICAn) and transmits the slave address.

<3> In the slave device if the address received matches the address (SVAn value) of a slave device**Note**, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device with the address matching the transmitted slave address sets the clock stretch state (SCLAn = 0) and issues an interrupt (INTIICAn: address match)**Note**.

<5> The timing at which the master device sets the clock stretch state changes to the 8th clock (WTIMn = 0).

<6> The slave device writes the data to transmit to the IICAn register and releases the clock stretch state set by the slave device.

<7> The master device releases the clock stretch state (WRELn = 1) and starts transferring data from the slave device to the master device.

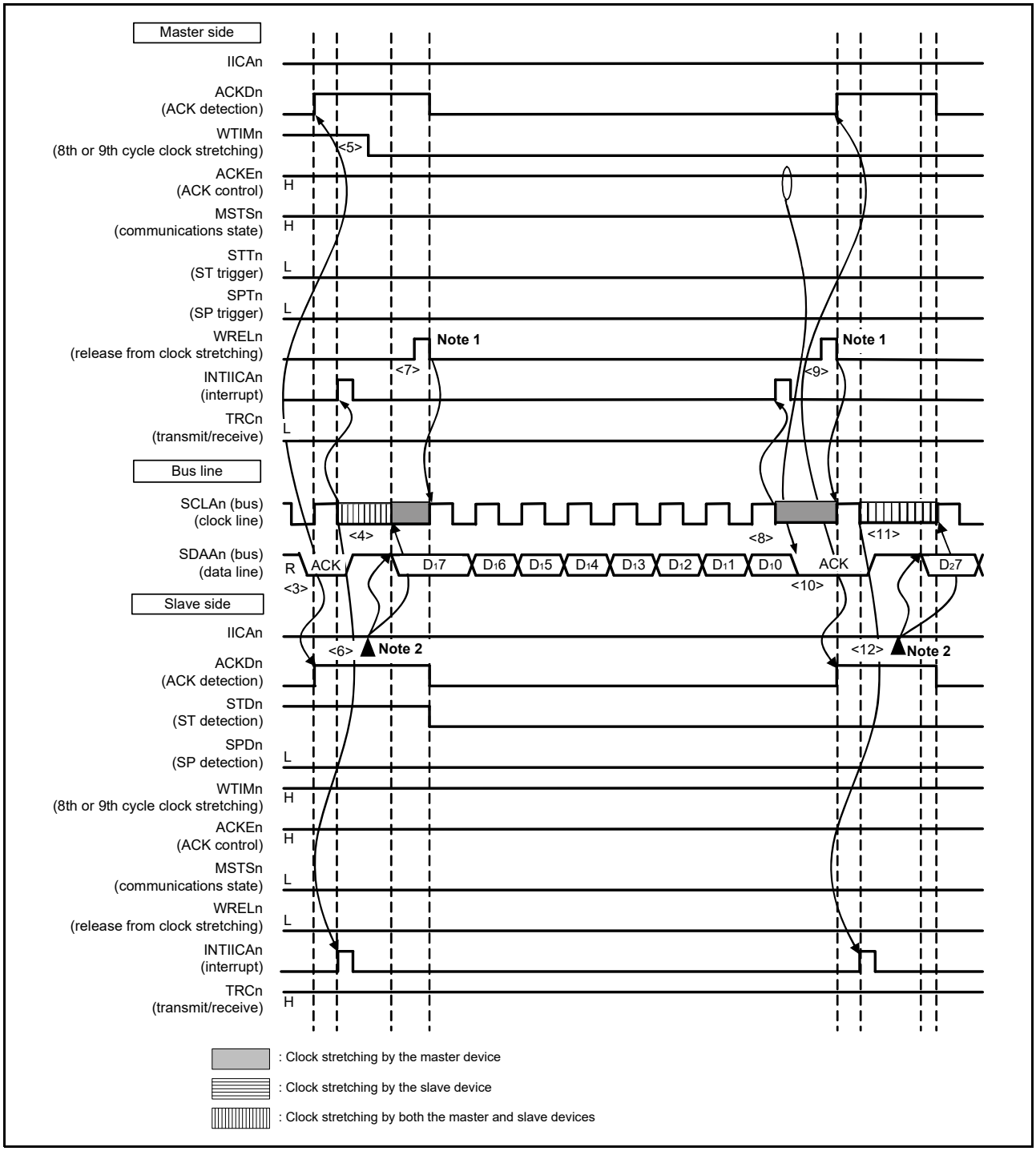
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set the clock stretch state. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <19> in **Figure 25 - 34** represent the entire procedure for communicating data using the I^2C bus.
1. Start condition to address to data in **Figure 25 - 34** shows the processing from <1> to <7>,
2. Address to data to data in **Figure 25 - 34** shows the processing from <3> to <12>, and
3. Data to data to stop condition in **Figure 25 - 34** shows the processing from <8> to <19>.

Remark 2. n = 0

Figure 25 - 34 Example of Slave to Master Communications (8th Cycle Clock Stretching Is Selected for the Master and 9th Cycle Clock Stretching Is Selected for the Slave) (2/3)

2. Address to data to data



Note 1. For release from the clock stretch state during reception by a master device, write FFH to IICAn or set the WRELn bit.

Note 2. For release from the clock stretch state during transmission by a slave device, write data to the IICAn register instead of setting the WRELn bit.

Remark n = 0

The meanings of <3> to <12> in **2. Address to data to data** in **Figure 25 - 34** are explained below.

<3> In the slave device if the address received matches the address (SVAn value) of a slave device**Note**, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device with the address matching the transmitted slave address sets the clock stretch state (SCLAn = 0) and issues an interrupt (INTIICAn: address match)**Note**.

<5> The master device changes the timing of clock stretching to the 8th clock (WTIMn = 0).

<6> The slave device writes the data to transmit to the IICA shift register n (IICAn) and releases the clock stretch state set by the slave device.

<7> The master device releases the clock stretch state (WRELn = 1) and starts transferring data from the slave device to the master device.

<8> The master device sets the clock stretch state (SCLAn = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICAn: end of transfer). Because of ACKEn = 1 in the master device, the master device then sends an ACK by hardware to the slave device.

<9> The master device reads the received data and releases the clock stretch state (WRELn = 1).

<10> The ACK is detected by the slave device (ACKDn = 1) at the rising edge of the 9th clock.

<11> The slave device sets the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICAn: end of transfer).

<12> By the slave device writing the data to transmit to the IICAn register, the clock stretch state set by the slave device is released. The slave device then starts transferring data to the master device.

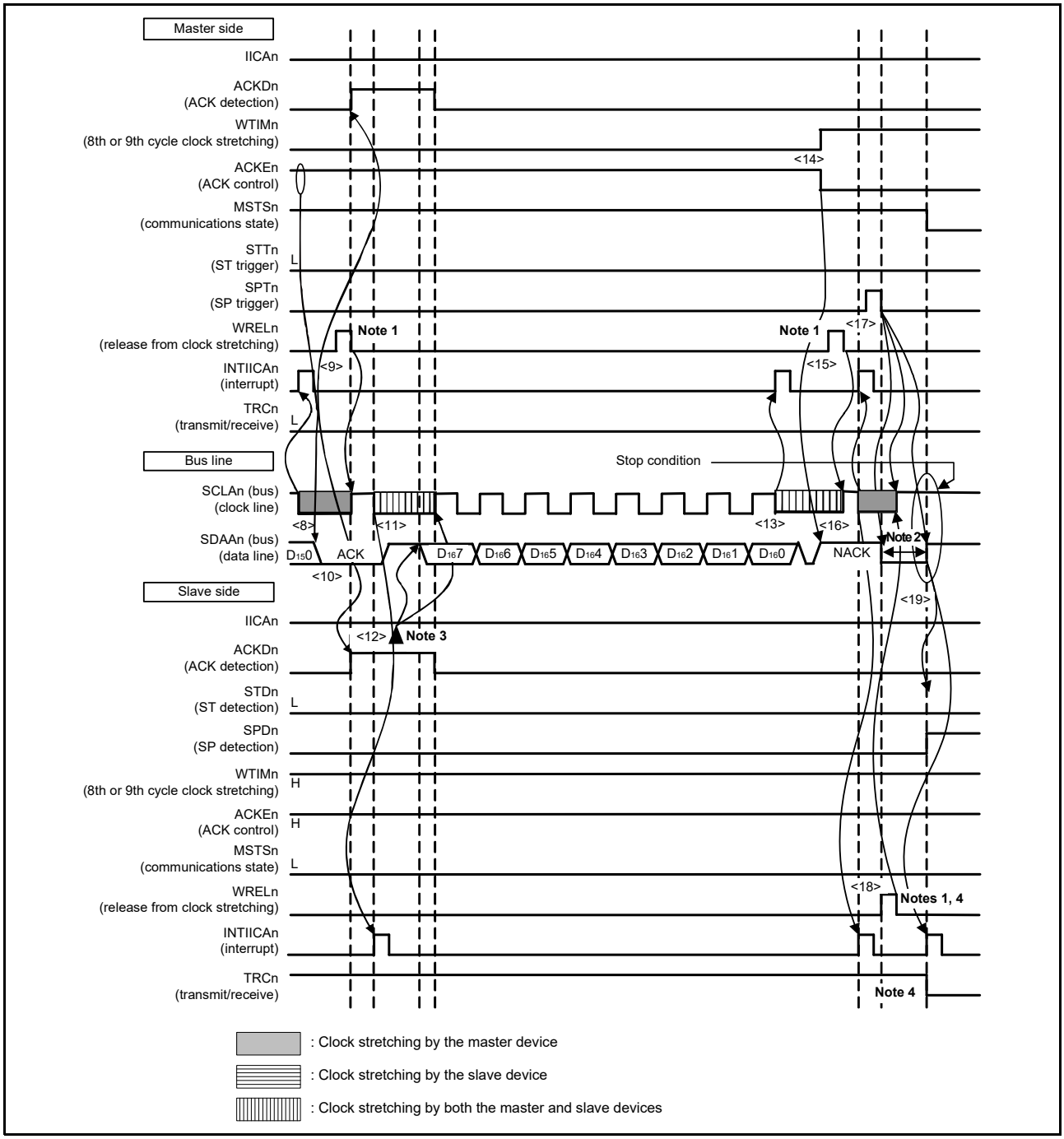
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set the clock stretch state. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <19> in **Figure 25 - 34** represent the entire procedure for communicating data using the I^2C bus.
1. Start condition to address to data in **Figure 25 - 34** shows the processing from <1> to <7>,
2. Address to data to data in **Figure 25 - 34** shows the processing from <3> to <12>, and
3. Data to data to stop condition in **Figure 25 - 34** shows the processing from <8> to <19>.

Remark 2. n = 0

Figure 25 - 34 Example of Slave to Master Communications (8th Cycle Clock Stretching Is Changed to 9th Cycle Clock Stretching for the Master and 9th Cycle Clock Stretching Is Selected for the Slave) (3/3)

3. Data to data to stop condition



Note 1. For release from the clock stretch state, write FFH to IICAn or set the WRELn bit.

Note 2. Make sure that the time between the rise of the SCLAn pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

Note 3. For release from the clock stretch state during transmission by a slave device, write data to the IICAn register instead of setting the WRELn bit.

Note 4. If the clock stretch state during transmission by a slave device is released by setting the WRELn bit, the TRCn flag will be cleared.

Remark n = 0

The meanings of <8> to <19> in **3. Data to data to stop condition** in **Figure 25 - 34** are explained below.

<8> The master device sets the clock stretch state (SCLAn = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICAn: end of transfer). Because of ACKEn = 0 in the master device, the master device then sends an ACK by hardware to the slave device.

<9> The master device reads the received data and releases the clock stretch state (WRELn = 1).

<10> The ACK is detected by the slave device (ACKDn = 1) at the rising edge of the 9th clock.

<11> The slave device sets the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICAn: end of transfer).

<12> By the slave device writing the data to transmit to the IICA register, the clock stretch state set by the slave device is released. The slave device then starts transferring data to the master device.

<13> The master device issues an interrupt (INTIICAn: end of transfer) at the falling edge of the 8th clock, and sets the clock stretch state (SCLAn = 0). Because ACK control (ACKEn = 1) is performed, the bus data line is at the low level (SDAAn = 0) at this stage.

<14> The master device sets NACK as the response (ACKEn = 0) and changes the timing at which it sets the clock stretch state to the 9th clock (WTIMn = 1).

<15> If the master device releases the clock stretch state (WRELn = 1), the slave device detects the NACK (ACK = 0) at the rising edge of the 9th clock.

<16> The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<17> When the master device issues a stop condition (SPTn = 1), the bus data line is cleared (SDAAn = 0) and the master device releases the clock stretch state. The master device then waits until the bus clock line is set (SCLAn = 1).

<18> The slave device acknowledges the NACK, halts transmission, and releases the clock stretch state (WRELn = 1) to end communication. Once the slave device releases the clock stretch state, the bus clock line is set (SCLAn = 1).

<19> Once the master device recognizes that the bus clock line is set (SCLAn = 1) and after the stop condition setup time has elapsed, the master device sets the bus data line (SDAAn = 1) and issues a stop condition (i.e. SCLAn =1 and SDAAn changes from 0 to 1). The slave device detects the generated stop condition and slave device issue an interrupt (INTIICAn: stop condition).

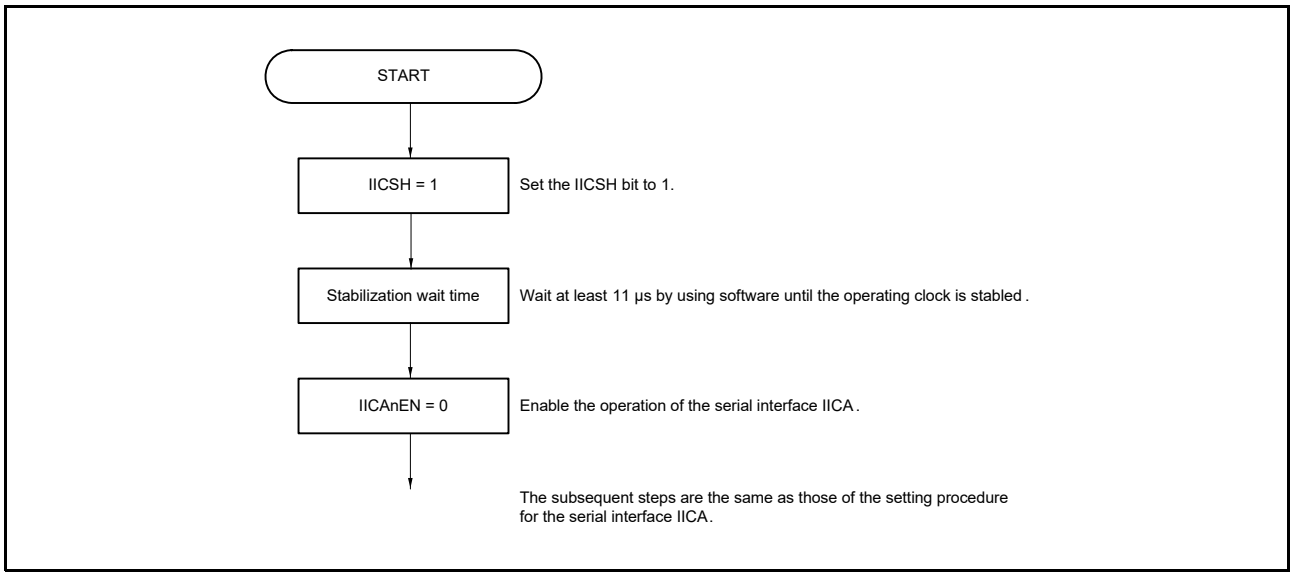
Remark 1. <1> to <19> in **Figure 25 - 34** represent the entire procedure for communicating data using the I^2C bus.
1. Start condition to address to data in **Figure 25 - 34** shows the processing from <1> to <7>,
2. Address to data to data in **Figure 25 - 34** shows the processing from <3> to <12>, and
3. Data to data to stop condition in **Figure 25 - 34** shows the processing from <8> to <19>.

Remark 2. n = 0

25.7 SMBus usage procedure

Figure 25 - 35 shows the procedure to select the SMBus 3.0 input threshold as the input threshold for the IICA interface.

Figure 25 - 35 SMBus Usage Procedure



Section 26 Digital Addressable Lighting Interface (DALI)

26.1 Overview

The digital addressable lighting interface (DALI) module complies with DALI, an international open standard lighting control communication protocol, that allows communications connecting devices such as electronic ballasts, LED lighting power supplies, switches and sensors, from different manufacturers. This module meets the international standard IEC 62386-101 Edition 1.0/2.0/2.1 (DALI), that includes software control.

Table 26 - 1 lists the specifications of the DALI module.

Table 26 - 1 Specifications of the DALI Module

Parameter	Description
Designed for compliance with	IEC 62386-101 Edition1.0/2.0/2.1 (DALI)[Note]
Communication mode	• Control devices (input devices and application controllers)/control gears • Transmission, reception, transmission/reception
Communication format	DALI data format
Serial data	• MSB-first • Receive data length: Minimum 1 bit, maximum 256 bits • Transmit data length: Selectable from 8, 16, 17, 20, 24, 32, 64, 128, and 256 bits
Interrupt output	• Stop condition detection interrupt (INTSDD) • DALI error interrupt (INTED) • Collision detection interrupt (INTCLD) • Bus power down detection interrupt (INTBPD) • Falling-edge detection interrupt (INTFED) • Read request interrupt for receive data (INTRD) • Write request interrupt for transmit data (INTTD)
Error detection	• Manchester framing error • Overrun error • Frame size violation error • Bit timing violation error
Adjustment/correction	• Width correction for DALITxD0 output waveform • Adjustment of the edge allowable area for the DALIRxD0 input signal, width correction for DALIRxD0 input waveform

Note Larger-number parts of the standard, such as IEC 62386-102 and IEC 62386-103, can be achieved by using a software stack each.

Figure 26 - 1 shows the block diagram of the DALI module.

Figure 26 - 1 Block Diagram of the DALI Module

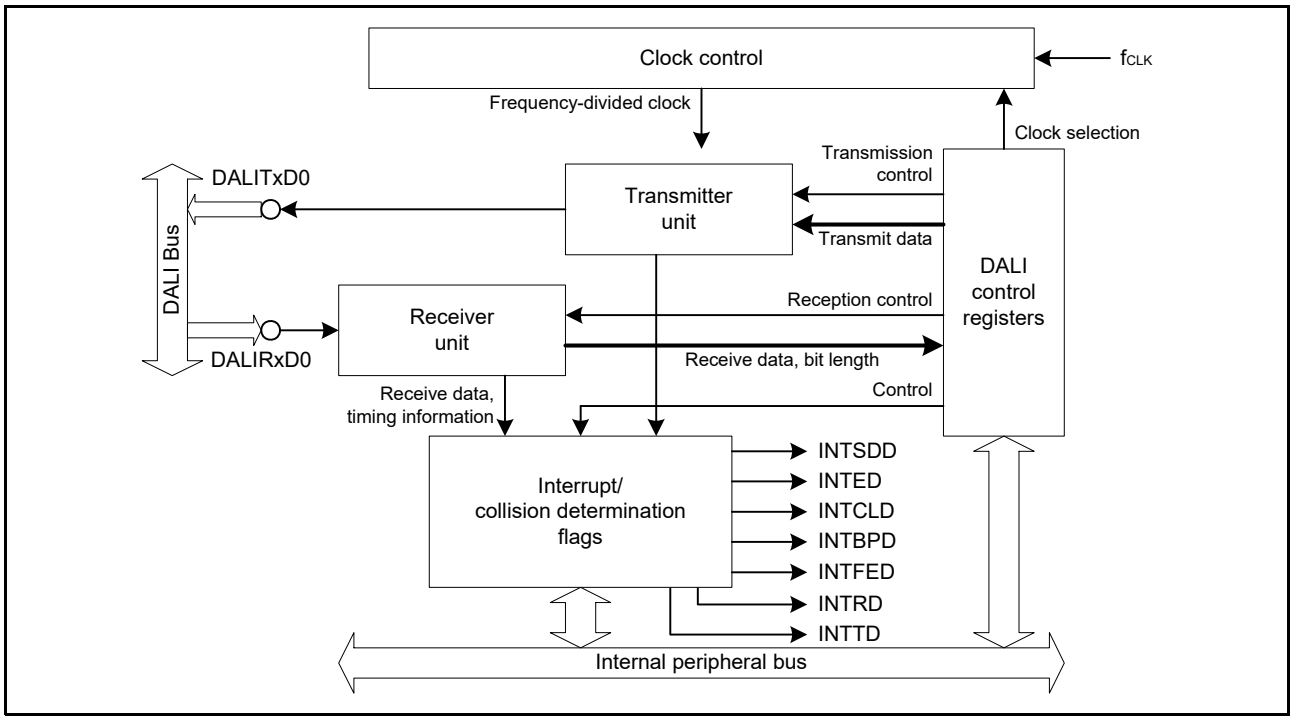


Table 26 - 2 lists the pin configuration of the DALI module.

Table 26 - 2 Pin Configuration of the DALI Module

Pin Name	I/O	Function
DALIRxD0	Input	DALI reception pin
DALITxD0	Output	DALI transmission pin

26.2 Registers for Controlling the Digital Addressable Lighting Interface

The following registers are used to control the digital addressable lighting interface.

- Peripheral enable register 1 (PER1)
- Peripheral reset control register 1 (PRR1)
- DALI configuration register 1 (CNFR1)
- DALI configuration register 2 (CNFR2)
- DALI control register 1 (CTR1)
- DALITxD0 control register 1 (TXDCTR1)
- DALI transmit control register 1 (TRSTR1)
- DALI bit timing violation threshold register 1 (BTVTHR1)
- DALI bit timing violation threshold register 2 (BTVTHR2)
- DALI bit timing violation threshold register 3 (BTVTHR3)
- DALI bit timing violation threshold register 4 (BTVTHR4)
- DALI collision threshold register 1 (COLTHR1)
- DALI collision threshold register 2 (COLTHR2)
- DALI collision threshold register 3 (COLTHR3)
- DALI collision threshold register 4 (COLTHR4)
- DALI collision threshold register 5 (COLTHR5)
- DALI transmit data registers 1H, 1L (TDR1H, TDR1L)
- DALI reception data registers 1H, 1L (RDR1H, RDR1L)
- DALI status register 1 (STR1)
- DALI status register 2 (STR2)
- DALI collision register 1 (COLR1)
- DALI flag error clearing register 1 (FECR1)
- DALI software reset register 1 (SWRR1)
- DALITxD0 waveform adjustment register 1 (TXWR1)
- DALIRxD0 waveform adjustment register 1 (RXWR1)
- DALI reception timing adjustment register 0 (FTDC0)
- Port mode registers xx (PMxx) (xx = 0, 5)
- Port registers xx (Pxx) (xx = 0, 5)
- Port input mode registers xx (PIMxx) (xx = 0, 5)
- Port output mode registers xx (POMxx) (xx = 0, 5)
- Port mode control A registers xx (PMCAxx) (xx = 0)

26.2.1 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. To use the DALI module, be sure to set bit 0 (DALIEN) of this register to 1. The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 26 - 2 Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	<3>	2	1	<0>
PER1	DACEN	0	PGACMPEN	TML32EN	DTCEN	0	0	DALIEN

DALIEN	Control of supply of an input clock to the DALI module
0	Stops supply of an input clock. • The 2nd SFRs used by the DALI module cannot be written. • When a 2nd SFR used by the DALI module is read, the value returned is 00H or 0000H.
1	Enables supply of an input clock. • The 2nd SFRs used by the DALI module can be read and written.

Caution 1. When setting the DALI module, make sure that the setting of the DALIEN bit is 1 first.
Caution 2. Be sure to clear bits 6, 2, and 1 to 0.

26.2.2 Peripheral reset control register 1 (PRR1)

The PRR1 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. To place the DALI module in the reset state, set bit 0 (DALIRES) of this register to 1. The PRR1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 26 - 3 Format of Peripheral Reset Control Register 1 (PRR1)

Address: F00FBH
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	3	2	1	<0>
PRR1	DACRES	0	PGACMPRES	TML32RES	0	0	0	DALIRES

DALIRES	Control resetting of the DALI module
0	The DALI module is released from the reset state.
1	The DALI module is in the reset state.

Caution Be sure to clear bits 6 and 3 to 1 to 0.

26.2.3 DALI configuration register 1 (CNFR1)

This register is used to set the basic operation of the DALI module. The following settings can be made.

- Clock
- Bit rate
- Extended mode
- Transmission data length

This register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 00FFH.

Figure 26 - 4 Format of DALI Configuration Register 1 (CNFR1) (1/2)

Address: F04D2H
After reset: 00FFH
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
CNFR1	CHL[3:0]				EXM	0	CKS[1:0]	

7	6	5	4	3	2	1	0
BR[7:0]							

CHL[3:0]				Character length
0	0	0	0	8 bits
0	0	0	1	16 bits
0	0	1	0	24 bits
0	0	1	1	32 bits
0	1	0	0	20 bits
0	1	0	1	17 bits
0	1	1	×	Setting prohibited
1	0	0	0	64 bits
1	0	0	1	128 bits
1	0	1	0	256 bits
1	0	1	1	Setting prohibited
1	1	×	×	Setting prohibited

The CHL[3:0] bits set the transmission data length to a value from among 8, 16, 17, 20, 24, 32, 64, 128, and 256 bits.

EXM	Extended mode selection
0	Non-extended mode
1	Extended mode, that is, transmit and receive mode for data greater than 32 bits

This bit selects whether data are to be transmitted and received in units greater than 32 bits. For transmission and reception of data in units no greater than 32 bits, set this bit to 0 to select the non-extended mode. To handle data in units greater than 32 bits, set this bit to 1 to select the extended mode. In the non-extended mode, the INTRD and INTTD signals are invalid (fixed to the low level), and the settings of the CHL[3], CTR1.TDIE, CTR1.RDIE, STR1.URF, and FECR1.URFC bits are also invalid. The meanings indicated by the STR1.LFRF flag and the STR2.RDBL[8:6] bits depend on the setting of this bit, and the communications operation of the DALI module changes accordingly. For details, see **26.3.8 Control by setting the extended mode selection bit (CNFR1.EXM)**.

Figure 26 - 4 Format of DALI Configuration Register 1 (CNFR1) (2/2)

CKS[1:0]		Clock selection
0	0	f_{CLK} clock (Y = 0)
0	1	f_{CLK}/4 clock (Y = 1)
1	0	f_{CLK}/16 clock (Y = 2)
1	1	Setting prohibited
The CKS[1:0] bits select the f_{CLK} division ratio to produce the operating clock for the DALI module as 1/1, 1/4, or 1/16.		

BR[7:0]	Bit rate setting
—	The BR[7:0] bits are readable. The following formula shows the relation between these bits and the bit rate B, where the setting of these bits is N. $N = \frac{f_{CLK} \times 10^6}{512 \times 2^{2Y-1} \times B} - 1$ $Error\ (\%) = \left\{ \frac{f_{CLK} \times 10^6}{B \times 512 \times 2^{2Y-1} \times (N+1)} - 1 \right\} \times 100$ B: Bit rate in bps N: Setting of bit rate generator: $0 \le N \le 255$ Y: Value selected by CKS[1:0] bits Specify Y and N as follows. Select the clock signal by setting the CKS[1:0] bits so that the setting N for the bit rate generator is within the given range, with B of 1200 bps, the bit rate for DALI communications. The following table shows an example of settings for that bit rate.

bps	Operating Frequency of f_{CLK} (MHz)											
	8			16			32			48		
	Y	N	Error (%)	Y	N	Error (%)	Y	N	Error (%)	Y	N	Error (%)
1200	0	25	0.16	1	12	0.16	1	25	0.16	1	38	0.16

The DALI module operates in 256 clock cycles per one-bit period at the given bit rate, and its internal operation is per those cycles as the minimum unit. When Y and N are specified for the target bit rate of 1200 bps, the period of one cycle, excluding the error, is 3.25 µs. The actual cycle time can be calculated from the following equation.

$$\text{Cycle time of the operating clock for the DALI module (µs)} = \frac{(N+1) \times 2^{2Y}}{f_{CLK}}$$

f_{CLK}: Input frequency for the DALI module (MHz)

The cycle time affects the timing of sampling of the input and output signals. For details, see **26.3.5 Sampling timing of the DALIRxD0 input signal and bit length adjustment**.

Caution 1. Be sure to clear bit 10 to 0.

Caution 2. These bits must only be modified while at least one of the following conditions applies.

- **The settings of the CTR1.RE and CTR1.TE bits are both 0.**
- **The settings of the CTR1.RE bit and the STR1.BBF flag are 1 and 0, respectively.**
- **The settings of the CTR1.TE bit and the STR1.BBF flag are 1 and 0, respectively.**

Caution 3. Only the high-speed on-chip oscillator clock (f_{IH}), high-speed system clock (f_{MX}), or PLL clock (f_{PLL}) is selectable as the operating clock for the DALI module.

26.2.4 DALI configuration register 2 (CNFR2)

This register is used to set the basic operation of the DALI module. The main settings are for operations in relation to errors and collisions. This register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 26 - 5 Format of DALI Configuration Register 2 (CNFR2) (1/2)

Address: F04D4H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
CNFR2	0	0	0	0	0	0	BTV1DIS	MFEDIS

7	6	5	4	3	2	1	0
0	0	CDM0	CDE	TXWE	SGA	BTVM	BTVE

BTV1DIS	BTV1 (bit timing violation 1) area control
0	Edges in the grey area from an edge to BTV threshold value 1 are detected as bit timing violations.
1	Edges in the grey area from an edge to BTV threshold value 1 are not detected as bit timing violations.

MFEDIS	Manchester framing error (MFE) detection control
0	Manchester framing errors (MFEs) are detected.
1	Manchester framing errors (MFEs) are not detected.

CDM0	Collision detection condition
0	Destroy areas are detected as collisions.
1	Destroy areas and avoidance areas are detected as collisions.
The CDM0 bit sets the condition for collision detection. For details on collisions, see **26.3.4 Collisions**.	

CDE**Note 1**	Collision detection enable
0	Collision detection is disabled.
1	Collision detection is enabled.
The CDE bit enables or disables collision detection. Set this bit to 0 for backward frame transmission**Note 2**.	

TXWE	Width adjustment control for the DALITxD0 output waveform
0	Width adjustment for the DALITxD0 output waveform is disabled.
1	Width adjustment for the DALITxD0 output waveform is enabled.
The TXWE bit enables or disables width adjustment for the DALITxD0 output waveform. For details, see **26.3.6 Width adjustment for DALITxD0 output waveform**.	

Figure 26 - 5 Format of DALI Configuration Register 2 (CNFR2) (2/2)

SGA	Frequency margin control
0	The frequency margin of the DALIRxD0 input signal is not extended.
1	The frequency margin of the DALIRxD0 input signal is extended.
The SGA bit selects whether to extend the frequency margin of the DALIRxD0 input signal. For details, see **26.3.5 Sampling timing of the DALIRxD0 input signal and bit length adjustment**.	

BTVM	Bit timing violation detection mode
0	Edges generated in grey areas between a half bit area and a 2 half bit area are not detected as bit timing violations.
1	Edges generated in grey areas between a half bit area and a 2 half bit area are detected as bit timing violations.
The BTVM bit sets the violation area when the detection of bit timing violations is enabled. The violation area set by this bit is the area between point G (half bit) and point H (2 half bit) shown in **Figure 26 - 35**. This bit is valid when the setting of the BTVE bit is 1 and must only be modified while the setting of the BTVE bit is 0.	

BTVE**Note 1**	Bit timing violation detection enable
0	Detection of bit timing violations is disabled.
1	Detection of bit timing violations is enabled.
The BTVE bit enables or disables the detection of bit timing violations. The bit is effective when the setting of the CTR1.RE bit is 1.	

Note 1. Bit timing violations and collisions are violations that are newly defined in IEC 62386-101 Edition 2.0/2.1. If the specification is to be compliant with IEC 62386-101, -102 Edition 1.0 so does not require detection of these violations, set bits BTVE and CDE to 0.

Note 2. If the CDE bit is set to 1 during backward frame transmission, immediately after writing the TRSTR1 register, a collision of the backward frame might be detected. Set the CDE bit to 1 after backward frame transmission has been completed.

Caution 1. Be sure to clear bits 15 to 10, 7, and 6 to 0.

Caution 2. These bits must only be modified while at least one of the following conditions applies.

- **The settings of the CTR1.RE and CTR1.TE bits are both 0.**
- **The settings of the CTR1.RE bit and the STR1.BBF flag are 1 and 0, respectively.**
- **The settings of the CTR1.TE bit and the STR1.BBF flag are 1 and 0, respectively.**

26.2.5 DALI control register 1 (CTR1)

This register is used to enable or disable transmission, reception, and interrupt requests of the DALI module. This register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 26 - 6 Format of DALI Control Register 1 (CTR1) (1/2)

Address: F04E6H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
CTR1	0	TDIE	RDIE	FEIE	BPIE	CLIE	DEIE	SDIE

7	6	5	4	3	2	1	0
0	0	0	0	0	0	RE	TE

TDIE	INTTD output enable
0	INTTD output is disabled.
1	INTTD output is enabled.
The TDIE bit enables or disables the write request interrupt for data to be transmitted (INTTD). This bit is only valid when the setting of the CNFR1.EXM bit is 1.	

RDIE	INTRD output enable
0	INTRD output is disabled.
1	INTRD output is enabled.
The RDIE bit enables or disables the read request interrupt for received data (INTRD). This bit is only valid when the setting of the CNFR1.EXM bit is 1.	

FEIE	INTFED output enable
0	INTFED output is disabled.
1	INTFED output is enabled.
The FEIE bit enables or disables the falling edge detection interrupt (INTFED).	

BPIE	INTBPD output enable
0	INTBPD output is disabled.
1	INTBPD output is enabled.
The BPIE bit enables or disables the bus power down detection interrupt (INTBPD).	

CLIE	INTCLD output enable
0	INTCLD output is disabled.
1	INTCLD output is enabled.
The CLIE bit enables or disables the collision detection interrupt (INTCLD). This bit is only valid when the setting of the CNFR2.CDE bit is 1.	

Figure 26 - 6 Format of DALI Control Register 1 (CTR1) (2/2)

DEIE	INTED output enable
0	INTED output is disabled.
1	INTED output is enabled.
The DEIE bit enables or disables the DALI error interrupt (INTED).	

SDIE	INTSDD output enable
0	INTSDD output is disabled.
1	INTSDD output is enabled.
The SDIE bit enables or disables the stop bit detection interrupt (INTSDD). This bit is invalid when the settings of the TE and RE bits are both 0.	

RE	Receive enable
0	Reception is disabled.
1	Reception is enabled**Note 1**.
The RE bit enables or disables reception.	

TE	Transmit enable
0	Transmission is disabled**Note 2**.
1	Transmission is enabled**Note 1**.
The TE bit enables or disables transmission.	

Note 1. If the RE bit is changed from 0 to 1 while another DALI device is communicating, the DALI module attempts to receive data of that communications. To eliminate the risk of this occurring, check the STR1.BBF flag to confirm the DALI bus is in the idle state (STR1.BBF flag is 0) before setting RE bit to 1.

Note 2. If the TE bit is changed from 1 to 0 during transmission, the DALI module immediately stops the transmission.

Caution Be sure to clear bits 15 and 7 to 2 to 0.

26.2.6 DALITxD0 control register 1 (TXDCTR1)

This register is used to select the output settings for the transmission pin, DALITxD0, of the DALI module. This register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 26 - 7 Format of DALITxD0 Control Register 1 (TXDCTR1)

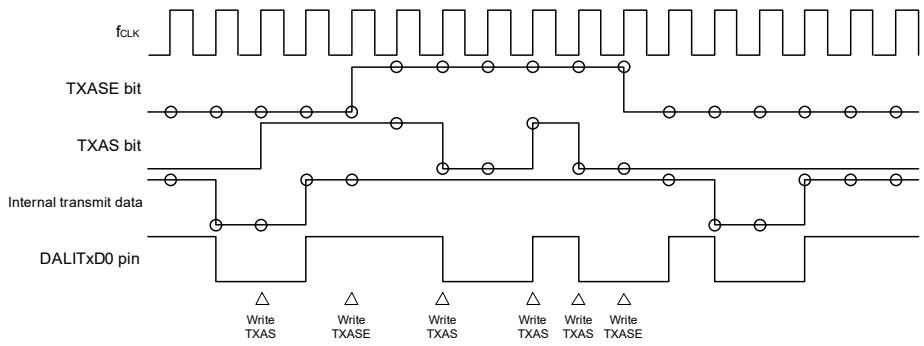
Address: F04E8H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TXDCTR1	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TXASE	TXAS

TXASE	DALITxD0 assertion enable
0	Internal data for transmission are output from the DALITxD0 pin.
1	The level specified by the TXAS bit is output from the DALITxD0 pin.

The TXASE bit selects whether to use the level specified by the TXAS bit as the level output by the DALITxD0 pin. The following figure shows the operation for control of the level output from the DALITxD0 pin.



TXAS	DALITxD0 assertion level
0	The output from the DALITxD0 pin is driven low.
1	The output from the DALITxD0 pin is driven high.

The TXAS bit selects the level output from the DALITxD0 pin as low or high. This bit is valid when the setting of the TXASE bit is 1.

Caution Be sure to clear bits 15 to 2 to 0.

26.2.7 DALI transmit control register 1 (TRSTR1)

This register is a trigger register that is used to start transmission by the DALI module. This register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 26 - 8 Format of DALI Transmit Control Register 1 (TRSTR1)

Address: F04E2H
After reset: 0000H
R/W: W

Symbol	15	14	13	12	11	10	9	8
TRSTR1	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	TRST

TRST	Transmission start trigger

The operation of this bit depends on the setting of the CNFR1.EXM bit.

[In the non-extended mode (CNFR1.EXM = 0)]

Writing 1 to this bit starts transmission. During transmission, the TRST bit cannot be set for a new transmission**Note**. This bit is only valid when the setting of the CTR1.TE bit is 1. The data length for transmission is specified by the CNFR1.CHL[2:0] bits. After writing 1 to this bit, data are transmitted with the MSB first.

[In the extended mode (CNFR1.EXM = 1)]

Writing 1 to this bit triggers the output of the write request interrupt for transmit data (INTTD) signal. The signal is output the number of times corresponding to the data length specified by the CNFR1.CHL[3:0] bits as indicated in the table below.

Setting of the CNFR1.CHL[3:0] Bits	Transmission Data Length	Number of Times the INTTD Signal Is Output
0000	8 bits	Once
0001	16 bits	Once
0010	24 bits	Once
0011	32 bits	Once
0100	20 bits	Once
0101	17 bits	Once
1000	64 bits	Twice
1001	128 bits	Four times
1010	256 bits	Eight times

Activate the DTC or CPU with the INTTD interrupt signal, then write the data for transmission in 16-bit units, starting with the higher-order bits to the TDR1H register, then the lower-order bits to the TDR1L register. After 1 is written to this bit, the first writing to the TDR1L register triggers the MSB-first transmission of the data with the length specified by the CNFR1.CHL[3:0] bits. During transmission, the TRST bit cannot be set for a new transmission**Note**. This bit is only valid when the setting of the CTR1.TE bit is 1.

Note The DALI module does not have a status flag to indicate the transmission state. Therefore, start by confirming that the STR1.BBF flag is 0, and then set the TRST bit to 1. After that, the STR1.BBF flag is set to 1, which indicates that transmission of the frame has started. When the transmission is completed, INTSDD is output and the STR1.BBF flag changes from 1 to 0.

Caution Be sure to clear bits 15 to 1 to 0.

26.2.8 DALI bit timing violation threshold register 1 (BTVTHR1)

This register is used to set ranges judged by the DALI module to be bit timing violations (BTVs). For details of the settings, see **26.3.3.4 Bit timing violation error**. This register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 4F00H.

Figure 26 - 9 Format of DALI Bit Timing Violation Threshold Register 1 (BTVTHR1)

Address: F04C0H
After reset: 4F00H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
BTVTHR1	BTV2[7:0]							

7	6	5	4	3	2	1	0
0	BTV1[6:0]						

BTV2[7:0]	Bit timing violation threshold 2
—	Specifies bit timing violation threshold value 2.
The BTV2[7:0] bits specify bit timing violation (BTV) threshold value 2. BTV threshold value 2 = (BTV2[7:0] + 1) × (two cycles of the operating clock for the DALI module, 6.5 µs typ.) Example: When the value set in the BTV2[7:0] bits is 4DH, BTV threshold value 2 = 507 µs. These bits can be set to values for times from 6.5 to 1664 µs (see **26.3.3.4 Bit timing violation error**). The setting must satisfy IEC 62386-101 Edition 2.0/2.1, taking frequency accuracy into account. The grey area defined by these bits does not exist with the initial BTV threshold value.		

BTV1[6:0]	Bit timing violation threshold 1
—	Specifies bit timing violation threshold value 1.
The BTV1[6:0] bits specify bit timing violation (BTV) threshold value 1. BTV threshold value 1 = BTV1[6:0] × (two cycles of the operating clock for the DALI module, 6.5 µs typ.) Example: When the value set in the BTV1[6:0] bits is 33H, BTV threshold value 1 = 331.5 µs. These bits can be set to values for times from 0 to 825.5 µs (see **26.3.3.4 Bit timing violation error**). The setting must satisfy IEC 62386-101 Edition 2.0/2.1, taking frequency accuracy into account. The grey area defined by these bits does not exist with the initial BTV threshold value. Whether to detect edges in the grey area from an edge to BTV threshold value 1 can be selected by setting the CNFR2.BTV1DIS bit.	

Caution 1. Be sure to clear bit 7 to 0.
Caution 2. These bits can only be modified when the settings of the CTR1.RE and CTR1.TE bits are both 0.

26.2.9 DALI bit timing violation threshold register 2 (BTVTHR2)

This register is used to set ranges judged by the DALI module to be bit timing violations (BTVs). For details of the settings, see **26.3.3.4 Bit timing violation error**. This register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 654FH.

Figure 26 - 10 Format of DALI Bit Timing Violation Threshold Register 2 (BTVTHR2)

Address: F04C2H
After reset: 654FH
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
BTVTHR2	BTV4[7:0]							

7	6	5	4	3	2	1	0
BTV3[7:0]							

BTV4[7:0]	Bit timing violation threshold 4
—	Specifies bit timing violation threshold value 4.
The BTV4[7:0] bits specify bit timing violation (BTV) threshold value 4. BTV threshold value 4 = BTV4[7:0] × (two cycles of the operating clock for the DALI module, 6.5 µs typ.) Example: When the value set in the BTV4[7:0] bits is 66H, BTV threshold value 4 = 663 µs. These bits can be set to values for times from 0 to 1657.5 µs (see **26.3.3.4 Bit timing violation error**). The setting must satisfy IEC 62386-101 Edition 2.0/2.1, taking frequency accuracy into account.		

BTV3[7:0]	Bit timing violation threshold 3
—	Specifies bit timing violation threshold value 3.
The BTV3[7:0] bits specify bit timing violation (BTV) threshold value 3. BTV threshold value 3 = (BTV3[7:0] + 1) × (two cycles of the operating clock for the DALI module, 6.5 µs typ.) Example: When the value set in the BTV3[7:0] bits is 4DH, BTV threshold value 3 = 507 µs. These bits can be set to values for times from 6.5 to 1664 µs (see **26.3.3.4 Bit timing violation error**). The setting must satisfy IEC 62386-101 Edition 2.0/2.1, taking frequency accuracy into account.	

Caution These bits can only be modified when the settings of the CTR1.RE and CTR1.TE bits are both 0.

26.2.10 DALI bit timing violation threshold register 3 (BTVTHR3)

This register is used to set a range judged by the DALI module to be a bit timing violation (BTV). For details of the settings, see **26.3.3.4 Bit timing violation error**. This register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 009DH.

Figure 26 - 11 Format of DALI Bit Timing Violation Threshold Register 3 (BTVTHR3)

Address: F04C4H
After reset: 009DH
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
BTVTHR3	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
BTV5[7:0]							

BTV5[7:0]	Bit timing violation threshold 5
—	Specifies bit timing violation threshold value 5.

The BTV5[7:0] bits specify bit timing violation (BTV) threshold value 5.
BTV threshold value 5 = (BTV5[7:0] + 1) × (two cycles of the operating clock for the DALI module, 6.5 µs typ.)
Example:
When the value set in the BTV5[7:0] bits is 9AH, BTV threshold value 5 = 1007.5 µs.
These bits can be set to values for times from 6.5 to 1664 µs (see **26.3.3.4 Bit timing violation error**). The setting must satisfy IEC 62386-101 Edition 2.0/2.1, taking frequency accuracy into account. The grey area defined by these bits does not exist with the initial BTV threshold value.

Caution 1. Be sure to clear bits 15 to 8 to 0.
Caution 2. These bits can only be modified when the settings of the CTR1.RE and CTR1.TE bits are both 0.

26.2.11 DALI bit timing violation threshold register 4 (BTVTHR4)

This register is used to set a range judged by the DALI module to be a bit timing violation (BTV) and as the time to represent a stop condition for this module. This register must therefore be set whenever DALI communications is to proceed regardless of whether bit timing violation detection is enabled or disabled. For details of the settings, see **26.3.3.4 Bit timing violation error** and **26.3.2 STOP condition detection and settling time**. This register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 00DBH.

Figure 26 - 12 Format of DALI Bit Timing Violation Threshold Register 4 (BTVTHR4)

Address: F04C6H
After reset: 00DBH
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
BTVTHR4	0	0	0	0	0	0	0	BTV6[8]

7	6	5	4	3	2	1	0
BTV6[7:0]							

BTV6[8:0]	Bit timing violation threshold 6
—	Specifies bit timing violation threshold value 6.
The BTV6[8:0] bits specify bit timing violation (BTV) threshold value 6. BTV threshold value 6 = BTV6[8:0] × (two cycles of the operating clock for the DALI module, 6.5 µs typ.) BTV threshold value 6 is also used in defining the stop condition. Example: When the value set in the BTV6[8:0] bits is 0D8H, BTV threshold value 6 = 1404 µs. These bits can be set to values for times from 1248 to 3321.5 µs. Do not set values for times less than 1248 µs, that is, equivalent to less than 1.5-bit widths. See **26.3.3.4 Bit timing violation error**. The setting must satisfy IEC 62386-101 Edition 2.0/2.1, taking frequency accuracy into account. The grey area defined by these bits with the initial BTV threshold value is treated as part of the stop condition waveform.	

Caution 1. Be sure to clear bits 15 to 9 to 0.
Caution 2. These bits can only be modified when the settings of the CTR1.RE and CTR1.TE bits are both 0.

26.2.12 DALI collision threshold register 1 (COLTHR1)

This register is used to set ranges judged by the DALI module to represent collisions. For details of the settings, see **26.3.4 Collisions**. This register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 380FH.

Figure 26 - 13 Format of DALI Collision Threshold Register 1 (COLTHR1)

Address: F04C8H
After reset: 380FH
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
COLTHR1	0	0	COL2[5:0]					

7	6	5	4	3	2	1	0
0	0	COL1[5:0]					

COL2[5:0]	Collision threshold 2
—	Specifies collision threshold value 2.
The COL2[5:0] bits specify collision (COL) threshold value 2. COL threshold value 2 = (COL2[5:0] + 1) × (two cycles of the operating clock for the DALI module, 6.5 µs typ.) Example: When the setting of the COL2[5:0] bits is 37H, COL threshold value 2 = 364 µs. These bits can be set to values for times from 6.5 to 416 µs (see **26.3.4 Collisions**). The setting must satisfy IEC 62386-101 Edition 2.0/2.1, taking frequency accuracy into account.	

COL1[5:0]	Collision threshold 1
—	Specifies collision threshold value 1.
The COL1[5:0] bits specify collision (COL) threshold value 1. COL threshold value 1 = COL1[5:0] × (two cycles of the operating clock for the DALI module, 6.5 µs typ.) Example: When the setting of the COL1[5:0] bits is 0FH, COL threshold value 1 = 97.5 µs. These bits can be set to values for times from 0 to 409.5 µs (see **26.3.4 Collisions**). The setting must satisfy IEC 62386-101 Edition 2.0/2.1, taking frequency accuracy into account.	

Caution 1. Be sure to clear bits 15, 14, 7, and 6 to 0.
Caution 2. These bits can only be modified when the settings of the CTR1.RE and CTR1.TE bits are both 0.

26.2.13 DALI collision threshold register 2 (COLTHR2)

This register is used to set ranges judged by the DALI module to represent collisions. For details of the settings, see **26.3.4 Collisions**. This register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 443CH.

Figure 26 - 14 Format of DALI Collision Threshold Register 2 (COLTHR2)

Address: F04CAH
After reset: 443CH
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
COLTHR2	0	COL4[6:0]						

7	6	5	4	3	2	1	0
0	COL3[6:0]						

COL4[6:0]	Collision threshold 4
—	Specifies collision threshold value 4.
The COL4[6:0] bits specify collision (COL) threshold value 4. COL threshold value 4 = (COL4[6:0] + 1) × (two cycles of the operating clock for the DALI module, 6.5 µs typ.) Example: When the setting of the COL4[6:0] bits is 43H, COL threshold value 4 = 442 µs. These bits can be set to values for times from 6.5 to 832 µs (see **26.3.4 Collisions**). The setting must satisfy IEC 62386-101 Edition 2.0/2.1, taking frequency accuracy into account.	

COL3[6:0]	Collision threshold 3
—	Specifies collision threshold value 3.
The COL3[6:0] bits specify collision (COL) threshold value 3. COL threshold value 3 = COL3[6:0] × (two cycles of the operating clock for the DALI module, 6.5 µs typ.) Example: When the setting of the COL3[6:0] bits is 3DH, COL threshold value 3 = 396.5 µs. These bits can be set to values for times from 0 to 825.5 µs (see **26.3.4 Collisions**). The setting must satisfy IEC 62386-101 Edition 2.0/2.1, taking frequency accuracy into account.	

Caution 1. Be sure to clear bits 15 and 7 to 0.
Caution 2. These bits can only be modified when the settings of the CTR1.RE and CTR1.TE bits are both 0.

26.2.14 DALI collision threshold register 3 (COLTHR3)

This register is used to set ranges judged by the DALI module to represent collisions. For details of the settings, see **26.3.4 Collisions**. This register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 7148H.

Figure 26 - 15 Format of DALI Collision Threshold Register 3 (COLTHR3)

Address: F04CCH
After reset: 7148H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
COLTHR3	0	COL6[6:0]						

7	6	5	4	3	2	1	0
0	COL5[6:0]						

COL6[6:0]	Collision threshold 6
—	Specifies collision threshold value 6.
The COL6[6:0] bits specify collision (COL) threshold value 6. COL threshold value 6 = (COL6[6:0] + 1) × (two cycles of the operating clock for the DALI module, 6.5 µs typ.) Example: When the setting of the COL6[6:0] bits is 70H, COL threshold value 6 = 734.5 µs. These bits can be set to values for times from 6.5 to 832 µs (see **26.3.4 Collisions**). The setting must satisfy IEC 62386-101 Edition 2.0/2.1, taking frequency accuracy into account.		

COL5[6:0]	Collision threshold 5
—	Specifies collision threshold value 5.
The COL5[6:0] bits specify collision (COL) threshold value 5. COL threshold value 5 = COL5[6:0] × (two cycles of the operating clock for the DALI module, 6.5 µs typ.) Example: When the setting of the COL5[6:0] bits is 49H, COL threshold value 5 = 474.5 µs. These bits can be set to values for times from 0 to 825.5 µs (see **26.3.4 Collisions**). The setting must satisfy IEC 62386-101 Edition 2.0/2.1, taking frequency accuracy into account.	

Caution 1. Be sure to clear bits 15 and 7 to 0.
Caution 2. These bits can only be modified when the settings of the CTR1.RE and CTR1.TE bits are both 0.

26.2.15 DALI collision threshold register 4 (COLTHR4)

This register is used to set ranges judged by the DALI module to represent collisions. For details of the settings, see **26.3.4 Collisions**. This register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 8879H.

Figure 26 - 16 Format of DALI Collision Threshold Register 4 (COLTHR4)

Address: F04CEH
After reset: 8879H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
COLTHR4	COL8[7:0]							

7	6	5	4	3	2	1	0
COL7[7:0]							

COL8[7:0]	Collision threshold 8
—	Specifies collision threshold value 8.
The COL8[7:0] bits specify collision (COL) threshold value 8. COL threshold value 8 = (COL8[7:0] + 1) × (two cycles of the operating clock for the DALI module, 6.5 µs typ.) Example: When the setting of the COL8[7:0] bits is 86H, COL threshold value 8 = 877.5 µs. These bits can be set to values for times from 6.5 to 1664 µs (see **26.3.4 Collisions**). The setting must satisfy IEC 62386-101 Edition 2.0/2.1, taking frequency accuracy into account.		

COL7[7:0]	Collision threshold 7
—	Specifies collision threshold value 7.
The COL7[7:0] bits specify collision (COL) threshold value 7. COL threshold value 7 = COL7[7:0] × (two cycles of the operating clock for the DALI module, 6.5 µs typ.) Example: When the setting of the COL7[7:0] bits is 7DH, COL threshold value 7 = 812.5 µs. These bits can be set to values for times from 0 to 1657.5 µs (see **26.3.4 Collisions**). The setting must satisfy IEC 62386-101 Edition 2.0/2.1, taking frequency accuracy into account.	

Caution These bits can only be modified when the settings of the CTR1.RE and CTR1.TE bits are both 0.

26.2.16 DALI collision threshold register 5 (COLTHR5)

This register is used to set a range judged by the DALI module to represent a collision. For details of the settings, see **26.3.4 Collisions**. This register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 008EH.

Figure 26 - 17 Format of DALI Collision Threshold Register 5 (COLTHR5)

Address: F04D0H
After reset: 008EH
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
COLTHR5	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
COL9[7:0]							

COL9[7:0]	Collision threshold 9
—	Specifies collision threshold value 9.
The COL9[7:0] bits specify collision (COL) threshold value 9. COL threshold value 9 = COL9[7:0] × (two cycles of the operating clock for the DALI module, 6.5 µs typ.) Example: When the setting of the COL9[7:0] bits is 91H, COL threshold value 9 = 942.5 µs. These bits can be set to values for times from 0 to 1657.5 µs (see **26.3.4 Collisions**). The setting must satisfy IEC 62386-101 Edition 2.0/2.1, taking frequency accuracy into account.	

Caution 1. Be sure to clear bits 15 to 8 to 0.
Caution 2. These bits can only be modified when the settings of the CTR1.RE and CTR1.TE bits are both 0.

26.2.17 DALI transmit data registers 1H, 1L (TDR1H, TDR1L)

These registers are used to set data for transmission from the DALI module. These registers can be set by a 16-bit memory manipulation instruction. The value of each register following a reset is 0000H.

Figure 26 - 18 Format of DALI Transmit Data Registers 1H, 1L (TDR1H, TDR1L) (1/2)

Address: F04DEH
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDR1H	DTDR[31:16]															

Bits 15 to 0	Transmit data [31:16]
DTDR [31:16]	16 higher-order bits of DALI data for transmission

Address: F04E0H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDR1L	DTDR[15:0]															

Bits 15 to 0	Transmit data [15:0]
DTDR [15:0]	16 lower-order bits of DALI data for transmission

Figure 26 - 18 Format of DALI Transmit Data Registers 1H, 1L (TDR1H, TDR1L) (2/2)

- In the non-extended mode (CNFR1.EXM = 0)
  The maximum data length the DALI module is able to transmit is 32 bits (DTDR[31:0]). The TDR1H register holds the 16 higher-order bits (DTDR[31:16]) and the TDR1L register holds the 16 lower-order bits (DTDR[15:0]) of data for transmission. When a transmission is triggered, that is, when 1 is written to the TRSTR1.TRST bit, data with the length specified by the CNFR1.CHL[2:0] bits are transmitted with the MSB first. Values written to bits not specified as part of the length is ignored. After input of the transmission trigger, do not write new values to these registers until output of the INTSDD signal (indicating the completion of transmission) or transmission is stopped due to output of the INTCLD signal.
  Examples:
  - When the CNFR1.CHL[3:0] bits are 0000B DTDR[7:0] are transmitted.
  - When the CNFR1.CHL[3:0] bits are 0010B DTDR[23:16] are transmitted, then DTDR[15:0] are transmitted.
- In the extended mode (CNFR1.EXM = 1)
  The maximum data length of the unit for transmission by the DALI module is extended to be up to 256 bits. Note that data for which the specified data length for transmission is no longer than 32 bits are treated the same as in the non-extended mode. Data with a length for transmission exceeding 32 bits are handled in 32-bit units. The TDR1H register holds the 16 higher-order bits (DTDR[31:16]) and the TDR1L register holds the 16 lower-order bits (DTDR[15:0]) of each 32-bit unit. When a transmission is triggered, that is, when 1 is written to the TRSTR1.TRST bit, the INTTD signal is output. In response to the INTTD signal, make sure to use the DTC or CPU to write new values, starting with the 16 higher-order bits to the TDR1H register, then the 16 lower-order bits to the TDR1L register. The INTTD signal is output the required number of times corresponding to the data length for transmission specified by the CNFR1.CHL[3:0] bits.

26.2.18 DALI reception data registers 1H, 1L (RDR1H, RDR1L)

These registers hold data received by the DALI module. These registers can be read by a 16-bit memory manipulation instruction. The value of each register following a reset is 0000H.

Figure 26 - 19 Format of DALI Reception Data Registers 1H, 1L (RDR1H, RDR1L)

Address: F04EEH
After reset: 0000H
R/W: R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDR1H	DRDR[31:16]															

Bits 15 to 0	Receive data [31:16]
DRDR [31:16]	16 higher-order bits of received DALI data

Address: F04F0H
After reset: 0000H
R/W: R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDR1L	DRDR[15:0]															

Bits 15 to 0	Receive data [15:0]
DRDR [15:0]	16 lower-order bits of received DALI data

When the stop condition is detected and the CTR1.RE bit is 1, the received data frame is stored in these registers with right alignment. The registers hold values sampled with 3/4-bit timing as the logical value of the received data.
Examples:
- When 24 bits of data have been received, the data are stored in the RDR1H.DRDR[23:16] and RDR1L.DRDR[15:0] bits.
- When 8 bits of data have been received, the data are stored in the RDR1L.DRDR[7:0] bits.

When received data have been stored from the internal shift register, the bits the shift register previously held are cleared. Bits that are not to be received in the internal shift register, are read as 0. Determine the valid bits by checking the data length for reception in the STR2.RDBL[8:0] bits.

26.2.19 DALI status register 1 (STR1)

This register is used to check the state of the DALI module. This register can be read by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 26 - 20 Format of DALI Status Register 1 (STR1) (1/3)

Address: F04F2H
After reset: 0000H
R/W: R

Symbol	15	14	13	12	11	10	9	8
STR1	0	0	0	0	0	URF	DAF	CDF

7	6	5	4	3	2	1	0
LFRF	BPDF	BBF	TENDF	RDRF	BTVF	OVF	MFEF

URF	Underrun error flag
0	No underrun error has occurred, 1 was written to the FECR1.URFC bit, or the setting of the CNFR1.EXM bit is 0.
1	An underrun error has occurred.
The URF flag indicates that an underrun error was detected. This flag is only valid in the extended mode (CNFR1.EXM = 1).	

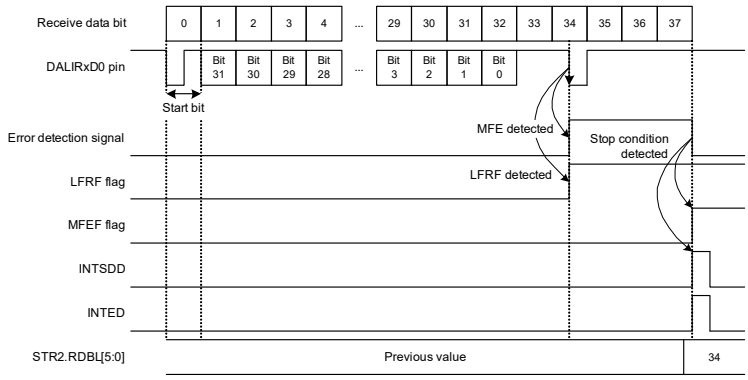
DAF	Destroy area flag
0	No collision occurred in the destroy area or 1 was written to the FECR1.DAFC bit.
1	A collision occurred in the destroy area.
When a collision is detected, the DAF flag indicates whether the collision occurred in the destroy area.	

CDF	Collision detection flag
0	No collision occurred or 1 was written to the FECR1.CDFC bit**Note 1**.
1	A collision occurred.
The CDF flag indicates whether a collision occurred.	

Figure 26 - 20 Format of DALI Status Register 1 (STR1) (2/3)

LFRF	Long frame reception flag
0	• In the non-extended mode (CNFR1.EXM = 0) 32 or fewer bits of data have been received, or 1 was written to the FECR1.LFRFC bit**Note 1**. • In the extended mode (CNFR1.EXM = 1) 256 or fewer bits of data have been received, or 1 was written to the FECR1.LFRFC bit**Notes 1, 2**.
1	• In the non-extended mode (CNFR1.EXM = 0) 33 or more bits of data have been received; that is, the reception of data bit [33] is confirmed. • In the extended mode (CNFR1.EXM = 1) 257 or fewer bits of data have been received; that is, the reception of data bit [257] is confirmed.

• In the non-extended mode (CNFR1.EXM = 0)
The LFRF flag is set when data bit [33] of the received data is recognized. If, however, data bit [33] is at the high level, and the high level is also detected for the following period of BTV threshold value 6, the DALI module judges the waveform to be part of a stop condition, and the LFRF flag is not set. If a falling edge is detected during the period of BTV threshold value 6, the DALI module judges the waveform to be an MFE, and the LFRF flag is set. For details, see the figure below.
• In the extended mode (CNFR1.EXM = 1)
The LFRF flag is set when data bit [257] of the received data is recognized. If, however, data bit [257] is at the high level, and the high level is also detected for the following period of BTV threshold value 6, the DALI module judges the waveform to be part of a stop condition, and the LFRF flag is not set. If a falling edge is detected during the period of BTV threshold value 6, the DALI module judges the waveform to be an MFE, and the LFRF flag is set. For details, see the figure below.



BPDF	Bus power down flag
0	This bit is set to 0 when any of the following conditions applies. • DALI bus power down was not detected. • 1 has been written to the FECR1.BPDFC bit**Note 1**. • The DALIRxD0 input signal has risen.
1	The DALIRxD0 input signal was at the low level for the period specified by the BTVTHR4.BTV6[8:0] bits.
The BPDF flag indicates that DALI bus power down was detected.	

Figure 26 - 20 Format of DALI Status Register 1 (STR1) (3/3)

BBF	Bus BUSY flag
0	The DALI bus is in the IDLE state.
1	The DALI bus is in the BUSY state.
The BBF flag indicates the state of usage of the DALI bus. When a collision occurs, the flag value is invalid. This flag is invalid when the settings of the CTR1.TE and CTR1.RE bits are both 0. [Clearing conditions] • Detection of a stop condition (the DALI bus is in the IDLE state) • Writing of 1 to the FECR1.BBFC bit**Note 1** • Incorrect detection of the falling edge on the DALI bus as the start bit. [Setting conditions] • Detection of a falling edge of the DALIRxD0 input signal. • The DALIRxD0 input signal is at the low level when the setting of the CTR1.TE or CTR1.RE bit is changed to 1 from 0.	

TENDF	Transmit end flag
0	The value 1 was written to the FECR1.TENDFC bit.
1	Frame transmission was completed.
The TENDF flag indicates that the DALI module has transmitted a frame.	

RDRF	Receive data register full flag
0	The RDR1L register was read**Note 3** or 1 was written to the FECR1.RDRFC**Note 1**.
1	Received data are stored in the RDR1L or RDR1H register.
The RDRF flag indicates that the RDR1L or RDR1H register contains unread received data.	

BTVF	Bit timing violation flag
0	No bit timing violation occurred or 1 was written to the FECR1.BTVFC bit**Note 1**.
1	A bit timing violation occurred.
The BTVF flag indicates that a bit timing violation was detected in received data.	

OVF	Overrun error flag
0	No overrun error has occurred or 1 was written to the FECR1.OVFC bit**Note 1**.
1	An overrun error has occurred.
The OVF flag indicates that an overrun error was detected.	

MFEF	Manchester framing error flag
0	No MFE has occurred or 1 was written to the FECR1.MFEFC bit**Note 1**.
1	An MFE has occurred.
While the setting of the CNFR2.MFEDIS bit is 0, the MFEF flag indicates that a Manchester framing error (MFE) was detected. While the setting of the CNFR2.MFEDIS bit is 1, the value of this flag is always 0.	

Note 1. If conditions for setting and clearing the flag are in contention, setting the flag takes priority.

Note 2. When the reception of a new frame starts while the value of the LFRF flag is 1, the INTRD signal is output simultaneously at the time of the start of reception. If this flag is set upon completion of reception, clear the flag before the reception of the next frame starts.

(Note and Caution are listed on the next page.)

Note 3. When the RDR1L register is read, the DALI module judges that reading of the received data has been completed. When the received data length is from 17 to 32 bits, the 16 higher-order bits (RDR1H register) must be read first. If the lower-order bits (RDR1L register) are read first, the RDRF flag will become 0 even though the RDR1H register has not been read.

Caution Be sure to clear bits 15 to 11 to 0.

26.2.20 DALI status register 2 (STR2)

This register is used to check the received data bit length, RDBL[8:0], of the DALI module. This register can be read by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 26 - 21 Format of DALI Status Register 2 (STR2)

Address: F04F4H
After reset: 0000H
R/W: R

Symbol	15	14	13	12	11	10	9	8
STR2	0	0	0	0	0	0	0	RDBL[8]

7	6	5	4	3	2	1	0
RDBL[7:0]							

RDBL[8:0]	Receive data bit length
—	Bit length of successfully received data

The RDBL[8:0] bits hold the bit length of successfully received data. When the value of the STR1.LFRF flag is 0, the RDBL[8:0] bits indicate the received data length without the start and stop bits. If reception is stopped because of an error or a collision, the RDBL[8:0] bits indicate the number of bits that were successfully received. This register is updated when a stop condition is detected. When reception is completed, check the size of the received frame by using this register. Note that when the value of the STR1.LFRF flag is 1, the value of this register is invalid. The maximum value of the RDBL[8:0] bits depends on the setting of the CNFR1.EXM bit. While the setting of the CNFR1.EXM bit is 0, the maximum value is 03FH. While the setting of the CNFR1.EXM bit is 1, the maximum value is 1FFH. In the non-extended mode (CNFR1.EXM = 0), the value of the RDBL[8:6] bits is fixed to 0.
Example:
The RDBL[8:0] bits of 000001000B indicates that 8 bits of data have been received.

26.2.21 DALI collision register 1 (COLR1)

This register is used to check the state of collisions occurring for the DALI module. This register can be read by a 16-bit memory manipulation instruction. The value of this register following a reset is 0800H.

Figure 26 - 22 Format of DALI Collision Register 1 (COLR1) (1/2)

Address: F04F6H
After reset: 0800H
R/W: R

Symbol	15	14	13	12	11	10	9	8
COLR1	0	0	TXDCV	RXDCEG	RXDMON	CLDAF	0	0

7	6	5	4	3	2	1	0
0	0	0	CDTF1	CDTF2[3:0]			

TXDCV	DALITxD0 collision status flag
0	Low
1	High
When a collision occurs and the value of the CLDAF flag is 0, the TXDCV bit holds the state (the level) of the DALITxD0 pin. If the value of the CLDAF flag is 1, this bit retains its previous value**Note 1**.	

RXDCEG	DALIRxD0 collision status flag
0	Falling edge
1	Rising edge
When a collision occurs and the value of the CLDAF flag is 0, the RXDCEG bit indicates the state (a rising or falling edge) of the DALIRxD0 pin that caused the collision. If the value of the CLDAF flag is 1, this bit retains its previous value**Note 1**.	

RXDMON	DALIRxD0 monitoring
The RXDMON bit monitors the DALIRxD0 input signal after to which width adjustment for the DALIRxD0 input waveform is applied. This bit can always be read regardless of the settings of the CTR1.TE and CTR1.RE bits. This bit holds the previous value during f_{CLK} stop.	

CLDAF	Collision factor flag
0	An edge on the DALIRxD0 pin occurred in an invalid area.
1	No edge on the DALIRxD0 pin occurred. A collision was detected in the last destroy area.
The CLDAF flag indicates the factor for collision occurrence. This flag being 0 indicates that an edge on the DALIRxD0 pin occurred with an invalid timing, that is, in neither a half bit nor a 2 half bit area. Check the CDTF1 and CDTF2[3:0] flags for the detailed timing of the collision. This flag being 1 indicates that the DALIRxD0 pin remained at the low level through and beyond the middle valid area with the label area 6 of **Figure 26 - 39** or that with the label area 10 of **Figure 26 - 40**.	

Figure 26 - 22 Format of DALI Collision Register 1 (COLR1) (2/2)

CDTF1	Collision detection timing flag 1
0	A collision was detected on the edge of a bit period boundary (**Figure 26 - 39 Collision Detection Timing 1**)Note 2.
1	A collision was detected as an edge in the middle of a bit period (**Figure 26 - 40 Collision Detection Timing 2**)Note 2.
The CDTF1 flag holds an indicator of the timing of collision detection.	

CDTF2[3:0]				Collision detection timing flag 2
0	0	0	0	After release from a reset
0	0	0	1	Collision detection timing 1
0	0	1	0	Collision detection timing 2
0	0	1	1	Collision detection timing 3
0	1	0	0	Collision detection timing 4
0	1	0	1	Collision detection timing 5
0	1	1	0	Collision detection timing 6
0	1	1	1	Collision detection timing 7Note 3
1	0	0	0	Collision detection timing 8Note 3
1	0	0	1	Collision detection timing 9Note 3
1	0	1	0	Collision detection timing 10Note 3
Other than above				Invalid
When the CLDAF flag is 0, the CDTF2[3:0] flags hold the timing of collision detection timing. When the CLDAF flag is 1, the CDTF2[3:0] flags retain their previous valueNote 1. The CDTF2[3:0] flags hold the timing (1 to 10) of collision detection as shown in **Figures 26 - 39** and **26 - 40**.				

Note 1. These bits and flags are not cleared by clearing the STR1.CDF flag. They are cleared by a reset or software reset.

Note 2. An edge on a bit period boundary is indicated by “*a” and an edge in the middle of a bit period is indicated by “*b” in **Figure 26 - 28**.

Note 3. When the CDTF1 flag is 0, this condition is not possible.

26.2.22 DALI flag error clearing register 1 (FECR1)

This register is used to clear error flags of the DALI module. This register can be written by a 16-bit memory manipulation instruction. When this register is read, 0 is always read. The value of this register following a reset is 0000H.

Figure 26 - 23 Format of DALI Flag Error Clearing Register 1 (FECR1) (1/2)

Address: F04FAH
After reset: 0000H
R/W: W

Symbol	15	14	13	12	11	10	9	8
FECR1	0	0	0	0	0	URFC	DAFC	CDFC

7	6	5	4	3	2	1	0
LFRFC	BPDFC	BBFC	TENDFC	RDRFC	BTVFC	OVFC	MFEFC

URFC	Underrun error flag clear
0	The STR1.URF flag is not cleared.
1	The STR1.URF flag is cleared.

DAFC	Destroy area flag clear
0	The STR1.DAF flag is not cleared.
1	The STR1.DAF flag is cleared.

CDFC	Collision detection flag clear
0	The STR1.CDF flag is not cleared.
1	The STR1.CDF flag is cleared.

LFRFC	Long frame reception flag clear
0	The STR1.LFRF flag is not cleared.
1	The STR1.LFRF flag is cleared.

BPDFC	Bus power down flag clear
0	The STR1.BPDF flag is not cleared.
1	The STR1.BPDF flag is cleared.

BBFC	Bus BUSY flag clear
0	The STR1.BBF flag is not cleared.
1	The STR1.BBF flag is cleared**Note**.

TENDFC	Transmit end flag clear
0	The STR1.TENDF flag is not cleared.
1	The STR1.TENDF flag is cleared.

Figure 26 - 23 Format of DALI Flag Error Clearing Register 1 (FECR1) (2/2)

RDRFC	Receive data register full flag clear
0	The STR1.RDRF flag is not cleared.
1	The STR1.RDRF flag is cleared.

BTVFC	Bit timing violation flag clear
0	The STR1.BTVF flag is not cleared.
1	The STR1.BTVF flag is cleared.

OVFC	Overrun error flag clear
0	The STR1.OVF flag is not cleared.
1	The STR1.OVF flag is cleared.

MFEFC	Manchester framing error flag clear
0	The STR1.MFEF flag is not cleared.
1	The STR1.MFEF flag is cleared.

Note Do not clear the STR1.BBF flag while the setting of either the CTR1.TE or the CTR1.RE bit is 1.

Caution Be sure to clear bits 15 to 11 to 0.

26.2.23 DALI software reset register 1 (SWRR1)

This register is used to initialize the data registers, the internal state, and the output pin of the DALI module. This register can be written by a 16-bit memory manipulation instruction. When this register is read, 0 is always read. The value of this register following a reset is 0000H.

Figure 26 - 24 Format of DALI Software Reset Register 1 (SWRR1)

Address: F04FCH
After reset: 0000H
R/W: W

Symbol	15	14	13	12	11	10	9	8
SWRR1	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	SWR

SWR	Software reset
Writing 1 to the SWR bit initializes the data registers, the internal state, and the output pin of the DALI module. Note that the control registers are not initialized. A software reset is completed in 1 cycle at f_{CLK}, after which transfer is ready for restarting from the next cycle. The following lists the registers, the interrupt request signals, and the output pin to be initialized. Registers: • RDR1L and RDR1H • STR1 and STR2 • COLR1 (except for the RXDMON bit and the CLDAF flag) Interrupt request signals: INTSDD, INTFED, INTBPD, INTCLD, INTED, INTRD, and INTTD An output pin of the DALI module: DALITxD0	

Caution Be sure to clear bits 15 to 2 to 0.

26.2.24 DALITxD0 waveform adjustment register 1 (TXWR1)

This register is used to adjust the DALITxD0 output waveform. This register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 003FH.

Figure 26 - 25 Format of DALITxD0 Waveform Adjustment Register 1 (TXWR1)

Address: F04D6H
After reset: 003FH
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TXWR1	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	TXLW[6:0]						

TXLW[6:0]	Low-level width of the DALITxD0 output waveform
—	DALITxD0 pin width at low level
Setting of these bits enables adjusting the widths at high and low level of the Manchester code transmitted from the DALITxD0 pin in units of two cycles of the operating clock for the DALI module (6.5 µs typ.). This can be used to correct for deviations in the timing characteristics of the rising or falling edges of a circuit connected to the DALI bus. The following shows the widths at high and low level while this function is on or off, that is, when the setting of the CNFR2.TXWE bit is 1 or 0, respectively. When the setting of the CNFR2.TXWE bit is 0: • Low-level width: 416 µs (reset value = 3FH) • High-level width: 416 µs (reset value = 3FH) When the setting of the CNFR2.TXWE bit is 1: • Low-level width: (TXLW[6:0] + 1) × 6.5 µs • High-level width: 832 µs – {(TXLW[6:0] + 1) × 6.5 µs} These bits can be set to values in the range from 20H (214.5 µs) to 5EH (617.5 µs). For details, see **26.3.6 Width adjustment for DALITxD0 output waveform**.	

Caution Be sure to clear bits 15 to 7 to 0.

26.2.25 DALIRxD0 waveform adjustment register 1 (RXWR1)

This register enables adjustment of sensing of the DALIRxD0 input waveform. This register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 3F00H.

Figure 26 - 26 Format of DALIRxD0 Waveform Adjustment Register 1 (RXWR1)

Address: F04D8H
After reset: 3F00H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
RXWR1	RSPOL	0	RSLEN[5:0]					

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	RSEN

RSPOL	Selection of the section of the DALIRxD0 input waveform for correction
0	Width for sensing of the low level of the DALIRxD0 input waveform is to be extended.
1	Width for sensing of the high level of the DALIRxD0 input waveform is to be extended.
The RSPOL bit specifies the part of the waveform sensing to be extended in width adjustment for the DALIRxD0 input waveform. This bit is valid when the setting of the RSEN bit is 1.	

RSLEN[5:0]	Extension width for the DALIRxD0 input waveform
00H to 3EH	Sets the extension width for the DALIRxD0 input waveform.
The RSLEN[5:0] bits specify the period by which sensing of the selected part of the DALIRxD0 input waveform is to be extended. These bits are valid when the setting of the RSEN bit is 1. The part of the waveform for which sensing is to be extended is set by the RSPOL bit. The extension period is set in units of one cycle of the operating clock for the DALI module (3.25 µs typ.). Set the value for the required extension period – 1 in these bits. Values in the following range can be set. 00H (= 3.25 µs) to 3EH (= 204.75 µs) Setting the value to 3FH applies no correction to the DALIRxD0 input waveform.	

RSEN	Width adjustment enable for the DALIRxD0 input waveform
0	Adjustment of width sensing for the DALIRxD0 input waveform is disabled.
1	Adjustment of width sensing for the DALIRxD0 input waveform is enabled.
The RSEN bit enables or disables the adjustment of width sensing for the DALIRxD0 input waveform. For details, see **26.3.7 Width adjustment for DALIRxD0 input waveform**.	

Caution Be sure to clear bits 14 and 7 to 1 to 0.

26.2.26 DALI reception timing adjustment register 0 (FTDC0)

This register adjusts the timing of sampling at the 1/4- and 3/4-bit timing points of the DALIRxD0 input signal. For details, see **26.3.5 Sampling timing of the DALIRxD0 input signal and bit length adjustment**. This register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 26 - 27 Format of DALI Reception Timing Adjustment Register 0 (FTDC0)

Address: F04E4H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
FTDC0	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	IST

Caution Be sure to clear bits 15 to 1 to 0.

26.2.27 Registers for controlling the port functions multiplexed with the DALI inputs and outputs

Set the following registers to control the port functions multiplexed with the DALI inputs and outputs.

- Port mode registers xx (PMxx)
- Port registers xx (Pxx)
- Port input mode registers xx (PIMxx)
- Port output mode registers xx (POMxx)
- Port mode control A registers xx (PMCAxx)

For details, see the following sections.

- **7.3.1 Port mode registers xx (PMxx)**
- **7.3.2 Port registers xx (Pxx)**
- **7.3.4 Port input mode registers xx (PIMxx)**
- **7.3.5 Port output mode registers xx (POMxx)**
- **7.3.7 Port mode control A registers xx (PMCAxx)**

When the pin multiplexed with DALITxD0 is to be used for DALI transmission, set the port mode register (PMxx) and port mode control A register (PMCAxx) bits corresponding to each port to 0. Furthermore, set the port register (Pxx) bit corresponding to each port to 1. The corresponding bit in the port output mode register xx (POMxx) can be set to 0 or 1.

When the pin multiplexed with DALIRxD0 is to be used for DALI reception, set the port mode control A register (PMCAxx) bit corresponding to each port to 0. Furthermore, set the port mode register (PMxx) bit corresponding to each port to 1. The corresponding bit in the port register (Pxx) can be set to 0 or 1.

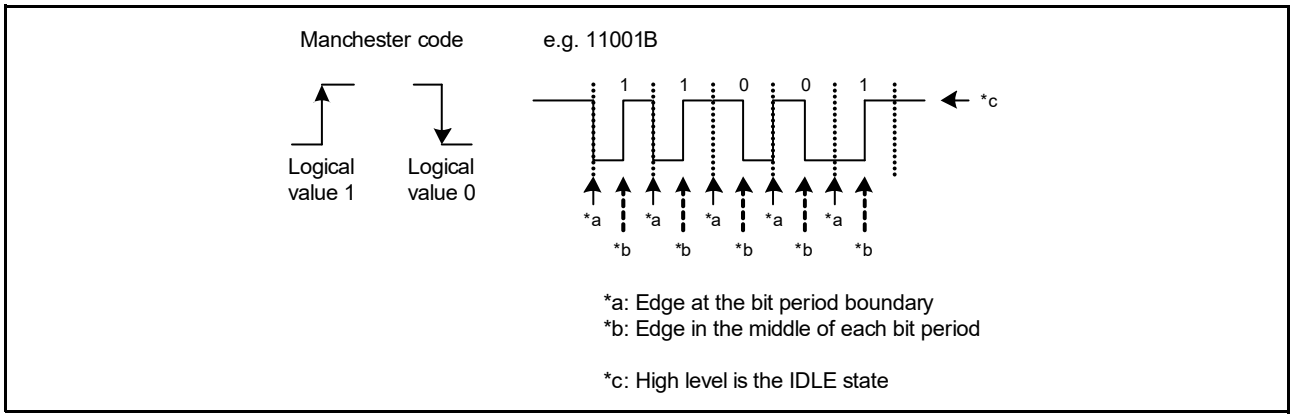
Remark xx = 0, 5
Note that PMCA5 is not present in the RL78/G24 products.

26.3 Functions of DALI Communications

26.3.1 Data format

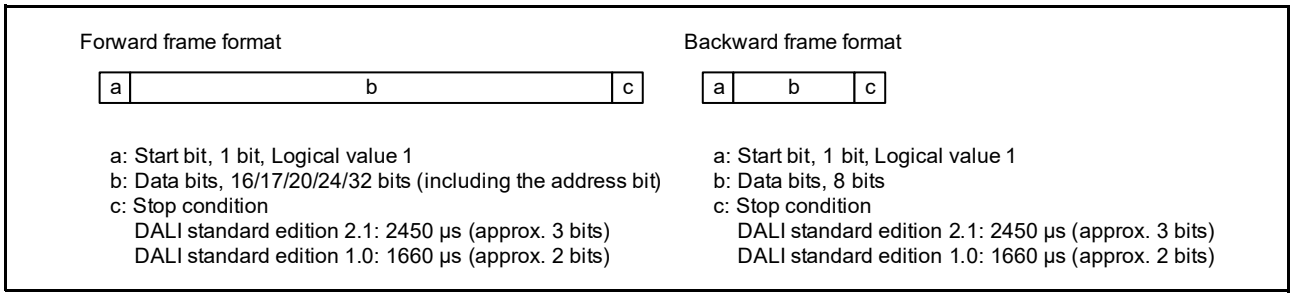
In DALI communications, bits are defined using Manchester code. Manchester code expresses logical value 1 or 0, not by voltage level (high or low), but by level change (rising or falling edge).

Figure 26 - 28 Manchester Code



Frames in DALI communications are individually defined as forward frames and backward frames. Forward frames are transmitted from an application controller to another, from an input device to an application controller, or from an application controller to a control gear. Backward frames are transmitted from a control gear or an input device to an application controller. A frame consists of a start bit, data bits, and a stop condition. The DALIRxD0 pin is used for reception and the DALITxD0 pin is used for transmission. Communications is performed in units of data frames.

Figure 26 - 29 DALI Data Format



26.3.2 STOP condition detection and settling time

When the DALI bus is fixed to high for the period of BTV threshold value 6, the DALI module determines that a stop condition was detected and outputs INTSDD[Note 1]. Check the STR1 register to determine if the stop condition is output from another DALI device or from the DALI module. When the STR1.TENDF flag is 1, the stop condition is transmitted from the DALI module[Note 2]. Because the high-level period of the DALI bus is measured from the last rising edge, the measurement start timing varies depending on the logical value of the last data bit (see **Figure 26 - 30 Stop Condition Detection**). Note that the stop condition definition is specified in the DALI standard as listed in **Table 26 - 3**.

Table 26 - 3 Stop Condition Definition

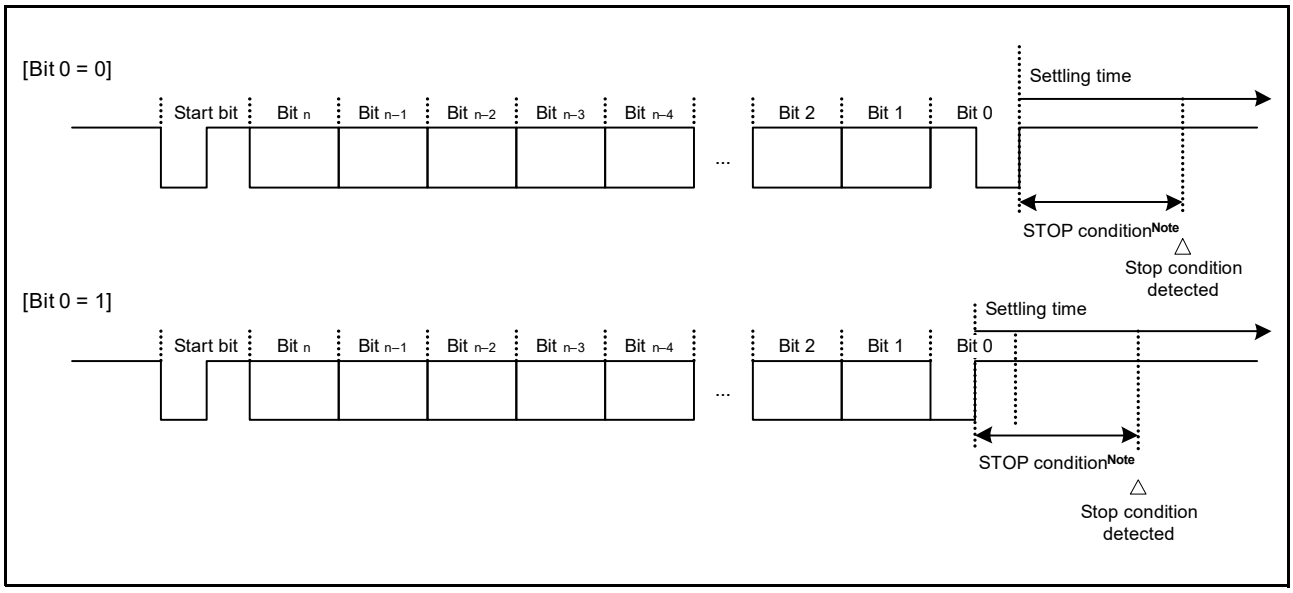
Standard	IEC 62386-101, -102 Edition 1.0	IEC 62386-101 Edition 2.0 Single-master	IEC 62386-101 Edition 2.0 Multi-master	IEC 62386-101 Edition 2.1
Stop condition definition	2 bits[Note]	2450 µs (min.)	2400 µs (min.)	2450 µs (min.)

Note The stop condition is measured from the rising edge of the DALIRxD0 input signal. If the logical value of the last data bit is 1, the stop condition starts from the rising edge in the middle of the last data bit period. The timing for the stop condition to occur is a half bit length earlier as compared with when the last data bit is 0. See **Figure 26 - 30 Stop Condition Detection**.

Note 1. Stop conditions are detected on the DALIRxD0 input signal after being synchronized with the operating clock for the DALI module in three cycles and then to which width adjustment for the DALIRxD0 input waveform is applied. The obtained signal can be checked by using the COLR1.RXDMON bit.

Note 2. For stop conditions, consider not only the delay described in **Note 1**, but also the delay from the DALITxD0 pin to the DALIRxD0 pin. When another device drives the DALIRxD0 pin low before the DALI module detects a stop condition, even if a transmission completes (the STR1.TENDF bit is 1), INTSDD is not output.

Figure 26 - 30 Stop Condition Detection



Note Time set in BTVTHR4.BTV6[8:0] is counted.

The receiver settling time between frames is defined in IEC 62386-101 Edition 2.0/2.1. The receiver settling time is defined in milliseconds and must be measured by software. Use the stop condition detection interrupt (INTSDD) and the falling edge detection interrupt (INTFED) for the measurement. In case of using the INTSDD as a trigger to measure the receiver settling time, the INTSDD is generated when a stop condition detection is completed, as shown in **Figure 26 - 30**. The period of BTV threshold value 6 must therefore be added for judgment. The timing at which the INTSDD is generated is defined by the BTVTHR4.BTV6[8:0] bits.

The transmitter settling time is also defined in IEC 62386-101 Edition 2.0/2.1 to avoid a bus collision. The transmitter

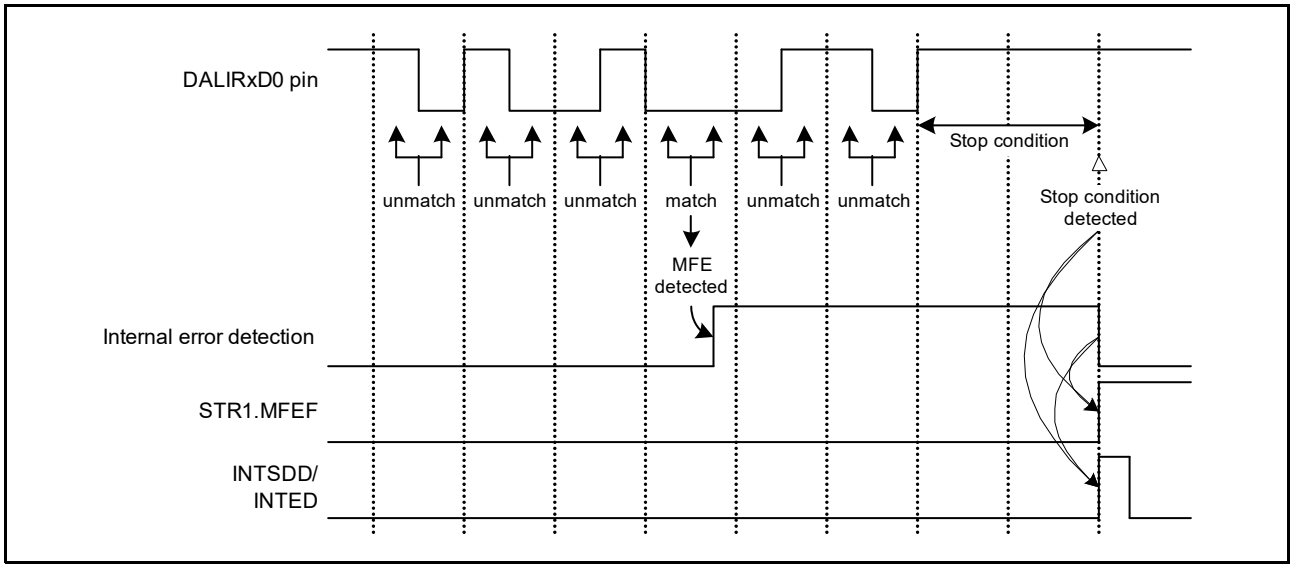
settling time is also defined in milliseconds and must be measured by software. INTSDD can be used as a trigger to start the measurement. In this case, add the period of BTV threshold value 6 for judgment. Use INTFED as a trigger to stop the settling time measurement.

26.3.3 DALI error detection

26.3.3.1 Manchester framing error (MFE)

Manchester coding is used for the bit definition of the DALI module. The values before and after the edge (1/4- and 3/4-bit timing) in the middle of a bit period are sampled and compared to identify the Manchester code. If the values are the same while the setting of the CNFR2.MFEDIS bit is 0, a Manchester framing error is judged as having occurred. While the setting of the CNFR2.MFEDIS bit is 1, Manchester framing errors are not checked.
For a bit where an MFE is detected, the value sampled at the 3/4-bit timing is stored in the DALI reception data registers 1H, 1L (RDR1H, RDR1L).
When the sampled value at 1/4-bit timing and the sampled value at 3/4-bit timing are both 0s, logical value 0 is stored in the data registers.
When the sampled value at 1/4-bit timing and the sampled value at 3/4-bit timing are both 1s, logical value 1 is stored in the data registers.
When an MFE is detected, the timing to output an interrupt (INTED) and for the flag to be set (STR1.MFEF) is when a stop condition is detected. This is the same as the INTSDD output timing.

Figure 26 - 31 Manchester Framing Error

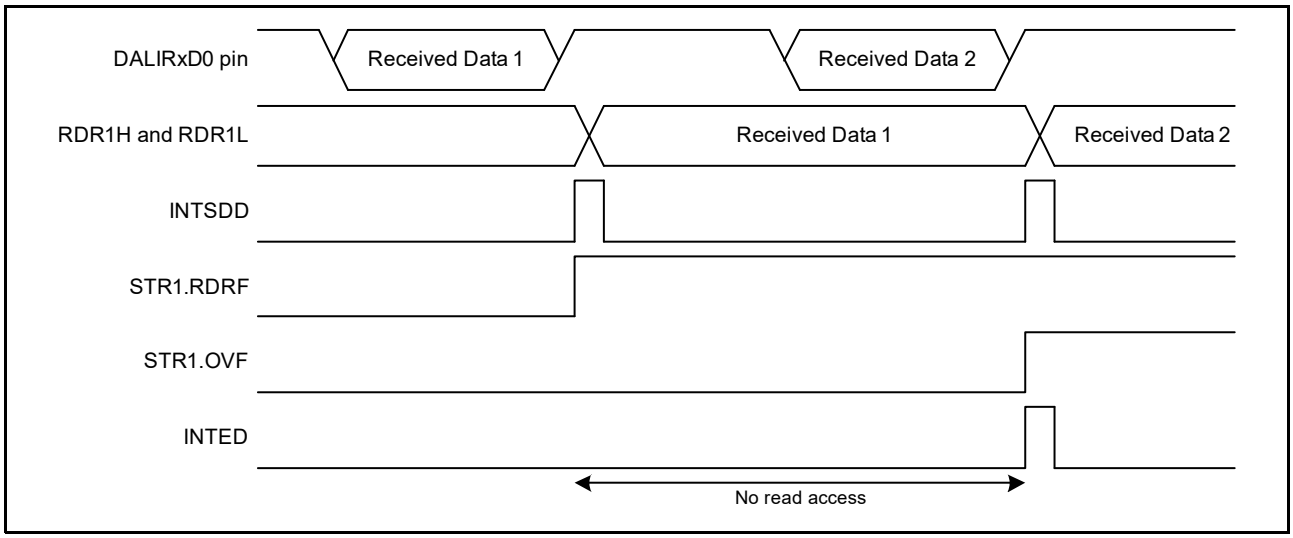


26.3.3.2 Overrun error

After a receive operation for the current data completes, if the receive operation for the next data completes before reading the received current data, an overrun error occurs.
When an overrun error occurs, the DALI module outputs INTSDD and INTED at the completion of receiving the next data (when a stop condition is detected) and sets the STR1.OVF flag.
The DALI module continues the receive operation even while the STR1.OVF flag is 1. The received data are overwritten in registers RDR1L and RDR1H and the received data width is overwritten to the STR2.RDBL[8:0] bits. Flags in the STR1 register are not overwritten. For example, if an MFE occurred in received data 1 (STR1.MFEF = 1) and no MFE occurred in next received data 2 (STR1.MFEF = 0), then the previous value of the STR1.MFEF flag remains (STR1.MFEF = 1).

1. In the non-extended mode (the setting of the CNFR1.EXM bit is 0)
   When INTED occurs, verify the error, clear the error flag, and read the received data where the error occurred. This prevents an overrun error when the next data reception completes.
2. In the extended mode (the setting of the CNFR1.EXM bit is 1)
   When INTED occurs, handle first the error in the same way as in the non-extended mode, then discard all the data in the received frame where an overrun error occurred. The reason is that up to which bit of the data in such a frame have been received successfully cannot be checked. This is caused by the set timing of the STR1.OVF flag, being when a stop condition is detected. Note that even if an overrun error occurred during frame reception, the data in the frame is received continuously until a stop condition is detected, and the INTRD signal is output every time 32 bits of the data are received.

Figure 26 - 32 Overrun Error



26.3.3.3 Frame size violation error

Frame size violation is newly defined in IEC 62386-101 Edition 2.0. When the DALI module receives an unspecified data size of a frame, it results in a frame size violation.

The DALI module does not restrict the receive data length. The maximum receive data sizes are respectively 32 bits and 256 bits in the non-extended mode and the extended mode, that are, the settings of the CNFR1.EXM bit are 0 and 1. Therefore, frame size violation is not determined by hardware. When a data reception completes, check the value of the STR2.RDBL[8:0] bits to ensure the received frame size is the specified data size and verify for a frame size violation with software.

The data size defined in IEC 62386-101 Edition 2.0 is as follows:

- Backward frame = 8 bits
- Forward frame = 16 bits, 20 bits[Note 1], 24 bits, and 32 bits[Note 1]

The DALI module continues to receive data until a stop condition is detected. When the received data are greater than 32 bits, the number of bits is counted until the internal counter reaches the maximum value of the counter. While the setting of the CNFR1.EXM bit is 0 and 1, the value is 63 and 511, respectively. When the counter reaches the maximum value, it overflows and restarts counting from 0. Therefore when the STR1.LFRF flag is 1, the value shown in these bits is invalid. When a stop condition is detected and the next data reception starts, the internal counter value is reset.

The set timing of the STR1.LFRF flag[Note 2] depends on the setting of the CNFR1.EXM bit. For details, see **26.2.17 DALI transmit data registers 1H, 1L (TDR1H, TDR1L)**.

Note 1. Reserved forward frames are defined in IEC 62386-101 Edition 2.0/2.1.

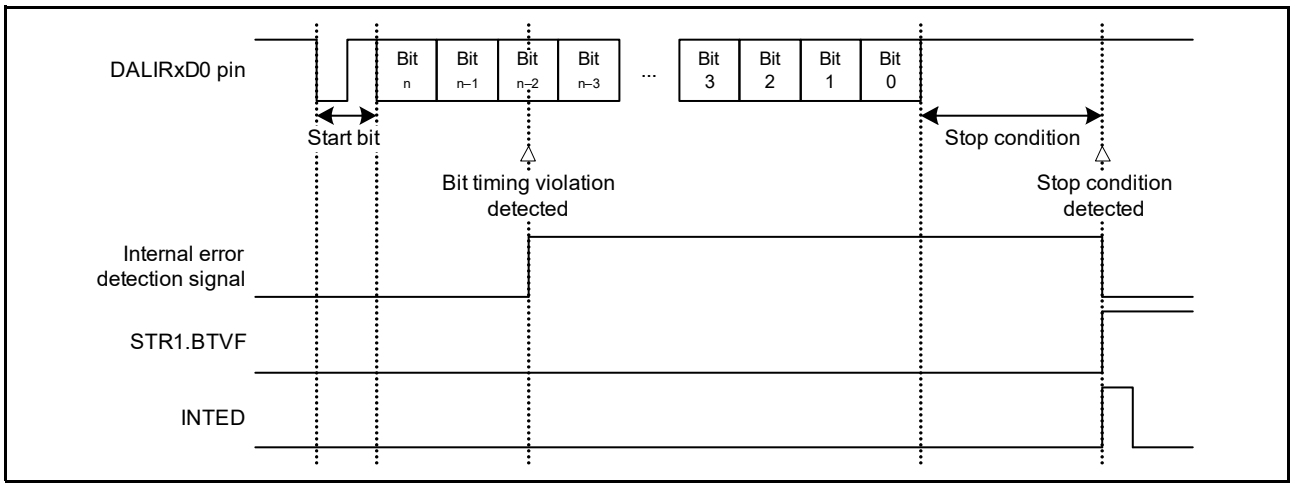
Note 2. The STR2.RDBL[8:0] bits are updated when a stop condition is detected. This flag reports incorrect long data that is received before a stop condition is detected.

26.3.3.4 Bit timing violation error

Bit timing violation is newly defined in IEC 62386-101 Edition 2.0. It defines an edge interval of received data waveforms. To enable the bit timing violation detection function, set the CNFR2.BTVE bit to 1.

When a bit timing violation occurs, the STR1.BTVF flag is set and INTED is output on a stop condition detection. Even if a bit timing violation occurs multiple times in a single frame, the STR1.BTVF flag is set and INTED is only output once when a stop condition is detected.

Figure 26 - 33 Bit Timing Violation Error



When any of the conditions is met as specified in the sections that follow, a bit timing violation occurs. The bit timing violation detection mode bit (CNFR2.BTVM) specifies whether condition 3 is used to check the violation occurrence. The BTV1 (bit timing violation 1) area control bit (CNFR2.BTV1DIS) specifies whether condition 4 is included for checking bit timing violations.
To adjust the range for checking the bit timing violation, set the bit timing violation threshold x bits (BTVx; x = 1 to 6). When the level of the DALI bus is driven high for the period of the BTV threshold value 6, it is determined as a stop condition. When the level of the DALI bus is driven low for 45 ms or more, it is determined as a bus power down. When the level is driven low for 550 ms or more, it is determined as a system failure. The DALI module then counts up to the value specified with the BTV threshold value 6, sets the STR1.BPDF flag, and outputs INTBPD. The bus power down and the system failure must be checked by software.

[Condition 1]
Relative to the falling edge at the start of the start bit or the rising or falling edge at the start of the data bit, the next rising or falling edge is within the bit timing violation as shown in **Figure 26 - 34**.

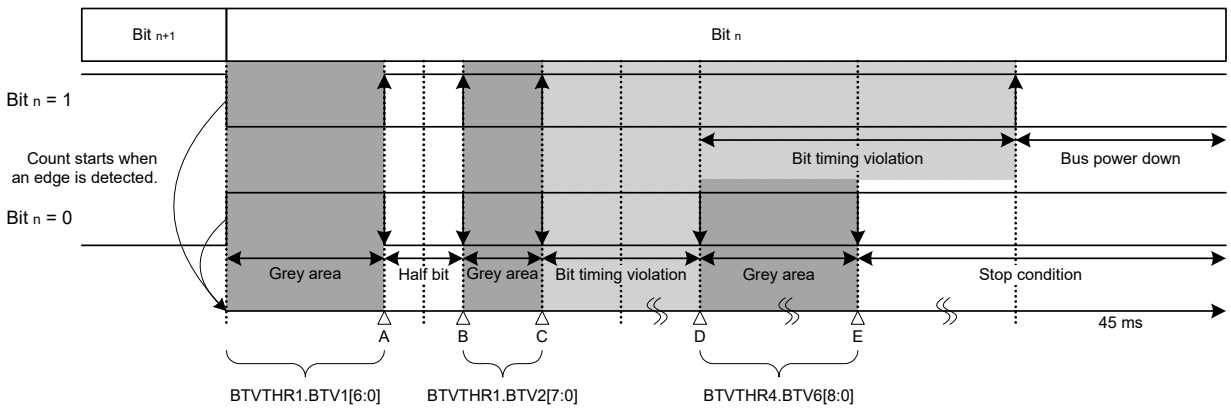
Caution The rising or falling edge at the start of a bit occurs when a bit, bit $_n$, and the previous bit, bit $_{n+1}$, are of the same value. If they have different values, it is not the condition for a bit timing violation.

[Condition 2]
Relative to the rising or falling edge of the data bit, the next rising or falling edge is within the bit timing violation as shown in **Figure 26 - 35**.

[Condition 3]
Relative to the rising or falling edge of the data bit, the next rising or falling edge is within the grey area, between points G and H as shown in **Figure 26 - 35**.

[Condition 4]
Relative to the falling edge at the start of the start bit or the rising or falling edge at the start of the data bit, the next rising or falling edge is within a grey area shown in **Figure 26 - 34** or in **Figure 26 - 35**.

Figure 26 - 34 Bit Timing Violation 1



Each of the grey areas is divided into adjacent areas with the value specified in the BTVx (x = 1 to 6) bits as a threshold value.
The following table shows the values specified in IEC 62386-101 Edition 2.0/2.1 and the initial values of the DALI module.

Point	A	B	C	D	E
DALI specification value	333.3 µs	500.0 µs	750.0 µs	1400.0 µs	2400.0 µs
Initial value of the DALI module[Note 1]	BTVTHR1.BTV1 [6:0] = 00H	BTVTHR1.BTV2[7:0] = 4FH		BTVTHR4.BTV6[8:0] = 0DBH	
	0 µs	520 µs		1423.5 µs	

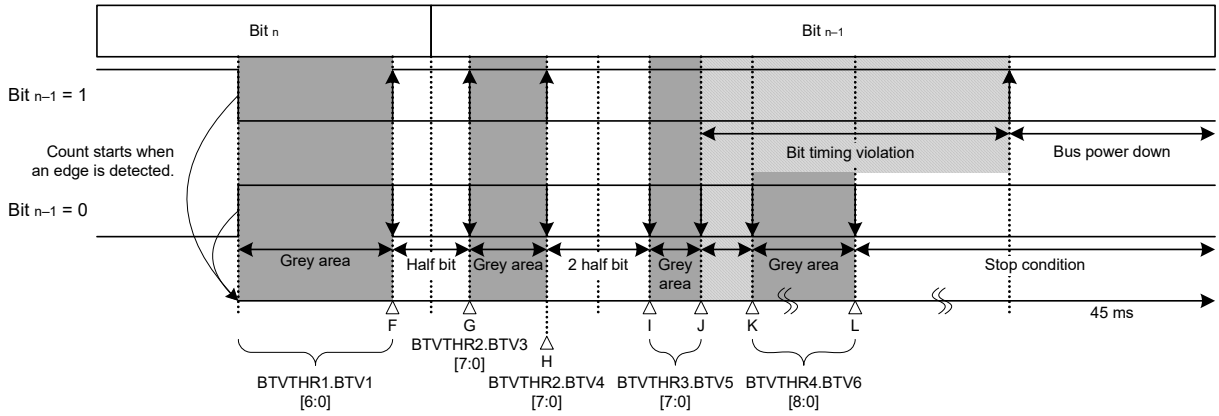
Definition of BTV areas for the period until the first rising edge in a bit period boundary starting with a fall:
Bit period boundary edge < Bit timing violation area[Note 2] ≤ BTV threshold value 1 (BTVTHR1.BTV1[6:0])
BTV threshold value 1 (BTVTHR1.BTV1[6:0]) < Half bit area ≤ BTV threshold value 2 (BTVTHR1.BTV2[7:0])
BTV threshold value 2 (BTVTHR1.BTV2[7:0]) < Bit timing violation area ≤ BTV threshold value 6 (BTVTHR4.BTV6[8:0])
BTV threshold value 6 (BTVTHR4.BTV6[8:0]) < Bus power down and system failure

Definition of BTV areas for the period until the first falling edge in a bit period boundary starting with a rise:
Bit period boundary edge < Bit timing violation area[Note 2] ≤ BTV threshold value 1 (BTVTHR1.BTV1[6:0])
BTV threshold value 1 (BTVTHR1.BTV1[6:0]) < Half bit area ≤ BTV threshold value 2 (BTVTHR1.BTV2[7:0])
BTV threshold value 2 (BTVTHR1.BTV2[7:0]) < Bit timing violation area < BTV threshold value 6 (BTVTHR4.BTV6[8:0])
BTV threshold value 6 (BTVTHR4.BTV6[8:0]) ≤ Stop condition

Note 1. Each value does not include frequency errors nor the errors due to bit rate setting.

Note 2. This is only checked while the setting of the CNFR2.BTV1DIS bit is 0.

Figure 26 - 35 Bit Timing Violation 2



Each of the grey areas is divided into adjacent areas with the value specified in the BTVx (x = 1 to 6) bits as a threshold value.
The following table shows the values specified in IEC 62386-101 Edition 2.0/2.1 and the initial values of the DALI module.

Point	F	G	H	I	J	K	L
DALI specification value	333.3 µs	500.0 µs	666.7 µs	1000.0 µs	1200.0 µs	1400.0 µs	2400.0 µs
Initial value of the DALI module[Note 1]	BTVTHR1.BTV1 [6:0] = 00H	BTVTHR2.BTV3 [7:0] = 4FH	BTVTHR2.BTV4 [7:0] = 65H	BTVTHR3.BTV5[7:0] = 9DH		BTVTHR4.BTV6[8:0] = 0DBH	
	0 µs	520 µs	656.5 µs	1027 µs		1423.5 µs	

Definition of BTV areas from a falling edge to the succeeding rising edge in the middle of a bit period while the setting of the CNFR2.BTVM bit is 0:
Middle edge < Bit timing violation area[Note 2] ≤ BTV threshold value 1 (BTVTHR1.BTV1[6:0])
BTV threshold value 1 (BTVTHR1.BTV1[6:0]) < Half bit area/2 half bit area ≤ BTV threshold value 5 (BTVTHR3.BTV5[7:0])
BTV threshold value 5 (BTVTHR3.BTV5[7:0]) < Bit timing violation area ≤ BTV threshold value 6 (BTVTHR4.BTV6[8:0])
BTV threshold value 6 (BTVTHR4.BTV6[8:0]) < Bus power down and system failure

Definition of BTV areas from a falling edge to the succeeding rising edge in the middle of a bit period while the setting of the CNFR2.BTVM bit is 1:
Middle edge < Bit timing violation area[Note 2] ≤ BTV threshold value 1 (BTVTHR1.BTV1[6:0])
BTV threshold value 1 (BTVTHR1.BTV1[6:0]) < Half bit area ≤ BTV threshold value 3 (BTVTHR2.BTV3[7:0])
BTV threshold value 3 (BTVTHR2.BTV3[7:0]) < Bit timing violation area ≤ BTV threshold value 4 (BTVTHR2.BTV4[7:0])
BTV threshold value 4 (BTVTHR2.BTV4[7:0]) < 2 half bit area ≤ BTV threshold value 5 (BTVTHR3.BTV5[7:0])
BTV threshold value 5 (BTVTHR3.BTV5[7:0]) < Bit timing violation area ≤ BTV threshold value 6 (BTVTHR4.BTV6[8:0])
BTV threshold value 6 (BTVTHR4.BTV6[8:0]) < Bus power down and system failure

Definition of BTV areas from a rising edge to the succeeding falling edge in the middle of a bit period while the setting of the CNFR2.BTVM bit is 0:
Middle edge < Bit timing violation area[Note 2] ≤ BTV threshold value 1 (BTVTHR1.BTV1[6:0])
BTV threshold value 1 (BTVTHR1.BTV1[6:0]) < Half bit area/2 half bit area ≤ BTV threshold value 5 (BTVTHR3.BTV5[7:0])
BTV threshold value 5 (BTVTHR3.BTV5[7:0]) < Bit timing violation area < BTV threshold value 6 (BTVTHR4.BTV6[8:0])
BTV threshold value 6 (BTVTHR4.BTV6[8:0]) ≤ Stop condition

Definition of BTV areas from a rising edge to the succeeding falling edge in the middle of a bit period while the setting of the CNFR2.BTVM bit is 1:
Middle edge < Bit timing violation area[Note 2] ≤ BTV threshold value 1 (BTVTHR1.BTV1[6:0])
BTV threshold value 1 (BTVTHR1.BTV1[6:0]) < Half bit area ≤ BTV threshold value 3 (BTVTHR2.BTV3[7:0])
BTV threshold value 3 (BTVTHR2.BTV3[7:0]) < Bit timing violation area ≤ BTV threshold value 4 (BTVTHR2.BTV4[7:0])
BTV threshold value 4 (BTVTHR2.BTV4[7:0]) < 2 half bit area ≤ BTV threshold value 5 (BTVTHR3.BTV5[7:0])
BTV threshold value 5 (BTVTHR3.BTV5[7:0]) < Bit timing violation area < BTV threshold value 6 (BTVTHR4.BTV6[8:0])
BTV threshold value 6 (BTVTHR4.BTV6[8:0]) ≤ Stop condition

Note 1. Each value does not include frequency errors nor the errors due to bit rate setting.
Note 2. This is only checked while the setting of the CNFR2.BTV1DIS bit is 0.

In the DALI module, the DALI specification values shown in **Figures 26 - 34** and **26 - 35** are set with BTVTHR1 to BTVTHR4 registers. The value set must comply with IEC 62386-101 Edition 2.0, taking into account frequency accuracy and sampling errors of the DALI module.

Table 26 - 4 lists settings of the BTVx (x = 1 to 6) bits.

Table 26 - 4 BTV Threshold Setting Values

Bit Name	Available Value Range [µs]	Initial Value	Description
BTVTHR1.BTV1 [6:0]	0 to 825.5	00H	Specify Point A in **Figure 26 - 34 Bit Timing Violation 1** and Point F in **Figure 26 - 35 Bit Timing Violation 2** within the available value range in 6.5 µs steps. The range from an edge to BTV threshold value 1 is determined as a bit timing violation. This range is described in the grey areas in **Figure 26 - 34 Bit Timing Violation 1** and **Figure 26 - 35 Bit Timing Violation 2**. The range from BTV threshold value 1 to BTV threshold value 2 in **Figure 26 - 34 Bit Timing Violation 1** and BTV threshold value 1 to BTV threshold value 3 in **Figure 26 - 35 Bit Timing Violation 2** is the half bit area.
BTVTHR1.BTV2 [7:0]	6.5 to 1664 (Condition: BTVTHR1.BTV1[6:0] < BTVTHR1.BTV2[7:0])	4FH	Specify Point B in **Figure 26 - 34 Bit Timing Violation 1** within the available value range in 6.5 µs steps. The range from BTV threshold value 1 to BTV threshold value 2 is the half bit area. The range from BTV threshold value 2 to BTV threshold value 6 is the bit timing violation.
BTVTHR2.BTV3 [7:0]	6.5 to 1664 (Condition: BTVTHR1.BTV1[6:0] + 1 < BTVTHR2.BTV3[7:0])	4FH	Specify Point G in **Figure 26 - 35 Bit Timing Violation 2** within the available value range in 6.5 µs steps. The range from BTV threshold value 1 to BTV threshold value 3 is the half bit area. The range from BTV threshold value 3 to BTV threshold value 4 is determined as a bit timing violation when the CNFR2.BTVM bit is 1. This range is described in the grey areas in **Figure 26 - 34 Bit Timing Violation 1** and **Figure 26 - 35 Bit Timing Violation 2**.
BTVTHR2.BTV4 [7:0]	0 to 1657.5 (Condition: BTVTHR2.BTV3[7:0] + 1 < BTVTHR2.BTV4[7:0])	65H	Specify Point H in **Figure 26 - 35 Bit Timing Violation 2** within the available value range in 6.5 µs steps. The range from BTV threshold value 3 to BTV threshold value 4 is determined as a bit timing violation when the CNFR2.BTVM bit is 1. This range is described in the grey areas in **Figure 26 - 34 Bit Timing Violation 1** and **Figure 26 - 35 Bit Timing Violation 2**. The range from BTV threshold value 4 to BTV threshold value 5 is the 2 half bit area.
BTVTHR3.BTV5 [7:0]	6.5 to 1664 (Condition: BTVTHR2.BTV4[7:0] + 1 < BTVTHR3.BTV5[7:0])	9DH	Specify a threshold value between Point I and Point J in **Figure 26 - 35 Bit Timing Violation 2** within the available value range in 6.5 µs steps. When the CNFR2.BTVM bit is 0, the range from BTV threshold value 1 to BTV threshold value 5 is the half bit or 2 half bit area, and the range from BTV threshold value 5 to BTV threshold value 6 is the bit timing violation. When the CNFR2.BTVM bit is 1, the range from BTV threshold value 4 to BTV threshold value 5 is the 2 half bit area, and the range from BTV threshold value 5 to BTV threshold value 6 is the bit timing violation.
BTVTHR4.BTV6 [8:0]	1254.5 to 3328 (Condition: BTVTHR1.BTV2[7:0] + 1 < BTVTHR4.BTV6[8:0] and BTVTHR3.BTV5[7:0] + 1 < BTVTHR4.BTV6[8:0])	0DBH	Specify Point D in **Figure 26 - 34 Bit Timing Violation 1** and a threshold value between Point K and Point L in **Figure 26 - 35 Bit Timing Violation 2** within the available value range in 6.5 µs steps. It is prohibited to set a value smaller than 1254.5 µs. The minimum available value is C0H. The range from BTV threshold value 2 to BTV threshold value 6 in **Figure 26 - 34 Bit Timing Violation 1** and BTV threshold value 5 to BTV threshold value 6 in **Figure 26 - 35 Bit Timing Violation 2** is the bit timing violation. The range beginning with the BTV threshold value 6 is determined as the stop condition or the bus power down.

26.3.3.5 Bus power down and system failure

When the level of the DALI bus is driven low for 45 ms or more, it is a bus power down. When the level is driven low for 550 ms or more, it is a system failure. In both cases, appropriate actions must be taken.

A bus power down is determined when the DALI module counts the DALI bus low-level period up to the period of BTV threshold value 6. Thereafter, the STR1.BPDF flag is set and INTBPD is output. After INTBPD is output, measure the low-level period of the DALI bus for at least 45 ms with software to determine whether it is a bus power down or not. Subsequently, measure the low-level period for at least 550 ms with software to determine whether it is a system failure or not.

When the DALI module detects a rising edge of the DALIRxD0 pin, it clears the STR1.BPDF flag and returns to the receive operation process. When a stop condition is detected after the receive operation starts, INTSDD is output. When INTSDD is output, stop the 45 ms or 550 ms measurement. When a probable bus power down is detected again after the receive operation starts, the STR1.BPDF flag is set to 1 again and INTBPD is output.

If the STR1.BPDF flag is reset to 0 after the 45 ms or 550 ms measurement, it means that the DALI bus was not held low during the measurement.

When the DALI bus is not held low during the 45 ms or 550 ms measurement, the DALI module returns to the receive operation process. In this case, the received data are not reliable, so discard the received data before a stop condition is detected after the STR1.BPDF flag changed from 1 to 0.

When INTSDD or INTBPD is not output during the 45 ms measurement and the STR1.BPDF flag is 1 after the measurement, the DALI bus is determined as a bus power down. Then, when INTSDD or INTBPD is not output during the subsequent 550 ms measurement and the STR1.BPDF flag is 1 after the measurement, the DALI bus is determined as a system failure.

The DALI module continues the receive operation until a stop condition is detected while the STR1.BPDF flag is 1. Therefore, to stop the receive operation, set bits CTR1.TE and CTR1.RE to 0 and issue a software reset by setting the SWRR1.SWR bit to 1. After the DALI bus is restored, set bits CTR1.TE and CTR1.RE to 1 to restart communications.

Figure 26 - 36 Bus Power Down and System Failure

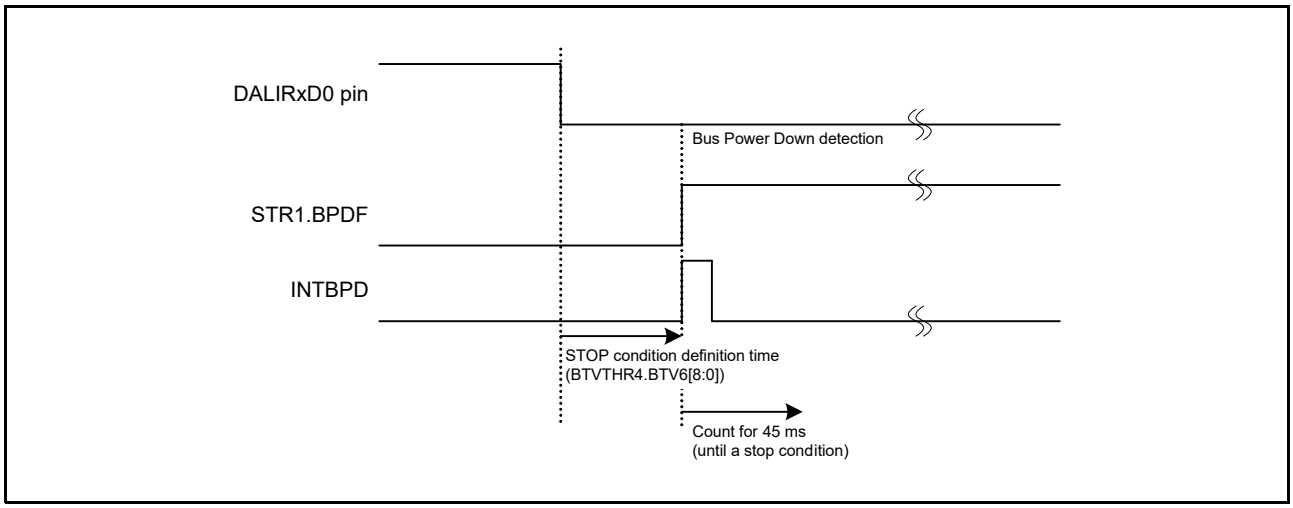
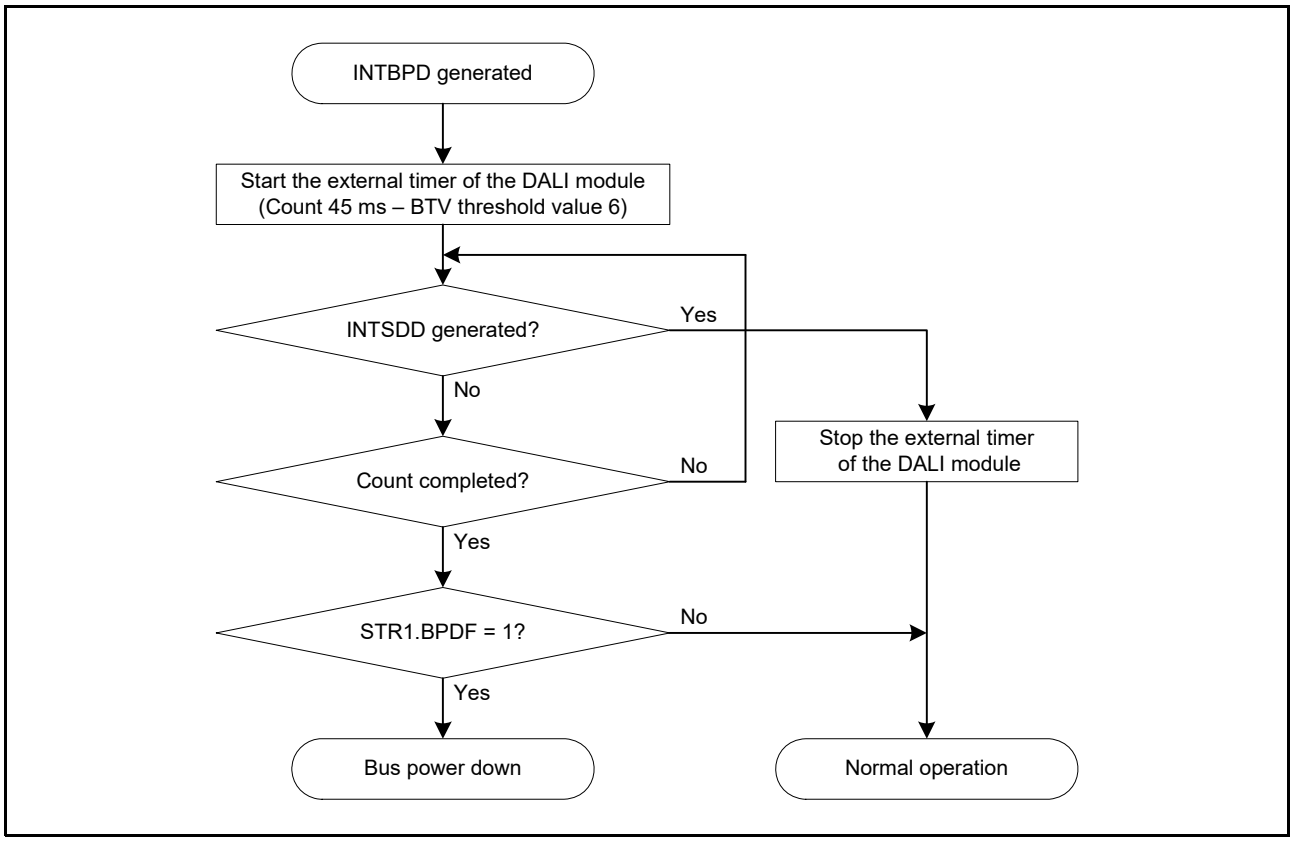


Figure 26 - 37 Flow of Bus Power Down and System Failure



26.3.3.6 Underrun error

In the extended mode (CNFR1.EXM = 1), an underrun error occurs when transmit data cannot be written in intended time, meaning unsuccessful data transmission.

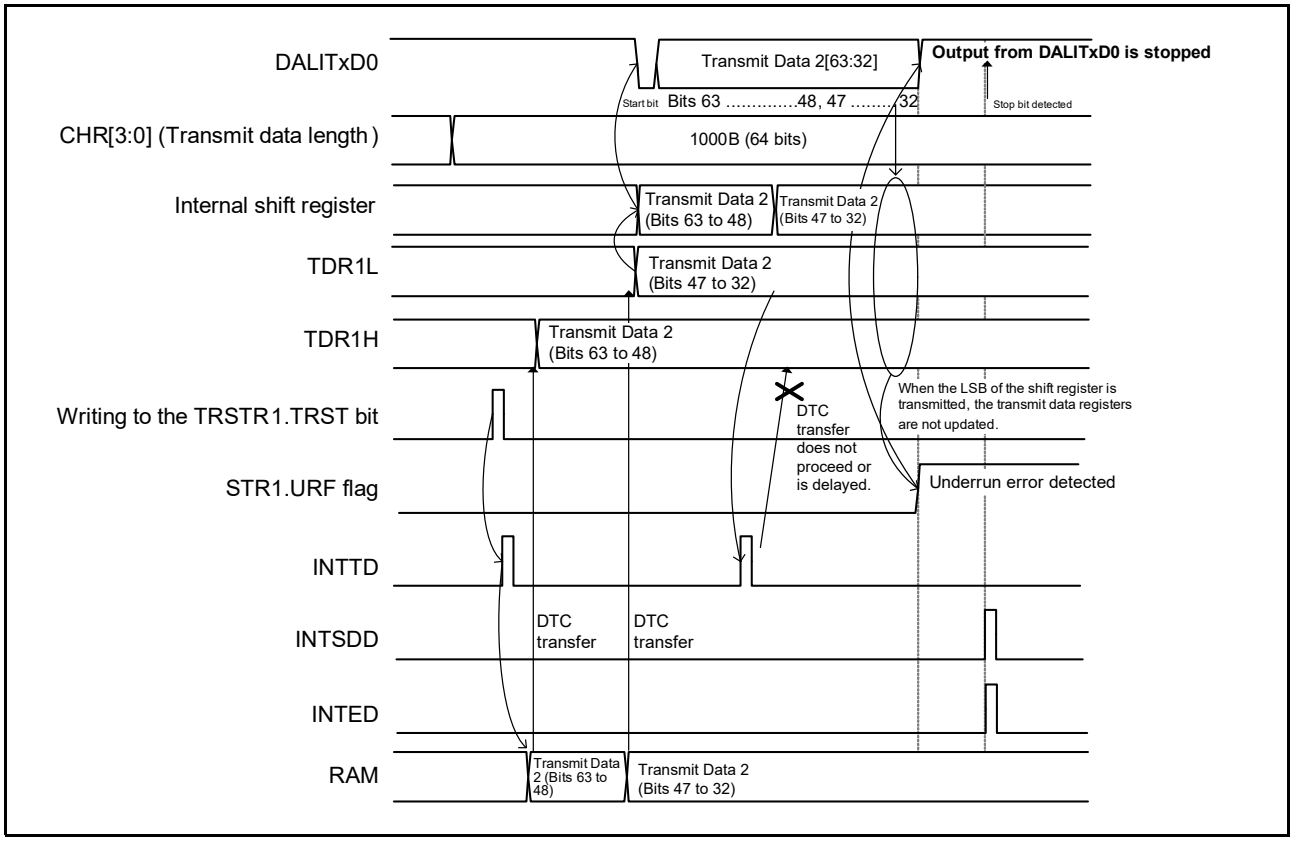
In the extended mode, transmit data are intended to be written to the DALI transmit data registers 1H, 1L (TDR1H, TDR1L) in response to the write request interrupt for transmit data (INTTD) signal. The written transmit data are stored in the internal shift register for transmission, then transmitted with the MSB-first. If transmit data cannot be written prior to the timing to be stored in the internal shift register for transmission, an underrun error is detected and the STR1.URF flag is set.

When an underrun error is detected, the output from the transmission pin (DALITxD0) is stopped and the DALI bus enters into the idle state, being set to the high level. This means that a stop condition is generated. When the stop condition is detected, the INTSDD and INTED signals are output. Note that the STR1.URF flag is not set and the INTED signal is not output if the setting of the transmit enable (CTR1.TE) bit is changed from 1 to 0 when a stop condition is generated following detection of an underrun error.

When INTED occurs, check the details of the error, clear the error flag, and stop the transmission by changing the setting of the CTR1.TE bit from 1 to 0.

While the value of the STR1.URF flag is 1, the INTTD signal is not output. In addition, writing 1 to the TRSTR1.TRST bit and writing to the TDR1L register are both invalid, that is, new transmission cannot start.

Figure 26 - 38 Underrun Error



26.3.4 Collisions

26.3.4.1 Overview of collisions

The DALI module supports multi-master communications. If two or more masters transmit data simultaneously during multi-master communications, a collision occurs on the bus. When the CNFR2.CDE bit is set to 1, the DALI module collision detection function is enabled and the collision detection interrupt (INTCLD) is generated. When a collision is detected during transmission, stop the transmission and perform the collision recovery or collision avoidance based on the collision occurrence condition.
For backward frame transmission, set the CNFR2.CDE bit to 0 and disable the collision detection function.
The DALI module detects a collision during the receive operation, where CTR1.RE bit = 1 and CTR1.TE bit = 0. If you do not want to detect a collision between multiple masters, set the CNFR2.CDE bit to 0.

26.3.4.2 Criteria for collision detection

Select the criteria for collision detection with the collision detection mode bit (CNFR2.CDM0).
Table 26 - 5 shows the relation between collision detection mode and collision detection interrupt (INTCLD).

Table 26 - 5 Criteria for Collision Detection Based on the Set Value of the CNFR2.CDM0 Bit

Set Value of the CNFR2.CDM0 Bit	Collision Detection Area[Note 1]			Collision Detection Interrupt (INTCLD) Condition[Note 2]
	Destroy	Avoidance	Valid	
0	Detected	Not detected	Not detected	Condition 1A or Condition 2A
1	Detected	Detected	Not detected	When either of the following conditions is met[Notes 3, 4] Condition 1A or Condition 1B Condition 2A or Condition 2B

Note 1. IEC 62386-101 Edition 2.0/2.1 defines the following areas:
- Destroy: Destroy area
- Avoidance area: Grey area
- Valid area: Valid half bit/Valid 2 half bit

The grey areas in the DALI standard are defined as the avoidance areas of the DALI module. In the areas, collision detection timing can be treated either as an error or not. While the setting of the CNFR2.CDM0 bit is 0, the avoidance areas (grey areas) are treated as valid areas. No edge occurrence in the areas sets the error flag STR1.CDF. While the setting of the CNFR2.CDM0 bit is 1, each edge occurrence in the avoidance areas (grey areas) sets the error flag STR1.CDF.

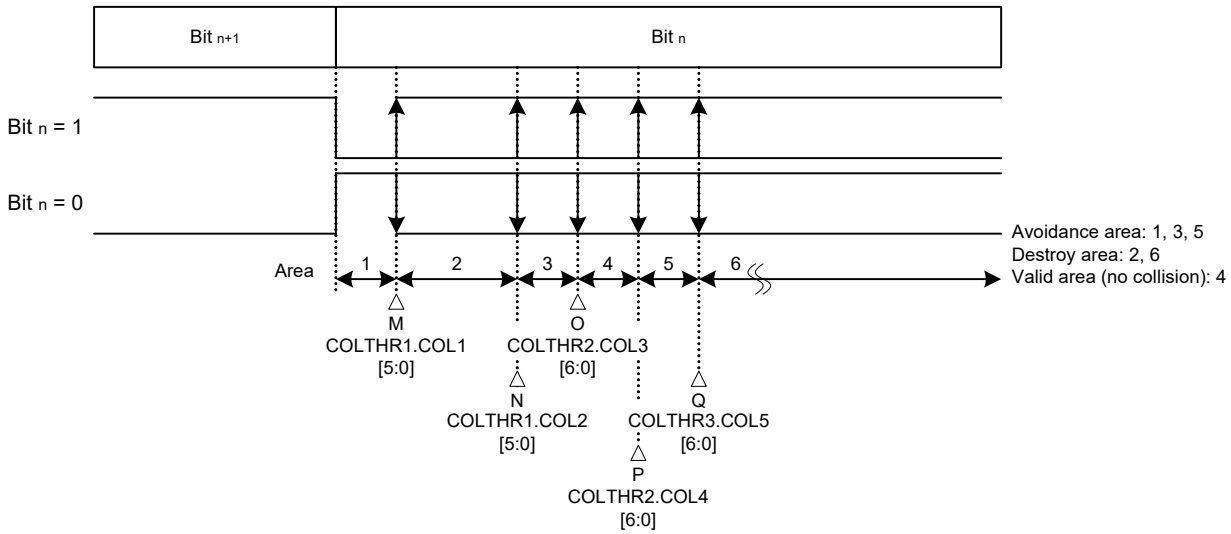
Note 2. For details on the conditions, see **Figure 26 - 39 Collision Detection Timing 1** and **Figure 26 - 40 Collision Detection Timing 2**.

Note 3. Adjust the timing to determine a collision by setting COLx (x = 1 to 9) bits. See **Figure 26 - 39 Collision Detection Timing 1** and **Figure 26 - 40 Collision Detection Timing 2**.

Note 4. When a collision detection interrupt (INTCLD) occurs, use the collision detection information to set any processing with software after the collision detection. For details, see **26.3.4.4 Collision detection information**.

The criteria for collision detection are valid when any of the conditions shown in **Figures 26 - 39** and **26 - 40** is met.

Figure 26 - 39 Collision Detection Timing 1



The following table shows the values specified in IEC 62386-101 Edition 2.0/2.1 and initial values of the DALI module.

Point	M	N	O	P	Q
Setting bit	COLTHR1.COL1 [5:0]	COLTHR1.COL2 [5:0]	COLTHR2.COL3 [6:0]	COLTHR2.COL4 [6:0]	COLTHR3.COL5 [6:0]
DALI specification value	100.0 µs	356.7 µs	400.0 µs	433.3 µs	476.7 µs
Initial value of the DALI module[Note]	97.5 µs	370.5 µs	390.0 µs	448.5 µs	468.0 µs

Criteria for collision detection

Condition 1A: Any of the following conditions is met.
- An edge is present in destroy area 2 and 6.
- The DALIRxD0 input signal is held low until destroy area 6 comes.

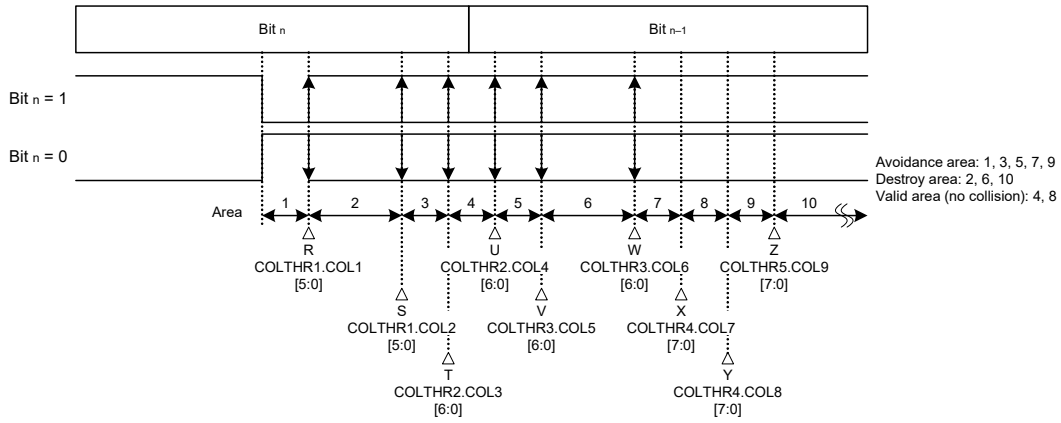
Condition 1B: The following condition is met.
- An edge is present in avoidance area 1, 3, and 5.

Relation between the collision detection areas and the collision threshold values

Bit period boundary edge < Area 1 (grey area/recovery area) ≤ COL threshold value 1 (COLTHR1.COL1[5:0])
COL threshold value 1 (COLTHR1.COL1[5:0]) < Area 2 (destroy area) ≤ COL threshold value 2 (COLTHR1.COL2[5:0])
COL threshold value 2 (COLTHR1.COL2[5:0]) < Area 3 (grey area) ≤ COL threshold value 3 (COLTHR2.COL3[6:0])
COL threshold value 3 (COLTHR2.COL3[6:0]) < Area 4 (half bit area) ≤ COL threshold value 4 (COLTHR2.COL4[6:0])
COL threshold value 4 (COLTHR2.COL4[6:0]) < Area 5 (grey area) ≤ COL threshold value 5 (COLTHR3.COL5[6:0])
COL threshold value 5 (COLTHR3.COL5[6:0]) < Area 6 (destroy area)

Note Each value does not include frequency errors nor the errors due to bit rate setting.

Figure 26 - 40 Collision Detection Timing 2



The following table shows the values specified in IEC 62386-101 Edition 2.0/2.1 and initial values of the DALI module.

Point	R	S	T	U	V	W	X	Y	Z
Setting bit	COLTHR1.COL1 [5:0]	COLTHR1.COL2 [5:0]	COLTHR2.COL3 [6:0]	COLTHR2.COL4 [6:0]	COLTHR3.COL5 [6:0]	COLTHR3.COL6 [6:0]	COLTHR4.COL7 [7:0]	COLTHR4.COL8 [7:0]	COLTHR5.COL9 [7:0]
DALI specification value	100.0 µs	356.7 µs	400.0 µs	433.3 µs	476.7 µs	723.3 µs	800.0 µs	866.7 µs	943.3 µs
Initial value of the DALI module[Note]	97.5 µs	370.5 µs	390.0 µs	448.5 µs	468.0 µs	741.0 µs	786.5 µs	890.5 µs	923.0 µs

Criteria for collision detection

Condition 2A: Any of the following conditions is met:
- An edge is present in destroy area 2, 6 and 10.
- The DALIRxD0 input signal is held low until destroy area 10 comes.

Condition 2B: The following condition is met:
- An edge is present in avoidance area 1, 3, 5, 7 and 9.

Relation between collision detection areas and collision threshold values

Middle edge < Area 1 (grey area/recovery area) ≤ COL threshold value 1 (COLTHR1.COL1[5:0])
COL threshold value 1 (COLTHR1.COL1[5:0]) < Area 2 (destroy area) ≤ COL threshold value 2 (COLTHR1.COL2[5:0])
COL threshold value 2 (COLTHR1.COL2[5:0]) < Area 3 (grey area/recovery area) ≤ COL threshold value 3 (COLTHR2.COL3[6:0])
COL threshold value 3 (COLTHR2.COL3[6:0]) < Area 4 (half bit area) ≤ COL threshold value 4 (COLTHR2.COL4[6:0])
COL threshold value 4 (COLTHR2.COL4[6:0]) < Area 5 (grey area/recovery area) ≤ COL threshold value 5 (COLTHR3.COL5[6:0])
COL threshold value 5 (COLTHR3.COL5[6:0]) < Area 6 (destroy area) ≤ COL threshold value 6 (COLTHR3.COL6[6:0])
COL threshold value 6 (COLTHR3.COL6[6:0]) < Area 7 (grey area/recovery area) ≤ COL threshold value 7 (COLTHR4.COL7[7:0])
COL threshold value 7 (COLTHR4.COL7[7:0]) < Area 8 (2 half bit area) ≤ COL threshold value 8 (COLTHR4.COL8[7:0])
COL threshold value 8 (COLTHR4.COL8[7:0]) < Area 9 (grey area/recovery area) ≤ COL threshold value 9 (COLTHR5.COL9[7:0])
COL threshold value 9 (COLTHR5.COL9[7:0]) < Area 10 (destroy area)

Note Each value does not include frequency errors nor the errors due to bit rate setting.

The DALI module sets the specification values shown in **Figures 26 - 39** and **26 - 40** in COLTHR1 to COLTHR5 registers. The value set must comply with IEC 62386-101 Edition 2.0/2.1, taking into account frequency accuracy and sampling errors of the DALI module. **Table 26 - 6** lists settings of the COLx (x = 1 to 9) bits.

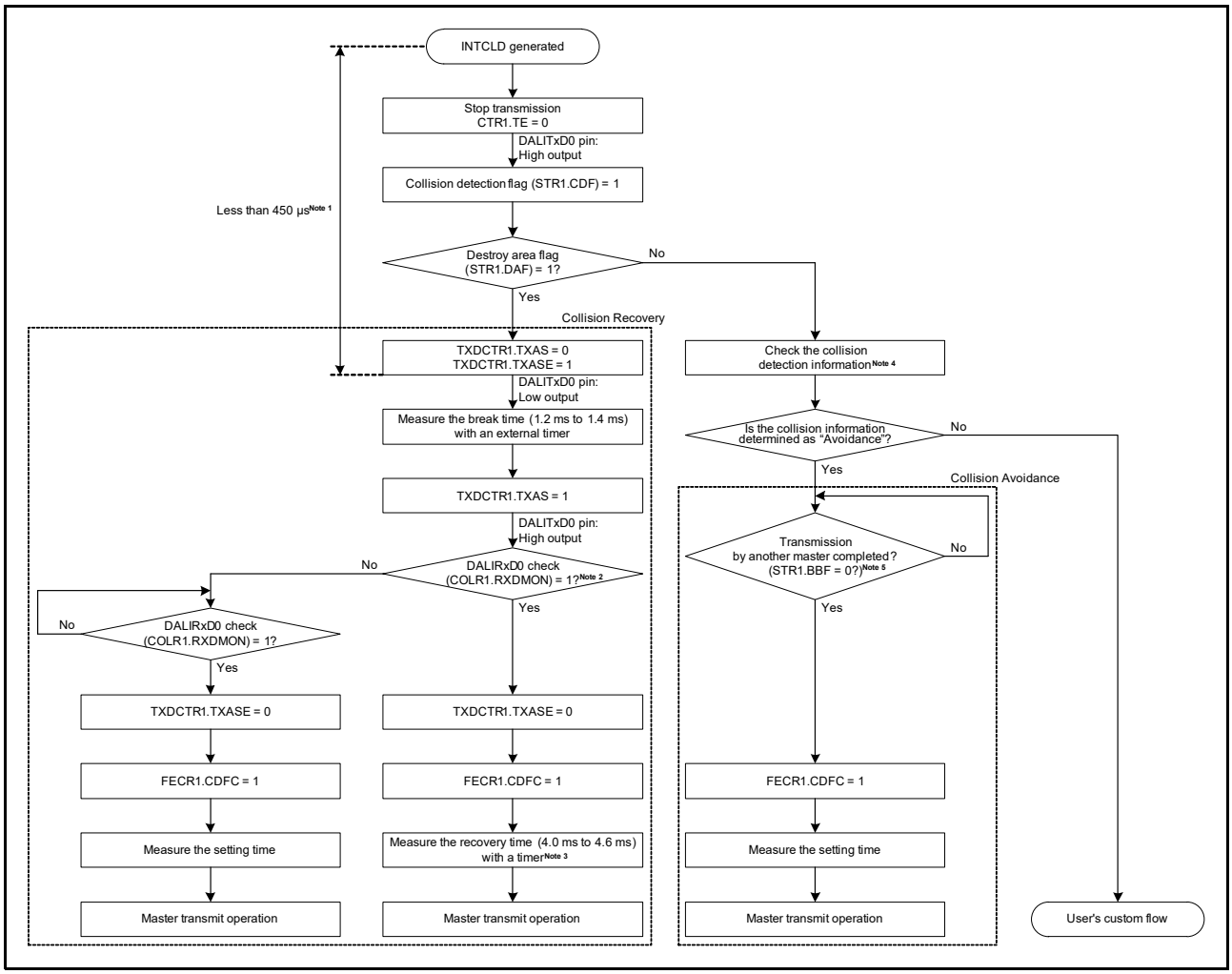
Table 26 - 6 COL Threshold Setting Values

Bit Name	Available Value Range [µs]	Initial Value	Description
COLTHR1.COL1 [5:0]	0 to 409.5	0FH	Specify Point M in **Figure 26 - 39 Collision Detection Timing 1** and Point R in **Figure 26 - 40 Collision Detection Timing 2** within the available value range in 6.5 µs steps. The range from an edge to COL threshold value 1 is the avoidance area and the range from COL threshold value 1 to COL threshold value 2 is the destroy area.
COLTHR1.COL2 [5:0]	6.5 to 416 (Condition: COLTHR1.COL1[5:0] + 1 < COLTHR1.COL2[5:0])	38H	Specify Point N in **Figure 26 - 39 Collision Detection Timing 1** and Point S in **Figure 26 - 40 Collision Detection Timing 2** within the available value range in 6.5 µs steps. The range from COL threshold value 1 to COL threshold value 2 is the destroy area and the range from COL threshold value 2 to COL threshold value 3 is the avoidance area.
COLTHR2.COL3 [6:0]	0 to 825.5 (Condition: COLTHR1.COL2[5:0] + 1 < COLTHR2.COL3[6:0])	3CH	Specify Point O in **Figure 26 - 39 Collision Detection Timing 1** and Point T in **Figure 26 - 40 Collision Detection Timing 2** within the available value range in 6.5 µs steps. The range from COL threshold value 2 to COL threshold value 3 is the avoidance area and the range from COL threshold value 3 to COL threshold value 4 is the valid area.
COLTHR2.COL4 [6:0]	6.5 to 832 (Condition: COLTHR2.COL3[6:0] + 1 < COLTHR2.COL4[6:0])	44H	Specify Point P in **Figure 26 - 39 Collision Detection Timing 1** and Point U in **Figure 26 - 40 Collision Detection Timing 2** within the available value range in 6.5 µs steps. The range from COL threshold value 3 to COL threshold value 4 is the valid area and the range from COL threshold value 4 to COL threshold value 5 is the avoidance area.
COLTHR3.COL5 [6:0]	0 to 825.5 (Condition: COLTHR2.COL4[6:0] + 1 < COLTHR3.COL5[6:0])	48H	Specify Point Q in **Figure 26 - 39 Collision Detection Timing 1** and Point V in **Figure 26 - 40 Collision Detection Timing 2** within the available value range in 6.5 µs steps. The range from COL threshold value 4 to COL threshold value 5 is the avoidance area and the range from COL threshold value 5 to COL threshold value 6 is the destroy area.
COLTHR3.COL6 [6:0]	6.5 to 832 (Condition: COLTHR3.COL5[6:0] + 1 < COLTHR3.COL6[6:0])	71H	Specify Point W in **Figure 26 - 40 Collision Detection Timing 2** within the available value range in 6.5 µs steps. The range from COL threshold value 5 to COL threshold value 6 is the destroy area and the range from COL threshold value 6 to COL threshold value 7 is the avoidance area.
COLTHR4.COL7 [7:0]	0 to 1657.5 (Condition: COLTHR3.COL6[6:0] + 1 < COLTHR4.COL7[7:0])	79H	Specify Point X in **Figure 26 - 40 Collision Detection Timing 2** within the available value range in 6.5 µs steps. The range from COL threshold value 6 to COL threshold value 7 is the avoidance area and the range from COL threshold value 7 to COL threshold value 8 is the valid area.
COLTHR4.COL8 [7:0]	6.5 to 1664 (Condition: COLTHR4.COL7[7:0] + 1 < COLTHR4.COL8[7:0])	88H	Specify Point Y in **Figure 26 - 40 Collision Detection Timing 2** within the available value range in 6.5 µs steps. The range from COL threshold value 7 to COL threshold value 8 is the valid area and the range from COL threshold value 8 to COL threshold value 9 is the avoidance area.
COLTHR5.COL9 [7:0]	0 to 1657.5 (Condition: COLTHR4.COL8[7:0] + 1 < COLTHR5.COL9[7:0])	8EH	Specify Point Z in **Figure 26 - 40 Collision Detection Timing 2** within the available value range in 6.5 µs steps. The range from COL threshold value 8 to COL threshold value 9 is the avoidance area and the range beginning with COL threshold value 9 is the destroy area.

26.3.4.3 Flow of the process on a collision occurrence

Figure 26 - 41 Flow of the Process on a Collision Occurrence shows the flow of collision recovery and collision avoidance on a collision occurrence.

Figure 26 - 41 Flow of the Process on a Collision Occurrence



Note 1. When starting the collision recovery, the multi-master transmitter forces the bus to be in active state within 450 µs before measuring the break time (T_{break}).

Note 2. Read the DALIRxD0 monitoring bit after waiting for the delay from DALITxD0 to DALIRxD0.

Note 3. To avoid collisions, it is strongly recommended that a multi-master transmitter starts its transmission at a random point between the minimum and maximum recovery time.

Note 4. Check the collision information based on the values in **Table 26 - 7 Collision Detection Information** to determine whether it is “Avoidance” or “Not avoidance”. If it is “Not avoidance”, go to a custom flow.

Note 5. When the CTR1.TE bit changes from 1 to 0, the STR1.BBF flag is valid until a stop condition is detected.

When the DALI module detects a collision during transmission, check the DALI status register 1 (STR1) and clear the flags that are set before starting the master transmission. See the collision recovery process or collision avoidance process in **Figure 26 - 41 Flow of the Process on a Collision Occurrence**. If you do not clear the collision detect flag (STR1.CDF), the collision information is not updated to new information and if the STR1.RDRF flag is 1, the DALI reception data registers 1H, 1L (RDR1H, RDR1L) also must be read.

When the DALI module detects a collision only during a reception, the processing shown in **Figure 26 - 41** is not required. Check the DALI status register 1 (STR1), clear the flags, and read the DALI reception data registers 1H, 1L (RDR1H, RDR1L) as needed.

26.3.4.4 Collision detection information

The DALI module saves the following information when a collision detection interrupt (INTCLD) occurs.

Table 26 - 7 Collision Detection Information

Collision Detection Information	Flag
State of the DALITxD0 pin on a collision occurrence	COLR1.TXDCV
DALIRxD0 state (rising or falling) on a collision occurrence	COLR1.RXDCEG
Collision factor (edge or not edge)	COLR1.CLDAF
Timing information on a collision occurrence	COLR1.CDTF1 and COLR1.CDTF2[3:0]

Caution The DALI module saves the collision detection information when a collision is detected.

The function to save the collision detection information is enabled when a collision occurs while the collision detection flag is not generated (STR1.CDF = 0). After the collision detection flag (STR1.CDF) is cleared, the next information is stored. Be sure to clear the collision detection flag (STR1.CDF).

Figure 26 - 42 shows an example of how to use the collision detection information when a collision occurred on an edge in the middle of a bit period (middle edge).

Figure 26 - 43 shows an example of how to use the collision detection information when a collision occurred on an edge at the bit period boundary (boundary edge).

Figure 26 - 44 shows an example of how to use the collision detection information when a collision occurred on a middle edge without a boundary edge.

Figure 26 - 42 Usage Example of Collision Detection Information 1

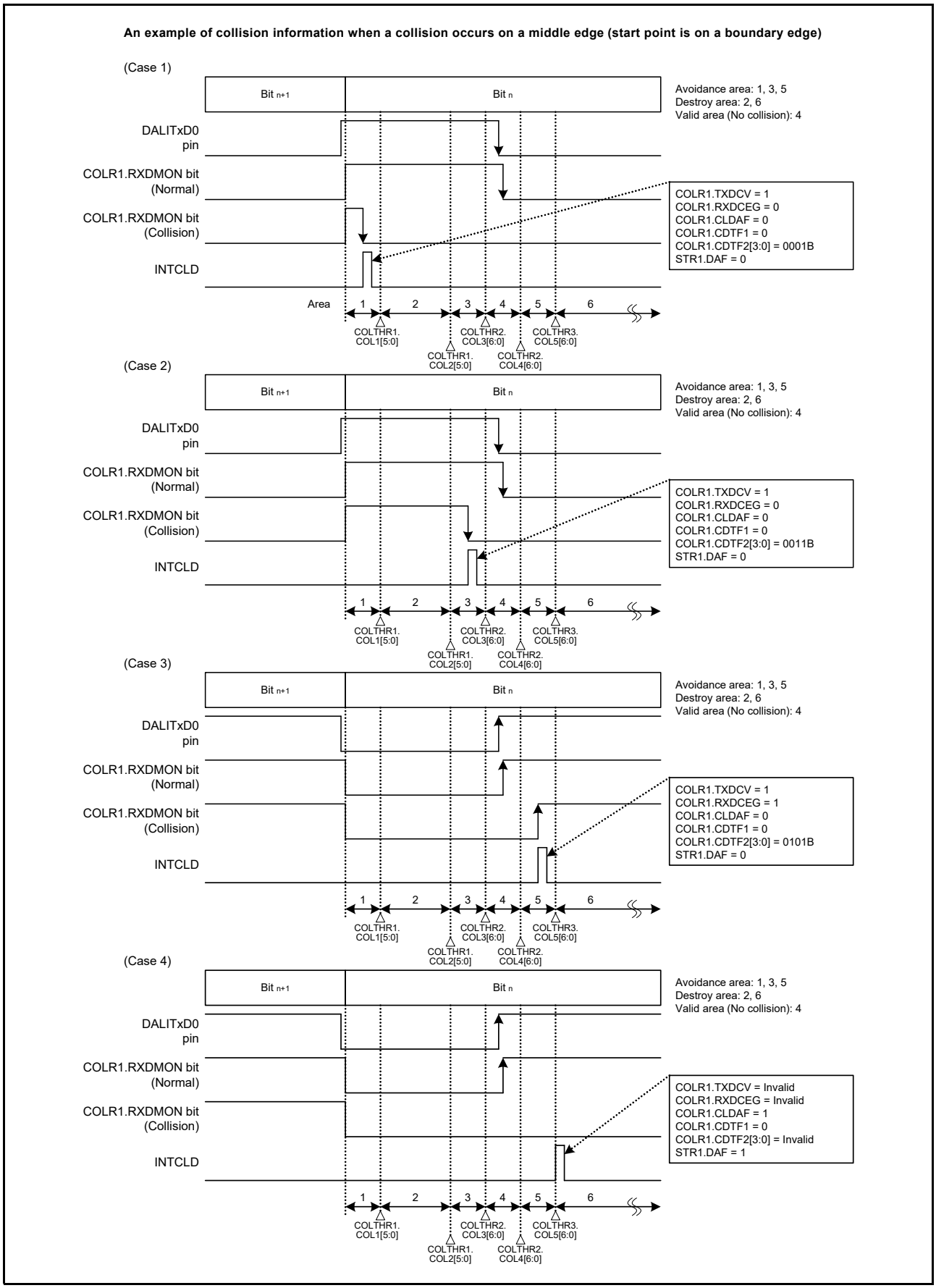


Figure 26 - 43 Usage Example of Collision Detection Information 2

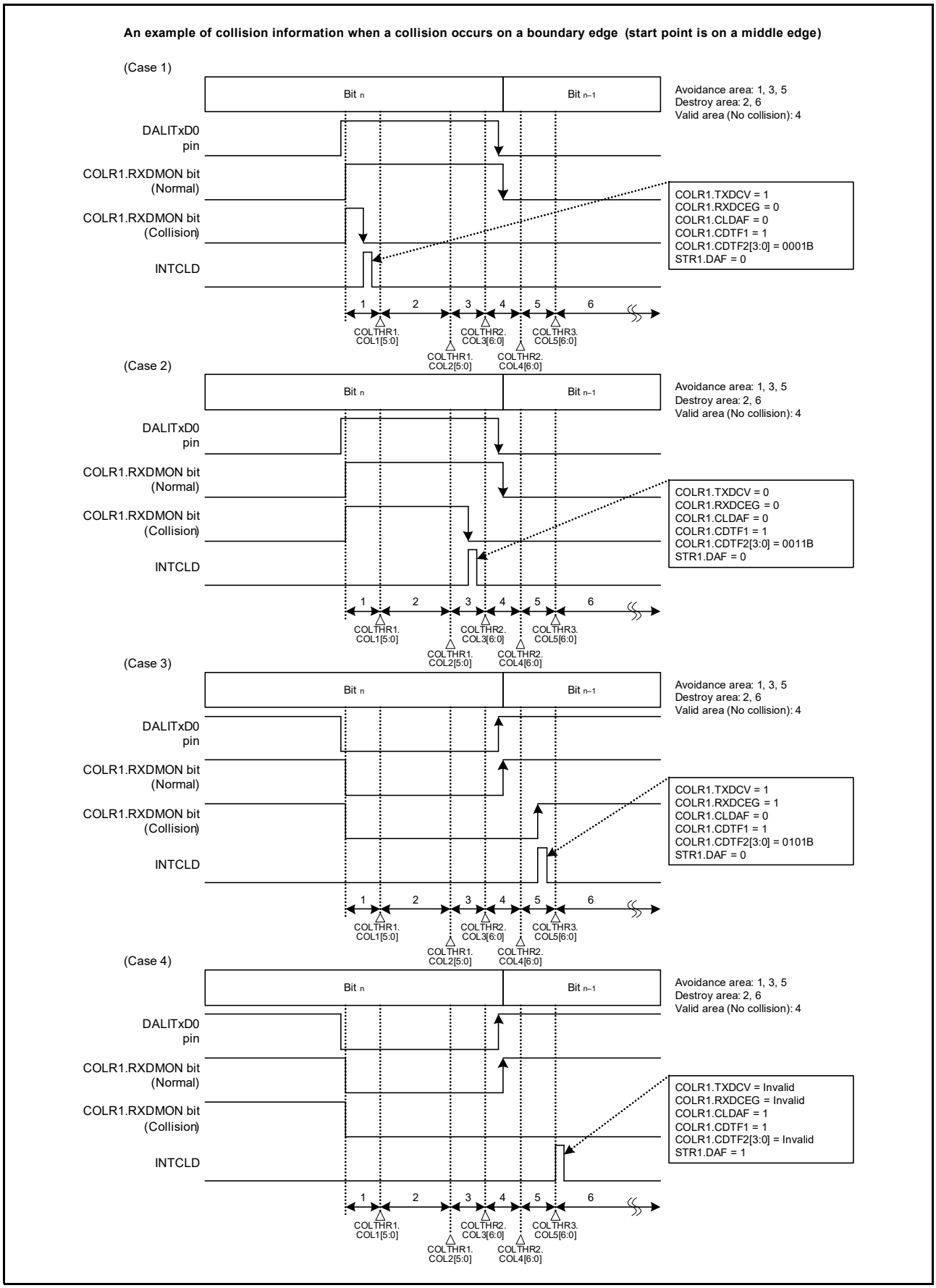
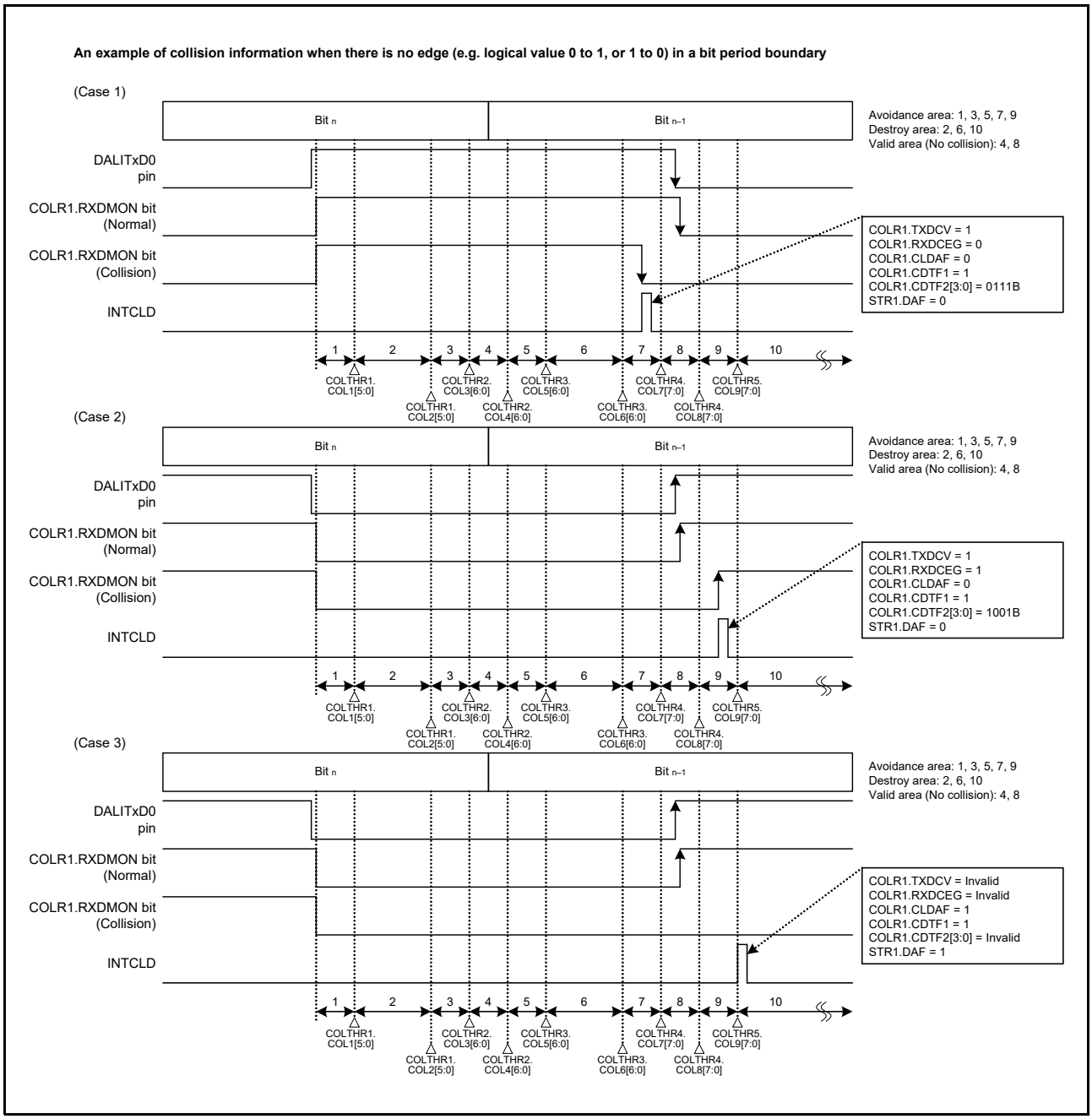


Figure 26 - 44 Usage Example of Collision Detection Information 3



26.3.5 Sampling timing of the DALIRxD0 input signal and bit length adjustment

The DALI module uses Manchester code to define bits. An edge always occurs in the middle of a 1-bit period (middle edge). The DALI module uses a middle edge to adjust the bit length and measure the sampling timing. 1/4-bit timing and 3/4-bit timing are defined before and after a middle edge and the value of the DALIRxD0 input signal after synchronization is sampled at these timings. The sampling value at 3/4-bit timing is used as the logical value.
If a middle edge occurs between 1/4- and 3/4-bit timing, the data bit is received. The waveform described in **26.3.5.1 Adjusting the acceptable range of the edges of DALIRxD0 input waveform** is used to detect an error of the bit length on a middle edge. An edge does not always occur on a boundary edge, so a boundary edge is not used as a reference.
The sampling points in the DALI module are defined in **Table 26 - 8 Sampling Points** and are not exactly 3/4-bit timing. The sampling points also vary depending on the values of the CNFR2.SGA and FTDC0.IST bits.

Table 26 - 8 Sampling Points

Definition	CNFR2.SGA = 0 and FTDC0.IST = 0	CNFR2.SGA = 1 and FTDC0.IST = 0	FTDC0.IST = 1, Regardless of the Setting of the CNFR2.SGA Bit
1/4-bit length 1/4-bit rate timing 1/4-bit sampling value 1/4-bit timing	Default value (1/4-bit length)	Default value – 16 cycles	Default value – 16 cycles
3/4-bit length 3/4-bit sampling value 3/4-bit timing	Default value (3/4-bit length + 1 cycle)	Default value + 25 cycles	Default value + 25 cycles

26.3.5.1 Adjusting the acceptable range of the edges of DALIRxD0 input waveform

The DALI module receives the high- or low-level period. To adjust the period, change the settings of the CNFR2.SGA and FTDC0.IST bits.

When the settings of both the CNFR2.SGA and FTDC0.IST bits are 0 (default):
High-/low-level period of Manchester code: 318.5 to 510.25 µs
When the settings of the CNFR2.SGA and FTDC0.IST bits are respectively 1 and 0:
High-/low-level period of Manchester code: 295.75 to 559 µs
When the setting of the FTDC0.IST bit is 1, regardless of the setting of the CNFR2.SGA bit:
High-/low-level period of Manchester code: 292.5 to 552.5 µs

26.3.6 Width adjustment for DALITxD0 output waveform

When a low- or high-level period of the DALITxD0 output waveform becomes shorter due to degradation of an external device such as a photocoupler, this function can adjust the low-level period of the DALITxD0 output waveform. The function calculates the reduction of the period due to degradation and enables the receiver to accept a waveform that is nearly 50% of duty.
The DALI module adjusts the low-level width of the DALITxD0 output waveform by setting the CNFR2.TXWE bit to 1. If an external device is not degraded and the DALITxD0 output waveform appears correct, disable this function (CNFR2.TXWE = 0) and use the TXWR1 register with the default value.

The DALITxD0 output waveform is adjusted as follows:
When the setting of the CNFR2.TXWE bit is 0:

- Low-level width: 416 µs (Reset value = 3FH)
- High-level width: 416 µs (Reset value = 3FH)

When the setting of the CNFR2.TXWE bit is 1:

- Low-level width:
  (Setting value of the TXWR1.TXLW[6:0] bit + 1) × (two cycles of the operating clock for the DALI module, 6.5 µs typ.)
- High-level width:
  832 µs – {(Setting value of the TXWR1.TXLW[6:0] bit + 1) × (two cycles of the operating clock for the DALI module, 6.5 µs typ.)}

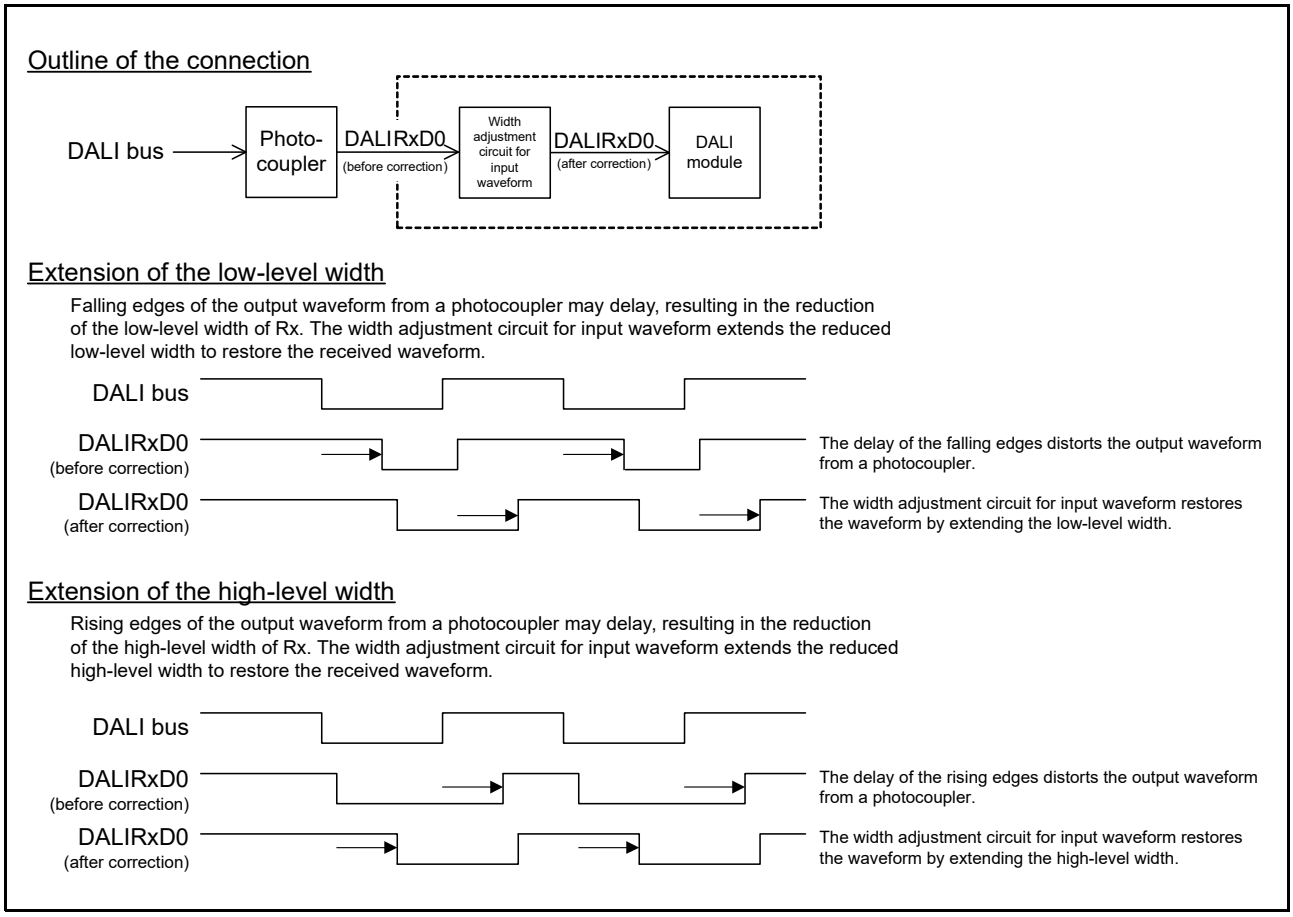
Values available for setting the TXWR1.TXLW[6:0] bits are from 20H (= 214.5 µs) to 5EH (= 617.5 µs).

26.3.7 Width adjustment for DALIRxD0 input waveform

26.3.7.1 Outline of the function

Width adjustment for DALIRxD0 input waveform is used to take measures against reduction of a low- or high-level period of the DALIRxD0 input waveform due to characteristics of an external device such as a photocoupler. This function corrects the widths of the input waveform by extending the low- or high-level width, making the duty factor of the received waveform nearer to 50%. **Figure 26 - 45** shows the outline of the operation.

Figure 26 - 45 Outline of the Operation of Width Adjustment for DALIRxD0 Input Waveform



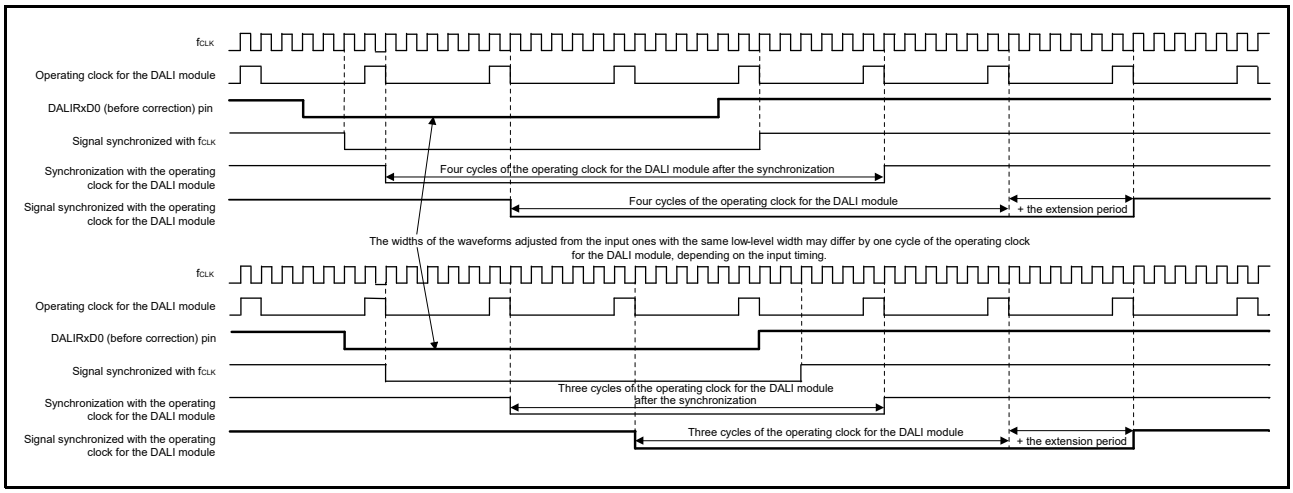
When this function is to be used, start DALI communications after setting the DALIRxD0 waveform adjustment register 1 (RXWR1) while the settings of both the CTR1.RE and CTR1.TE bits are 0.

The following shows the extension width set by the RXWR1.RSLEN[5:0] bits.
Extension width = (Value of RXWR1.RSLEN[5:0] bits + 1) × (one cycle of the operating clock for the DALI module, 3.25 µs typ.)

26.3.7.2 Operation

Width adjustment for DALIRxD0 input waveform synchronizes the DALIRxD0 input signal with fCLK, and then with the operating clock for the DALI module (3.25 µs typ.). The waveform of which the width is adjusted is therefore the waveform of the signal after being synchronized. The adjusted widths of the waveform consequently fluctuates by one cycle of the operating clock for the DALI module, depending on the timing at which the input signal is received. In addition, the synchronization and extension cause the delay of the adjusted waveform from the original one.

Figure 26 - 46 Example of the Synchronization and Extension by Width Adjustment for DALIRxD0 Input Waveform



If resuming communications from the initial setting is required due to the occurrence of any problem during communications, the initial setting can be performed by setting the RXWR1.RSEN bit to 0, even if the adjustment has been stopped while the width adjustment circuit for the DALIRxD0 input waveform was internally operating. Note that the correction value set to the RXWR1.RSLEN[5:0] bits must be the necessary minimum. For example, when multiple communication partners require different extension periods, set the correction value for the minimum extension period among them. The extension period exceeding the necessary minimum may cause malfunctions of the width adjustment circuit for the DALIRxD0 input waveform, leading abnormal DALI communications.

26.3.8 Control by setting the extended mode selection bit (CNFR1.EXM)

The communications operation of the DALI module depends on the setting of the extended mode selection bit (CNFR1.EXM).

26.3.8.1 Reception

The following factors of the reception only depends on the setting of the CNFR1.EXM bit; the INTRD signal output, the valid bit width of the STR2.RDBL[8:0] bits, and the set condition for the STR1.LFRF flag. The relationship among them is shown in **Table 26 - 9** and **Figure 26 - 47**.

Table 26 - 9 Factors of Reception Depending on the Setting of the CNFR1.EXM Bit

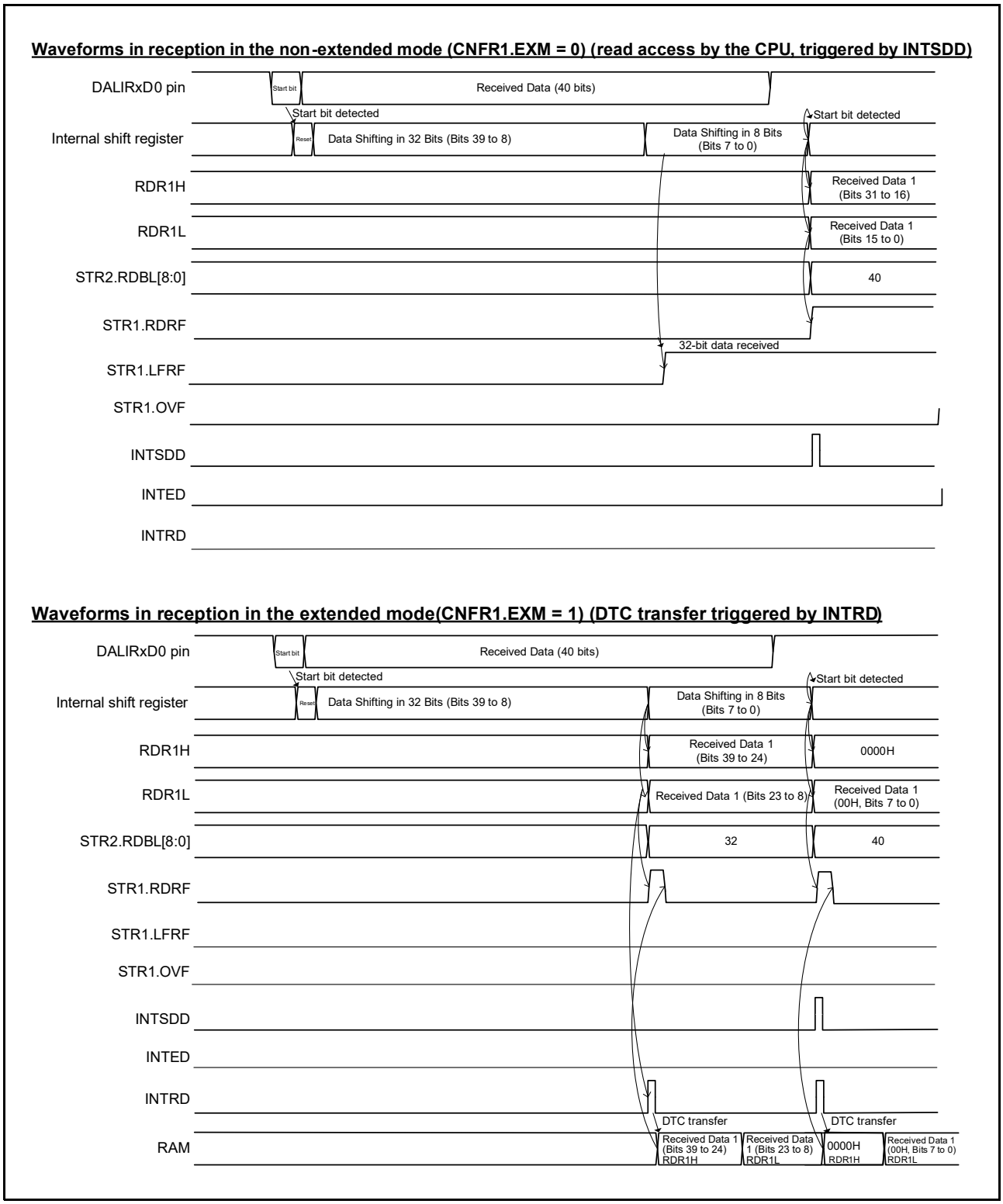
Setting of the CNFR1.EXM Bit	0 (Non-extended Mode)	1 (Extended Mode)
INTRD output	Fixed to the low level	Output while the setting of the CTR1.RDIE bit is 1
Valid bit width of the STR2.RDBL[8:0] bits	The STR2.RDBL[5:0] bits are valid. The STR2.RDBL[8:6] bits are fixed to 000B.	The STR2.RDBL[8:0] bits are valid.
Set condition for the STR1.LFRF flag	When data bit [33] is received	When data bit [257] is received

Note the following points on the reception in the extended mode.

- Output timing of the INTRD signal upon completion of reception
  The INTRD signal is output when received data are stored in the DALI reception data registers 1H, 1L (RDR1H, RDR1L). The storage is triggered by either of every reception of 32 bits of the data or the detection of a stop condition.
  Note that if the received data length is a multiple of 32, the data are not stored in the reception data registers when a stop condition is detected, because the storage of the data has been completed upon reception of 32 bits of the data. Thus, the timing of the INTRD signal output upon completion of reception depends on the received data length.
  Example: The number of times and the timing of the INTRD signal output
  For data length smaller than 32 bits: The signal is output once when a stop condition is detected.
  For data length of 32 bits: The signal is output once when 32 bits of the data are received, while it is not output when a stop condition is detected.
  For data length greater than 32 bits and smaller than 62 bits: The signal is output twice, when 32 bits of the data are received and when a stop condition is detected.

- Received data may include invalid bits
  When a frame with data length greater than 32 bits and also not a multiple of 32 is received, such as 40-bit length of the received data shown in **Figure 26 - 47**, the data stored in the DALI reception data registers 1H, 1L (RDR1H, RDR1L) include invalid bits. This is because the registers hold received data in order from the LSB. Therefore, for frames with data length greater than 32 bits, extract and use valid bits by checking the received data length in the STR2.RDBL[8:0] bits.

Figure 26 - 47 Waveforms in Reception Depending on the Setting of the CNFR1.EXM Bit



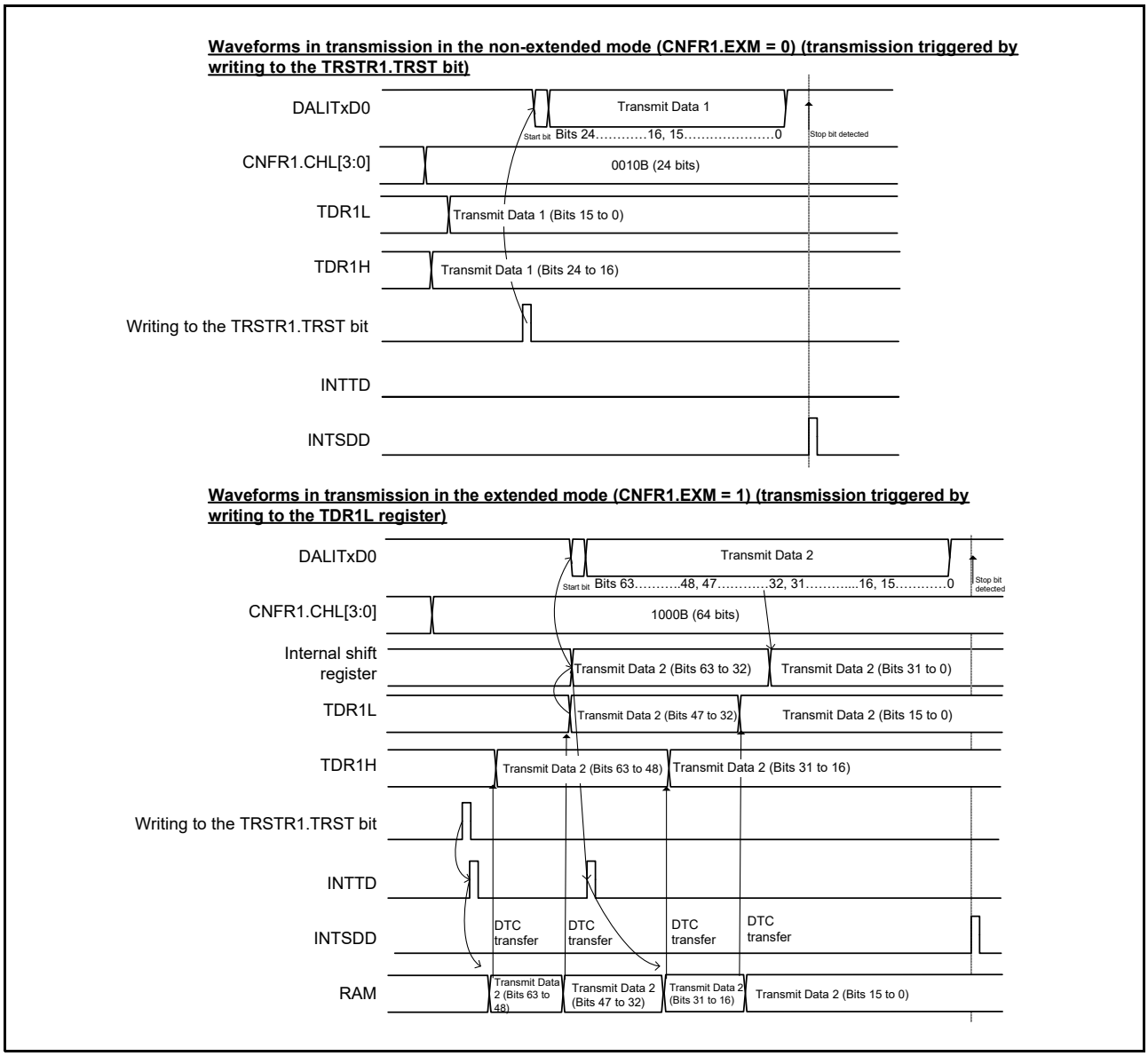
26.3.8.2 Transmission

The following factors of the transmission only depends on the setting of the CNFR1.EXM bit; the INTTD signal output, the character length, the transmission start trigger, and the underrun error detection function. The relationship among them is shown in **Table 26 - 10** and **Figure 26 - 48**.

Table 26 - 10 Factors of Transmission Depending on the Setting of the CNFR1.EXM Bit

Setting of the CNFR1.EXM Bit	0 (Non-extended Mode)	1 (Extended Mode)
INTTD output	Output fixed to the low level	Output while the setting of the CTR1.TDIE bit is 1
Character length	The CNFR1.CHL[2:0] bits are used. The setting of the CNFR1.CHL[3] bit is invalid.	The CNFR1.CHL[3:0] bits are used.
Transmission start trigger	Writing 1 to the TRSTR1.TRST bit	Writing to the TDR1L register
Underrun error detection function	Invalid	Valid

Figure 26 - 48 Waveforms in Transmission Depending on the Setting of the CNFR1.EXM Bit

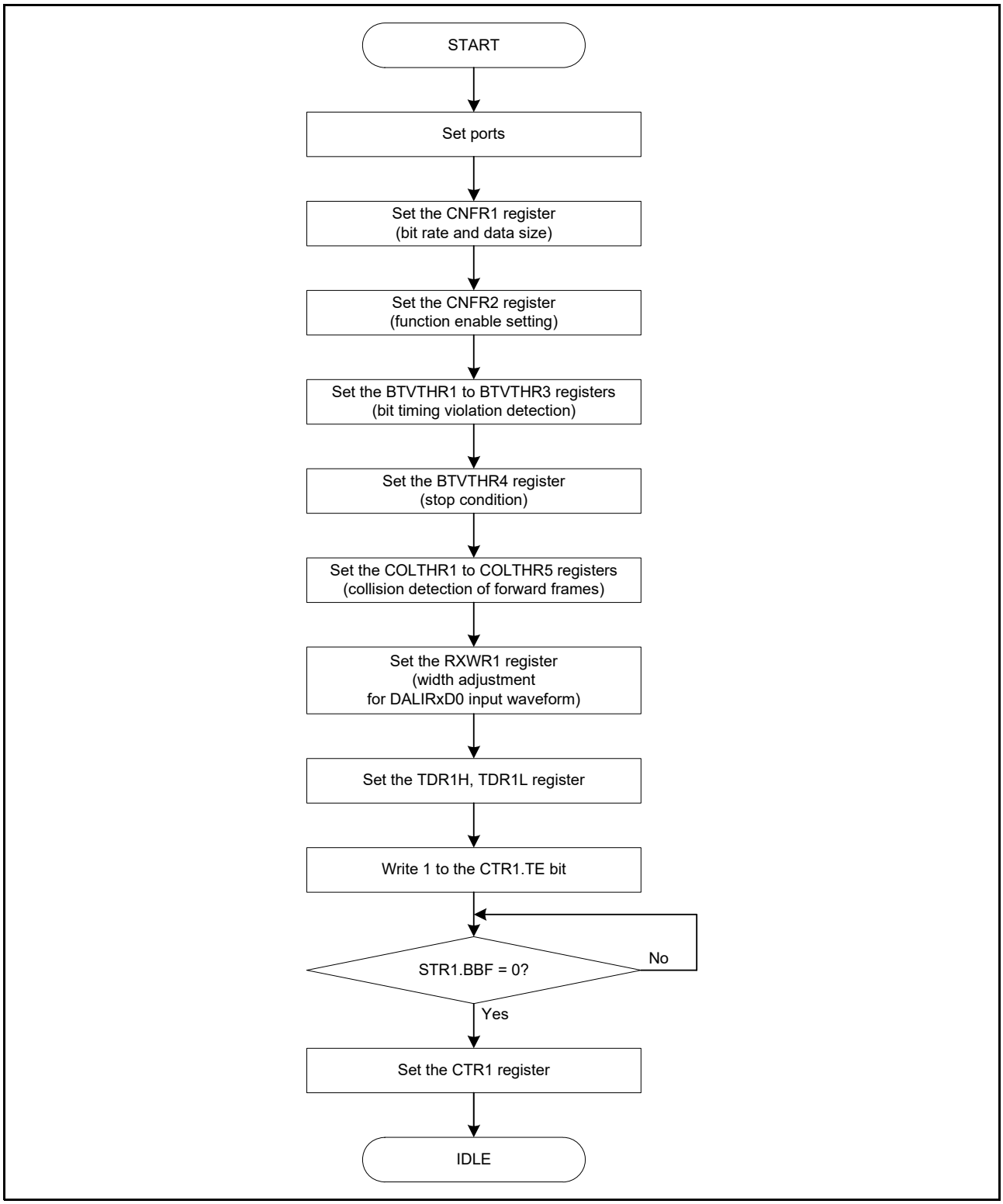


26.4 Operation

26.4.1 Initial setting

Before transmitting and receiving data, follow the flow in **Figure 26 - 49 Initial Setting Flow** to set each register.

Figure 26 - 49 Initial Setting Flow



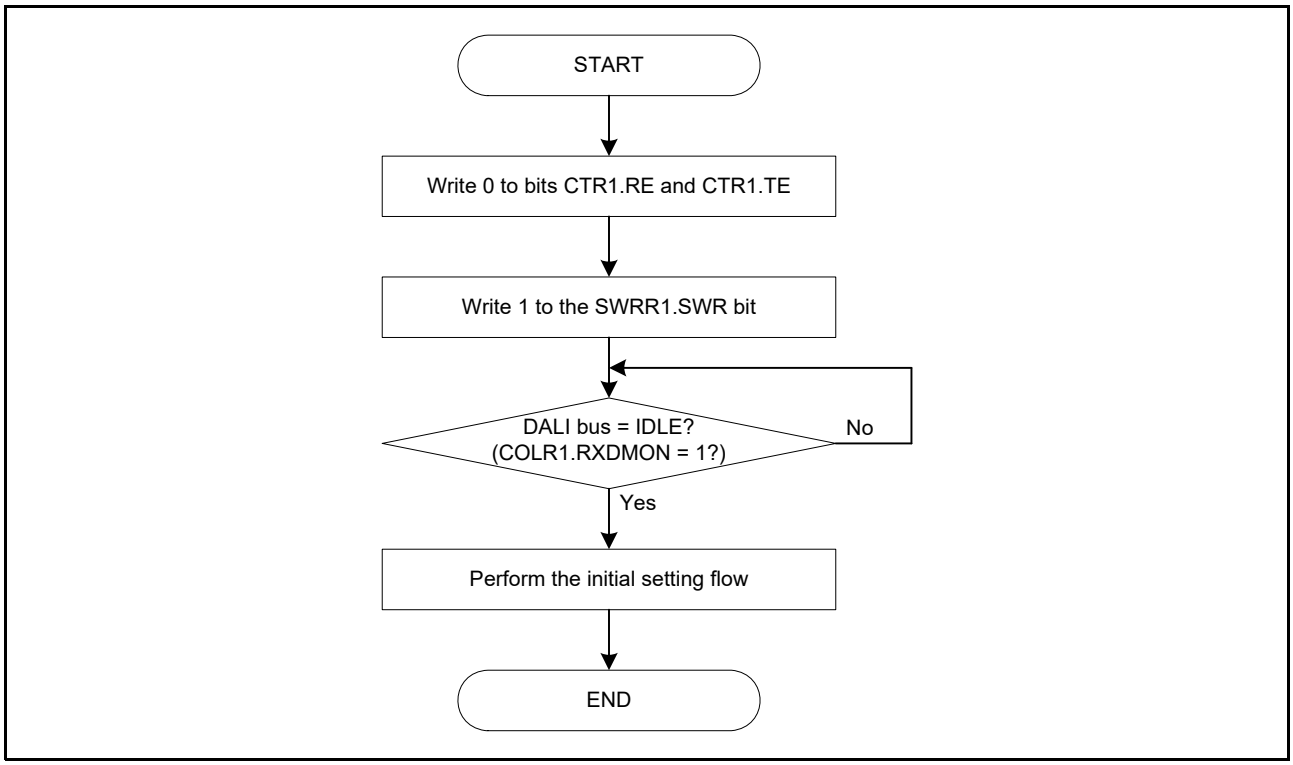
26.4.2 Software reset

When a reset is required for data registers, flag bits, internal registers, and internal counters, follow the flow in **Figure 26 - 50 Software Reset Flow**. Output pins of the DALI module are also reset. For target registers and output pins of the DALI module for software reset, see **26.2.23 DALI software reset register 1 (SWRR1)**.

26.4.2.1 Use example of software reset

1. To halt a transmission and reception
   Perform the flow in **Figure 26 - 50 Software Reset Flow** to halt a transmission and a reception.
2. To stop a reception during bus power down
   The DALI module continues to receive frames during a bus power down. To stop reception during a bus power down, perform the flow in **Figure 26 - 50 Software Reset Flow**. That is, after a DALI bus returns from the bus power down state, set the CTR1.TE and CTR1.RE bits to 1, then resume communications.
3. To stop a reception when receiving data greater than expected
   While the DALI module receives data greater than expected, INTSDD is not output. To stop the reception, confirm the STR1.LFRF flag is 1 and perform the flow in **Figure 26 - 50 Software Reset Flow**. That is, set the CTR1.RE bit to 0, and after a DALI bus restores, set the CTR1.RE bit to 1, then restart reception.

Figure 26 - 50 Software Reset Flow

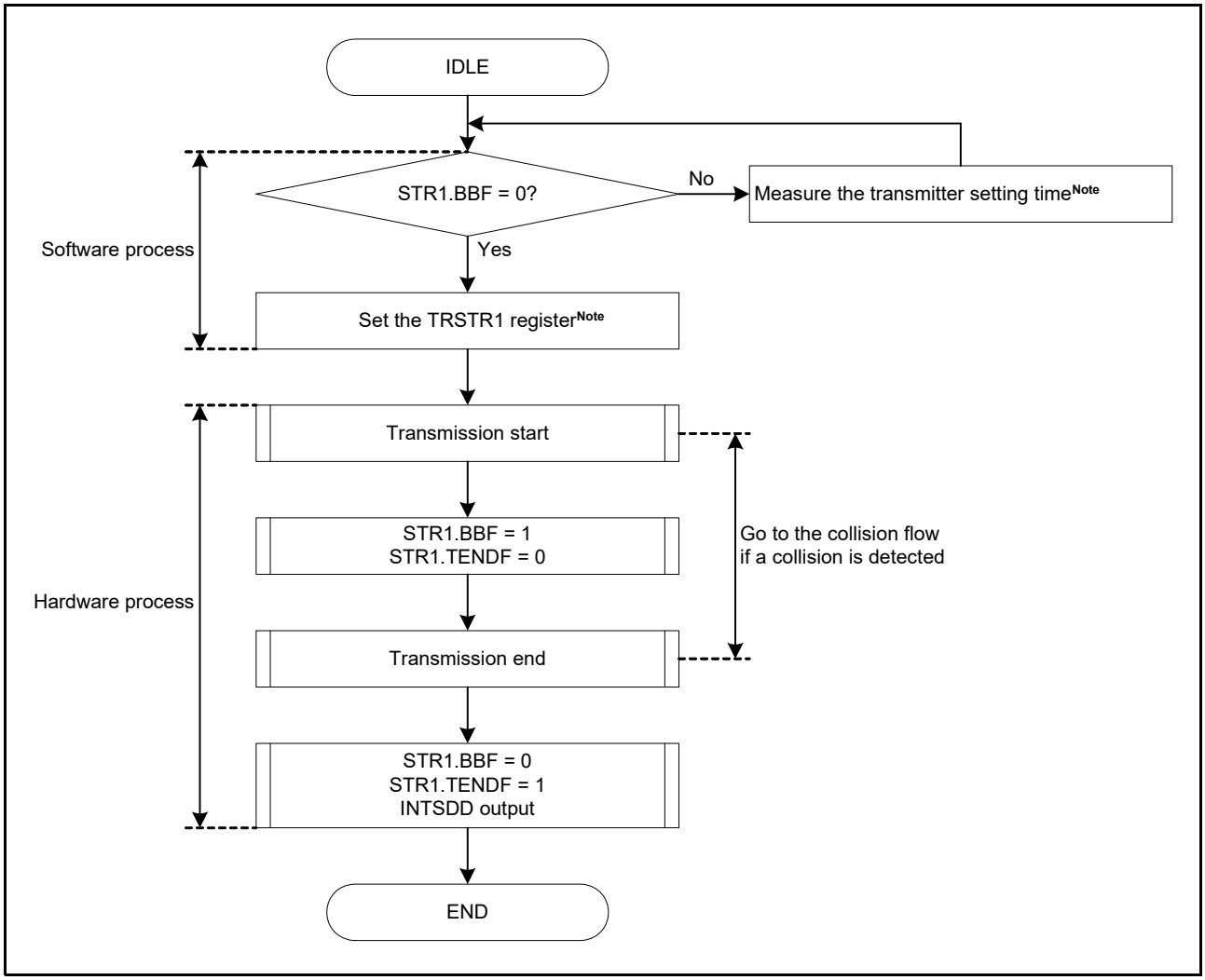


26.4.3 Transmission

26.4.3.1 Transmission flow in the non-extended mode

Figure 26 - 51 shows a transmission flow in DALI communications in the non-extended mode.

Figure 26 - 51 Transmission Flow in the Non-extended Mode (CNFR1.EXM = 0)

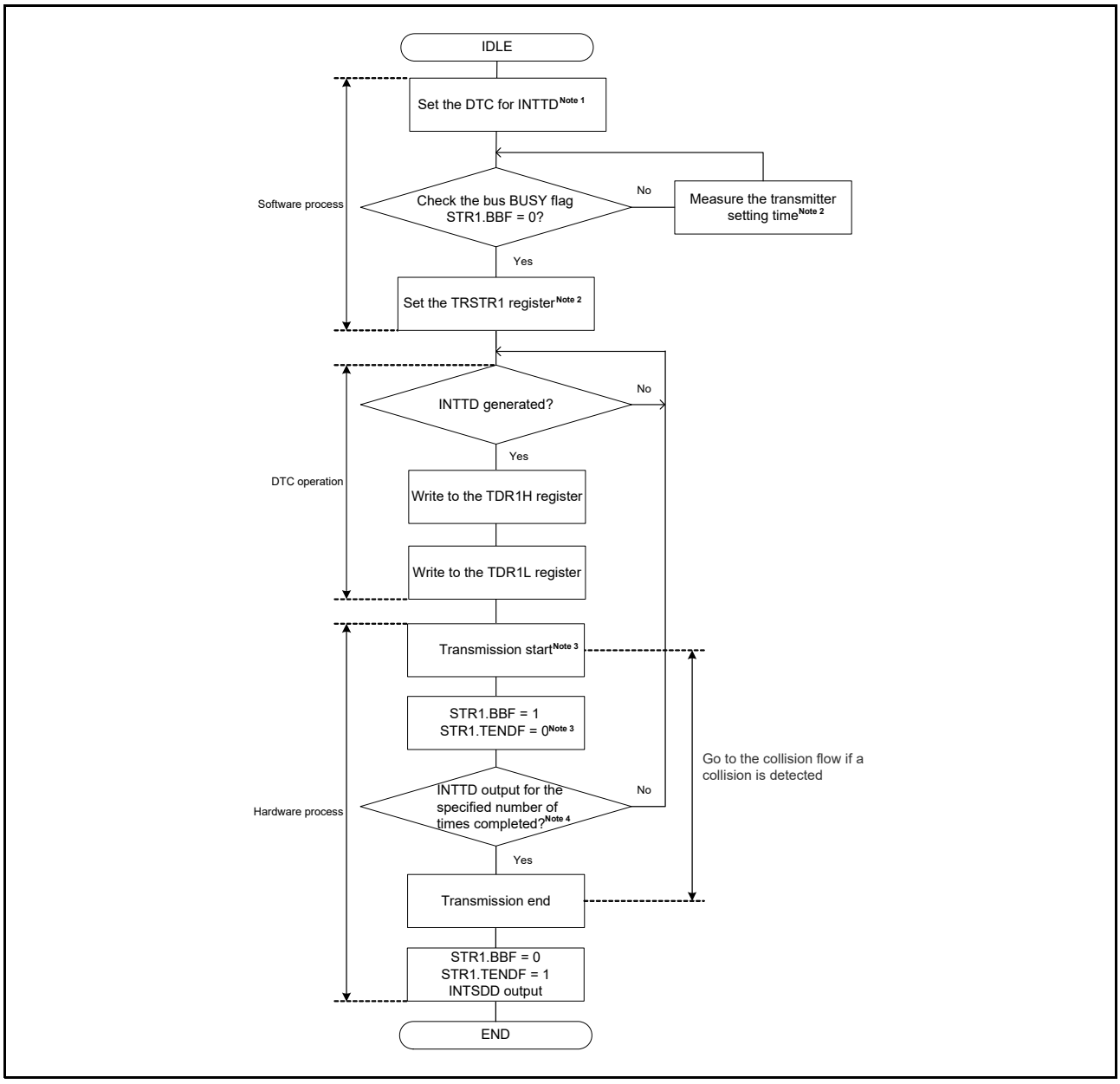


Note The transmitter setting time is defined in IEC 62386-101 Edition 2.0/2.1, which is not measured in the DALI module. When writing to the TRSTR1 register, follow the defined setting time.

26.4.3.2 Transmission flow in the extended mode

Figures 26 - 52 and **26 - 53** show transmission flows in DALI communications in the extended mode, using DTC transfer and by CPU processing, respectively. **Table 26 - 11** lists a setting example of the DTC.

Figure 26 - 52 Transmission Flow Using DTC Transfer



Note 1. For setting examples of the DTC, see **27.3.3 Vector table**.

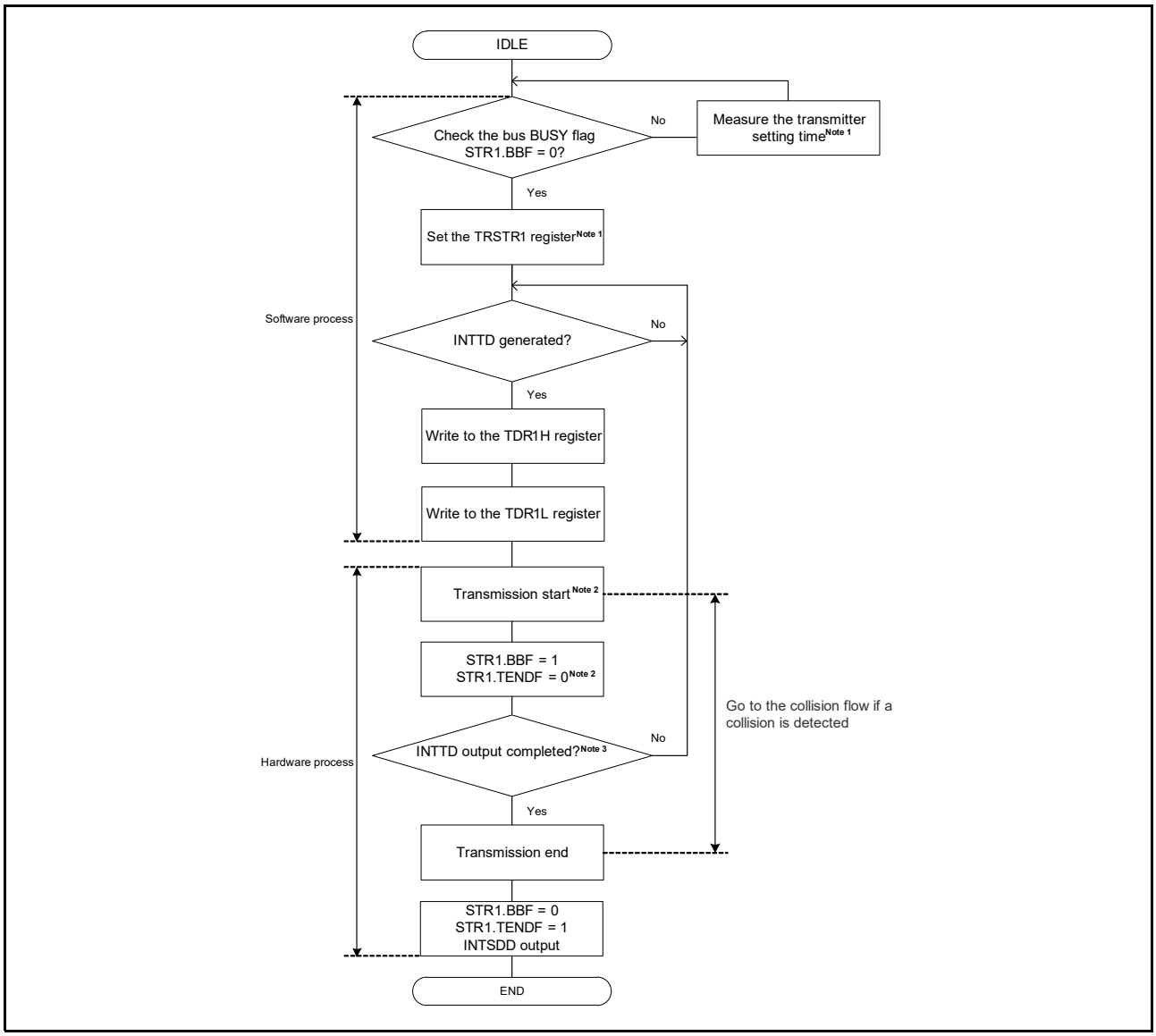
Note 2. The transmit time is defined in IEC 62386-101 Edition 2.0/2.1, which is not judged in this LSI. When setting the TRSTR1 register, follow the setting time defined in the standard.

Note 3. This step only applies to the first transmission.

Note 4. In the extended mode (CNFR1.EXM = 1), the number of times of the INTTD interrupt output depends on the setting of the CNFR1.CHL[3:0] bits. For details, see **26.4.5 Outputting an interrupt**.

Caution In the initial setting flow, factors of the control registers such as clock select, transmit data length, and transmit enabling, must be set appropriately. If such factors have been changed, initial setting must be implemented again.

Figure 26 - 53 Transmission Flow by CPU Processing



Note 1. The transmit time is defined in IEC 62386-101 Edition 2.0/2.1, which is not judged automatically in this LSI. When setting the TRSTR1 register, follow the setting time defined in the standard.

Note 2. This step only applies to the first transmission.

Note 3. In the extended mode (CNFR1.EXM = 1), the number of times of the INTTD interrupt output depends on the setting of the CNFR1.CHL[3:0] bits. For details, see **26.4.5 Outputting an interrupt**.

Caution **In the initial setting flow, factors of the control registers such as clock select, transmit data length, and transmit enabling, must be set appropriately. If such factors have been changed, initial setting must be implemented again.**

26.4.3.3 Setting example of the DTC to write transmit data

In the extended mode (CNFR1.EXM = 1), to activate the DTC using the write request interrupt for transmit data (INTTD) and transfer transmit data to the TDR1H and TDR1L registers, the DTC must be set appropriately in advance.

Table 26 - 11 lists a setting example of the DTC to write transmit data.

Table 26 - 11 Setting Example of the DTC to Write Transmit Data

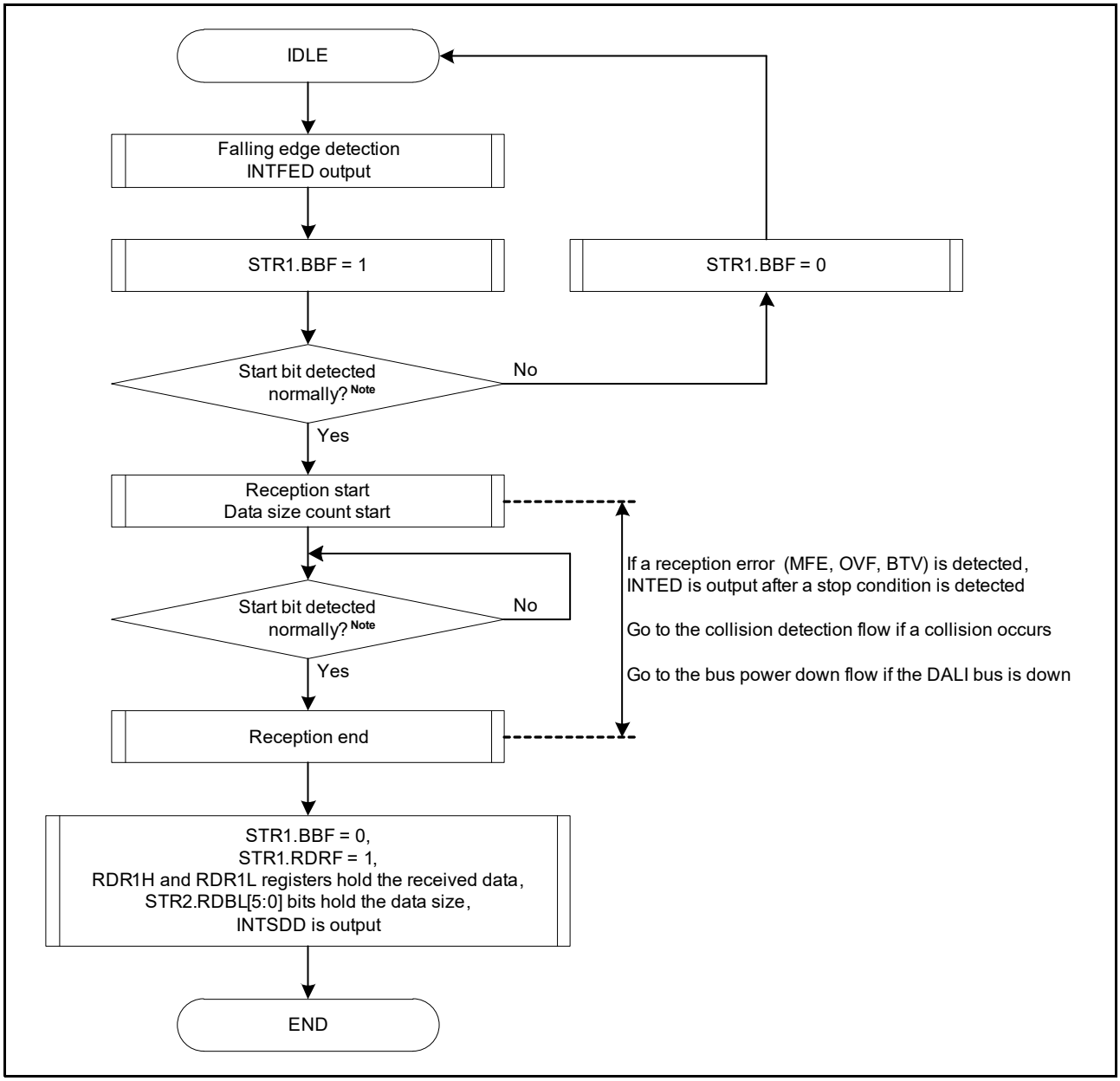
Registers in the DTC		Transmit Data Length			
		≤ 32 Bits	64 Bits	128 Bits	256 Bits
DTCCRj	SZ	1	1	1	1
	RPTINT	0	0	0	0
	CHNE	0	0	0	0
	DAMOD	0	0	0	0
	SAMOD	1	1	1	1
	RPTSEL	0	0	0	0
	MODE	0	0	0	0
DTBLSj		02H	02H	02H	02H
DTCCTj		01H	02H	04H	08H
DTSARj		RAM address	RAM address	RAM address	RAM address
DTDARj		Address of the TDR1H register	Address of the TDR1H register	Address of the TDR1H register	Address of the TDR1H register

26.4.4 Reception

26.4.4.1 Reception flow in the non-extended mode

Figure 26 - 54 shows a reception flow in DALI communications in the non-extended mode.

Figure 26 - 54 Reception Flow in the Non-extended Mode (CNFR1.EXM = 0)

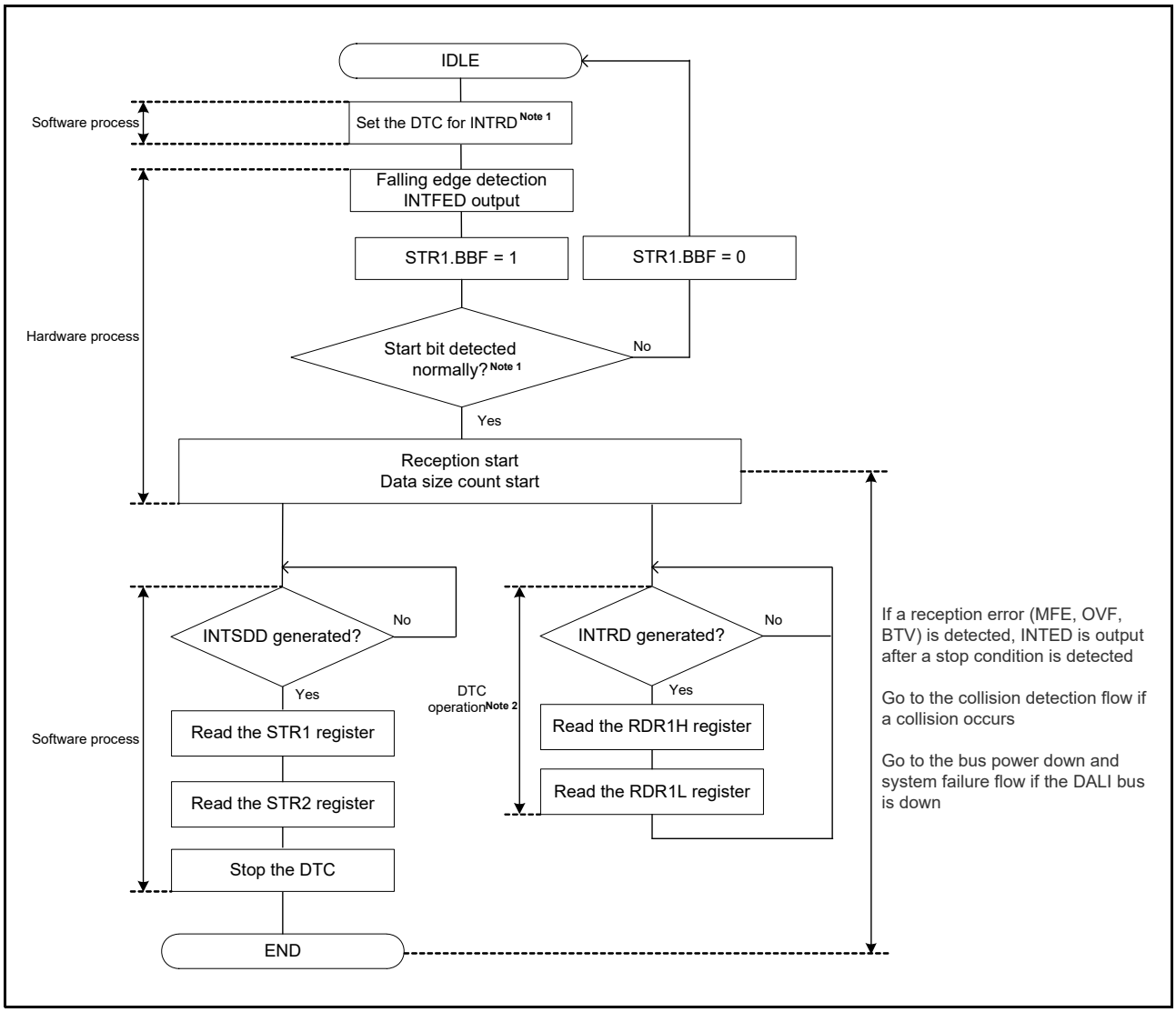


Note See **26.4.4.4 Detecting a start bit**.

26.4.4.2 Reception flow in the extended mode

Figures 26 - 55 and **26 - 56** show reception flows in DALI communications in the extended mode, using DTC transfer and by CPU processing, respectively. **Table 26 - 12** lists a setting example of the DTC.

Figure 26 - 55 Reception Flow Using DTC Transfer

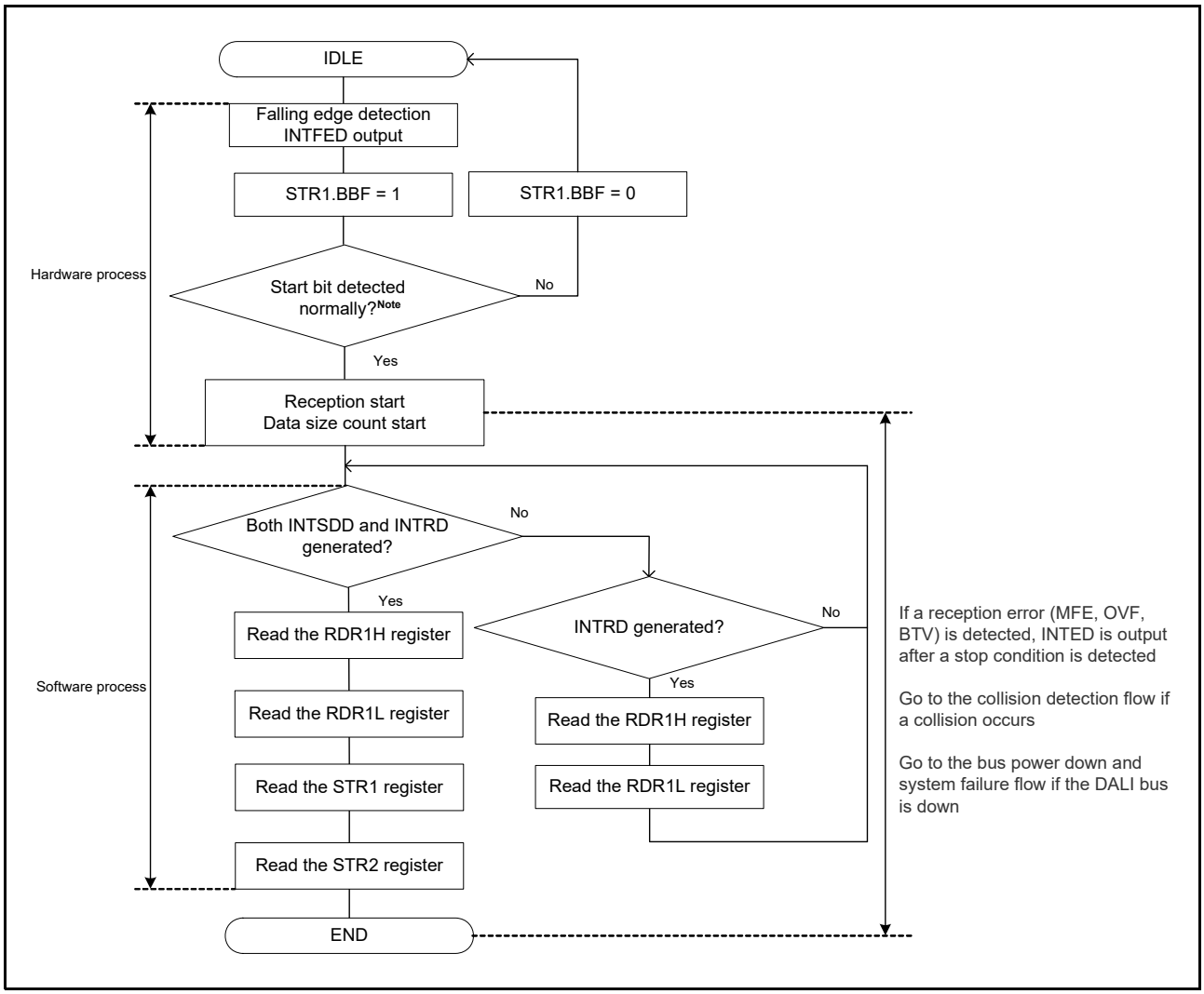


Note 1. See **26.4.3.3 Setting example of the DTC to write transmit data**.

Note 2. For setting examples of the DTC, see **27.4.3 Repeat mode**.

Caution **In the initial setting flow, factors of the control registers such as clock select, transmit data length, and transmit enabling, must be set appropriately. If such factors have been changed, initial setting must be implemented again.**

Figure 26 - 56 Reception Flow by CPU Processing



Note See **26.4.4.4 Detecting a start bit**.

Caution **In the initial setting flow, factors of the control registers such as clock select, transmit data length, and transmit enabling, must be set appropriately. If such factors have been changed, initial setting must be implemented again.**

26.4.4.3 Setting example of the DTC to read receive data

To activate the DTC using the read request interrupt for receive data (INTRD) and transfer receive data from the RDR1H and RDR1L registers, the DTC must be set appropriately in advance.
Assuming that receive data length is unknown until completion of reception, the DTC must be set anticipating that data with maximum receive data length may be transferred. When a frame shorter than the maximum receive data length is received, stop the DTC transfer in accordance with the stop condition detection interrupt (INTSDD).
Table 26 - 12 lists a setting example of the DTC for INTRD.

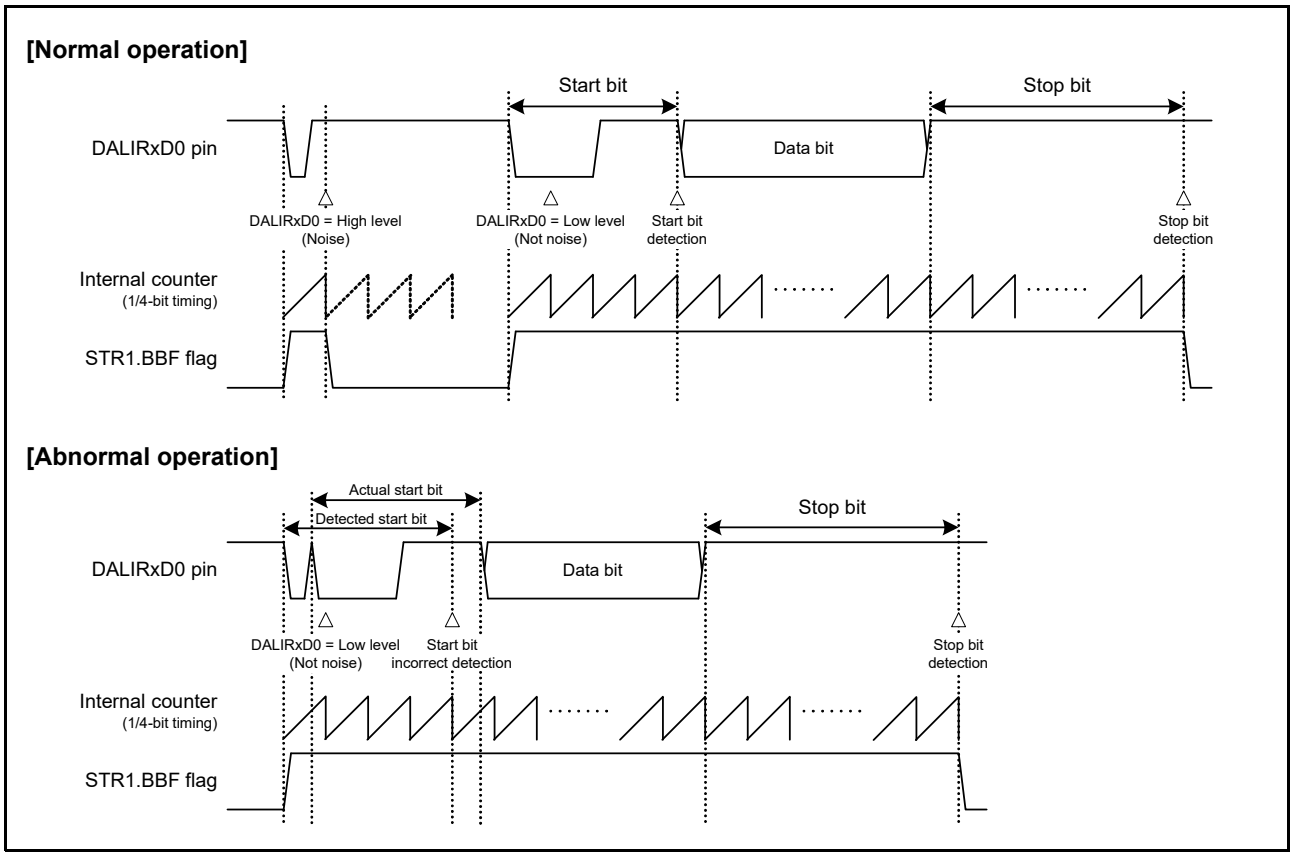
Table 26 - 12 Setting Example of the DTC for INTRD

Registers in the DTC		Receive Data Length (max.)
		256 Bits
DTCCRj	SZ	1
	RPTINT	0
	CHNE	0
	DAMOD	1
	SAMOD	0
	RPTSEL	0
	MODE	0
DTBLSj		02H
DTCCTj		08H
DTSARj		RAM address
DTDARj		Address of the RDR1H register

26.4.4.4 Detecting a start bit

The DALI module has a function that can prevent from incorrectly detecting a start bit due to noise.
When a falling edge is input to the DALIRxD0 pin, a receive operation starts. The STR1.BBF flag is then set. When a high level on the DALIRxD0 pin is detected at the 1/4-bit timing, the falling edge determines that it was caused by noise. The STR1.BBF flag is then cleared and the internal state returns to IDLE.
This function removes noise generated during the period from falling edge detection through to 1/4-bit timing. However, if a start bit is generated during this period, the values of each bit are sampled at a maximum of 1/4-bit length earlier (see **Figure 26 - 57 Detecting a Start Bit**). In that case, a Manchester framing error, bit timing violation, or a frame size violation might be the cause.

Figure 26 - 57 Detecting a Start Bit



26.4.5 Outputting an interrupt

26.4.5.1 INTFED, INTSDD, INTED

When INTSDD or INTED is generated, check the related flags and the received data.
INTSDD and INTFED are used for measurement with an external timer of the DALI module.

Figure 26 - 58 Procedure for Software Processing on the Generation of INTFED, INTSDD, or INTED (1/2)

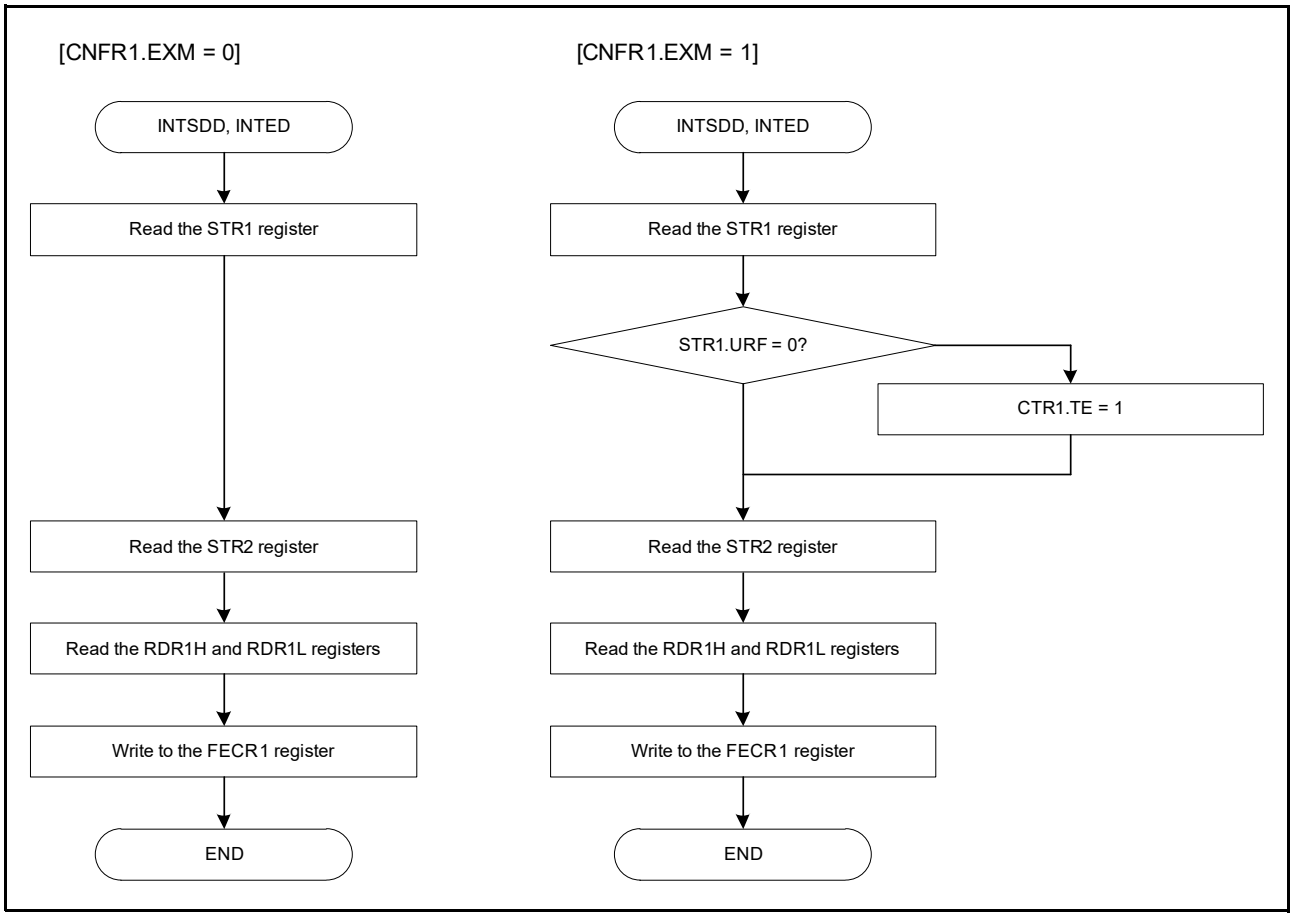
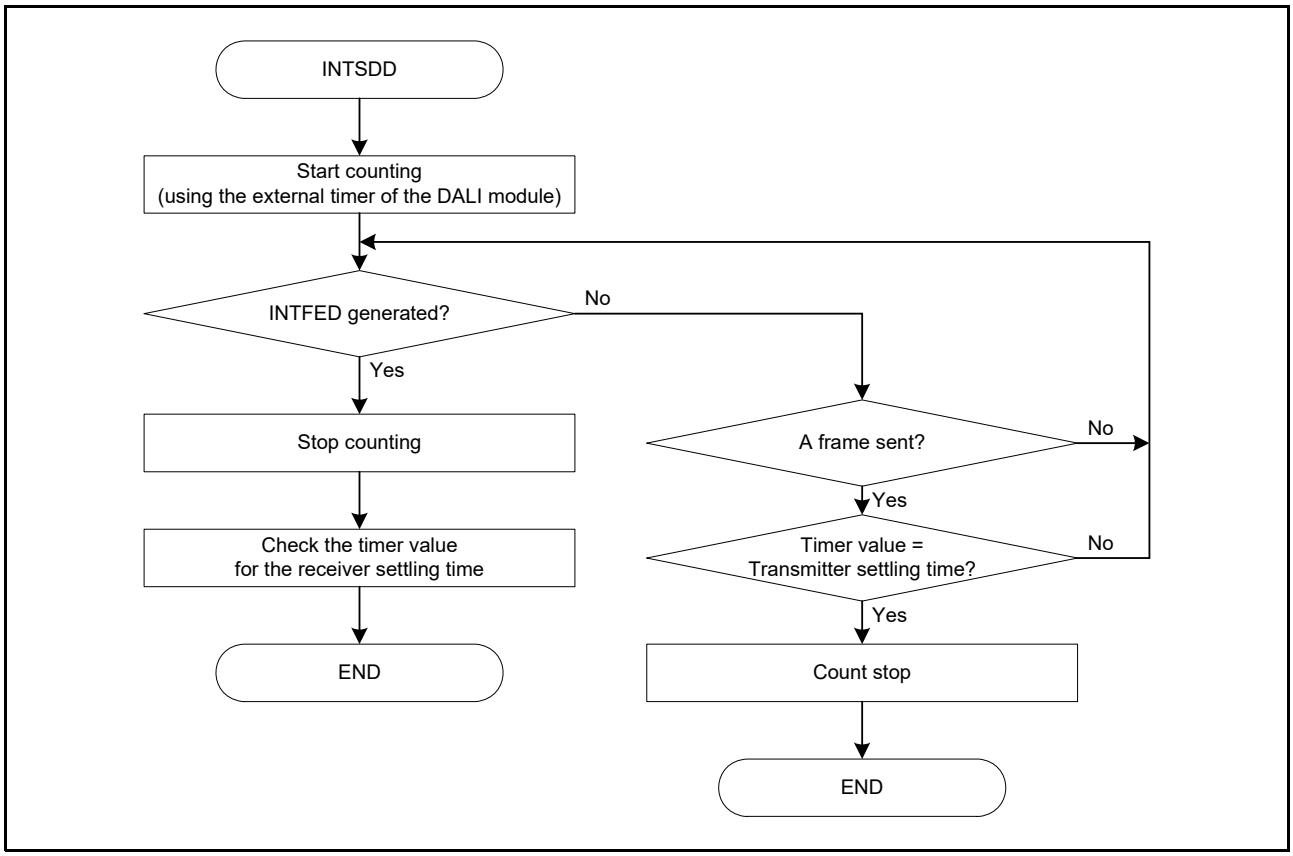


Figure 26 - 58 Procedure for Software Processing on the Generation of INTFED, INTSDD, or INTED (2/2)



26.4.5.2 INTCLD

See **Figure 26 - 41 Flow of the Process on a Collision Occurrence**.

26.4.5.3 INTBPD

See **Figure 26 - 37 Flow of Bus Power Down and System Failure**.

26.4.5.4 INTRD

See **Figure 26 - 55 Reception Flow Using DTC Transfer** and **Figure 26 - 56 Reception Flow by CPU Processing**.

26.4.5.5 INTTD

See **Figure 26 - 52 Transmission Flow Using DTC Transfer** and **Figure 26 - 53 Transmission Flow by CPU Processing**.

26.5 Usage Notes

26.5.1 Erroneous recognition of receive data length

When a high level of MFE continues for the period of the BTV threshold value 6 during data reception, the DALI module recognizes that a stop condition is detected but it does not detect MFE. That is, INTSDD is output but INTED is not output and STR1.MFEF flag is not set. When the stop condition is defined as 3-bit length, for example, if bits [10:8] of 24-bit data are driven high, the DALI module recognizes bits [7:0] as 8-bit data, bits [10:8] as the stop condition, bit [11] as the start bit of the next frame, and bits [24:12] as data of the next frame. In this case, the bits are recognized as the second frame, and bits [24:11] are detected as invalid data by the frame size violation. However, the data that is recognized as the first frame (bits [7:0]) is determined as normal backward frame reception. When 17 bits of data (bits [16:0]) are received and the last bit to be received, bit [0], has an MFE such that it is fixed to the high level, the DALI module recognizes that bit as a stop condition, so it fails to detect the MFE. The DALI module therefore treats the data as 16 bits of correctly received data, resulting in erroneous recognition of the data length. Thus, in such cases, higher-level communications processing is required to judge whether or not the data have in fact been correctly received. If the CTR1.RE bit is changed from 0 to 1 while another DALI device is transmitting a frame, a data reception starts during the frame transmission. Set the CTR1.RE bit while the DALI bus is idle. Verify the STR1.BBF flag or the COLR1.RXDMON bit to determine if the DALI bus is idle.

26.5.2 Receiving data greater than expected

The DALI module continues the receive operation until it detects a stop condition.
When receiving a frame longer than expected, the DALI module sets the STR1.LFRF flag when it recognizes bit [33] or [257] of the receive data while the setting of the CNFR1.EXM bit is 0 or 1, respectively, and continues reception. Data length is counted up to the maximum value of the STR2.RDBL[5:0] bits which is 63. However, if the stop condition cannot be detected before the counter reaches the maximum value, the counter overflows and restarts counting from 0. In this case, even if the STR1.LFRF flag is 1, the value of the STR2.RDBL[5:0] bits is invalid. The value is not required for the receive data length.
When the stop condition cannot be detected for a long period, the STR1.BBF flag remains 1 and no interrupt signal, such as INTSDD, is asserted. In this case, confirm the STR1.LFRF flag is 1, meaning it receives incorrect long data, and stop the receive operation of the DALI module by setting the CTR1.RE bit to 0. After normal DALI bus state is restored, issue a software reset by setting the SWRR1.SWR bit to 1 and restart the receive operation.

26.5.3 Oscillation error and sampling error

In the DALI module, the operating clock is generated by dividing the frequency of f_{CLK}. The data from the DALI bus is input synchronously with the operating clock of the DALI module. For this reason, set the bit timing violation detection threshold value and the collision detection threshold value, taking into account the f_{CLK} oscillation errors and the DALI module sampling errors. The sampling error of the DALI module is one cycle of the operating clock for the DALI module (3.25 µs typ.) at a maximum. The DALI module is designed with the oscillation errors within ±1.5%.

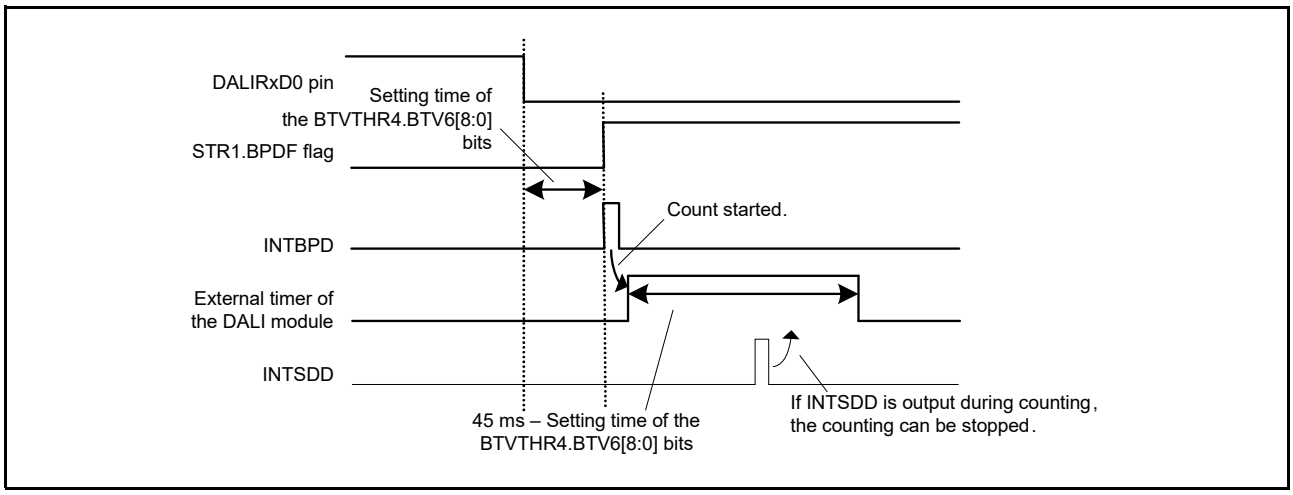
26.5.4 Using an external timer of the DALI module

Some functions are achieved by using an external timer of the DALI module so that updates and changes to the DALI standard can be easily reflected. The following sections describe functions that use external timer.

26.5.4.1 Determination of bus power down and system failure

The DALI module measures a low level of the DALI bus for the period of the BTV threshold value 6 and outputs INTBPD. Based on the DALI standard, bus power down is when a low-level period continues for 45 ms or more, and system failure is when a low-level period continues for 550 ms or more. Therefore, start the external timer of the DALI module by INTBPD and measure the periods of {45 ms – (BTV threshold value 6)} and {550 ms – (BTV threshold value 6)}. Check the STR1.BPDF flag value after completion of the measurement to ensure the DALI bus is held low during the measurement. See **26.3.3.5 Bus power down and system failure** for information about the bus power down operation of the DALI module.

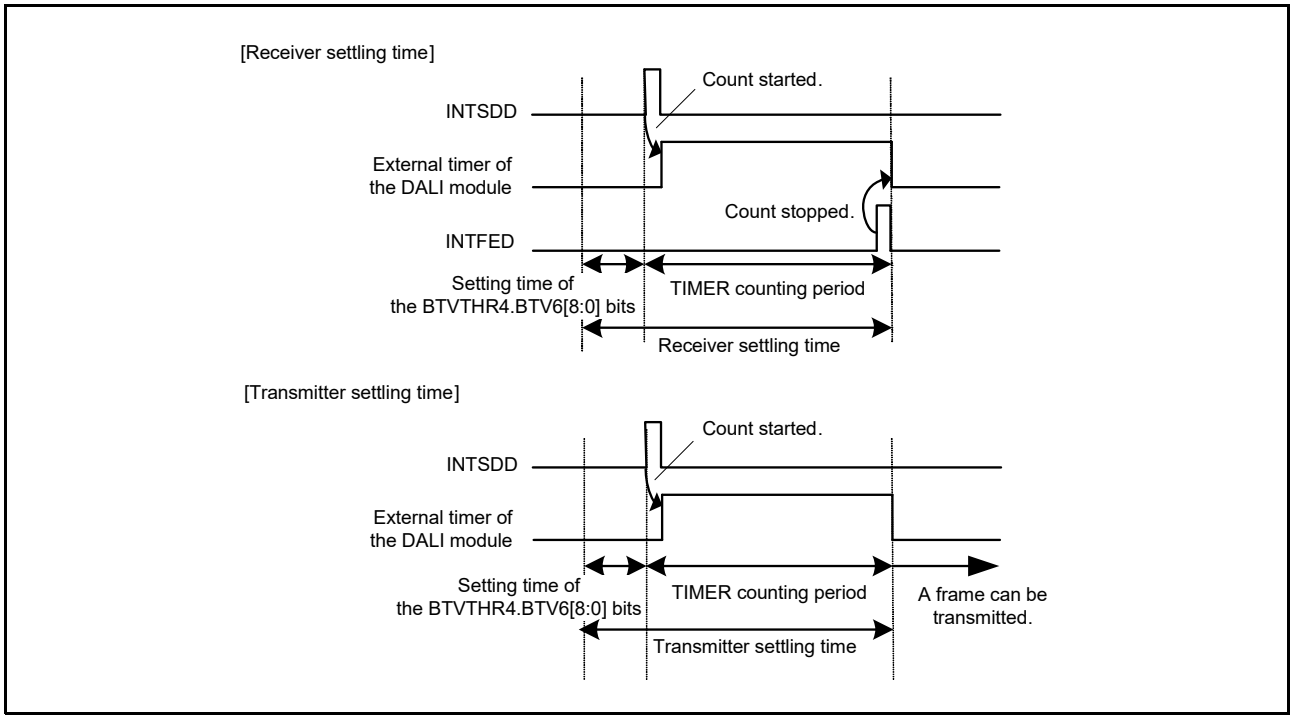
Figure 26 - 59 Example of Bus Power Down Detection



26.5.4.2 Measuring settling time

The settling time is defined in the DALI standard. Use INTSDD and INTFED that the DALI module outputs to measure the settling time as specified in international standard IEC 62386-101. INTSDD is output when a stop condition, the time specified by BTVTHR4.BTV6[8:0], is detected at the DALIRxD0 pin of the DALI module. Even while the receive operation is disabled, that is, the setting of the CTR1.RE bit is 0, the DALI module monitors the DALIRxD0 pin, and interrupts such as INTFED are generated.
INTFED is output when the first falling edge is detected after the DALI bus IDLE (high level) state. The DALI module outputs INTFED before it recognizes the start bit. Therefore, if the first falling edge from the DALI bus IDLE state is noise, then INTFED is output, but the DALI module might not transition to the receive operation state. INTFED is not used for internal operation, but is used for measuring the settling time.

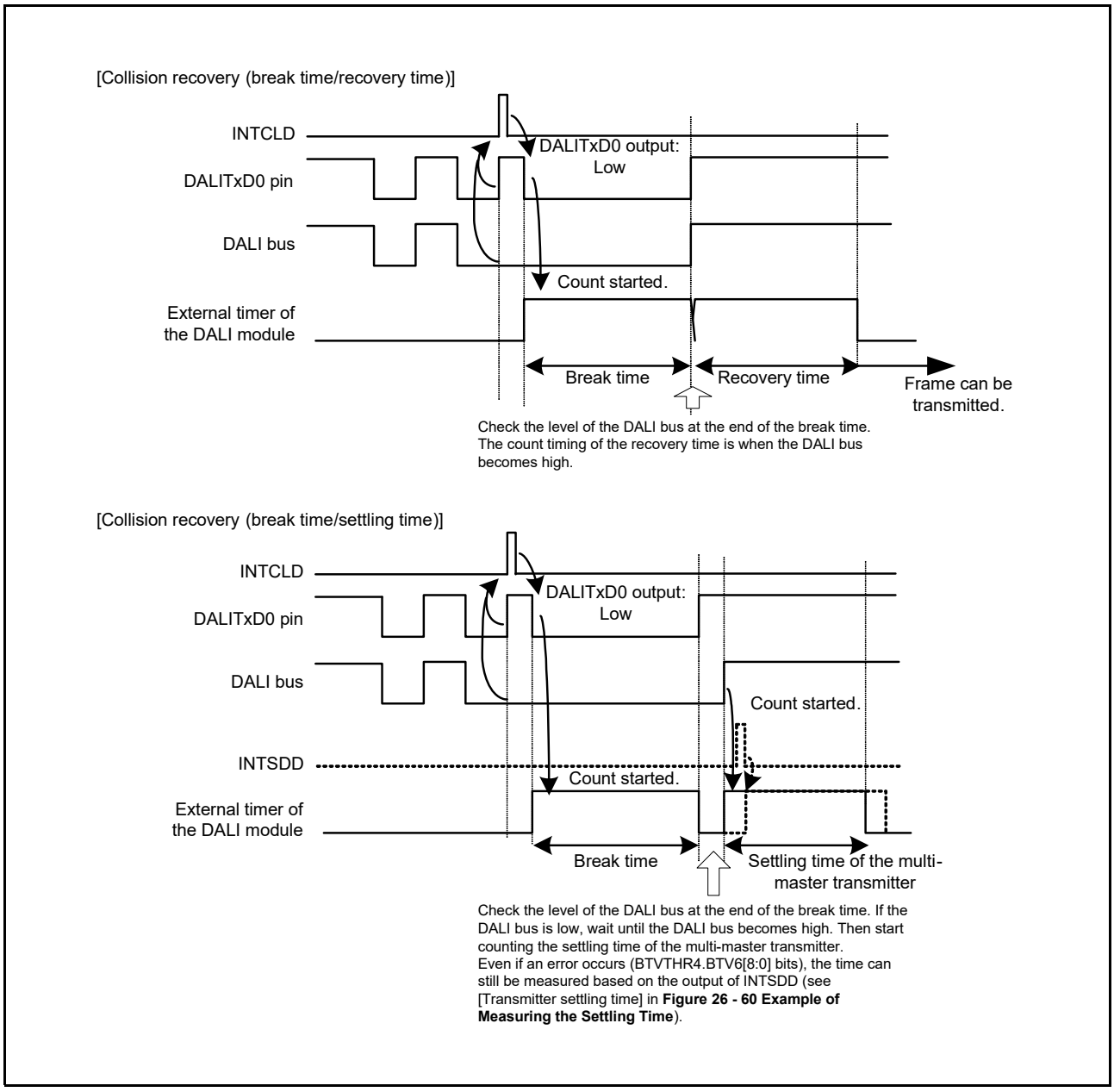
Figure 26 - 60 Example of Measuring the Settling Time



26.5.4.3 Transmission control on a collision occurrence

When the CNFR2.CDE bit is set to 1, collision detection starts. Even if a collision is detected, the DALI module continues to transmit. The DALI module must be controlled as described in **26.3.4 Collisions**.
The break time, T_{break}, the recovery time for collision recovery, $T_{recover}$, and the multi-master transmitter settling time must be measured.

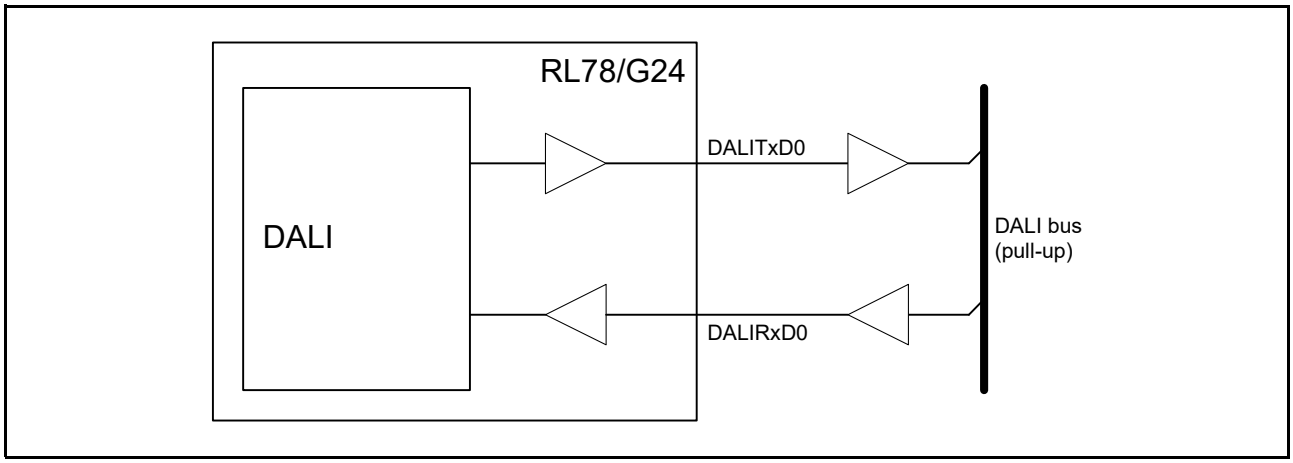
Figure 26 - 61 Example of Collision Detection



26.5.5 Example of external device connection

Connect pins DALITxD0 and DALIRxD0 to the DALI bus as shown in **Figure 26 - 62**.
Design an external circuit to satisfy the international standard IEC 62386-101 (Edition 2.0 2014-11/Edition 2.1 2018-05).

Figure 26 - 62 Example of External Device Connection



26.5.6 Notes on selecting external devices

Connecting a DALI bus to the RL78/G24 must be done via an external circuit for voltage conversion and, if required, insulation. Such an external circuit may include photocouplers and transistors. Note that the rising and falling edges of the waveforms output from such a circuit may delay, resulting in deviation of the high- and low-level width. Consider this in designing an external circuit. Furthermore, design an external circuit to satisfy the DALI standard on the DALI bus side and to operate within the permissible range of this module, taking voltage fluctuation and aging degradation into account. The DALI module supports width adjustment for both DALITxD0 output waveform and DALIRxD0 input waveform, by using which the range of the transmission and reception characteristics can be extended.

Adjustable range of the width of DALITxD0 output waveform: 214.5 to 617.5 µs
Acceptable range of the edges of DALIRxD0 input waveform:
When the settings of both the CNFR2.SGA and FTDC0.IST bits are 0 (default):
High-/low-level period of Manchester code: 318.5 to 510.25 µs
When the settings of the CNFR2.SGA and FTDC0.IST bits are respectively 1 and 0:
High-/low-level period of Manchester code: 295.75 to 559 µs
When the setting of the FTDC0.IST bit is 1, regardless of the setting of the CNFR2.SGA bit:
High-/low-level period of Manchester code: 292.5 to 552.5 µs
Adjustable range of the width of DALIRxD0 input waveform: 0 to 208 µs

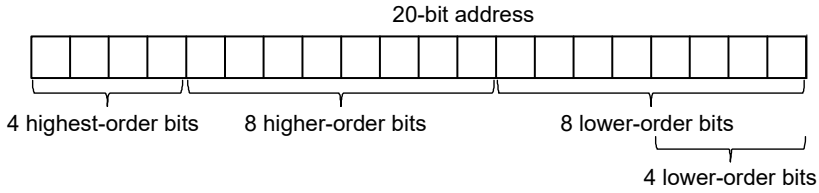
Caution 1. The adjustable range of the width of DALIRxD0 input waveform can be achieved by correcting the acceptable range of the edges of DALIRxD0 input waveform, in accordance with the deviation of the rising and falling edges of the waveform output from the external circuit.
Caution 2. None of the given ranges include errors in oscillation clock frequencies.

For details, see the following sections and documents.

- **26.3.6 Width adjustment for DALITxD0 output waveform**
- **26.3.5.1 Adjusting the acceptable range of the edges of DALIRxD0 input waveform**
- **26.3.7 Width adjustment for DALIRxD0 input waveform**
- IEC 62386-101 standard
- Data sheets of the photocouplers and transistors to be used

Section 27 Data Transfer Controller (DTC)

The term “8 higher-order bits of the address” in this section indicates bits 15 to 8 of 20-bit address as shown below.



Unless otherwise specified, the 4 highest-order address bits all become 1 (values are of the form FxxxxH).

27.1 Functions of DTC

The data transfer controller (DTC) transfers data between memories without using the CPU. The DTC is activated by a peripheral function interrupt and transfers data. The DTC and CPU use the same bus, and the DTC takes priority over the CPU in using the bus.

Table 27 - 1 lists the DTC specifications.

Table 27 - 1 DTC Specifications (1/2)

Item		Specification
Activation sources		42 sources (20-pin products)/47 sources (24- and 25-pin products)/52 sources (30- and 32-pin products)/53 sources (40-, 44-, 48-, 52-, and 64-pin products)
Allocatable control data		24 sets
Address space available for use with DTC transfer	Address space	64 Kbytes (F0000H to FFFFFH), excluding general-purpose registers
	Source	Special function registers (SFRs), RAM area (excluding general-purpose registers), mirror area**Note**, data flash memory area**Note**, extended special function registers (2nd SFRs)
	Destination	Special function registers (SFRs), RAM area (excluding general-purpose registers), extended special function registers (2nd SFRs)
Maximum number of transfers	Normal mode	256 times
	Repeat mode	255 times
Maximum size of block to be transferred	Normal mode (8-bit transfer)	256 bytes
	Normal mode (16-bit transfer)	512 bytes
	Repeat mode	255 bytes
Unit of transfers		8 bits/16 bits
Transfer mode	Normal mode	Transfers end on completion of the transfer causing the DTCCTj register value to change from 1 to 0.
	Repeat mode	On completion of the transfer causing the DTCCTj register value to change from 1 to 0, the repeat area address is initialized and the DTRLDj register value is reloaded to the DTCCTj register to continue transfers.
Address control	Normal mode	Fixed or incremented
	Repeat mode	Addresses of the area not selected as the repeat area are fixed or incremented.

Table 27 - 1 DTC Specifications (2/2)

Item		Specification
Priority of activation sources		Refer to **Table 27 - 3 DTC Activation Sources and Vector Addresses**.
Interrupt request	Normal mode	When the data transfer causing the DTCCTj register value to change from 1 to 0 is performed, the activation source interrupt request is generated for the CPU, and interrupt handling is performed on completion of the data transfer.
	Repeat mode	When the data transfer causing the DTCCTj register value to change from 1 to 0 is performed while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled), the activation source interrupt request is generated for the CPU, and interrupt handling is performed on completion of the data transfer.
Transfer start		When the DTCENi[7:0] bits in the DTCENi registers are 1 (activation enabled), data transfer is started each time the corresponding DTC activation sources are generated.
Transfer stop	Normal mode	• The DTCENi[7:0] bits are set to 0 (activation disabled). • The data transfer causing the DTCCTj register value to change from 1 to 0 is completed.
	Repeat mode	• The DTCENi[7:0] bits are set to 0 (activation disabled). • The data transfer causing the DTCCTj register value to change from 1 to 0 is completed while the RPTINT bit is 1 (interrupt generation enabled).

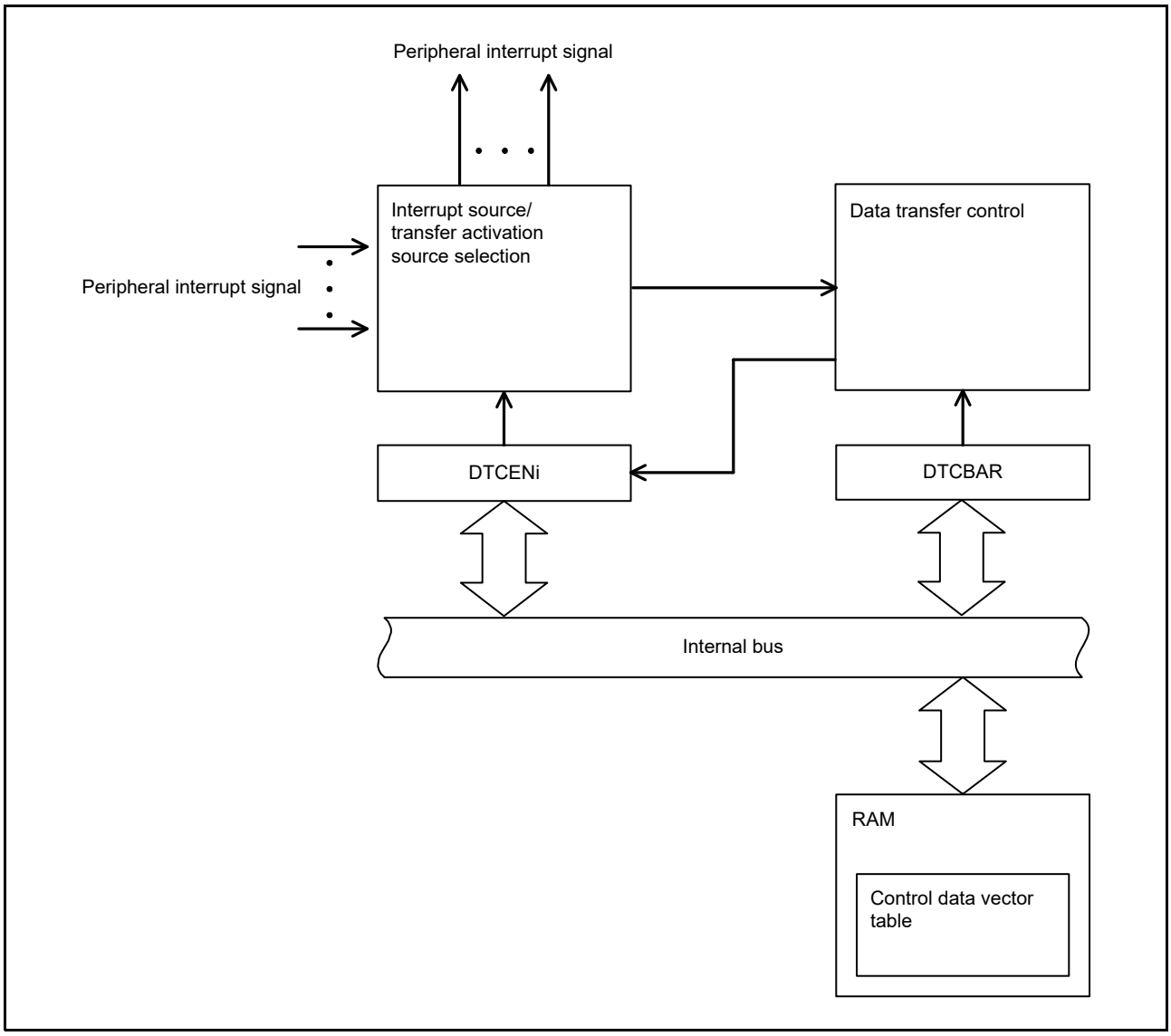
Note In the HALT mode or SNOOZE mode, these areas cannot be set as the sources for DTC transfer since the flash memory is stopped.

Remark i = 0 to 6, j = 0 to 23

27.2 Configuration of DTC

Figure 27 - 1 shows the DTC block diagram.

Figure 27 - 1 DTC Block Diagram



Remark i = 0 to 6

27.3 Registers to Control the DTC

The following registers are used to control the DTC.

- Peripheral enable register 1 (PER1)
- DTC activation enable registers i (DTCENi) (i = 0 to 6)
- DTC base address register (DTCBAR)

The DTC control data are listed below.
The DTC control data are allocated in the DTC control data area in RAM.
The DTCBAR register is used to set the 256-byte area, including the DTC control data area and the DTC vector table area where the start addresses for the control data are stored.

- DTC control registers j (DTCCRj) (j = 0 to 23)
- DTC block size registers j (DTBLSj) (j = 0 to 23)
- DTC transfer count registers j (DTCCTj) (j = 0 to 23)
- DTC transfer count reload registers j (DTRLDj) (j = 0 to 23)
- DTC source address registers j (DTSARj) (j = 0 to 23)
- DTC destination address registers j (DTDARj) (j = 0 to 23)

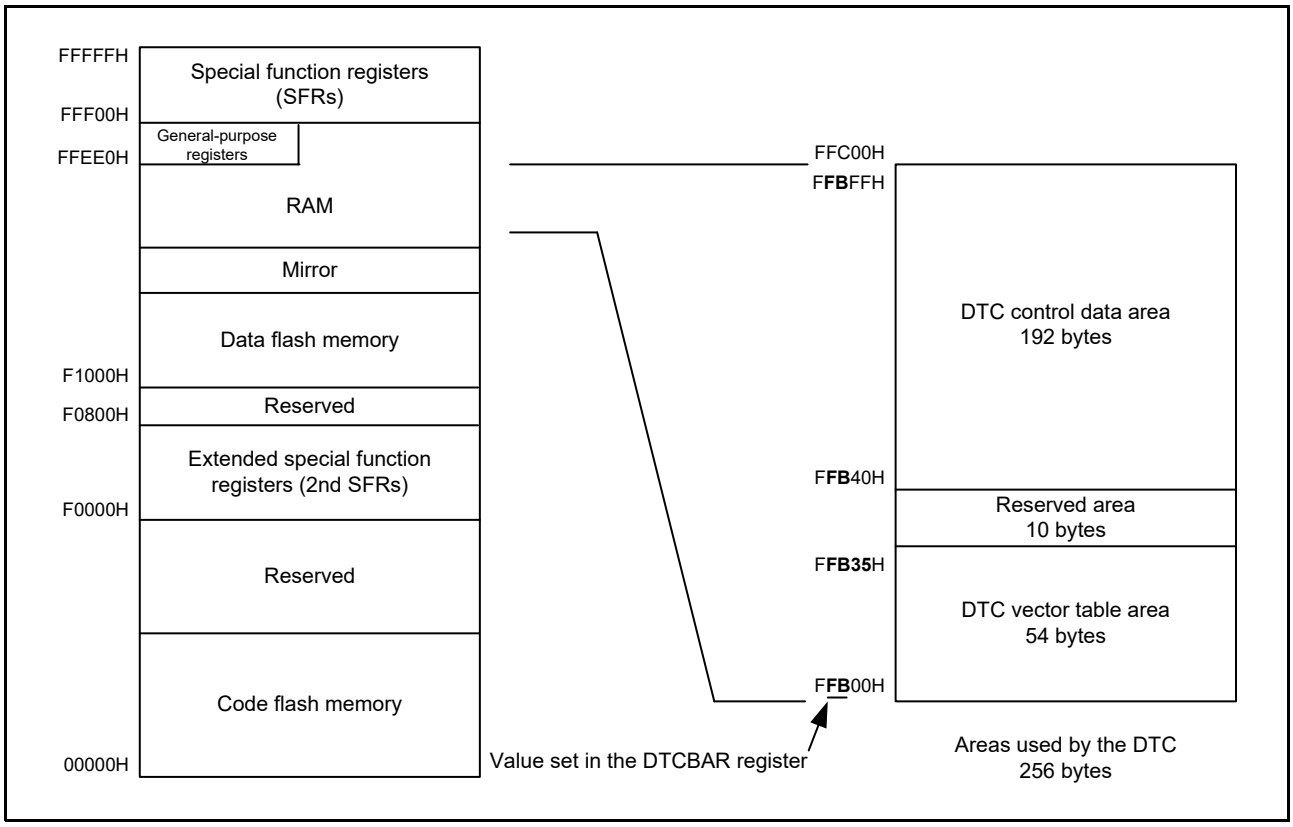
27.3.1 Allocation of DTC control data area and DTC vector table area

The DTCBAR register is used to set the 256-byte area where DTC control data and the vector table are allocated within the RAM area.

Figure 27 - 2 shows a memory map example when the DTCBAR register is set to FBH.

In the 192-byte DTC control data area, the space not used by the DTC can be used as RAM.

Figure 27 - 2 Memory Map Example When the DTCBAR Register Is Set to FBH



The areas where the DTC control data and vector table can be allocated differ depending on the product.

Caution 1. It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space as the DTC control data area or DTC vector table area.

Caution 2. Make sure the stack area, the DTC control data area, and the DTC vector table area do not overlap.

Caution 3. The internal RAM area in the following products cannot be used as the DTC control data area or DTC vector table area when using the on-chip debugging trace function.
R7F101GxE (x = 7, 8, 9, A, B, E, F, G): FD300H to FD6FFH
R7F101GxG (x = 7, 8, 9, A, B, E, F, G): FD300H to FD6FFH

27.3.2 Control data allocation

The control data are allocated beginning with each start address in the order: registers DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, and DTDARj (j = 0 to 23).

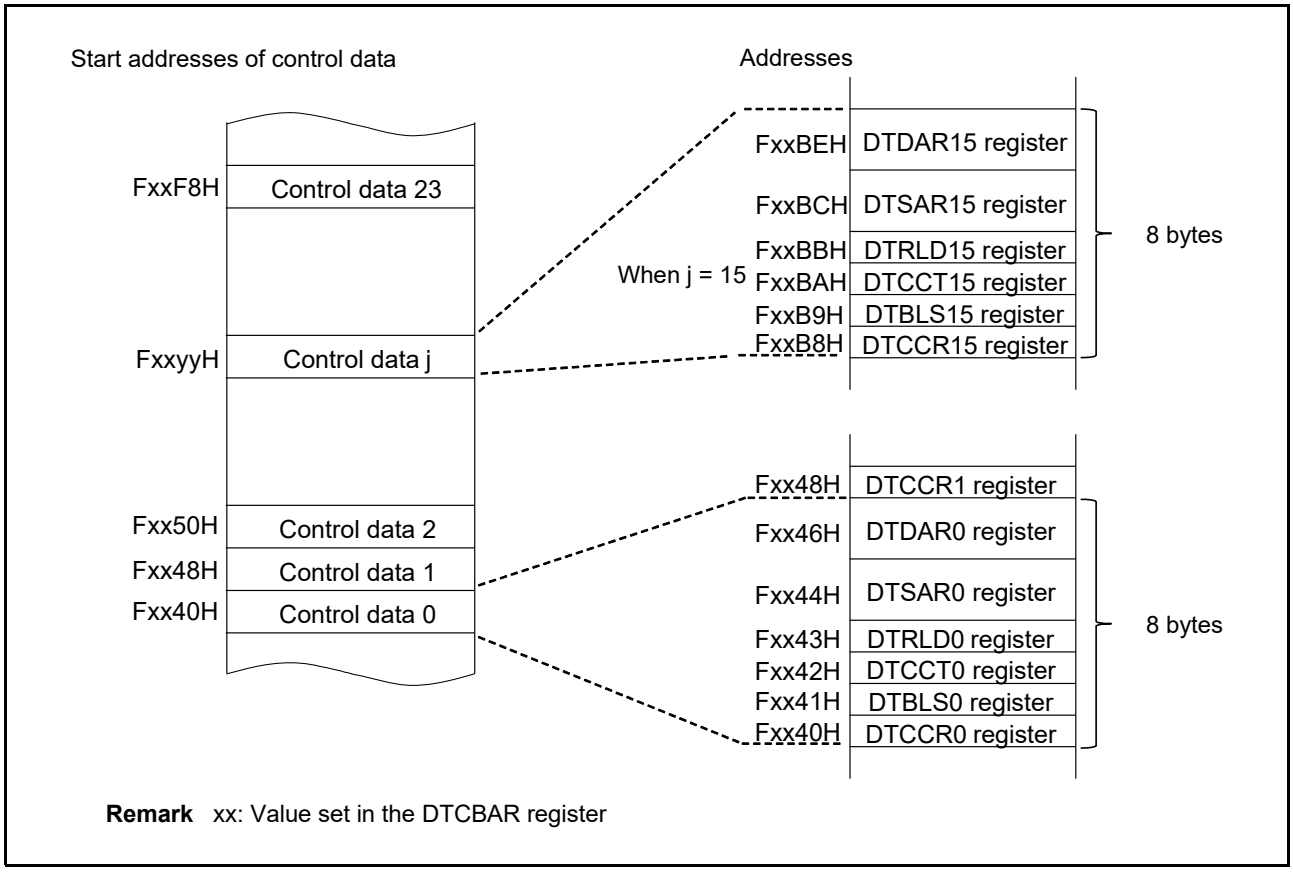
The 8 higher-order bits of start addresses 0 to 23 are set in the DTCBAR register, and the 8 lower-order bits are separately set according to the vector table assigned to each activation source.

Figure 27 - 3 shows control data allocation.

Caution 1. Change the data in registers DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, and DTDARj when the corresponding bit among the DTCENi[7:0] bits (i = 0 to 6) in the DTCENi register is set to 0 (activation disabled).

Caution 2. Do not access DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, or DTDARj by using DTC transfer.

Figure 27 - 3 Control Data Allocation



Remark xx: Value set in the DTCBAR register

Table 27 - 2 Start Addresses of Control Data j

j	Address	j	Address
11	Fxx98H	23	FxxF8H
10	Fxx90H	22	FxxF0H
9	Fxx88H	21	FxxE8H
8	Fxx80H	20	FxxE0H
7	Fxx78H	19	FxxD8H
6	Fxx70H	18	FxxD0H
5	Fxx68H	17	FxxC8H
4	Fxx60H	16	FxxC0H
3	Fxx58H	15	FxxB8H
2	Fxx50H	14	FxxB0H
1	Fxx48H	13	FxxA8H
0	Fxx40H	12	FxxA0H

Remark xx: Value set in the DTCBAR register

27.3.3 Vector table

When the DTC is activated, one control data are selected according to the data read from the vector table which has been assigned to each activation source, and the selected control data are read from the DTC control data area. **Table 27 - 3** lists the DTC activation sources and vector addresses. One byte of the vector table is assigned to each activation source, and the offset values from 40H to F8H at which each set of control data starts are stored in each of the locations to select each of the 24 control data sets. The 8 higher-order bits of the vector addresses are set in the DTCBAR register, and 00H to 35H are allocated to the 8 lower-order bits corresponding to the activation source.

Caution Change the start address of the DTC control data area to be set in the vector table when the corresponding bit among the DTCENi[7:0] bits (i = 0 to 6) in the DTCENi register is set to 0 (activation disabled).

Figure 27 - 4 Start Addresses of Control Data and Vector Table

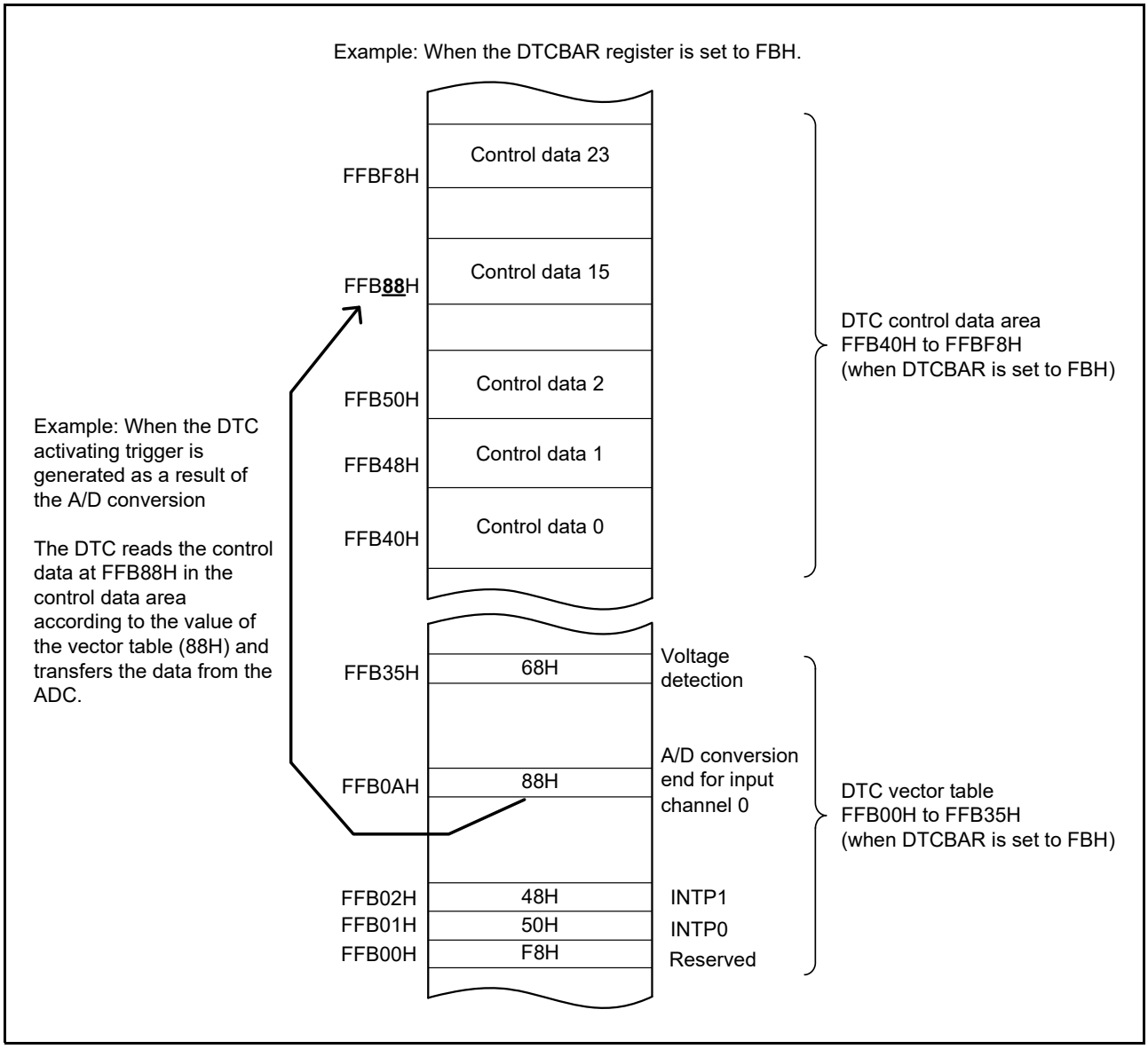


Table 27 - 3 DTC Activation Sources and Vector Addresses (1/2)

DTC Activation Source (Interrupt Request Source)	Source No.	Vector Address	Priority
Reserved	0	Address set in the DTCBAR register + 00H	Highest
INTP0	1	Address set in the DTCBAR register + 01H	
INTP1 Note 3	2	Address set in the DTCBAR register + 02H	
INTP2 Note 3	3	Address set in the DTCBAR register + 03H	
INTP3 Note 3	4	Address set in the DTCBAR register + 04H	
INTP4 Note 3	5	Address set in the DTCBAR register + 05H	
INTP5 Note 2	6	Address set in the DTCBAR register + 06H	
INTP6	7	Address set in the DTCBAR register + 07H	
INTP7	8	Address set in the DTCBAR register + 08H	
Key input Note 4	9	Address set in the DTCBAR register + 09H	
A/D conversion end for input channel 0	10	Address set in the DTCBAR register + 0AH	
UART0 reception transfer end/CSI01 transfer end or buffer empty/IIC01 transfer end Note 2	11	Address set in the DTCBAR register + 0BH	
UART0 transmission transfer end/CSI00 transfer end or buffer empty/IIC00 transfer end Note 2	12	Address set in the DTCBAR register + 0CH	
UART1 reception transfer end/CSI11 transfer end or buffer empty/IIC11 transfer end	13	Address set in the DTCBAR register + 0DH	
UART1 transmission transfer end/CSI10 transfer end or buffer empty/IIC10 transfer end	14	Address set in the DTCBAR register + 0EH	
UART2 reception transfer end/CSI21 transfer end or buffer empty/IIC21 transfer end	15	Address set in the DTCBAR register + 0FH	
UART2 transmission transfer end/CSI20 transfer end or buffer empty/IIC20 transfer end	16	Address set in the DTCBAR register + 10H	
DALI reception transfer end	17	Address set in the DTCBAR register + 11H	
DALI transmission transfer end	18	Address set in the DTCBAR register + 12H	
DALI bus power down detection	19	Address set in the DTCBAR register + 13H	
End of channel 0 timer array unit counting or capture	20	Address set in the DTCBAR register + 14H	
End of channel 1 timer array unit counting or capture	21	Address set in the DTCBAR register + 15H	
End of channel 2 timer array unit counting or capture	22	Address set in the DTCBAR register + 16H	
End of channel 3 timer array unit counting or capture	23	Address set in the DTCBAR register + 17H	
RD2 timer compare match A0	24	Address set in the DTCBAR register + 18H	
RD2 timer compare match B0	25	Address set in the DTCBAR register + 19H	
RD2 timer compare match C0	26	Address set in the DTCBAR register + 1AH	
RD2 timer compare match D0	27	Address set in the DTCBAR register + 1BH	
RD2 timer compare match A1	28	Address set in the DTCBAR register + 1CH	
RD2 timer compare match B1	29	Address set in the DTCBAR register + 1DH	
RD2 timer compare match C1	30	Address set in the DTCBAR register + 1EH	
RD2 timer compare match D1	31	Address set in the DTCBAR register + 1FH	
Timer RD2 extended complementary PWM cycle peak detection or timer RD20 timer-KB PWM output gating period detection	32	Address set in the DTCBAR register + 20H	
Timer RD2 extended complementary PWM carrier period detection or timer RD21 timer-KB PWM output gating period detection	33	Address set in the DTCBAR register + 21H	
Completion of FAA operations	34	Address set in the DTCBAR register + 22H	
RG2 timer compare match A	35	Address set in the DTCBAR register + 23H	
RG2 timer compare match B	36	Address set in the DTCBAR register + 24H	
RG2 timer compare match C	37	Address set in the DTCBAR register + 25H	
RG2 timer compare match D	38	Address set in the DTCBAR register + 26H	
RJ0 timer underflow	39	Address set in the DTCBAR register + 27H	
End of TMKB30 counting	40	Address set in the DTCBAR register + 28H	
End of TMKB31 counting	41	Address set in the DTCBAR register + 29H	
End of TMKB32 counting Note 2	42	Address set in the DTCBAR register + 2AH	
FAA timing compare match 0	43	Address set in the DTCBAR register + 2BH	
Fixed-cycle signal of realtime clock/alarm match detection Note 3	44	Address set in the DTCBAR register + 2CH	
Interval signal detection of 32-bit interval timer	45	Address set in the DTCBAR register + 2DH	
Comparator detection 0	46	Address set in the DTCBAR register + 2EH	
Comparator detection 1	47	Address set in the DTCBAR register + 2FH	
Comparator detection 2	48	Address set in the DTCBAR register + 30H	
Comparator detection 3 Note 2	49	Address set in the DTCBAR register + 31H	Lowest

Table 27 - 3 DTC Activation Sources and Vector Addresses (2/2)

DTC Activation Source (Interrupt Request Source)	Source No.	Vector Address	Priority
A/D conversion end for input channel 1 Note 5	50	Address set in the DTCBAR register + 32H	Highest
A/D conversion end for input channel 2 Note 5	51	Address set in the DTCBAR register + 33H	↕
A/D conversion end for input channel 3 Note 5	52	Address set in the DTCBAR register + 34H	↕
Voltage detection Note 1	53	Address set in the DTCBAR register + 35H	Lowest

Note 1. When bit 6 (LVD0SEL) of the user option byte (000C1H) is set to 0 or when bit 6 (LVD1SEL) of the voltage detection level register (LVIS) is set to 0

Note 2. For 24- to 64-pin products only.

Note 3. For 30- to 64-pin products only.

Note 4. For 40- to 64-pin products only.

Note 5. When bit 0 (ADVMOD) of A/D converter mode register 3 (ADM3) is set to 1

27.3.4 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. If the DTC is to be used, be sure to set bit 3 (DTCEN) to 1. The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 27 - 5 Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	<3>	2	1	<0>
PER1	DACEN	0	PGACMPEN	TML32EN	DTCEN	0	0	DALIEN

DTCEN	Control of supply of an input clock to the DTC
0	Stops supply of an input clock. • The SFRs used by the DTC cannot be written.
1	Enables supply of an input clock. • The DTC can operate.

27.3.5 DTC control registers j (DTCCRj) (j = 0 to 23)

The DTCCRj registers are used to control the DTC operating mode.

Figure 27 - 6 Format of DTC Control Register j (DTCCRj) (1/2)

Address: Refer to **27.3.2 Control data allocation**.
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DTCCRj	0	SZ	RPTINT	CHNE	DAMOD	SAMOD	RPTSEL	MODE

SZ	Transfer data size selection
0	8 bits
1	16 bits

RPTINT	Enabling/disabling repeat mode interrupts
0	Interrupt generation disabled
1	Interrupt generation enabled
The setting of the RPTINT bit is invalid when the MODE bit is 0 (normal mode).	

CHNE	Enabling/disabling chain transfers
0	Chain transfers disabled
1	Chain transfers enabled
Set the CHNE bit in the DTCCR23 register to 0 (chain transfers disabled).	

DAMOD	Transfer destination address control
0	Fixed
1	Incremented
The setting of the DAMOD bit is invalid when the MODE bit is 1 (repeat mode) and the RPTSEL bit is 0 (transfer destination is the repeat area).	

SAMOD	Transfer source address control
0	Fixed
1	Incremented
The setting of the SAMOD bit is invalid when the MODE bit is 1 (repeat mode) and the RPTSEL bit is 1 (transfer source is the repeat area).	

RPTSEL	Repeat area selection
0	Transfer destination is the repeat area.
1	Transfer source is the repeat area.
The setting of the RPTSEL bit is invalid when the MODE bit is 0 (normal mode).	

Figure 27 - 6 Format of DTC Control Register j (DTCCRj) (2/2)

MODE	Transfer mode selection
0	Normal mode
1	Repeat mode

Caution Do not access the DTCCRj register by using DTC transfer.

27.3.6 DTC block size registers j (DTBLSj) (j = 0 to 23)

The DTBLSj registers are used to set the block size of the data to be transferred by one activation.

Figure 27 - 7 Format of DTC Block Size Register j (DTBLSj)

Address: Refer to **27.3.2 Control data allocation**.
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DTBLSj	DTBLSj7	DTBLSj6	DTBLSj5	DTBLSj4	DTBLSj3	DTBLSj2	DTBLSj1	DTBLSj0

DTBLSj	Transfer block size	
	8-bit transfer	16-bit transfer
00H	256 bytes	512 bytes
01H	1 byte	2 bytes
02H	2 bytes	4 bytes
03H	3 bytes	6 bytes
• • •	• • •	• • •
FDH	253 bytes	506 bytes
FEH	254 bytes	508 bytes
FFH	255 bytes	510 bytes

Caution Do not access the DTBLSj register by using DTC transfer.

27.3.7 DTC transfer count registers j (DTCCTj) (j = 0 to 23)

The DTCCTj registers are used to set the number of DTC data transfers. The value is decremented by 1 each time DTC transfer is activated once.

Figure 27 - 8 Format of DTC Transfer Count Register j (DTCCTj)

Address: Refer to **27.3.2 Control data allocation**.
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DTCCTj	DTCCTj7	DTCCTj6	DTCCTj5	DTCCTj4	DTCCTj3	DTCCTj2	DTCCTj1	DTCCTj0

DTCCTj	Number of transfers
00H	256 times
01H	Once
02H	2 times
03H	3 times
• • •	• • •
FDH	253 times
FEH	254 times
FFH	255 times

Caution Do not access the DTCCTj register by using DTC transfer.

27.3.8 DTC transfer count reload registers j (DTRLDj) (j = 0 to 23)

The DTRLDj registers are used to set the initial value of the transfer count register in repeat mode. Since the value of each DTRLDj register is reloaded to the corresponding DTCCTj register in repeat mode, set the same value as the initial value of the DTCCTj register.

Figure 27 - 9 Format of DTC Transfer Count Reload Register j (DTRLDj)

Address: Refer to **27.3.2 Control data allocation**.
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DTRLDj	DTRLDj7	DTRLDj6	DTRLDj5	DTRLDj4	DTRLDj3	DTRLDj2	DTRLDj1	DTRLDj0

Caution Do not access the DTRLDj register by using DTC transfer.

27.3.9 DTC source address registers j (DTSARj) (j = 0 to 23)

The DTSARj registers are used to specify the transfer source address for data transfer. When the SZ bit in the DTCCRj register is set to 1 (the transfer data size is 16 bits), the lowest-order bit is ignored and the address is handled as an even address.

Figure 27 - 10 Format of DTC Source Address Register j (DTSARj)

Address: Refer to **27.3.2 Control data allocation**.
After reset: Undefined
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
DTSARj	DTSARj15	DTSARj14	DTSARj13	DTSARj12	DTSARj11	DTSARj10	DTSARj9	DTSARj8

7	6	5	4	3	2	1	0
DTSARj7	DTSARj6	DTSARj5	DTSARj4	DTSARj3	DTSARj2	DTSARj1	DTSARj0

Caution 1. Do not set the general-purpose register (FFEE0H to FFEFFH) space as the transfer source address.
Caution 2. Do not access the DTSARj register by using DTC transfer.

27.3.10 DTC destination address registers j (DTDARj) (j = 0 to 23)

The DTDARj registers are used to specify the transfer destination address for data transfer. When the SZ bit in the DTCCRj register is set to 1 (the transfer data size is 16 bits), the lowest-order bit is ignored and the address is handled as an even address.

Figure 27 - 11 Format of DTC Destination Address Register j (DTDARj)

Address: Refer to **27.3.2 Control data allocation**.
After reset: Undefined
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
DTDARj	DTDARj15	DTDARj14	DTDARj13	DTDARj12	DTDARj11	DTDARj10	DTDARj9	DTDARj8

7	6	5	4	3	2	1	0
DTDARj7	DTDARj6	DTDARj5	DTDARj4	DTDARj3	DTDARj2	DTDARj1	DTDARj0

Caution 1. Do not set the general-purpose register (FFEE0H to FFEFFH) space as the transfer destination address.
Caution 2. Do not access the DTDARj register by using DTC transfer.

27.3.11 DTC activation enable registers i (DTCENi) (i = 0 to 6)

The DTCENi registers are 8-bit registers which enable or disable DTC activation by interrupt sources. **Table 27 - 4** lists the correspondences between interrupt sources and the DTCENi[7:0] bits. The DTCENi register can be set by a 1-bit or 8-bit memory manipulation instruction.

Caution 1. Modify the DTCENi[7:0] bits if an activation source corresponding to the bit has not been generated.

Caution 2. Do not access the DTCENi register by using DTC transfer.

Caution 3. The assigned functions differ depending on the product. For the bits to which no function is assigned, be sure to set their values to 0.

Figure 27 - 12 Format of DTC Activation Enable Register i (DTCENi) (1/2)

Address: F02E8H (DTCEN0), F02E9H (DTCEN1), F02EAH (DTCEN2), F02EBH (DTCEN3), F02ECH (DTCEN4), F02EDH (DTCEN5), F02EEH (DTCEN6)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DTCENi	DTCENi7	DTCENi6	DTCENi5	DTCENi4	DTCENi3	DTCENi2	DTCENi1	DTCENi0

DTCENi7	DTC activation enable i7
0	Activation disabled
1	Activation enabled
The DTCENi7 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi6	DTC activation enable i6
0	Activation disabled
1	Activation enabled
The DTCENi6 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi5	DTC activation enable i5
0	Activation disabled
1	Activation enabled
The DTCENi5 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi4	DTC activation enable i4
0	Activation disabled
1	Activation enabled
The DTCENi4 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi3	DTC activation enable i3
0	Activation disabled
1	Activation enabled
The DTCENi3 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

Figure 27 - 12 Format of DTC Activation Enable Register i (DTCENi) (2/2)

DTCENi2	DTC activation enable i2
0	Activation disabled
1	Activation enabled
The DTCENi2 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi1	DTC activation enable i1
0	Activation disabled
1	Activation enabled
The DTCENi1 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi0	DTC activation enable i0
0	Activation disabled
1	Activation enabled
The DTCENi0 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

Table 27 - 4 Correspondences between Interrupt Sources and DTCENi[7:0] Bits

Register	DTCENi7 Bit	DTCENi6 Bit	DTCENi5 Bit	DTCENi4 Bit	DTCENi3 Bit	DTCENi2 Bit	DTCENi1 Bit	DTCENi0 Bit
DTCEN0	Reserved	INTP0	INTP1Note 3	INTP2Note 3	INTP3Note 3	INTP4Note 3	INTP5Note 2	INTP6
DTCEN1	INTP7	Key inputNote 4	A/D conversion end for input channel 0	UART0 reception transfer end/CSI01 transfer end or buffer empty/IIC01 transfer endNote 2	UART0 transmission transfer end/CSI00 transfer end or buffer empty/IIC00 transfer endNote 2	UART1 reception transfer end/CSI11 transfer end or buffer empty/IIC11 transfer end	UART1 transmission transfer end/CSI10 transfer end or buffer empty/IIC10 transfer end	UART2 reception transfer end/CSI21 transfer end or buffer empty/IIC21 transfer end
DTCEN2	UART2 transmission transfer end/CSI20 transfer end or buffer empty/IIC20 transfer end	DALI reception transfer end	DALI transmission transfer end	DALI bus power down detection	End of channel 0 timer array unit counting or capture	End of channel 1 timer array unit counting or capture	End of channel 2 timer array unit counting or capture	End of channel 3 timer array unit counting or capture
DTCEN3	RD2 timer compare match A0	RD2 timer compare match B0	RD2 timer compare match C0	RD2 timer compare match D0	RD2 timer compare match A1	RD2 timer compare match B1	RD2 timer compare match C1	RD2 timer compare match D1
DTCEN4	Valley detection in RD2 timer extended complementary PWM mode	Crest detection in RD2 timer extended complementary PWM mode	Completion of FAA operations	RG2 timer compare match A	RG2 timer compare match B	RG2 timer compare match C	RG2 timer compare match D	RJ0 timer underflow
DTCEN5	End of TMKB30 counting	End of TMKB31 counting	End of TMKB32 counting	FAA timing compare match 0	Fixed-cycle signal of realtime clock/alarm match detection	Interval signal detection of 32-bit interval timer	Comparator detection 0	Comparator detection 1
DTCEN6	Comparator detection 2	Comparator detection 3	A/D conversion end for input channel 1Note 5	A/D conversion end for input channel 2Note 5	A/D conversion end for input channel 3Note 5	Voltage detectionNote 1	Reserved	Reserved

(Notes, Caution, and Remark are listed on the next page.)

Note 1. When bit 6 (LVD0SEL) of the user option byte (000C1H) is set to 0 or when bit 6 (LVD1SEL) of the voltage detection level register (LVIS) is set to 0

Note 2. For 24- to 64-pin products only.

Note 3. For 30- to 64-pin products only.

Note 4. For 40- to 64-pin products only.

Note 5. When bit 0 (ADVMOD) of A/D converter mode register 3 (ADM3) is set to 1

Caution **For the bits to which no function is assigned, be sure to set their values to 0.**

Remark i = 0 to 6

27.3.12 DTC base address register (DTCBAR)

The DTCBAR register is an 8-bit register used to set the following addresses: the vector address where the start address of the DTC control data area is stored and the address of the DTC control data area. The value of the DTCBAR register is handled as the 8 higher-order bits to generate a 16-bit address. The DTCBAR register can be set by an 8-bit memory manipulation instruction.

Caution 1. Change the DTCBAR register value with all DTC activation sources set to activation disabled.
Caution 2. Do not rewrite the DTCBAR register more than once.
Caution 3. Do not access the DTCBAR register by using DTC transfer.
Caution 4. For the allocation of the DTC control data area and the DTC vector table area, refer to the cautions in 27.3.1 Allocation of DTC control data area and DTC vector table area.

Figure 27 - 13 Format of DTC Base Address Register (DTCBAR)

Address: F02E0H
After reset: FDH
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DTCBAR	DTCBAR7	DTCBAR6	DTCBAR5	DTCBAR4	DTCBAR3	DTCBAR2	DTCBAR1	DTCBAR0

27.4 DTC Operation

When the DTC is activated, the DTC reads control data from the DTC control data area, proceeds with data transfer according to the control data, and writes back the control data after data transfer to the DTC control data area. Twenty-four sets of control data can be stored in the DTC control data area, which allows 24 types of data transfers to be performed.

There are two transfer modes (normal mode and repeat mode) and two transfer sizes (8-bit transfer and 16-bit transfer).

When the CHNE bit in the DTCCRj (j = 0 to 23) register is set to 1 (chain transfers enabled), multiple control data are read and data transfers are continuously performed by one activation source (chain transfers).

A transfer source address is specified by the 16-bit register DTSARj, and a transfer destination address is specified by the 16-bit register DTDARj. The values in registers DTSARj and DTDARj are separately incremented or fixed according to the control data after the data transfer.

27.4.1 Activation sources

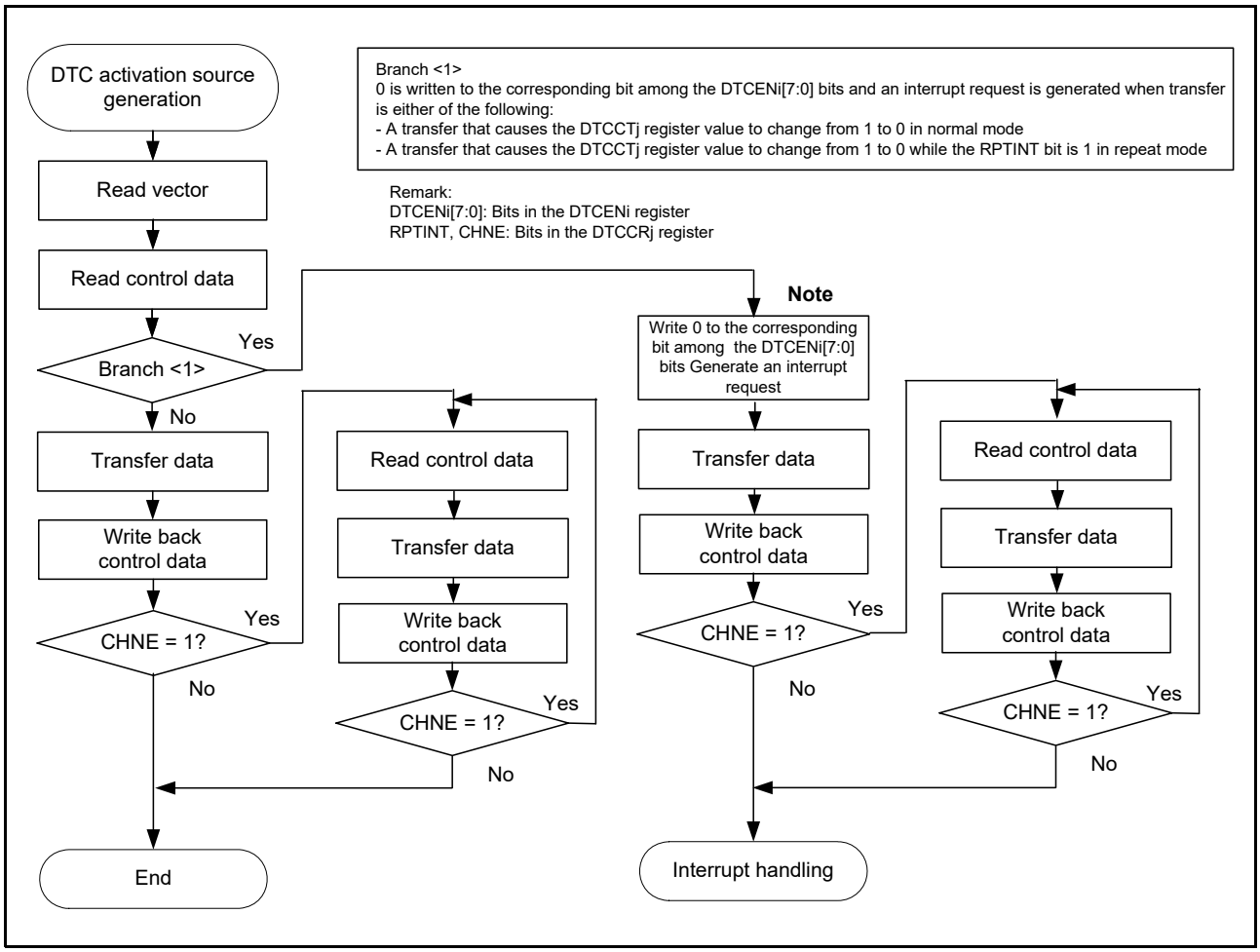
The DTC is activated by an interrupt signal from the peripheral functions. The interrupt signals to activate the DTC are selected with the DTCENi (i = 0 to 6) register.

The DTC sets the corresponding bit among the DTCENi[7:0] bits in the DTCENi register to 0 (activation disabled) during operation when the setting of data transfer (the first transfer in chain transfers) is either of the following:

- A transfer that causes the DTCCTj (j = 0 to 23) register value to change to 0 in normal mode
- A transfer that causes the DTCCTj register value to change to 0 while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled) in repeat mode

Figure 27 - 14 shows a DTC internal operation flowchart.

Figure 27 - 14 DTC Internal Operation Flowchart



Note 0 is not written to the corresponding bit among the DTCENi[7:0] bits for data transfers activated by the setting to enable chain transfers (the CHNE bit is 1). Also, no interrupt request is generated.

Remark i = 0 to 6, j = 0 to 23

27.4.2 Normal mode

One to 256 bytes of data are transferred by one activation during 8-bit transfer and 2 to 512 bytes during 16-bit transfer. The number of transfers can be 1 to 256 times. When the data transfer causing the DTCCTj (j = 0 to 23) register value to change to 0 is performed, the DTC generates an interrupt request corresponding to the activation source to the interrupt controller during DTC operation, and sets the corresponding bit among the DTCENi[7:0] bits (i = 0 to 6) in the DTCENi register to 0 (activation disabled).

Table 27 - 5 shows the register functions in normal mode. **Figure 27 - 15** shows data transfers in normal mode.

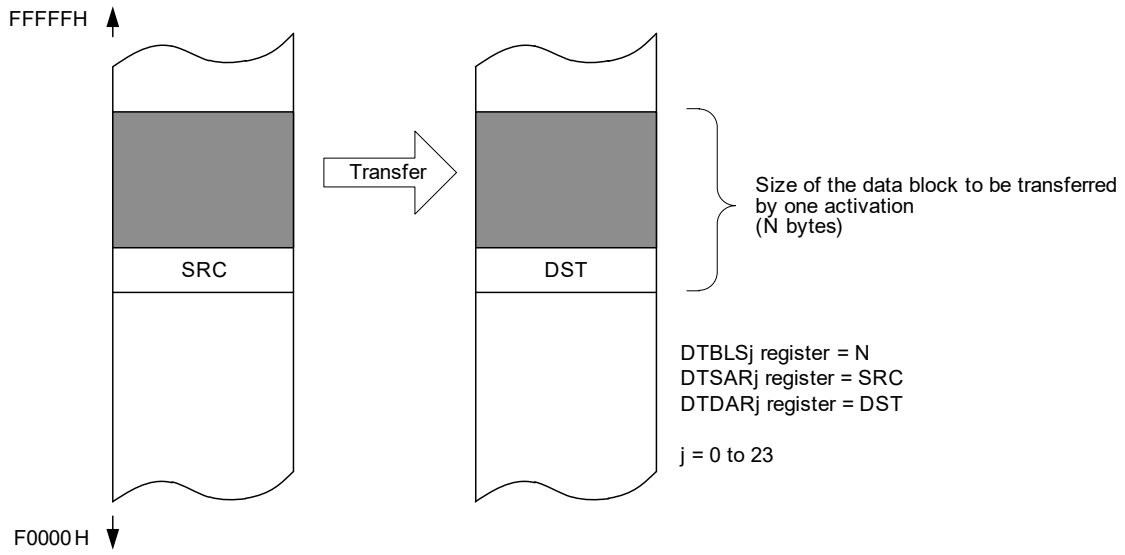
Table 27 - 5 Register Functions in Normal Mode

Register Name	Register Symbol	Function
DTC block size register j	DTBLSj	Size of the data block to be transferred by one activation
DTC transfer count register j	DTCCTj	Number of data transfers
DTC transfer count reload register j	DTRLDj	Not used**Note**
DTC source address register j	DTSARj	Data transfer source address
DTC destination address register j	DTDARj	Data transfer destination address

Note Initialize this register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

Remark j = 0 to 23

Figure 27 - 15 Data Transfers in Normal Mode

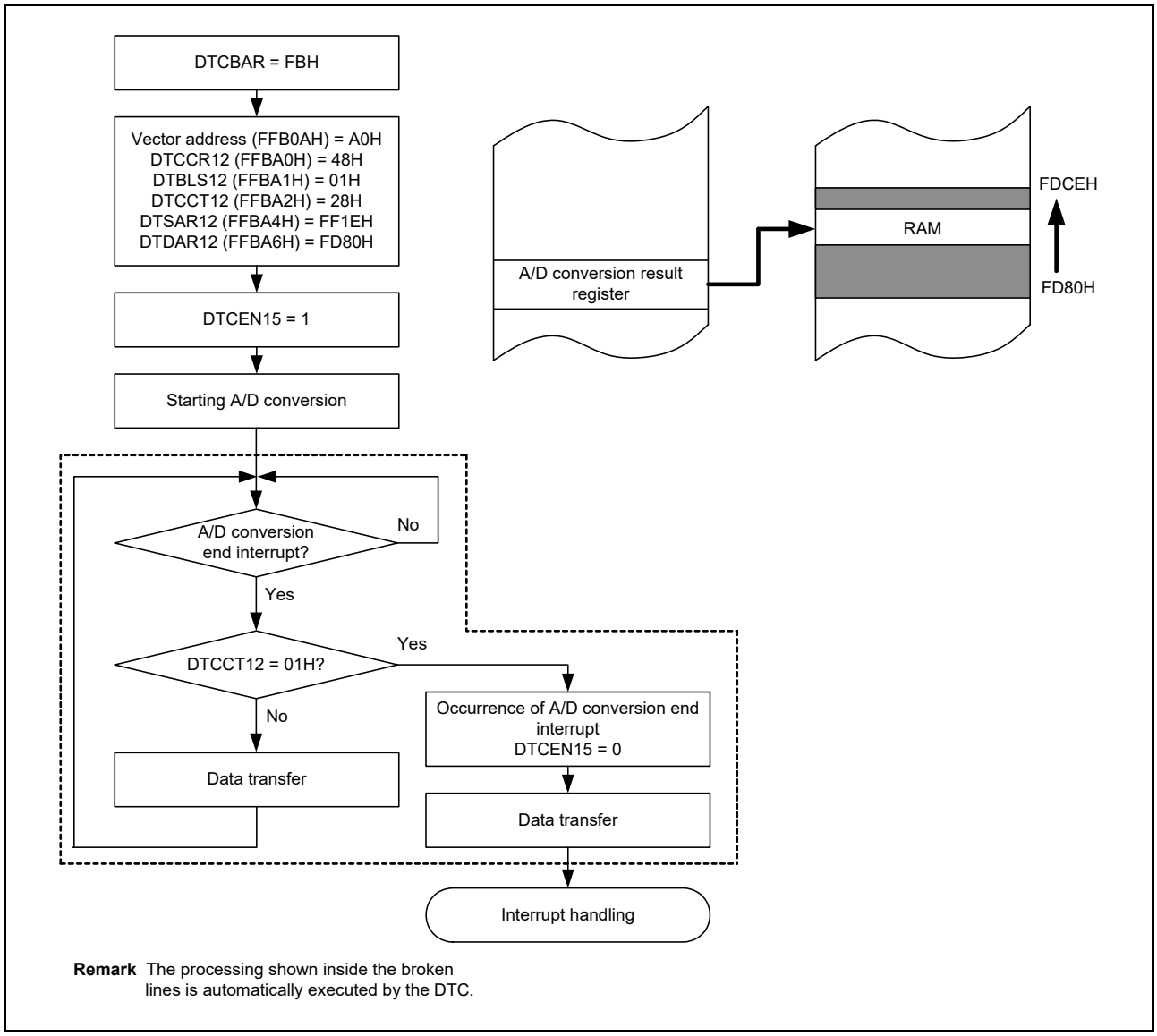


DTCCRj Register Setting				Source Address Control	Destination Address Control	Source Address after Transfer	Destination Address after Transfer
DAMOD	SAMOD	RPTSEL	MODE				
0	0	×	0	Fixed	Fixed	SRC	DST
0	1	×	0	Incremented	Fixed	SRC + N	DST
1	0	×	0	Fixed	Incremented	SRC	DST + N
1	1	×	0	Incremented	Incremented	SRC + N	DST + N

×: 0 or 1

1. Example 1 of using normal mode: Consecutively capturing A/D conversion results
   The DTC is activated by an A/D conversion end interrupt and the value of the A/D conversion result register is transferred to RAM.
   - The vector address is FFB0AH and control data are allocated at FFBA0H to FFBA7H.
   - Transfers 2-byte data of the A/D conversion result register (FFF1EH, FFF1FH) to the 80-byte area at addresses from FFD80H to FFDCFH of RAM 40 times.

Figure 27 - 16 Example 1 of Using Normal Mode: Consecutively Capturing A/D Conversion Results

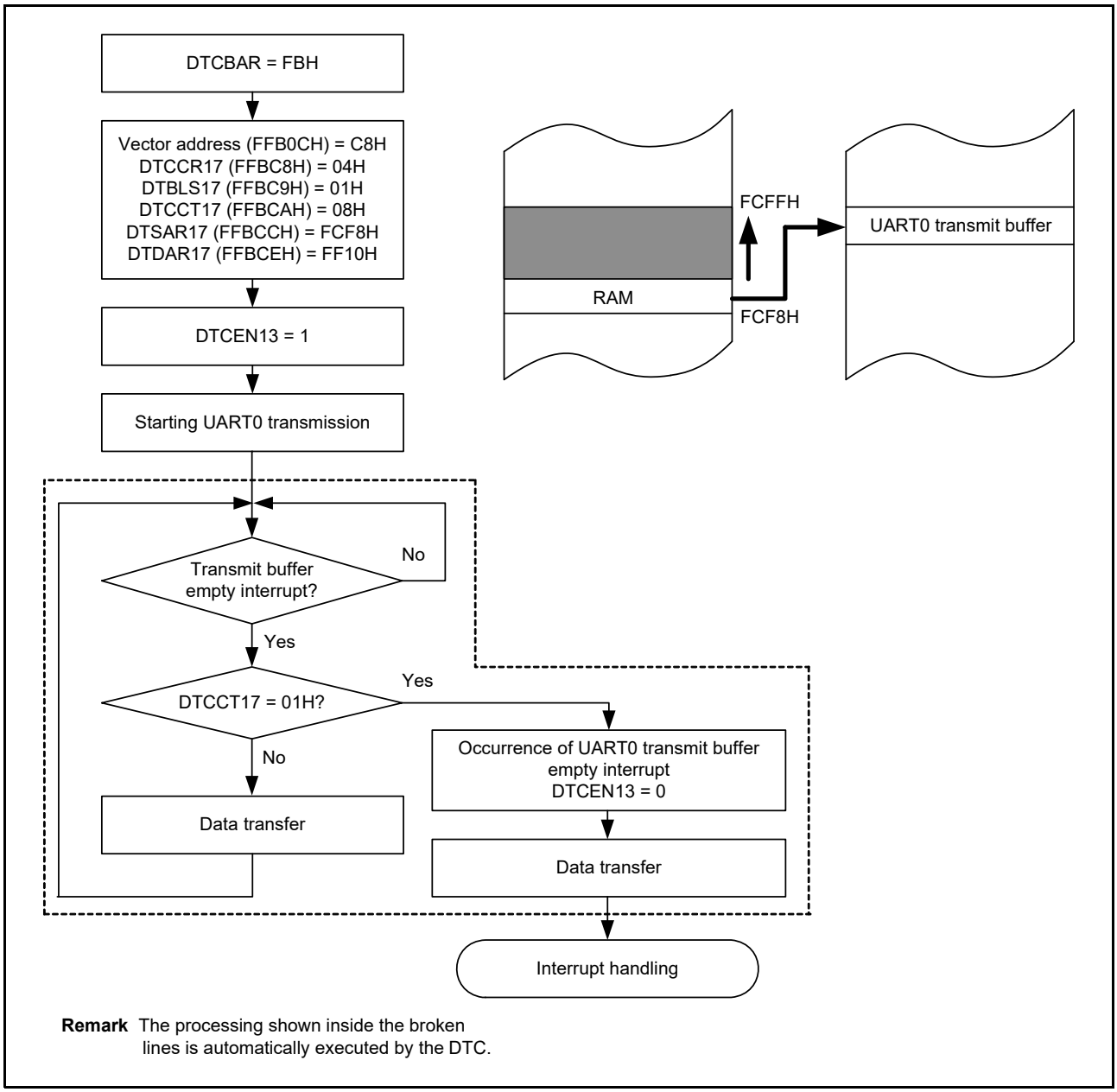


Remark The processing shown inside the broken lines is automatically executed by the DTC.

The value of the DTRLD12 register is not used because of normal mode, but initialize the register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

2. Example 2 of using normal mode: UART0 consecutive transmission
   The DTC is activated by a UART0 transmit buffer empty interrupt and the value of RAM is transferred to the UART0 transmit buffer.
   - The vector address is FFB0CH and control data are allocated at FFBC8H to FFBCFH.
   - Transfers 8 bytes of data at addresses from FFCF8H to FFCFFH of RAM to the UART0 transmit buffer (FFF10H).

Figure 27 - 17 Example 2 of Using Normal Mode: UART0 Consecutive Transmission



Remark The processing shown inside the broken lines is automatically executed by the DTC.

The value of the DTRLD17 register is not used because of normal mode, but initialize the register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.
Start the first UART0 transmission by software. The second and subsequent transmissions proceed automatically by using transmit buffer empty interrupts to activate the DTC.

27.4.3 Repeat mode

One to 255 bytes of data are transferred by one activation. Either of the transfer source or destination should be specified as the repeat area. The number of transfers can be 1 to 255 times. On completion of the specified number of transfers, the DTCCTj (j = 0 to 23) register and the address specified for the repeat area are initialized to continue transfers. When the data transfer causing the DTCCTj register value to change to 0 is performed while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled), the DTC generates an interrupt request corresponding to the activation source to the interrupt controller during DTC operation, and sets the corresponding bit among the DTCENi[7:0] bits (i = 0 to 6) in the DTCENi register to 0 (activation disabled). When the RPTINT bit in the DTCCRj register is 0 (interrupt generation disabled), no interrupt request is generated even if the data transfer causing the DTCCTj register value to change to 0 is performed. Also, the corresponding bit among the DTCENi[7:0] bits is not set to 0.

Table 27 - 6 lists the register functions in repeat mode. **Figure 27 - 18** shows data transfers in repeat mode.

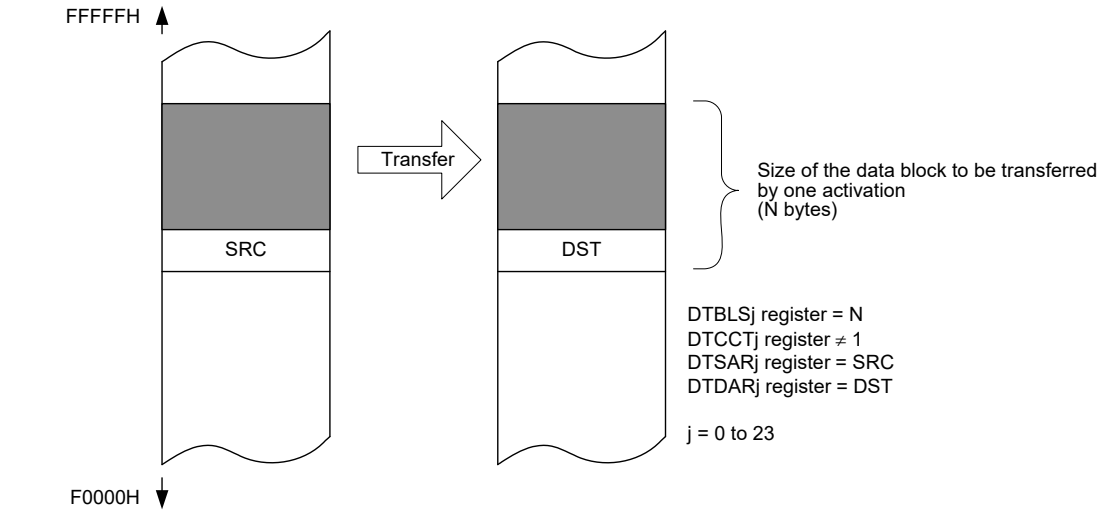
Table 27 - 6 Register Functions in Repeat Mode

Register Name	Register Symbol	Function
DTC block size register j	DTBLSj	Size of the data block to be transferred by one activation
DTC transfer count register j	DTCCTj	Number of data transfers
DTC transfer count reload register j	DTRLDj	This register value is reloaded to the DTCCTj register (the number of transfers is initialized).
DTC source address register j	DTSARj	Data transfer source address
DTC destination address register j	DTDARj	Data transfer destination address

Remark j = 0 to 23

Figure 27 - 18 Data Transfers in Repeat Mode

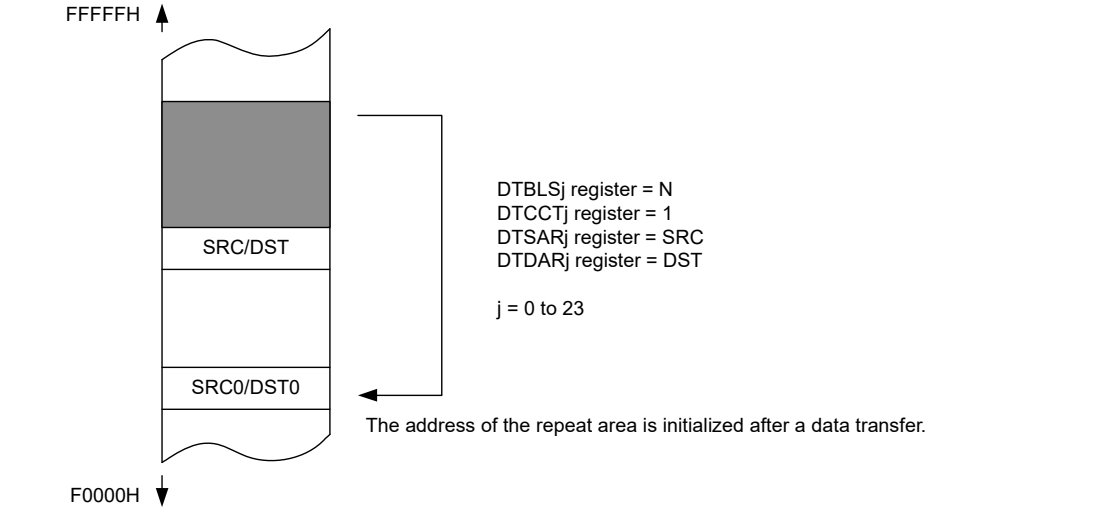
a) DTCCTj register ≠ 01H



DTCCRj Register Setting				Source Address Control	Destination Address Control	Source Address after Transfer	Destination Address after Transfer
DAMOD	SAMOD	RPTSEL	MODE				
0	×	1	1	Repeat area	Fixed	SRC + N	DST
1	×	1	1	Repeat area	Incremented	SRC + N	DST + N
×	0	0	1	Fixed	Repeat area	SRC	DST + N
×	1	0	1	Incremented	Repeat area	SRC + N	DST + N

×: 0 or 1

b) DTCCTj register = 01H



DTCCRj Register Setting				Source Address Control	Destination Address Control	Source Address after Transfer	Destination Address after Transfer
DAMOD	SAMOD	RPTSEL	MODE				
0	×	1	1	Repeat area	Fixed	SRC0	DST
1	×	1	1	Repeat area	Incremented	SRC0	DST + N
×	0	0	1	Fixed	Repeat area	SRC	DST0
×	1	0	1	Incremented	Repeat area	SRC + N	DST0

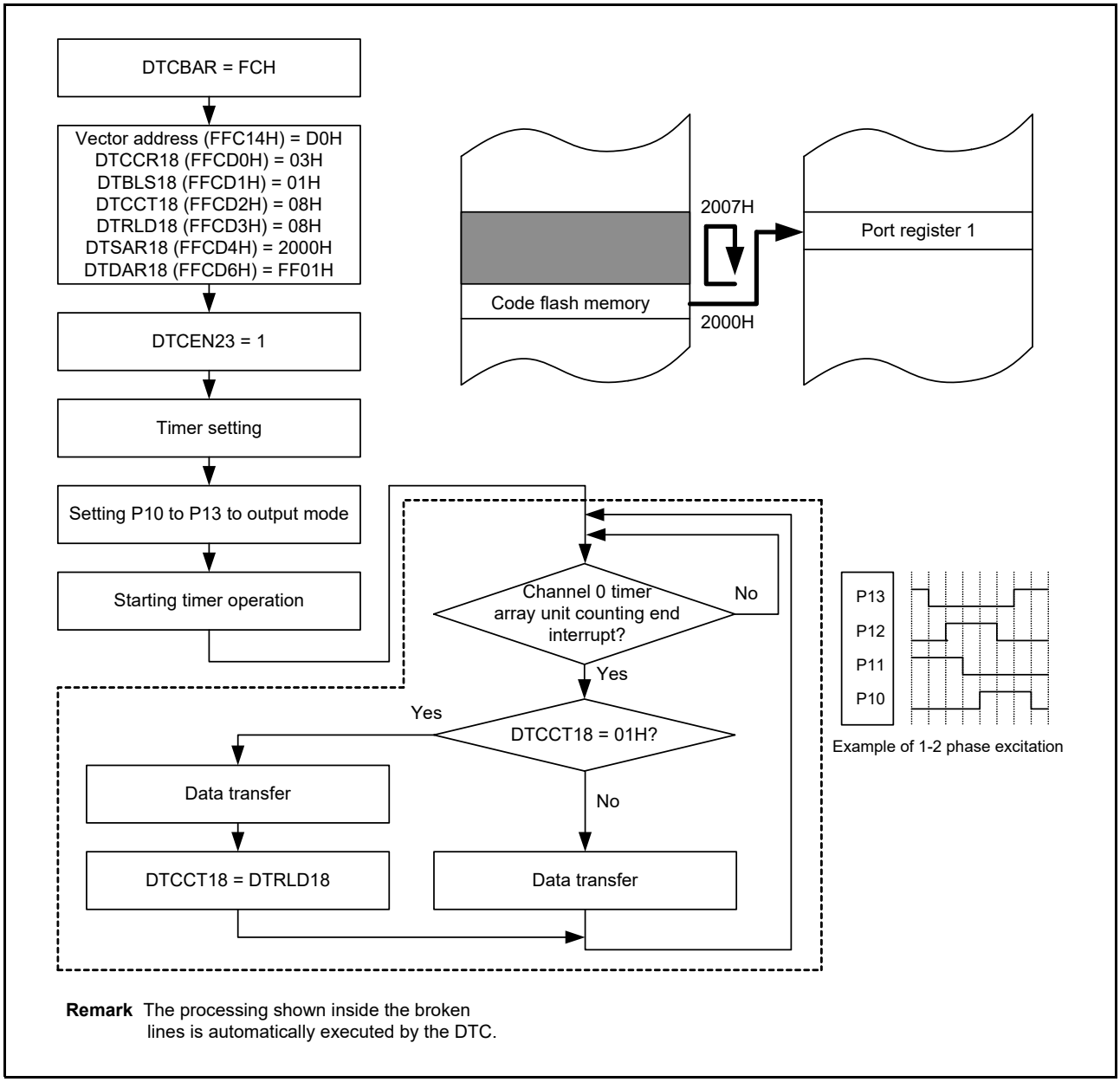
SRC0: Initial source address value
DST0: Initial destination address value
×: 0 or 1

Caution 1. When repeat mode is used, the 8 lower-order bits of the initial value for the repeat area address must be 00H.

Caution 2. When repeat mode is used, the data size of the repeat area must be set to 255 bytes or less.

1. Example 1 of using repeat mode: Outputting stepping motor control pulses using port pins
   The DTC is activated using the interval timer function of the channel 0 timer array unit, and the patterns of the motor control pulse stored in the code flash memory are transferred to the general-purpose port pins.
   - The vector address is FFC14H and control data are allocated at FFCD0H to FFCD7H.
   - Transfers 8-byte data at addresses from 02000H to 02007H of the code flash memory from the mirror area (F2000H to F2007H) to port register 1 (FFF01H).
   - A repeat mode interrupt is disabled.

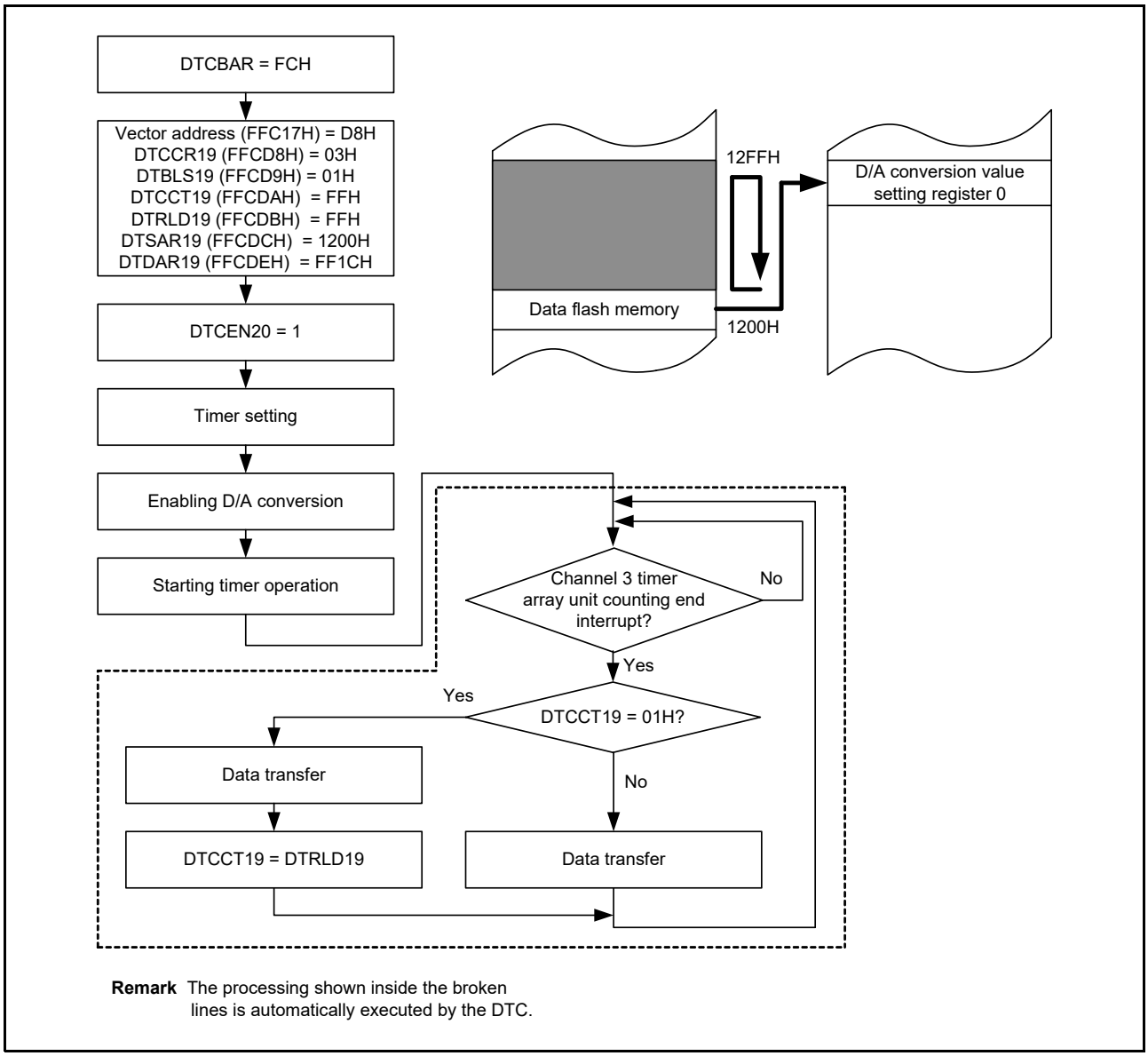
Figure 27 - 19 Example 1 of Using Repeat Mode: Outputting Stepping Motor Control Pulses Using Port Pins



To stop the output, stop the timer first and then clear the DTCEN23 bit.

2. Example 2 of using repeat mode: Outputting a sine wave using the D/A converter
   The DTC is activated using an interrupt of the interval timer function of the channel 3 timer array unit, and the table of the sine wave stored in the data flash memory is transferred to the D/A conversion value setting register 0 (F0334H).
   The timer interval time is set to the D/A output setup time.
   - The vector address is FFC17H and control data are allocated at FFCD8H to FFCDFH.
   - Transfers 255-byte data at addresses from F1200H to F12FEH of the data flash memory to the D/A conversion value setting register 0 (F0334H).
   - A repeat mode interrupt is disabled.

Figure 27 - 20 Example 2 of Using Repeat Mode: Outputting a Sine Wave Using the D/A Converter



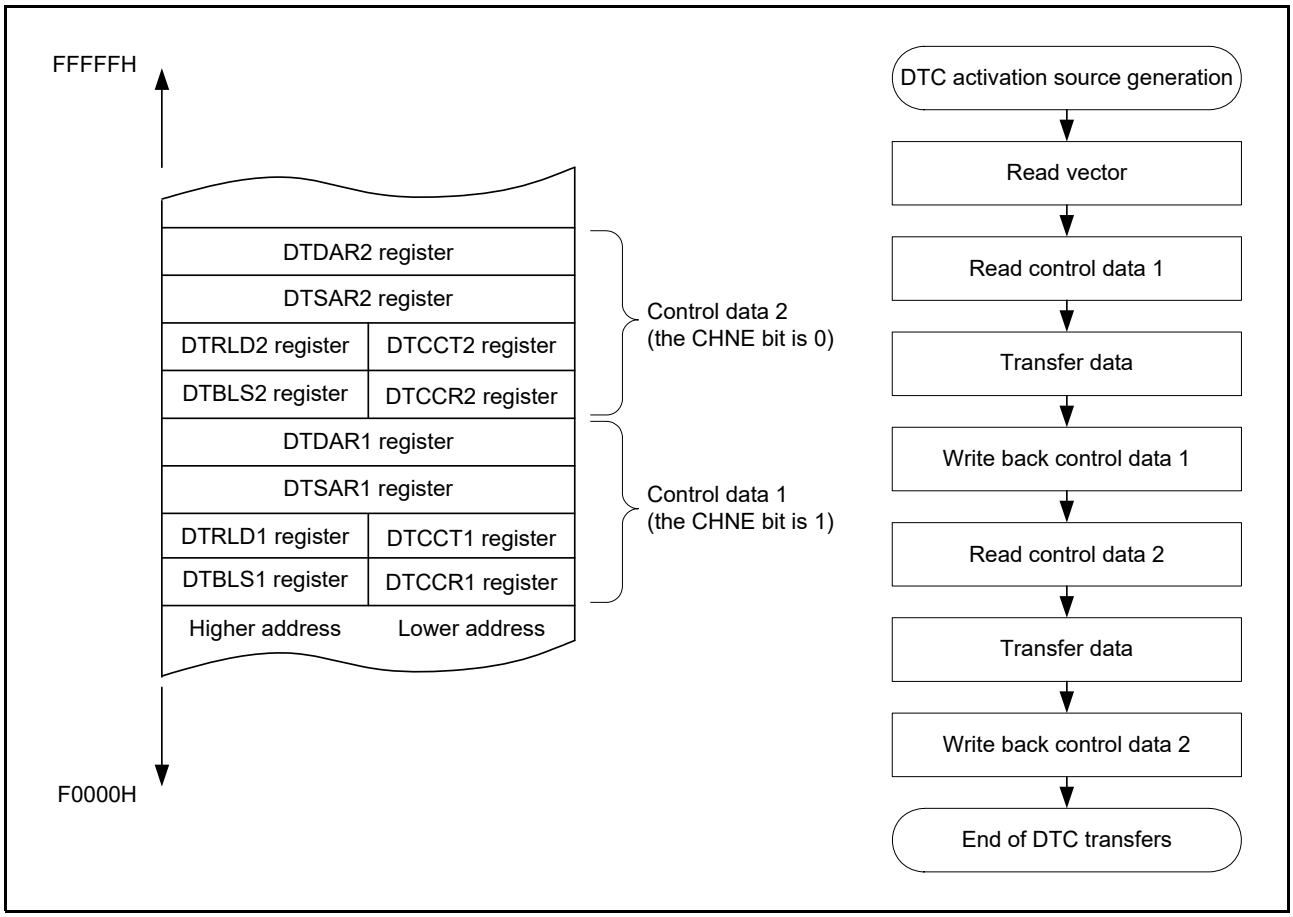
Remark The processing shown inside the broken lines is automatically executed by the DTC.

To stop the output, stop the timer first and then clear the DTCEN20 bit.

27.4.4 Chain transfers

When the CHNE bit in the DTCCRj (j = 0 to 23) register is 1 (chain transfers enabled), multiple data transfers can be continuously performed by one activation source.
When the DTC is activated, one control data are selected according to the data read from the vector address corresponding to the activation source, and the selected control data are read from the DTC control data area.
When the CHNE bit of the control data is 1 (chain transfers enabled), the next control data immediately following the current control data are read to continue a transfer after the current transfer is completed. This operation is repeated until the data transfer with the control data for which the CHNE bit is 0 (chain transfers disabled) is completed.
When chain transfers are performed using multiple control data, the number of transfers set for the first control data is enabled, and the number of transfers set for the second and subsequent control data to be processed will be invalid.
Figure 27 - 21 shows data transfers during chain transfers.

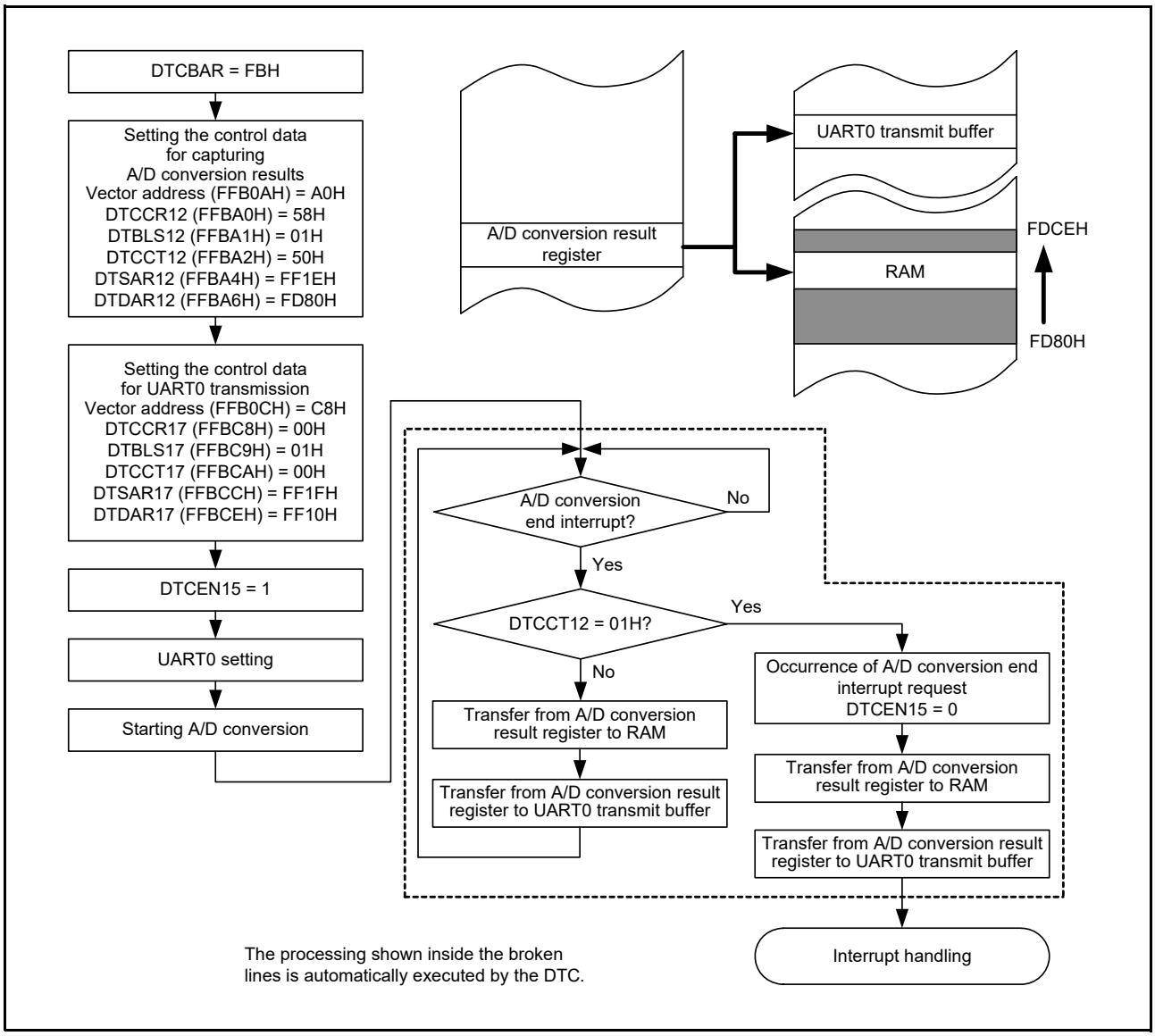
Figure 27 - 21 Data Transfers during Chain Transfers



Caution 1. Set the CHNE bit in the DTCCR23 register to 0 (chain transfers disabled).
Caution 2. During chain transfers, the corresponding bit among the DTCENi[7:0] bits (i = 0 to 6) in the DTCENi register is not set to 0 (activation disabled) for the second and subsequent transfers. Also, no interrupt request is generated.

1. Example of using chain transfers: Consecutively capturing A/D conversion results and UART0 transmission
   The DTC is activated by an A/D conversion end interrupt and A/D conversion results are transferred to RAM, and then transmitted using the UART0.
   - The vector address is FFB0AH.
   - The control data for capturing A/D conversion results are allocated at FFBA0H to FFBA7H.
   - The control data for UART0 transmission are allocated at FFBA8H to FFBAFH.
   - Transfers 2-byte data of the A/D conversion result register (FFF1FH, FFF1EH) to the area at addresses from FFD80H to FFDCFH of RAM, and transfers the 1 higher-order byte (FFF1FH) of the A/D conversion result register to the UART transmit buffer (FFF10H).

Figure 27 - 22 Example of Using Chain Transfers: Consecutively Capturing A/D Conversion Results and UART0 Transmission



27.5 Points for Caution When the DTC Is to Be Used

27.5.1 Setting DTC control data and vector table

- Do not access the DTC extended special function registers (2nd SFRs), the DTC control data area, the DTC vector table area, or the general-purpose register (FFEE0H to FFEFFH) space by using DTC transfer.
- Modify the DTC base address register (DTCBAR) while all DTC activation sources are set to activation disabled.
- Do not rewrite the DTC base address register (DTCBAR) twice or more.
- Modify the data of the DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, or DTDARj register when the corresponding bit among the DTCENi[7:0] bits in the DTCENi (i = 0 to 6) register is 0 (activation disabled).
- Modify the start address of the DTC control data area to be set in the vector table when the corresponding bit among the DTCENi[7:0] bits in the DTCENi (i = 0 to 6) register is 0 (activation disabled).

27.5.2 Allocation of DTC control data area and DTC vector table area

The areas where the DTC control data and vector table can be allocated differ depending on the product and usage conditions.

- It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space as the DTC control data area or DTC vector table area.
- Make sure the stack area, the DTC control data area, and the DTC vector table area do not overlap.
- The internal RAM area in the following products cannot be used as the DTC control data area or DTC vector table area when using the on-chip debugging trace function.
  R7F101GxE (x = 7, 8, 9, A, B, E, F, G): FD300H to FD6FFH
  R7F101GxG (x = 7, 8, 9, A, B, E, F, G): FD300H to FD6FFH
- Initialize the DTRLDj register (j = 0 to 23) to 00H even in normal mode when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.
- The area where debug monitor programs are allocated when using the on-chip debugging function cannot be used as the DTC control data area or DTC vector table area. For details, see **40.4 Allocation of Memory Spaces to User Resources**.

27.5.3 DTC pending instruction

Even if a DTC transfer request is generated, DTC transfer is held pending immediately after the following instructions. Also, the DTC is not activated between PREFIX instruction code and the instruction immediately after that code.

- Call/return instruction
- Unconditional branch instruction
- Conditional branch instruction
- Read access instruction for code flash memory
- Bit manipulation instructions for IFxx, MKxx, PRxx, and PSW, and an 8-bit manipulation instruction that has the ES register as the operand
- Instruction for accessing the data flash memory
- Multiply/divide/multiply & accumulate instruction (excluding MULU)

Caution 1. When a DTC transfer request is received, all interrupt requests are held pending until DTC transfer is completed.

Caution 2. While the DTC is held pending by the DTC pending instruction, all interrupt requests are held pending.

27.5.4 Operation when accessing data flash memory space

When accessing the data flash memory space after execution of an instruction from the start of DTC data transfer, a wait for three clock cycles will be inserted to the instruction after the DTC data transfer.

Instruction 1
DTC data transfer
Instruction 2 ← A wait for three clock cycles is inserted.
MOV A, ! Data Flash memory space

27.5.5 Number of DTC execution clock cycles

Table 27 - 7 lists the operations following DTC activation and required number of cycles for each operation.

Table 27 - 7 Operations following DTC Activation and Required Number of Cycles for Each Operation

Vector Read	Control Data		Data Read	Data Write
	Read	Write-back		
1	4	**Note 1**	**Note 2**	**Note 2**

Note 1. For the number of clock cycles required for control data write-back operation, refer to **Table 27 - 8 Number of Clock Cycles Required for Control Data Write-back Operation**.

Note 2. For the number of clock cycles required for data read/write operation, refer to **Table 27 - 9 Number of Clock Cycles Required for One Data Read/Write Operation**.

Table 27 - 8 Number of Clock Cycles Required for Control Data Write-back Operation

DTCCR Register Setting				Address Setting		Control Register to Be Written Back				Number of Clock Cycles
DAMOD	SAMOD	RPTSEL	MODE	Source	Destination	DTCCTj Register	DTRLDj Register	DTSARj Register	DTDARj Register	
0	0	×	0	Fixed	Fixed	Written back	Written back	Not written back	Not written back	1
0	1	×	0	Incremented	Fixed	Written back	Written back	Written back	Not written back	2
1	0	×	0	Fixed	Incremented	Written back	Written back	Not written back	Written back	2
1	1	×	0	Incremented	Incremented	Written back	Written back	Written back	Written back	3
0	×	1	1	Repeat area	Fixed	Written back	Written back	Written back	Not written back	2
1	×	1	1		Incremented	Written back	Written back	Written back	Written back	3
×	0	0	1	Fixed	Repeat area	Written back	Written back	Not written back	Written back	2
×	1	0	1	Incremented		Written back	Written back	Written back	Written back	3

Remark j = 0 to 23; ×: 0 or 1

Table 27 - 9 Number of Clock Cycles Required for One Data Read/Write Operation

Operation	RAM	Code Flash Memory	Data Flash Memory	Special Function Register (SFR)	Extended Special Function Register (2nd SFR)	
					No Wait State	Wait States
Data read	1	2	4	1	1	1 + number of wait cycles**Note**
Data write	1	—	—	1	1	1 + number of wait cycles**Note**

Note The number of wait cycles differs depending on the specifications of the register to be accessed in the extended special function register (2nd SFR) area.

27.5.6 DTC response time

Table 27 - 10 lists the DTC response time. The DTC response time is the time from the DTC activation source being detected until DTC transfer starts. It does not include the number of DTC execution clock cycles.

Table 27 - 10 DTC Response Time

	Minimum Time	Maximum Time
Response time	3 cycles	19 cycles

Note that the response from the DTC may be further delayed under the following cases. The number of delayed clock cycles differs depending on the condition.

- Execution of an instruction from the internal RAM
  Maximum response time: 20 cycles
- Execution of a DTC pending instruction (refer to **27.5.3 DTC pending instruction**)
  Maximum response time: Maximum response time for each condition + execution clock cycles for the instruction to be held pending under the given condition.
- Access to the TRJ0 register, which entails a further wait
  Maximum response time: Maximum response time for each condition + 1 cycle

Remark 1 cycle: 1/fCLK (fCLK: CPU/peripheral hardware clock)

27.5.7 DTC activation sources

- After inputting a DTC activation source, do not input the same activation source again until DTC transfer is completed.
- While a DTC activation source is generated, do not manipulate the DTC activation enable bit corresponding to the source.
- When DTC activation sources are in contention, the CPU checks their priority levels to determine which is to be used to activate the DTC on receiving the requests for DTC transfer. For details on the priority levels of activation sources, refer to **27.3.3 Vector table**.
- When DTC activation is enabled in either of the following states, DTC transfer is started and an interrupt is generated after completion of the transfer. Therefore, check the monitor flag (CiMON) of the comparator i before enabling DTC activation as required.
  a) State 1: All of the following conditions are met.
     – Comparator i is set to an interrupt request on one-edge detection (CiEDG = 0)
     – Comparator i is set to an interrupt request on the rising edge (CiEPO = 0)
     – IVCMP > comparator i reference voltage
  b) State 2: All of the following conditions are met.
     – Comparator i is set to an interrupt request on one-edge detection (CiEDG = 0)
     – Comparator i is set to an interrupt request on the falling edge (CiEPO = 1)
     – IVCMP < comparator i reference voltage

Caution The minus-side input signal for comparator i is selected as follows
i = 0: IVREF0 or D/A converter output 0
i = 1: IVREF0, D/A converter output 0, or D/A converter output 1
i = 2: IVREF0, IVREF1, D/A converter output 0, or D/A converter output 2
(when the DACONF bit of the DAM2 register is set to 1)
i = 3: IVREF0, IVREF1, D/A converter output 0, or D/A converter output 1

27.5.8 Operation in standby mode

State	DTC Operation
HALT mode	Able to operate[Note 1]
STOP mode	DTC activation sources can be accepted[Note 2]
SNOOZE mode	Able to operate[Notes 3, 4, 5, 6]

Note 1. When the subsystem clock is selected as f_{CLK}, operation is disabled if the RTCLPC bit of the OSMC register is 1.

Note 2. In the STOP mode, detecting a DTC activation source enables transition to SNOOZE mode and DTC transfer. After completion of transfer, the chip returns to the STOP mode. However, since the code flash memory and the data flash memory are stopped during the SNOOZE mode, the flash memory cannot be set as the transfer source.

Note 3. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock is selected as f_{CLK}.

Note 4. When a transfer end interrupt from the CSIp in SNOOZE mode is being used as the DTC activation source, use the transfer end interrupt to release the chip from the SNOOZE mode and start processing by the CPU after completion of DTC transfer, or use a chain transfer to make the settings for reception by the CSIp (writing 1 to the STm0 bit of the serial channel stop register m (STm), writing 0 to the SWCm bit of the serial standby control register m (SSCm), setting the SSCm register, and writing 1 to the SSm0 bit of the serial channel start register m (SSm)) again.

Note 5. When a transfer end interrupt from the UARTq in SNOOZE mode is being used as the DTC activation source, use the transfer end interrupt to release the chip from the SNOOZE mode and start processing by the CPU after completion of DTC transfer, or use a chain transfer to make the settings for reception by the UARTq (writing 1 to the STm1 bit of the serial channel stop register m (STm), writing 0 to the SWCm bit of the serial standby control register m (SSCm), setting the SSCm register, and writing 1 to the SSm1 bit of the serial channel start register m (SSm)) again.

Note 6. When an A/D conversion end interrupt from the A/D converter in SNOOZE mode is being used as the DTC activation source, use the A/D conversion end interrupt to release the chip from the SNOOZE mode and start processing by the CPU after completion of DTC transfer, or use a chain transfer to make the settings for the SNOOZE mode function of the A/D converter (writing 1 to the AWC bit of A/D converter mode register 2 after having written 0 to it) again.

Remark 20-, 24-, and 25-pin products: p = 20; m = 1; q = 0
30- to 64-pin products: p = 00, 20; m = 0, 1; q = 0

Section 28 Event Link Controller (ELC)

28.1 Functions of ELC

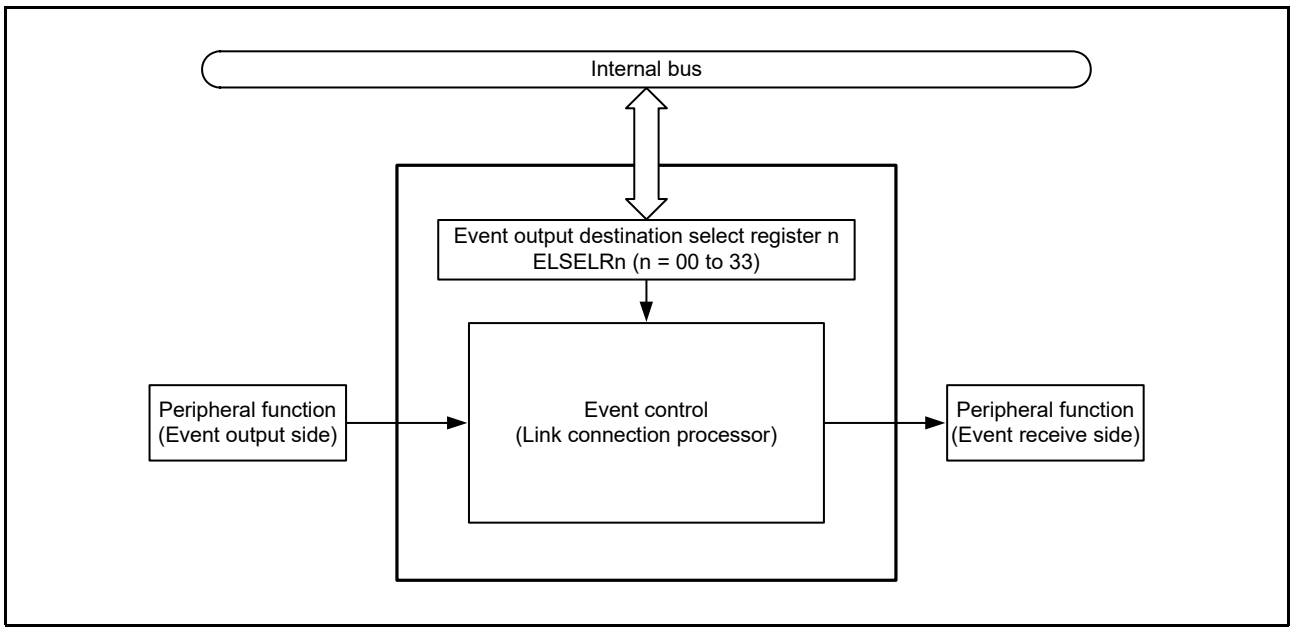
The event link controller (ELC) mutually connects (links) events output from each peripheral function. By linking events, it becomes possible to coordinate operation between peripheral functions directly without going through the CPU.
The ELC has the following functions.

- Capable of directly linking event signals from 34 types (40-, 44-, 48-, 52-, and 64-pin products), 32 types (30- and 32-pin products), 28 types (24- and 25-pin products), or 26 types (20-pin products) of peripheral functions to specified peripheral functions
- Event signals can be used as activation sources for operating any one of 19 types of peripheral functions

28.2 Configuration of ELC

Figure 28 - 1 shows the ELC block diagram.

Figure 28 - 1 ELC Block Diagram



28.3 Registers to Control the ELC

The following registers are used to control the ELC.

- Event output destination select registers n (ELSELRn) (n = 00 to 33)

28.3.1 Event output destination select registers n (ELSELRn) (n = 00 to 33)

Each ELSELRn register links an event signal to trigger the operation of an event-receiving peripheral function (link destination peripheral function) after signal 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 (n = 00 to 33) register during a period when no event output peripheral functions are generating event signals.
Table 28 - 1 lists the correspondence between ELSELRn (n = 00 to 33) registers and peripheral functions, and **Table 28 - 2** lists the correspondence between values set to ELSELRn (n = 00 to 33) registers and operation of link destination peripheral functions at reception.

Figure 28 - 2 Format of Event Output Destination Select Register n (ELSELRn)

Address: F0300H (ELSELR00) to F0321H (ELSELR33)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELSELRn	0	0	0	ELSELn4	ELSELn3	ELSELn2	ELSELn1	ELSELn0

ELSELn4	ELSELn3	ELSELn2	ELSELn1	ELSELn0	Event Link Selection
0	0	0	0	0	Event link disabled
0	0	0	0	1	Select operation of peripheral function 1 to link**Note**
0	0	0	1	0	Select operation of peripheral function 2 to link**Note**
0	0	0	1	1	Select operation of peripheral function 3 to link**Note**
0	0	1	0	0	Select operation of peripheral function 4 to link**Note**
0	0	1	0	1	Select operation of peripheral function 5 to link**Note**
0	0	1	1	0	Select operation of peripheral function 6 to link**Note**
0	0	1	1	1	Select operation of peripheral function 7 to link**Note**
0	1	0	0	0	Select operation of peripheral function 8 to link**Note**
0	1	0	0	1	Select operation of peripheral function 9 to link**Note**
0	1	0	1	0	Select operation of peripheral function 10 to link**Note**
0	1	0	1	1	Select operation of peripheral function 11 to link**Note**
0	1	1	0	0	Select operation of peripheral function 12 to link**Note**
0	1	1	0	1	Select operation of peripheral function 13 to link**Note**
0	1	1	1	0	Select operation of peripheral function 14 to link**Note**
0	1	1	1	1	Select operation of peripheral function 15 to link**Note**
1	0	0	0	0	Select operation of peripheral function 16 to link**Note**
1	0	0	0	1	Select operation of peripheral function 17 to link**Note**
1	0	0	1	0	Select operation of peripheral function 18 to link**Note**
1	0	0	1	1	Select operation of peripheral function 19 to link**Note**
Other than above					Setting prohibited

Note See **Table 28 - 2 Correspondence between Values Set to ELSELRn (n = 00 to 33) Registers and Operation of Link Destination Peripheral Functions at Reception**.

Table 28 - 1 Correspondence between ELSELRn (n = 00 to 33) 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 1Note 2	INTP1
ELSELR02	External interrupt edge detection 2Note 2	INTP2
ELSELR03	External interrupt edge detection 3Note 2	INTP3
ELSELR04	External interrupt edge detection 4Note 2	INTP4
ELSELR05	External interrupt edge detection 5Note 1	INTP5
ELSELR06	Key return signal detectionNote 3	INTKR
ELSELR07	RTC fixed-cycle signal/alarm match detection	INTRTC
ELSELR08	Timer RD2 counter 0 input capture A/compare match A	INTTRD0
ELSELR09	Timer RD2 counter 0 input capture B/compare match B	INTTRD0
ELSELR10	Timer RD2 counter 1 input capture A/compare match A	INTTRD1
ELSELR11	Timer RD2 counter 1 input capture B/compare match B	INTTRD1
ELSELR12	Timer RD2 counter 1 underflow	TRD1 underflow signal
ELSELR13	Timer RJ underflow/end of pulse width measurement period/end of pulse period measurement period	INTTRJ0
ELSELR14	Timer RG2 input capture A/compare match A	INTTRG
ELSELR15	Timer RG2 input capture B/compare match B	INTTRG
ELSELR16	Timer RG2 compare match C	INTTRG
ELSELR17	Timer RG2 compare match D	INTTRG
ELSELR18	32-bit interval timer channel 0 interval signal detection	ELCITL0
ELSELR19	End of counting or capturing by channel 0 of the timer array unit	INTTM00
ELSELR20	End of counting or capturing by channel 1 of the timer array unit	INTTM01
ELSELR21	End of counting or capturing by channel 2 of the timer array unit	INTTM02
ELSELR22	End of counting or capturing by channel 3 of the timer array unit	INTTM03
ELSELR23	Comparator detection 0	INTCMP0
ELSELR24	Comparator detection 1	INTCMP1
ELSELR25	Comparator detection 2	INTCMP2
ELSELR26	Comparator detection 3Note 1	INTCMP3
ELSELR27	FAA timing compare match 3	INTTIMEC3
ELSELR28	FAA timing compare match 4	INTTIMEC4
ELSELR29	FAA timing compare match 5	INTTIMEC5
ELSELR30	A/D conversion end 0	ELCAD0
ELSELR31	A/D conversion end 1	ELCAD1
ELSELR32	A/D conversion end 2	ELCAD2
ELSELR33	A/D conversion end 3	ELCAD3

Note 1. This is only applicable to the 24- to 64-pin products.
Note 2. This is only applicable to the 30- to 64-pin products.
Note 3. This is only applicable to the 40- to 64-pin products.

Table 28 - 2 Correspondence between Values Set to ELSELRn (n = 00 to 33) Registers and Operation of Link Destination Peripheral Functions at Reception

Bits ELSELn4 to ELSELn0 in the ELSELRn Registers	Link Destination Number	Link Destination Peripheral Function	Operation When Receiving Event
00001B	1	A/D converter	A/D conversion starts
00010B	2	Timer input of timer array unit channel 0Note 1	Delay counter, input pulse interval measurement, external event counter
00011B	3	Timer input of timer array unit channel 1Note 2	Delay counter, input pulse interval measurement, external event counter
00100B	4	Timer RJ	Count source
00101B	5	Timer RG2	TRGIOB input capture
00110B	6	Timer RD2 event input 0	TRDIOD0 input capture, pulse output forced cutoff
00111B	7	Timer RD2 event input 1	TRDIOD1 input capture, pulse output forced cutoff
01000B	8	D/A converter 0 (DAC0)Note 3	Realtime output
01001B	9	D/A converter 1 (DAC1)Note 3	Realtime output
01010B	10	D/A converter 2 (DAC2)Note 3	Realtime output
01011B	11	Timer RD2 PWM option unit A (PWMOPA)	Pulse output forced cutoff
01100B	12	FAA	Input event detection interrupt 0
01101B	13	FAA	Input event detection interrupt 1
01110B	14	FAA	Input event detection interrupt 2
01111B	15	FAA	Input event detection interrupt 3
10000B	16	FAA	Input event detection interrupt 4
10001B	17	FAA	Input event detection interrupt 5
10010B	18	FAA	Input event detection interrupt 6
10011B	19	FAA	Input event detection interrupt 7

Note 1. To select the timer input of timer array unit 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 I/O select register 0 (TIOS0).

Note 2. To select the timer input of timer array unit 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 I/O select register 0 (TIOS0).

Note 3. When entering the STOP mode while the realtime output event mode for D/A conversion is enabled, disable linking of ELC events before entering STOP mode.

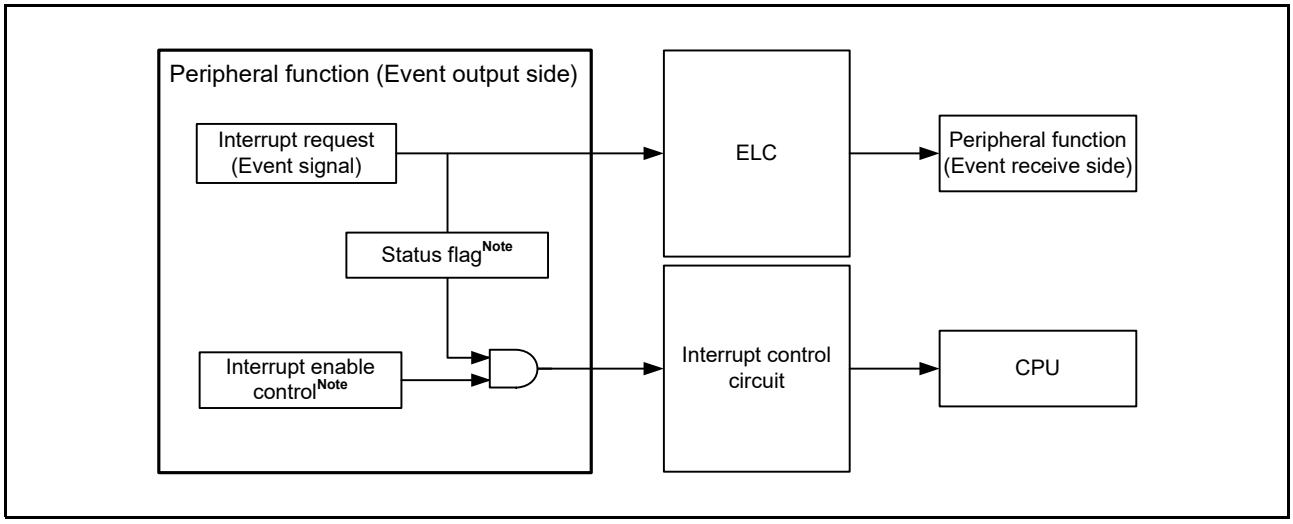
28.4 ELC Operation

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 28 - 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 28 - 2 Correspondence between Values Set to ELSELRn (n = 00 to 33) Registers and Operation of Link Destination Peripheral Functions at Reception**.

Figure 28 - 3 Relationship between Interrupt Handling and ELC



Note Not available depending on the peripheral function.

Table 28 - 3 lists the response of peripheral functions that receive events.

Table 28 - 3 Response of Peripheral Functions That Receive Events (1/2)

Event Receiver No.	Event Link Destination Function	Operation after Event Reception	Response
1	A/D converter	A/D conversion starts	The edge is detected 3 or 4 cycles of fCLK after an ELC event is generated.
2	Timer input of timer array unit channel 0	Delay counter, input pulse width measurement, external event counter	The edge is detected 3 or 4 cycles of fCLK after an ELC event is generated.
3	Timer input of timer array unit channel 1	Delay counter, input pulse width measurement, external event counter	The edge is detected 3 or 4 cycles of fCLK 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 RG2	TRGIOB input capture	A count start trigger is generated 2 or 3 cycles of fCLK after an ELC event is generated.
6	Timer RD2 event input 0	TRDIOD0 input capture	A count start trigger is generated 2 or 3 cycles of the timer RD2 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 RD2 operating clock after an ELC event is generated.
7	Timer RD2 event input 1	TRDIOD1 input capture	A count start trigger is generated 2 or 3 cycles of the timer RD2 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 RD2 operating clock after an ELC event is generated.
8	D/A converter 0 (DAC0)	Realtime output (channel 0)	To synchronize asynchronous events from ELC, receive the event during the period (1 cycle or more and less than 2 cycles), and start the D/A conversion operation in the next cycle.
9	D/A converter 1 (DAC1)	Realtime output (channel 1)	To synchronize asynchronous events from ELC, receive the event during the period (1 cycle or more and less than 2 cycles), and start the D/A conversion operation in the next cycle.
10	D/A converter 2 (DAC2)	Realtime output (channel 2)	To synchronize asynchronous events from ELC, receive the event during the period (1 cycle or more and less than 2 cycles), and start the D/A conversion operation in the next cycle.
11	Timer RD2 PWM option unit A (PWMOPA)	Pulse output forced cutoff	To synchronize asynchronous events from ELC, receive the event during the period (1 cycle or more and less than 2 cycles), and start the pulse output forced cutoff operation in the next cycle.
12	FAA	Input event detection interrupt 0	An event from the ELC is directly used as the input event of the FAA.
13	FAA	Input event detection interrupt 1	An event from the ELC is directly used as the input event of the FAA.
14	FAA	Input event detection interrupt 2	An event from the ELC is directly used as the input event of the FAA.
15	FAA	Input event detection interrupt 3	An event from the ELC is directly used as the input event of the FAA.
16	FAA	Input event detection interrupt 4	An event from the ELC is directly used as the input event of the FAA.

Table 28 - 3 Response of Peripheral Functions That Receive Events (2/2)

Event Receiver No.	Event Link Destination Function	Operation after Event Reception	Response
17	FAA	Input event detection interrupt 5	An event from the ELC is directly used as the input event of the FAA.
18	FAA	Input event detection interrupt 6	An event from the ELC is directly used as the input event of the FAA.
19	FAA	Input event detection interrupt 7	An event from the ELC is directly used as the input event of the FAA.

Section 29 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.

		20-pin	24- and 25-pin	30- and 32-pin	40- and 44-pin	48-, 52-, and 64-pin
Maskable interrupts	External	6	8	12	13	15
	Internal	46	55			

29.1 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, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H). 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. For default priority, see **Table 29 - 1 Interrupt Source List**. 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.

29.2 Interrupt Sources and Configuration

Interrupt sources include maskable interrupts and a software interrupt. In addition, they also have up to seven reset sources (see **Table 29 - 1 Interrupt Source List**). The vector codes that specify 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 29 - 1 Interrupt Source List (1/4)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2	64-pin	52-pin	48-pin	44-pin	40-pin	32-pin	30-pin	25-pin	24-pin	20-pin
		Name	Trigger													
Maskable	0	INTWDTI	Watchdog timer interval Note 3 (75% of overflow time + 1/4 fIL)	Internal	00004H	<a>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	1	INTLVI	Voltage detection Note 4		00006H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	2	INTP0	Pin input edge detection	External	00008H	<b>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	3	INTP1			0000AH		✓	✓	✓	✓	✓	✓	✓	—	—	—
		INTAD1	End of A/D input channel 1 conversion Note 8	Internal		<a>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	4	INTP2	Pin input edge detection	External	0000CH	<b>	✓	✓	✓	✓	✓	✓	✓	—	—	—
		INTAD2	End of A/D input channel 2 conversion Note 8	Internal		<a>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	5	INTP3	Pin input edge detection	External	0000EH	<b>	✓	✓	✓	✓	✓	✓	✓	—	—	—
	6	INTP4			00010H		✓	✓	✓	✓	✓	✓	✓	—	—	—
	7	INTP5			00012H		✓	✓	✓	✓	✓	✓	✓	✓	✓	—
	8	INTST2/ INTCSI20/ INTIIC20	UART2 transmission transfer end or buffer empty interrupt/CSI20 transfer end or buffer empty interrupt/IIC20 transfer end	Internal	00014H	<a>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	9	INTSR2/ INTCSI21/ INTIIC21	UART2 reception transfer end/CSI21 transfer end or buffer empty interrupt/IIC21 transfer end		00016H		✓	✓	✓	✓	✓	**Note 5**	**Note 5**	**Note 5**	**Note 5**	**Note 5**
	10	INTSRE2	UART2 reception communication error occurrence		00018H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	11	INTFAAE	FAA end interrupt		0001AH		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	12	INTTIMEC0	FAA timing compare match 0		0001CH		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	13	INTST0/ INTCSI00/ INTIIC00	UART0 transmission transfer end or buffer empty interrupt/CSI00 transfer end or buffer empty interrupt/IIC00 transfer end		0001EH		✓	✓	✓	✓	✓	✓	✓	✓	✓	—
	14	INTTM00	End of timer channel 00 count or capture		00020H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	15	INTSRE0	UART0 reception communication error occurrence		00022H		✓	✓	✓	✓	✓	✓	✓	✓	✓	—
		INTTM01H	End of timer channel 01 count or capture (at higher 8-bit timer operation)				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	16	INTST1/ INTCSI10/ INTIIC10	UART1 transmission transfer end or buffer empty interrupt/CSI10 transfer end or buffer empty interrupt/IIC10 transfer end		00024H		✓	**Note 6**	**Note 6**	**Note 6**	**Note 6**	**Note 6**	**Note 6**	**Note 6**	**Note 6**	**Note 6**
	17	INTSR1/ INTCSI11/ INTIIC11	UART1 reception transfer end/CSI11 transfer end or buffer empty interrupt/IIC11 transfer end		00026H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 29 - 1 Interrupt Source List (2/4)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2	64-pin	52-pin	48-pin	44-pin	40-pin	32-pin	30-pin	25-pin	24-pin	20-pin
		Name	Trigger													
Maskable	18	INTSRE1	UART1 reception communication error occurrence	Internal	00028H	<a>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		INTTM03H	End of timer channel 03 count or capture (at higher 8-bit timer operation)				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	19	INTIICA0	End of IICA0 communication		0002AH		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	20	INTSR0/ INTCSI01/ INTIIC01	UART0 reception transfer end/CSI01 transfer end or buffer empty interrupt/IIC01 transfer end		0002CH		✓	✓	✓	**Note 7**	**Note 7**	**Note 7**	**Note 7**	**Note 7**	**Note 7**	—
	21	INTTM01	End of timer channel 01 count or capture (at 16-bit/lower 8-bit timer operation)		0002EH		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	22	INTTM02	End of timer channel 02 count or capture		00030H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	23	INTTM03	End of timer channel 03 count or capture (at 16-bit/lower 8-bit timer operation)		00032H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	24	INTAD0	End of A/D input channel 0 conversion Note 8		00034H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	25	INTRTC	Fixed-cycle signal of realtime clock/alarm match detection		00036H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	26	INTITL	Interval signal of 32-bit interval timer detection		00038H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	27	INTKR	Key return signal detection	External	0003AH	<c>	✓	✓	✓	✓	✓	—	—	—	—	—
	28	INTTD	DALI transmission transfer end	Internal	0003CH	<a>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		INTTRJ0	Timer RJ interrupt				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	29	INTTRD0	Timer RD2 counter 0 input capture, compare match, overflow, underflow interrupt		0003EH		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	30	INTTRD1	Timer RD2 counter 1 input capture, compare match, overflow, underflow interrupt		00040H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	31	INTRD	DALI reception transfer end		00042H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		INTTRG	Timer RG2 input capture, compare match, overflow, underflow interrupt				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	32	INTED/ INTCLD/ INTBPD	DALI error interrupt/collision detection interrupt/bus power-down detection interrupt		00044H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		INTTRX	Timer RX overflow detection				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 29 - 1 Interrupt Source List (3/4)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2	64-pin	52-pin	48-pin	44-pin	40-pin	32-pin	30-pin	25-pin	24-pin	20-pin
		Name	Trigger													
Maskable	33	INTP20	Pin input edge detection	External	00046H	<b>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	34	INTP21	Pin input edge detection		00048H		✓	✓	✓	✓	✓	✓	✓	✓	✓	—
	35	INTP6	Pin input edge detection		0004AH		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		INTTMKBSTR10	16-bit timer KB31 forced output stop 1 or 2 activation interrupt 0	Internal		<a>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	36	INTP7	Pin input edge detection	External	0004CH	<b>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		INTTMKBSTP10	16-bit timer KB31 forced output stop 1 or 2 termination interrupt 0	Internal		<a>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	37	INTP8	Pin input edge detection	External	0004EH	<b>	✓	✓	✓	—	—	—	—	—	—	—
		INTTMKBSTR11	16-bit timer KB31 forced output stop 1 or 2 activation interrupt 1	Internal		<a>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	38	INTP9	Pin input edge detection	External	00050H	<b>	✓	✓	✓	—	—	—	—	—	—	—
		INTTMKBSTP11	16-bit timer KB31 forced output stop 1 or 2 termination interrupt 1	Internal		<a>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	39	INTFL	Reserved	—	00052H	<b>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	40	INTP10	Pin input edge detection	External	00054H		✓	✓	—	—	—	—	—	—	—	—
		INTCMP0	Comparator detection 0	Internal			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	41	INTP11	Pin input edge detection	External	00056H		✓	✓	—	—	—	—	—	—	—	—
		INTCMP1	Comparator detection 1	Internal			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	42	INTFAATRAP	FAA illegal instruction interrupt		00058H	<a>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	43	INTCMP2	Comparator detection 2		0005AH	<b>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	44	INTCMP3	Comparator detection 3		0005CH		✓	✓	✓	✓	✓	✓	✓	✓	✓	—
	45	INTTMKB0	End of 16-bit timer KB30 count		0005EH	<a>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	46	INTTMKB1	End of 16-bit timer KB31 count		00060H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	47	INTTMKB2	End of 16-bit timer KB32 count		00062H		✓	✓	✓	✓	✓	✓	✓	✓	✓	—
	48	INTSDD	DALI stop condition detection interrupt		00064H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		INTGCR	Timer RG2 phase counting clearance detection interrupt				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	49	INTFED	DALI falling edge detection interrupt		00066H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		INTPMC	Timer RG2 phase change detection interrupt				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	50	INTTMKBSTR00	16-bit timer KB30 forced output stop 1 or 2 activation interrupt 0		00068H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 29 - 1 Interrupt Source List (4/4)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2	64-pin	52-pin	48-pin	44-pin	40-pin	32-pin	30-pin	25-pin	24-pin	20-pin
		Name	Trigger													
Maskable	51	INTTMKBSTP00	16-bit timer KB30 forced output stop 1 or 2 termination interrupt 0	Internal	0006AH	<a>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	52	INTTMKBSTR01	16-bit timer KB30 forced output stop 1 or 2 activation interrupt 1		0006CH		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	53	INTTMKBSTP01	16-bit timer KB30 forced output stop 1 or 2 termination interrupt 1		0006EH		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	54	INTTMKBSTR20	16-bit timer KB32 forced output stop 1 or 2 activation interrupt 0		00070H		✓	✓	✓	✓	✓	✓	✓	✓	✓	—
	55	INTTMKBSTP20	16-bit timer KB32 forced output stop 1 or 2 termination interrupt 0		00072H		✓	✓	✓	✓	✓	✓	✓	✓	✓	—
	56	INTTMKBSTR21	16-bit timer KB32 forced output stop 1 or 2 activation interrupt 1		00074H		✓	✓	✓	✓	✓	✓	✓	✓	✓	—
	57	INTTMKBSTP21	16-bit timer KB32 forced output stop 1 or 2 termination interrupt 1		00076H		✓	✓	✓	✓	✓	✓	✓	✓	✓	—
	58	INTTIMEC1	FAA timing compare match 1		00078H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	59	INTTIMEC2	FAA timing compare match 2		0007AH		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	60	INTAD3	End of A/D input channel 3 conversion Note 8		0007CH		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Software	—	BRK	Execution of BRK instruction	—	0007EH	<d>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Reset	—	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$ pin input	—	00000H	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		POR	Power-on-reset				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		LVD	Voltage detection				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		WDT	Overflow of watchdog timer				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		TRAP	Execution of illegal instruction				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		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 60 indicates the lowest priority.

Note 2. Basic configuration types <a> to <d> correspond to <a> to <d> in **Figure 29 - 1 Basic Configuration of Interrupt Function**.

Note 3. When the value of bit 7 (WDTINT) in the user option byte (000C0H) is 1.

Note 4. When the value of bit 6 (LVD1SEL) in the voltage detection level register (LVIS) is 0 or the value of bit 7 (LVD0SEL) in the user option byte (000C1H) is 1.

Note 5. INTSR2 is only present in this product.

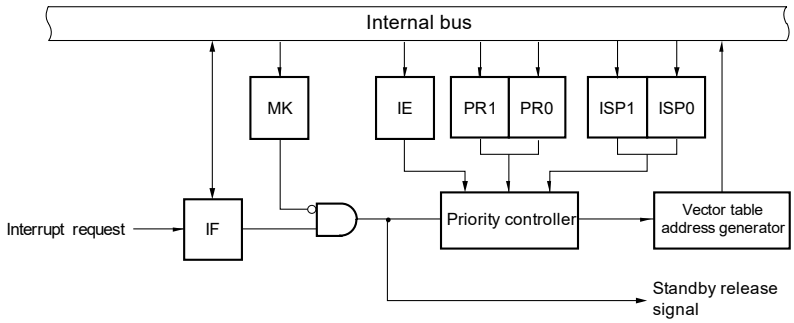
Note 6. INTST1 is only present in this product.

Note 7. INTSR0 is only present in this product.

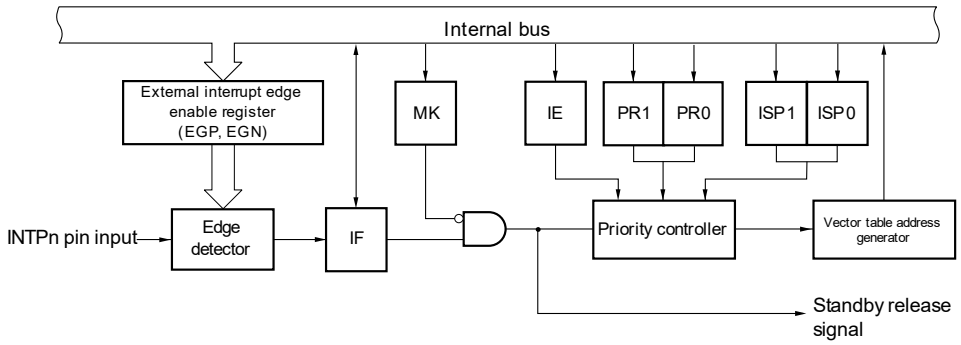
Note 8. When the value of bit 0 (ADVMOD) in the A/D converter mode register 3 (ADM3) is 1.

Figure 29 - 1 Basic Configuration of Interrupt Function (1/2)

<a> Internal maskable interrupt



<b> External maskable interrupt (INTPn)

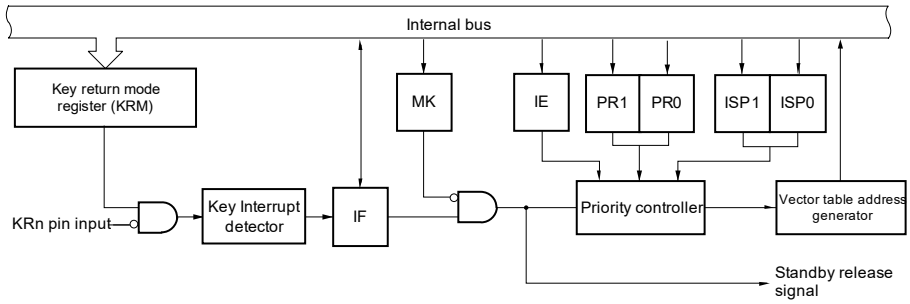


Remark 1. IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

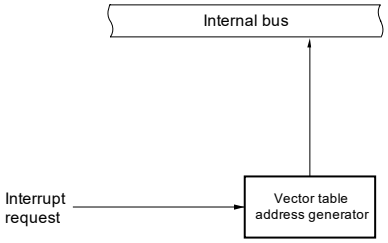
Remark 2. 20-pin: n = 0, 6, 7, and 20; m = 0 to 2
24- and 25-pin: n = 0, 5 to 7, 20, and 21; m = 0 to 3
30-, 32-, 40-, and 44-pin: n = 0 to 7, 20, and 21; m = 0 to 3
48-pin: n = 0 to 9, 20, and 21; m = 0 to 3
52- and 64-pin: n = 0 to 11, 20, and 21; m = 0 to 3

Figure 29 - 1 Basic Configuration of Interrupt Function (2/2)

<c> External maskable interrupt (INTKR)



<d> Software interrupt



Remark 1. 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 2. 40- and 44-pin: n = 0 to 3
48-pin: n = 0 to 5
52- and 64-pin: n = 0 to 7

29.3 Registers to Control the Interrupt Functions

The following registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H)
- Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H)
- Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H)
- External interrupt rising edge enable registers (EGP0, EGP1), external interrupt falling edge enable registers (EGN0, EGN1)
- Program status word (PSW)
- Port mode registers xx (PMxx) (xx = 0 to 5, 7, 12, 14)
- Port mode control A registers xx (PMCAxx) (xx = 2)

Table 29 - 2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 29 - 2 Flags Corresponding to Interrupt Request Sources (1/4)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		64-pin	52-pin	48-pin	44-pin	40-pin	32-pin	30-pin	25-pin	24-pin	20-pin
		Register		Register		Register										
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTLVI	LVIIF Note 1		LVIMK Note 1		LVIPR0, LVIPR1 Note 1		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP0	PIF0		PMK0		PPR00, PPR10		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP1	PIF1 Note 24		PMK1 Note 24		PPR01, PPR11 Note 24		✓	✓	✓	✓	✓	✓	✓	—	—	—
INTAD1	ADIF1 Note 24		ADMK1 Note 24		ADPR01, ADPR11 Note 24		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP2	PIF2 Note 25		PMK2 Note 25		PPR02, PPR12 Note 25		✓	✓	✓	✓	✓	✓	✓	—	—	—
INTAD2	ADIF2 Note 25		ADMK2 Note 25		ADPR02, ADPR12 Note 25		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP3	PIF3		PMK3		PPR03, PPR13		✓	✓	✓	✓	✓	✓	✓	—	—	—
INTP4	PIF4		PMK4		PPR04, PPR14		✓	✓	✓	✓	✓	✓	✓	—	—	—
INTP5	PIF5		PMK5		PPR05, PPR15		✓	✓	✓	✓	✓	✓	✓	✓	✓	—

Table 29 - 2 Flags Corresponding to Interrupt Request Sources (2/4)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		64-pin	52-pin	48-pin	44-pin	40-pin	32-pin	30-pin	25-pin	24-pin	20-pin
		Register		Register		Register										
INTST2	STIF2Note 2	IF0H	STMK2Note 2	MK0H	STPR02, STPR12Note 2	PR00H, PR10H	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTCSI20	CSIIF20Note 2		CSIMK20Note 2		CSIPR020, CSIPR120Note 2		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTIIC20	IICIF20Note 2		IICMK20Note 2		IICPR020, IICPR120Note 2		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTSR2	SRIF2Note 3		SRMK2Note 3		SRPR02, SRPR12Note 3		✓	✓	✓	✓	✓	Note 21	Note 21	Note 21	Note 21	Note 21
INTCSI21	CSIIF21Note 3		CSIMK21Note 3		CSIPR021, CSIPR121Note 3		✓	✓	✓	✓	✓	Note 21	Note 21	Note 21	Note 21	Note 21
INTIIC21	IICIF21Note 3		IICMK21Note 3		IICPR021, IICPR121Note 3		✓	✓	✓	✓	✓	Note 21	Note 21	Note 21	Note 21	Note 21
INTSRE2	SREIF2		SREMK2		SREPR02, SREPR12		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTFAAE	FAAEIF		FAAEMK		FAAEPR0, FAAEPR1		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTIMEC0	TIMECIF0		TIMECMK0		TIMECPR00, TIMECPR10		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTST0	STIF0Note 4		STMK0Note 4		STPR00, STPR10Note 4		✓	✓	✓	✓	✓	✓	✓	✓	✓	—
INTCSI00	CSIIF00Note 4		CSIMK00Note 4		CSIPR000, CSIPR100Note 4		✓	✓	✓	✓	✓	✓	✓	✓	✓	—
INTIIC00	IICIF00Note 4		IICMK00Note 4		IICPR000, IICPR100Note 4		✓	✓	✓	✓	✓	✓	✓	✓	✓	—
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTSRE0	SREIF0Note 5		SREMK0Note 5		SREPR00, SREPR10Note 5		✓	✓	✓	✓	✓	✓	✓	✓	✓	—
INTTM01H	TMIF01HNote 5		TMMK01HNote 5		TMPR001H, TMPR101HNote 5		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTST1	STIF1Note 6	IF1L	STMK1Note 6	MK1L	STPR01, STPR11Note 6	PR01L, PR11L	✓	Note 22	Note 22	Note 22	Note 22	Note 22	Note 22	Note 22	Note 22	Note 22
INTCSI10	CSIIF10Note 6		CSIMK10Note 6		CSIPR010, CSIPR110Note 6		✓	Note 22	Note 22	Note 22	Note 22	Note 22	Note 22	Note 22	Note 22	Note 22
INTIIC10	IICIF10Note 6		IICMK10Note 6		IICPR010, IICPR110Note 6		✓	Note 22	Note 22	Note 22	Note 22	Note 22	Note 22	Note 22	Note 22	Note 22
INTSR1	SRIF1Note 7		SRMK1Note 7		SRPR01, SRPR11Note 7		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTCSI11	CSIIF11Note 7		CSIMK11Note 7		CSIPR011, CSIPR111Note 7		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTIIC11	IICIF11Note 7		IICMK11Note 7		IICPR011, IICPR111Note 7		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTSRE1	SREIF1Note 8		SREMK1Note 8		SREPR01, SREPR11Note 8		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTM03H	TMIF03HNote 8		TMMK03HNote 8		TMPR003H, TMPR103HNote 8		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTIICA0	IICAIF0		IICAMK0		IICAPR00, IICAPR10		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTSR0	SRIF0Note 9		SRMK0Note 9		SRPR00, SRPR10Note 9		✓	✓	✓	Note 23	Note 23	Note 23	Note 23	Note 23	Note 23	—
INTCSI01	CSIIF01Note 9		CSIMK01Note 9		CSIPR001, CSIPR101Note 9		✓	✓	✓	Note 23	Note 23	Note 23	Note 23	Note 23	Note 23	—
INTIIC01	IICIF01Note 9		IICMK01Note 9		IICPR001, IICPR101Note 9		✓	✓	✓	Note 23	Note 23	Note 23	Note 23	Note 23	Note 23	—
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTM02	TMIF02		TMMK02		TMPR002, TMPR102		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 29 - 2 Flags Corresponding to Interrupt Request Sources (3/4)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		64-pin	52-pin	48-pin	44-pin	40-pin	32-pin	30-pin	25-pin	24-pin	20-pin
		Register		Register		Register										
INTAD0	ADIF0	IF1H	ADMK0	MK1H	ADPR00, ADPR10	PR01H, PR11H	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTRTC	RTCIF		RTCMK		RTCPR0, RTCPR1		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTITL	ITLIF		ITLMK		ITLPR0, ITLPR1		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTKR	KRIF		KRMK		KRPR0, KRPR1		✓	✓	✓	✓	✓	—	—	—	—	—
INTTD	TDIF Note 10		TDMK Note 10		TDPR0, TDPR1 Note 10		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTRJ0	TRJIF0 Note 10		TRJMK0 Note 10		TRJPR00, TRJPR10 Note 10		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTRD0	TRDIF0		TRDMK0		TRDPR00, TRDPR10		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTRD1	TRDIF1		TRDMK1		TRDPR01, TRDPR11		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTRD	RDIF Note 11		RDMK Note 11		RDPR0, RDPR1 Note 11		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTRG	TRGIF Note 11		TRGMK Note 11		TRGPR0, TRGPR1 Note 11		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTED	EDIF Note 12	IF2L	EDMK Note 12	MK2L	EDPR0, EDPR1 Note 12	PR02L, PR12L	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTCLD	CLDIF Note 12		CLDMK Note 12		CLDPR0, CLDPR1 Note 12		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTBPD	BPDIF Note 12		BPDMK Note 12		BPDPR0, BPDPR1 Note 12		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTRX	TRXIF Note 12		TRXMK Note 12		TRXPR0, TRXPR1 Note 12		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP20	PIF20		PMK20		PPR020, PPR120		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP21	PIF21		PMK21		PPR021, PPR121		✓	✓	✓	✓	✓	✓	✓	✓	✓	—
INTP6	PIF6 Note 13		PMK6 Note 13		PPR06, PPR16 Note 13		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTMKBSTR10	TMKBSTRIF10 Note 13		TMKBSTRMK10 Note 13		TMKBSTRPR010, TMKBSTRPR110 Note 13		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP7	PIF7 Note 14		PMK7 Note 14		PPR07, PPR17 Note 14		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTMKBSTP10	TMKBSTPIF10 Note 14		TMKBSTPMK10 Note 14		TMKBSTPPR010, TMKBSTPPR110 Note 14		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP8	PIF8 Note 15		PMK8 Note 15		PPR08, PPR18 Note 15		✓	✓	✓	—	—	—	—	—	—	—
INTTMKBSTR11	TMKBSTRIF11 Note 15		TMKBSTRMK11 Note 15		TMKBSTRPR011, TMKBSTRPR111 Note 15		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP9	PIF9 Note 16		PMK9 Note 16		PPR09, PPR19 Note 16		✓	✓	✓	—	—	—	—	—	—	—
INTTMKBSTP11	TMKBSTPIF11 Note 16		TMKBSTPMK11 Note 16		TMKBSTPPR011, TMKBSTPPR111 Note 16		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTFL	FLIF		FLMK		FLPR0, FLPR1		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP10	PIF10 Note 17	IF2H	PMK10 Note 17	MK2H	PPR010, PPR110 Note 17	PR02H, PR12H	✓	✓	—	—	—	—	—	—	—	—
INTCMP0	CMPIF0 Note 17		CMPMK0 Note 17		CMPPR00, CMPPR10 Note 17		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP11	PIF11 Note 18		PMK11 Note 18		PPR011, PPR111 Note 18		✓	✓	—	—	—	—	—	—	—	—
INTCMP1	CMPIF1 Note 18		CMPMK1 Note 18		CMPPR01, CMPPR11 Note 18		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTFAATRAP	FAATRAPIF		FAATRAPMK		FAATRAPPR0, FAATRAPPR1		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTCMP2	CMPIF2		CMPMK2		CMPPR02, CMPPR12		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTCMP3	CMPIF3		CMPMK3		CMPPR03, CMPPR13		✓	✓	✓	✓	✓	✓	✓	✓	✓	—

Table 29 - 2 Flags Corresponding to Interrupt Request Sources (4/4)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		64-pin	52-pin	48-pin	44-pin	40-pin	32-pin	30-pin	25-pin	24-pin	20-pin
		Register		Register		Register										
INTTMKB0	TMKBIF0	IF2H	TMKBMK0	MK2H	TMKBPR00, TMKBPR10	PR02H, PR12H	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTMKB1	TMKBIF1		TMKBMK1		TMKBPR01, TMKBPR11		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTMKB2	TMKBIF2		TMKBMK2		TMKBPR02, TMKBPR12		✓	✓	✓	✓	✓	✓	✓	✓	✓	—
INTSDD	SDDIF Note 19	IF3L	SDDMK Note 19	MK3L	SDDPR0, SDDPR1 Note 19	PR03L, PR13L	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTGCR	GCRIF Note 19		GCRMK Note 19		GCRPR0, GCRPR1 Note 19		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTFED	FEDIF Note 20		FEDMK Note 20		FEDPR0, FEDPR1 Note 20		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTPMC	PMCIF Note 20		PMCMK Note 20		PMCPR0, PMCPR1 Note 20		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTMKBSTR00	TMKBSTRIF00		TMKBSTRMK00		TMKBSTRPR000, TMKBSTRPR100		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTMKBSTP00	TMKBSTPIF00		TMKBSTPMK00		TMKBSTPPR000, TMKBSTPPR100		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTMKBSTR01	TMKBSTRIF01		TMKBSTRMK01		TMKBSTRPR001, TMKBSTRPR101		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTMKBSTP01	TMKBSTPIF01		TMKBSTPMK01		TMKBSTPPR001, TMKBSTPPR101		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTMKBSTR20	TMKBSTRIF20		TMKBSTRMK20		TMKBSTRPR020, TMKBSTRPR120		✓	✓	✓	✓	✓	✓	✓	✓	✓	—
INTTMKBSTP20	TMKBSTPIF20		TMKBSTPMK20		TMKBSTPPR020, TMKBSTPPR120		✓	✓	✓	✓	✓	✓	✓	✓	✓	—
INTTMKBSTR21	TMKBSTRIF21	IF3H	TMKBSTRMK21	MK3H	TMKBSTRPR021, TMKBSTRPR121	PR03H, PR13H	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
INTTMKBSTP21	TMKBSTPIF21		TMKBSTPMK21		TMKBSTPPR021, TMKBSTPPR121		✓	✓	✓	✓	✓	✓	✓	✓	✓	—
INTTIMEC1	TIMECIF1		TIMECMK1		TIMECPR01, TIMECPR11		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTIMEC2	TIMECIF2		TIMECMK2		TIMECPR02, TIMECPR12		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTAD3	ADIF3		ADMK3		ADPR03, ADPR13		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note 1. The DLVD0F and DLVD1F bits of the LVIM register can be used to confirm which voltage detector has issued the given interrupt, LVD0 or LVD1. For details, see **34.3.1 Voltage detection register (LVIM)**.

Note 2. If any of the interrupt sources INTST2, INTCSI20, and INTIIC20 is generated, bit 0 of the IF0H register is set to 1. Bit 0 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.

Note 3. If any of the interrupt sources INTSR2, INTCSI21, and INTIIC21 is generated, bit 1 of the IF0H register is set to 1. Bit 1 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.

Note 4. If any 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 5. Do not use a UART0 reception error interrupt and an interrupt of channel 1 of the TAU (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. If the UART0 reception error interrupt is not used (EOC01 = 0), UART0 and channel 1 of the TAU (at higher 8-bit timer operation) can be used at the same time. If either of the interrupt sources INTSRE0 and INTTM01H is generated, bit 7 of the IF0H register is set to 1. Bit 7 of the MK0H, PR00H, and PR10H registers supports these two interrupt sources.

Note 6. If any of the interrupt sources INTST1, INTCSI10, and INTIIC10 is generated, bit 0 of the IF1L register is set to 1. Bit 0 of the MK1L, PR01L, and PR11L registers supports these three interrupt sources.

Note 7. If any of the interrupt sources INTSR1, INTCSI11, and INTIIC11 is generated, bit 1 of the IF1L register is set to 1. Bit 1 of the MK1L, PR01L, and PR11L registers supports these three interrupt sources.

Note 8. Do not use a UART1 reception error interrupt and an interrupt of channel 3 of the TAU (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. If the UART1 reception error interrupt is not used (EOC03 = 0), UART1 and channel 3 of the TAU (at higher 8-bit timer operation) can be used at the same time. If either of the interrupt sources INTSRE1 and INTTM03H is generated, bit 2 of the IF1L register is set to 1. Bit 2 of the MK1L, PR01L, and PR11L registers supports these two interrupt sources.

Note 9. If any of the interrupt sources INTSR0, INTCSI01, and INTIIC01 is generated, bit 4 of the IF1L register is set to 1. Bit 4 of the MK1L, PR01L, and PR11L registers supports these three interrupt sources.

Note 10. If either of the interrupt sources INTTD and INTTRJ0 is generated, bit 4 of the IF1H register is set to 1. Bit 4 of the MK1H, PR01H, and PR11H registers supports these two interrupt sources.

Note 11. If either of the interrupt sources INTRD and INTTRG is generated, bit 7 of the IF1H register is set to 1. Bit 7 of the MK1H, PR01H, and PR11H registers supports these two interrupt sources.

Note 12. If any of the interrupt sources INTED, INTCLD, INTBPD, and INTTRX is generated, bit 0 of the IF2L register is set to 1. Bit 0 of the MK2L, PR02L, and PR12L registers supports these four interrupt sources.

Note 13. If either of the interrupt sources INTP6 and INTTMKBSTR10 is generated, bit 3 of the IF2L register is set to 1. Bit 3 of the MK2L, PR02L, and PR12L registers supports these two interrupt sources.

Note 14. If either of the interrupt sources INTP7 and INTTMKBSTP10 is generated, bit 4 of the IF2L register is set to 1. Bit 4 of the MK2L, PR02L, and PR12L registers supports these two interrupt sources.

Note 15. If either of the interrupt sources INTP8 and INTTMKBSTR11 is generated, bit 5 of the IF2L register is set to 1. Bit 5 of the MK2L, PR02L, and PR12L registers supports these two interrupt sources.

Note 16. If either of the interrupt sources INTP9 and INTTMKBSTP11 is generated, bit 6 of the IF2L register is set to 1. Bit 6 of the MK2L, PR02L, and PR12L registers supports these two interrupt sources.

Note 17. Do not use the external interrupt INTP10 and the internal interrupt INTCMP0 at the same time because they share the same flag as interrupt request sources. If an interrupt from either the INTP10 or INTCMP0 source is generated, bit 0 of the IF2H register is set to 1. Bit 0 of the MK2H, PR02H, and PR12H registers supports these two interrupt sources.

Note 18. Do not use the external interrupt INTP11 and the internal interrupt INTCMP1 at the same time because they share the same flag as interrupt request sources. If an interrupt from either the INTP11 or INTCMP1 source is generated, bit 1 of the IF2H register is set to 1. Bit 1 of the MK2H, PR02H, and PR12H registers supports these two interrupt sources.

Note 19. If either of the interrupt sources INTSDD and INTGCR is generated, bit 0 of the IF3L register is set to 1. Bit 0 of the MK3L, PR03L, and PR13L registers supports these two interrupt sources.

Note 20. If either of the interrupt sources INTFED and INTPMC is generated, bit 1 of the IF3L register is set to 1. Bit 1 of the MK3L, PR03L, and PR13L registers supports these two interrupt sources.

Note 21. INTSR2 is only present in this product.

Note 22. INTST1 is only present in this product.

Note 23. INTSR0 is only present in this product.

Note 24. If either of the interrupt sources INTP1 and INTAD1 is generated, bit 3 of the IF0L register is set to 1. Bit 3 of the MK0L, PR00L, and PR10L registers supports these two interrupt sources.

Note 25. If either of the interrupt sources INTP2 and INTAD2 is generated, bit 4 of the IF0L register is set to 1. Bit 4 of the MK0L, PR00L, and PR10L registers supports these two interrupt sources.

29.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H)

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 the given interrupt request is acknowledged, a reset signal is generated, or an instruction is executed. 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, IF2H, IF3L, and IF3H 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, and the IF3L and IF3H registers are combined to form 16-bit registers IF0, IF1, IF2, and IF3, they can be set by a 16-bit memory manipulation instruction. The value of each register following a reset is 00H.

Remark Executing an instruction that writes data to this register increases the number of cycles for instruction execution by 2.

Figure 29 - 2 Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H) (1/2)

Address: FFFE0H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	PIF5	PIF4	PIF3	PIF2 ADIF2	PIF1 ADIF1	PIF0	LVIIF	WDTIIF

Address: FFFE1H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	SREIF0 TMIF01H	TMIF00	STIF0 CSIIF00 IICIF00	TIMECIF0	FAAEIF	SREIF2	SRIF2 CSIIF21 IICIF21	STIF2 CSIIF20 IICIF20

Address: FFFE2H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	TMIF03	TMIF02	TMIF01	SRIF0 CSIIF01 IICIF01	IICAIF0	SREIF1 TMIF03H	SRIF1 CSIIF11 IICIF11	STIF1 CSIIF10 IICIF10

Address: FFFE3H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1H	RDIF TRGIF	TRDIF1	TRDIF0	TDIF TRJIF0	KRIF	ITLIF	RTCIF	ADIF0

Figure 29 - 2 Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H) (2/2)

Address: FFFD0H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2L	FLIF	PIF9 TMKBSTPIF 11	PIF8 TMKBSTRIF 11	PIF7 TMKBSTPIF 10	PIF6 TMKBSTRIF 10	PIF21	PIF20	EDIF CLDIF BPDIF TRXIF

Address: FFFD1H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2H	TMKBIF2	TMKBIF1	TMKBIF0	CMPIF3	CMPIF2	FAATRAPIF	PIF11 CMPIF1	PIF10 CMPIF0

Address: FFFD2H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF3L	TMKBSTPIF 20	TMKBSTRIF 20	TMKBSTPIF 01	TMKBSTRIF 01	TMKBSTPIF 00	TMKBSTRIF 00	FEDIF PMCIF	SDDIF GCRIF

Address: FFFD3H
After reset: 00H
R/W: R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
IF3H	0	0	0	ADIF3	TIMECIF2	TIMECIF1	TMKBSTPIF 21	TMKBSTRIF 21

xxIFx	Interrupt request flag
0	No interrupt request signal is generated
1	Indicates the generation of the interrupt request signal and the interrupt request currently being in the active state.

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 29 - 2 Flags Corresponding to Interrupt Request Sources. 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.

29.3.2 Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt. The MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, and MK3H 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, and the MK2L and MK2H registers, and the MK3L and MK3H registers are combined to form 16-bit registers MK0, MK1, MK2, and MK3, they can be set by a 16-bit memory manipulation instruction. The value of each register following a reset is FFH.

Remark Executing an instruction that writes data to this register increases the number of cycles for instruction execution by 2.

Figure 29 - 3 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H) (1/2)

Address: FFFE4H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	PMK5	PMK4	PMK3	PMK2 ADMK2	PMK1 ADMK1	PMK0	LVIMK	WDTIMK

Address: FFFE5H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	SREMK0 TMMK01H	TMMK00	STMK0 CSIMK00 IICMK00	TIMECMK0	FAAEMK	SREMK2	SRMK2 CSIMK21 IICMK21	STMK2 CSIMK20 IICMK20

Address: FFFE6H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	TMMK03	TMMK02	TMMK01	SRMK0 CSIMK01 IICMK01	IICAMK0	SREMK1 TMMK03H	SRMK1 CSIMK11 IICMK11	STMK1 CSIMK10 IICMK10

Address: FFFE7H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1H	RDMK TRGMK	TRDMK1	TRDMK0	TDMK TRJMK0	KRMK	ITLMK	RTCMK	ADMK0

Figure 29 - 3 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H) (2/2)

Address: FFFD4H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2L	FLMK	PMK9 TMKBSTPMK11	PMK8 TMKBSTRMK11	PMK7 TMKBSTPMK10	PMK6 TMKBSTRMK10	PMK21	PMK20	EDMK CLDMK BPDMK TRXMK

Address: FFFD5H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2H	TMKBMK2	TMKBMK1	TMKBMK0	CMPMK3	CMPMK2	FAATRAPMK	PMK11 CMPMK1	PMK10 CMPMK0

Address: FFFD6H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK3L	TMKBSTPMK20	TMKBSTRMK20	TMKBSTPMK01	TMKBSTRMK01	TMKBSTPMK00	TMKBSTRMK00	FEDMK PMCMK	SDDMK GCRMK

Address: FFFD7H
After reset: FFH
R/W: R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
MK3H	1	1	1	ADMK3	TIMECMK2	TIMECMK1	TMKBSTPMK21	TMKBSTRMK21

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 29 - 2 Flags Corresponding to Interrupt Request Sources. Be sure to set bits that are not available to the initial value.

29.3.3 Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H)

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, 2H, 3L, or 3H). The PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, and PR13H 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 PR03L and PR03H registers, the PR10L and PR10H registers, the PR11L and PR11H registers, the PR12L and PR12H registers, and the PR13L and PR13H registers are combined to form 16-bit registers PR00, PR01, PR02, PR03, PR10, PR11, PR12, and PR13, they can be set by a 16-bit memory manipulation instruction. The value of each register following a reset is FFH.

Remark Executing an instruction that writes data to this register increases the number of cycles for instruction execution by 2.

Figure 29 - 4 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H) (1/4)

Address: FFFE8H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	PPR05	PPR04	PPR03	PPR02 ADPR02	PPR01 ADPR01	PPR00	LVIPR0	WDTIPR0

Address: FFFE9H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00H	SREPR00 TMPR001H	TMPR000	STPR00 CSIPR000 IICPR000	TIMECPR00	FAAEPR0	SREPR02	SRPR02 CSIPR021 IICPR021	STPR02 CSIPR020 IICPR020

Address: FFFEAH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01L	TMPR003	TMPR002	TMPR001	SRPR00 CSIPR001 IICPR001	IICAPR00	SREPR01 TMPR003H	SRPR01 CSIPR011 IICPR011	STPR01 CSIPR010 IICPR010

Figure 29 - 4 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H) (2/4)

Address: FFFEBH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01H	RDPR0 TRGPR0	TRDPR01	TRDPR00	TDPR0 TRJPR00	KRPR0	ITLPR0	RTCPR0	ADPR00

Address: FFFD8H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02L	FLPR0	PPR09 TMKBSTPPR011	PPR08 TMKBSTRPR011	PPR07 TMKBSTPPR010	PPR06 TMKBSTRPR010	PPR021	PPR020	EDPR0 CLDPR0 BPDPR0 TRXPR0

Address: FFFD9H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02H	TMKBPR02	TMKBPR01	TMKBPR00	CMPPR03	CMPPR02	FAATRAPPR0	PPR011 CMPPR01	PPR010 CMPPR00

Address: FFFDAH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR03L	TMKBSTPPR020	TMKBSTRPR020	TMKBSTPPR001	TMKBSTRPR001	TMKBSTPPR000	TMKBSTRPR000	FEDPR0 PMCPR0	SDDPR0 GCRPR0

Address: FFFDBH
After reset: FFH
R/W: R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
PR03H	1	1	1	ADPR03	TIMECPR02	TIMECPR01	TMKBSTPPR021	TMKBSTRPR021

Figure 29 - 4 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H) (3/4)

Address: FFFECH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	PPR15	PPR14	PPR13	PPR12 ADPR12	PPR11 ADPR11	PPR10	LVIPR1	WDTIPR1

Address: FFFEDH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10H	SREPR10 TMPR101H	TMPR100	STPR10 CSIPR100 IICPR100	TIMECPR10	FAAEPR1	SREPR12	SRPR12 CSIPR121 IICPR121	STPR12 CSIPR120 IICPR120

Address: FFFEEH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	TMPR103	TMPR102	TMPR101	SRPR10 CSIPR101 IICPR101	IICAPR10	SREPR11 TMPR103H	SRPR11 CSIPR111 IICPR111	STPR11 CSIPR110 IICPR110

Address: FFFEFH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11H	RDPR1 TRGPR1	TRDPR11	TRDPR10	TDPR1 TRJPR10	KRPR1	ITLPR1	RTCPR1	ADPR10

Address: FFFDCH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12L	FLPR1	PPR19 TMKBSTPPR111	PPR18 TMKBSTRPR111	PPR17 TMKBSTPPR110	PPR16 TMKBSTRPR110	PPR121	PPR120	EDPR1 CLDPR1 BPDPR1 TRXPR1

Address: FFFDDH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12H	TMKBPR12	TMKBPR11	TMKBPR10	CMPPR13	CMPPR12	FAATRAPPR1	PPR111 CMPPR11	PPR110 CMPPR10

Figure 29 - 4 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H) (4/4)

Address: FFFDEH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR13L	TMKBSTPPR120	TMKBSTRPR120	TMKBSTPPR101	TMKBSTRPR101	TMKBSTPPR100	TMKBSTRPR100	FEDPR1 PMCPR1	SDDPR1 GCRPR1

Address: FFFDFH
After reset: FFH
R/W: R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
PR13H	1	1	1	ADPR13	TIMECPR12	TIMECPR11	TMKBSTPPR121	TMKBSTRPR121

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 29 - 2 Flags Corresponding to Interrupt Request Sources. Be sure to set bits that are not available to the initial value.

29.3.4 External interrupt rising edge enable registers (EGP0, EGP1), external interrupt falling edge enable registers (EGN0, EGN1)

These registers specify the valid edge for INTP0 to INTP11. The EGP0, EGP1, EGN0, and EGN1 registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each register following a reset is 00H.

Figure 29 - 5 Format of External Interrupt Rising Edge Enable Registers (EGP0, EGP1) and External Interrupt Falling Edge Enable Registers (EGN0, EGN1)

Address: FFF38H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FFF39H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

Address: FFF3AH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
EGP1	EGP21	EGP20	0	0	EGP11	EGP10	EGP9	EGP8

Address: FFF3BH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
EGN1	EGN21	EGN20	0	0	EGN11	EGN10	EGN9	EGN8

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 11, 20, and 21)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 29 - 3 shows the ports corresponding to the EGPn and EGNn bits.

Table 29 - 3 Interrupt Request Signals Corresponding to EGPn and EGNn Bits

Detection Enable Bit		Interrupt Request Signal	64-pin	52-pin	48-pin	30-, 32-, 40-, and 44-pin	24- and 25-pin	20-pin
EGP0	EGN0	INTP0	✓	✓	✓	✓	✓	✓
EGP1	EGN1	INTP1	✓	✓	✓	✓	—	—
EGP2	EGN2	INTP2	✓	✓	✓	✓	—	—
EGP3	EGN3	INTP3	✓	✓	✓	✓	—	—
EGP4	EGN4	INTP4	✓	✓	✓	✓	—	—
EGP5	EGN5	INTP5	✓	✓	✓	✓	✓	—
EGP6	EGN6	INTP6	✓	✓	✓	✓	✓	✓
EGP7	EGN7	INTP7	✓	✓	✓	✓	✓	✓
EGP8	EGN8	INTP8	✓	✓	✓	—	—	—
EGP9	EGN9	INTP9	✓	✓	✓	—	—	—
EGP10	EGN10	INTP10	✓	✓	—	—	—	—
EGP11	EGN11	INTP11	✓	✓	—	—	—	—
EGP20	EGN20	INTP20	✓	✓	✓	✓	✓	✓
EGP21	EGN21	INTP21	✓	✓	✓	✓	✓	—

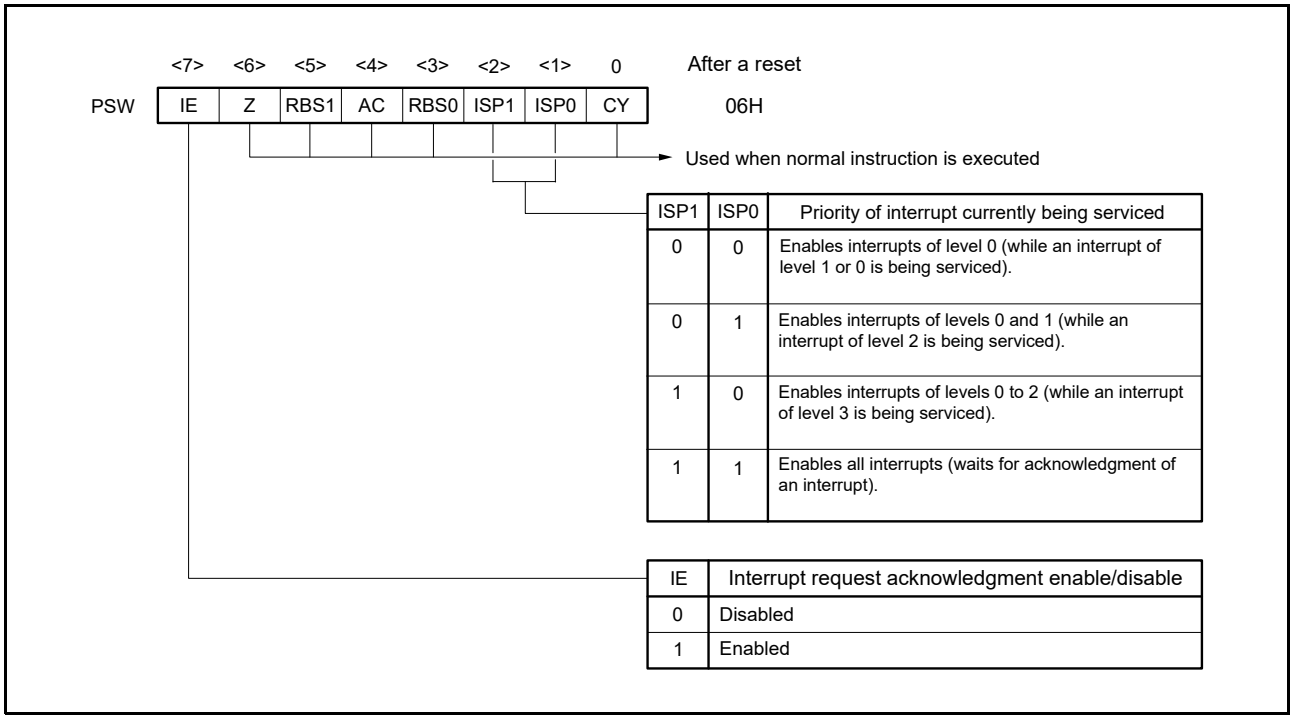
Caution **When the input port pins used for the external interrupt functions are switched to the output mode, the INTPn interrupt might be generated upon detection of a valid edge. When switching the input port pins to the output mode, set the port mode register xx (PMxx) to 0 after disabling the edge detection (by setting EGPn and EGNn to 0).**

Remark 1. For edge detection ports, see **Section 7 Port Functions**.
Remark 2. n = 0 to 11, 20, and 21

29.3.5 Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current state for an interrupt request. The IE flag that sets maskable interrupt enable/disable and the ISP[1:0] flags that control 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 ISP[1:0] 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. The value of this register following a reset is 06H.

Figure 29 - 6 Configuration of Program Status Word



ISP1	ISP0	Priority of interrupt currently being serviced
0	0	Enables interrupts of level 0 (while an interrupt of level 1 or 0 is being serviced).
0	1	Enables interrupts of levels 0 and 1 (while an interrupt of level 2 is being serviced).
1	0	Enables interrupts of levels 0 to 2 (while an 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

29.3.6 Registers for controlling the port functions multiplexed with the interrupt inputs

Set the following registers to control the port functions multiplexed with the interrupt inputs.

- Port mode registers xx (PMxx)
- Port mode control A registers xx (PMCAxx)

For details, see **7.3.1 Port mode registers xx (PMxx)** and **7.3.7 Port mode control A registers xx (PMCAxx)**.

When the pins multiplexed with INTP0 to INTP11, INTP20, and INTP21 are to be used for interrupt inputs, set the corresponding bits in the given port mode registers xx (PMxx) to 1 and those in port mode control A registers xx (PMCAxx) to 0.

Remark xx = 0 to 5, 7, 12, 14
Note that PMCA3 to PMCA5 and PMCA7 are not present in the RL78/G24 products.

29.4 Interrupt Servicing Operations

29.4.1 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 required from acceptance of an interrupt request in the form of the generation of a maskable interrupt request until vectored interrupt servicing proceeds are in the range indicated in **Table 29 - 4** below. For the interrupt request acknowledgment timing, see **Figures 29 - 8** and **29 - 9**.

Table 29 - 4 Times until Vectored Interrupt Servicing

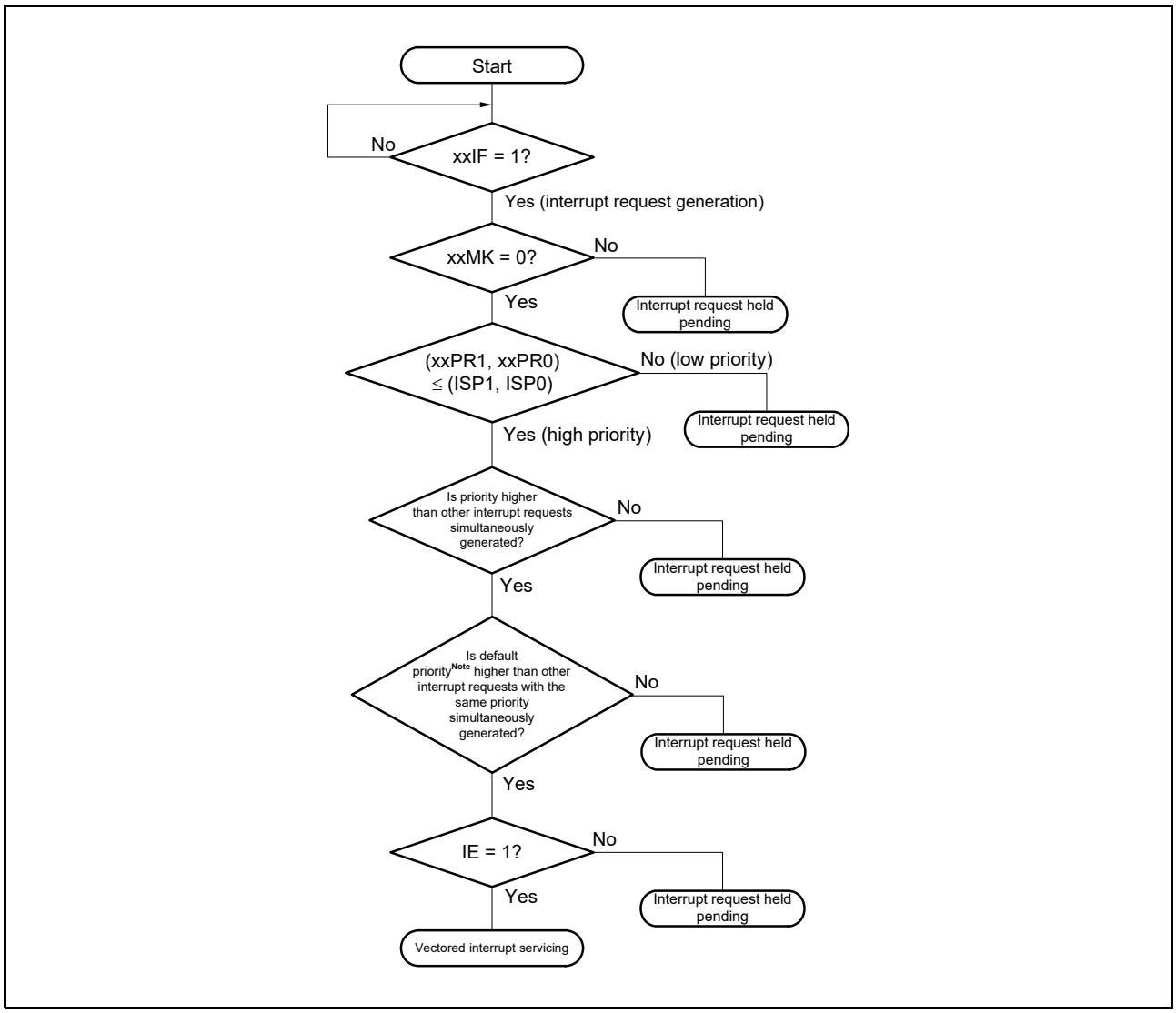
	Minimum Time	Maximum Time[Note]
Servicing time (when PFBE = 0)	9 clock cycles	16 clock cycles
Servicing time (when PFBE = 1)	11 clock cycles	20 clock cycles

Note Maximum time does not apply when an instruction from the internal RAM area is executed.

Remark 1 clock cycle: 1/f_{CLK} (f_{CLK}: CPU clock frequency)

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 interrupt 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 29 - 7** shows the interrupt request acknowledgment 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 to 0, and the contents of the priority specification flag corresponding to the acknowledged interrupt are transferred to the ISP[1:0] flags. The vector table data determined for each interrupt request is loaded into the PC and program control branches to the specified servicing. Restoring from an interrupt is possible by using the RETI instruction.

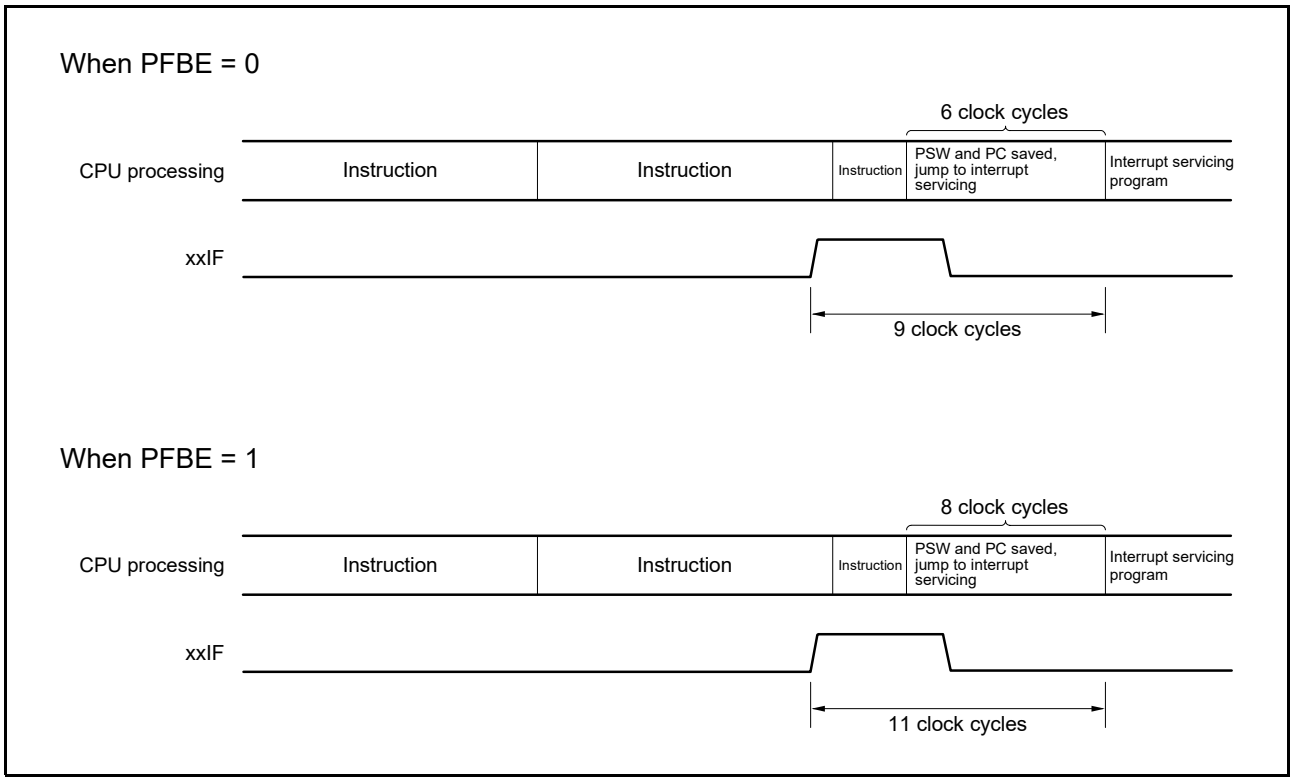
Figure 29 - 7 Interrupt Request Acknowledgment Processing Algorithm



Note For the default priority, refer to **Table 29 - 1 Interrupt Source List**.

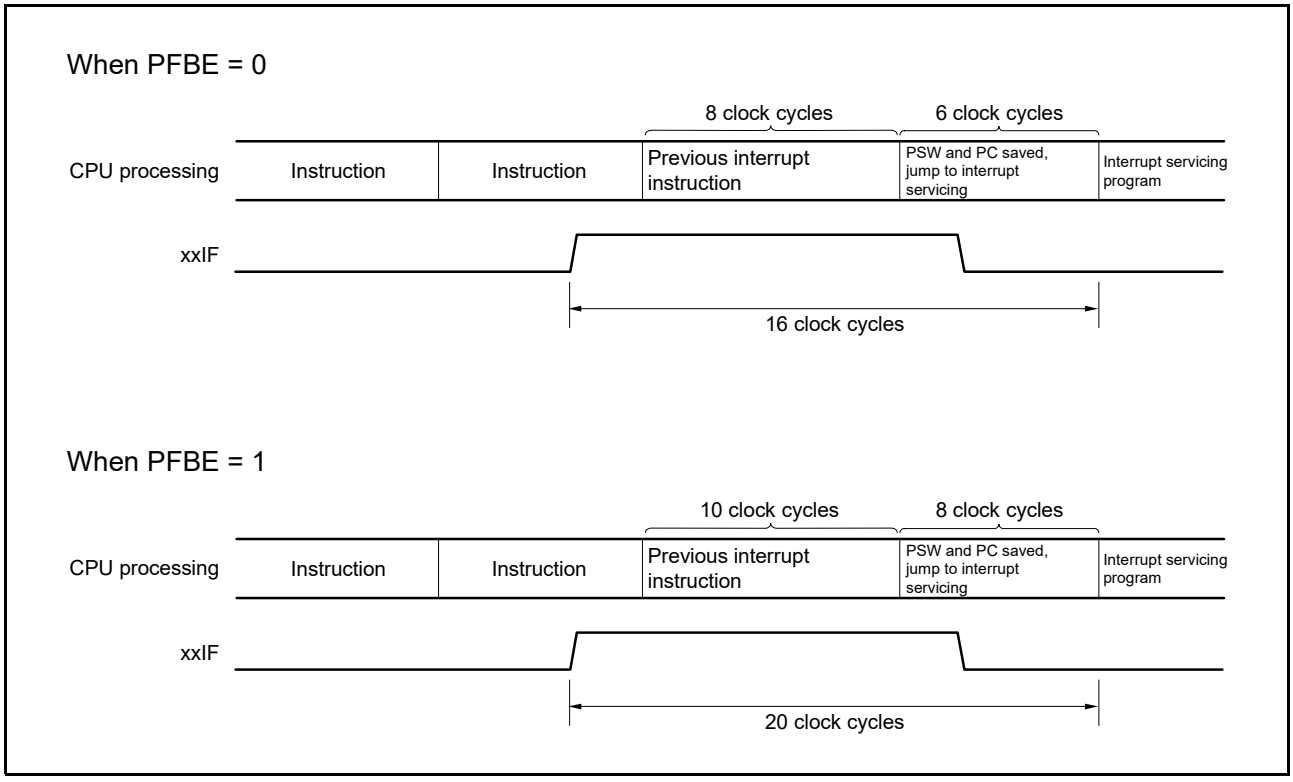
Remark xxIF: Interrupt request flag
xxMK: Interrupt mask flag
xxPR0: Priority specification flag 0
xxPR1: Priority specification flag 1
IE: Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)
ISP[1:0]: Flags that indicate the priority level of the interrupt currently being serviced (see **Figure 29 - 6 Configuration of Program Status Word**)

Figure 29 - 8 Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock cycle: 1/fCLK (fCLK: CPU clock)

Figure 29 - 9 Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock cycle: 1/fCLK (fCLK: CPU clock)

29.4.2 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 to 0, the contents of the vector table (0007EH, 0007FH) are loaded into the PC, and program control branches to the specified servicing. Restoring from a software interrupt is possible by using the RETB instruction.

Caution The RETI instruction cannot be used for restoring from the software interrupt.

29.4.3 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 the IE flag to 1 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 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.

Table 29 - 5 shows relationship between interrupt requests enabled for multiple interrupt servicing and **Figure 29 - 10** shows examples of multiple interrupt servicing.

Table 29 - 5 Relationship between Interrupt Requests Enabled for Multiple Interrupt Servicing during Interrupt Servicing

Multiple Interrupt Request / Maskable Interrupt		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	
Processing in progress	ISP[1:0] = 00B	✓	—	—	—	—	—	—	—	✓
	ISP[1:0] = 01B	✓	—	✓	—	—	—	—	—	✓
	ISP[1:0] = 10B	✓	—	✓	—	✓	—	—	—	✓
Waiting for reception	ISP[1:0] = 11B	✓	—	✓	—	✓	—	✓	—	✓

Remark 1. ✓: Multiple interrupt servicing enabled

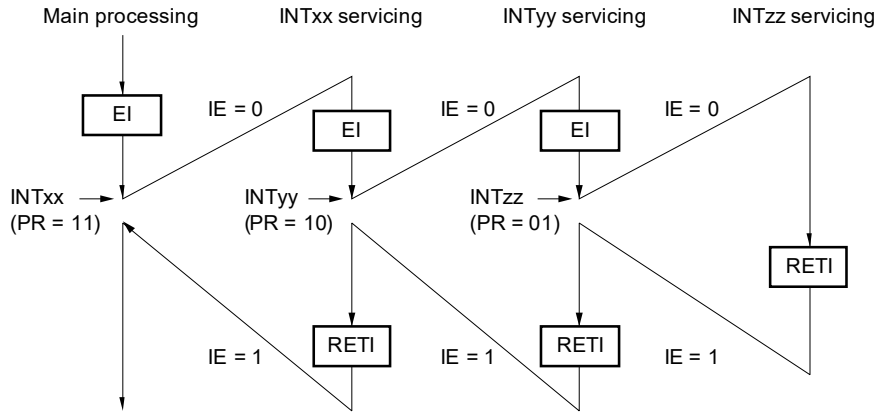
Remark 2. —: Multiple interrupt servicing disabled

Remark 3. ISP[1:0], and IE are flags contained in the PSW.
ISP[1:0] = 00B: An interrupt of level 1 or level 0 is being serviced.
ISP[1:0] = 01B: An interrupt of level 2 is being serviced.
ISP[1:0] = 10B: An interrupt of level 3 is being serviced.
ISP[1:0] = 11B: Wait for an interrupt acknowledgment (all interrupts are 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, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, and PR13H 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)

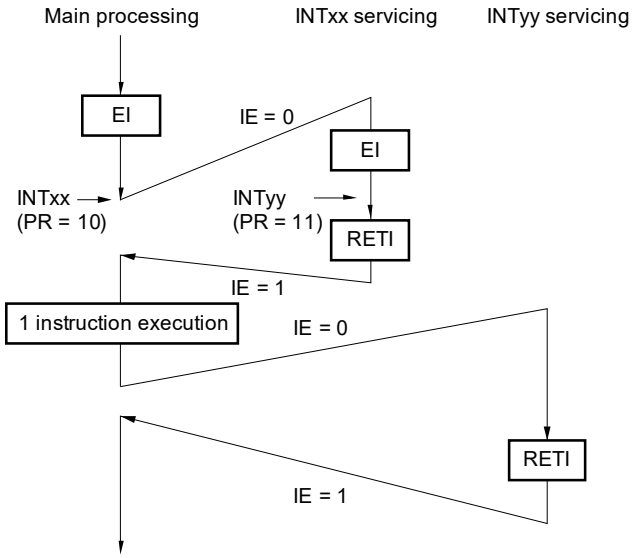
Figure 29 - 10 Examples of Multiple Interrupt Servicing (1/2)

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

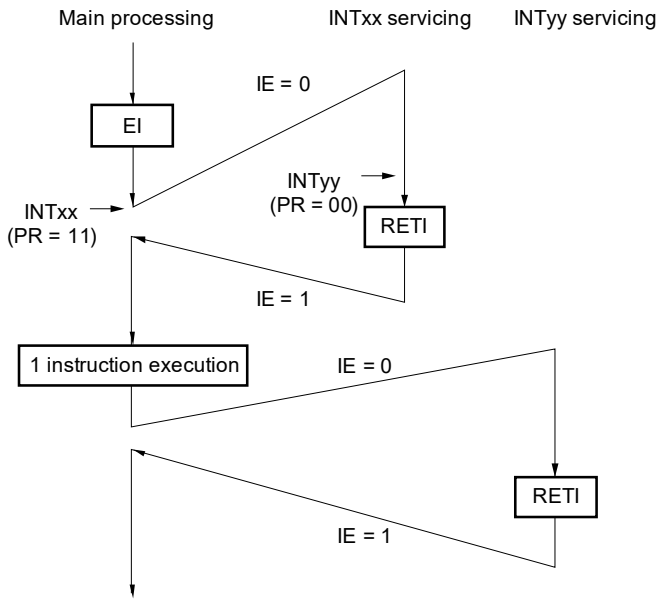


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

Remark 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 29 - 10 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.

Remark 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.

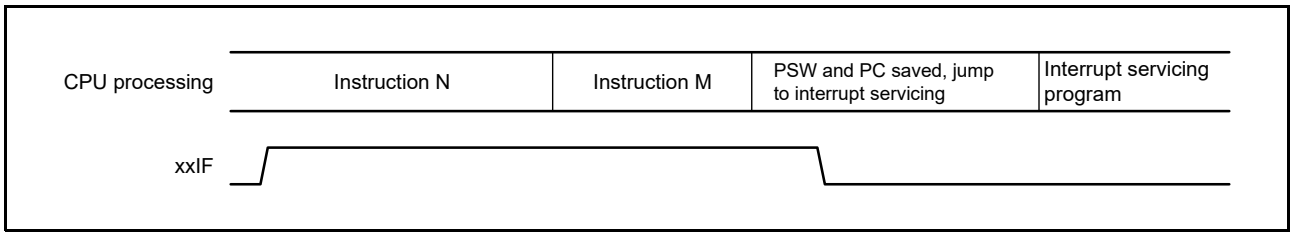
29.4.4 Interrupt request held pending

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 (instructions that hold interrupt requests pending) 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
- Write instructions for the IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, and PR13H registers

Figure 29 - 11 shows the timing at which interrupt requests are held pending.

Figure 29 - 11 Holding Interrupt Requests Pending



Remark 1. Instruction N: Instruction to hold interrupt requests pending
Remark 2. Instruction M: Instruction other than the instruction to hold interrupt requests pending

Section 30 Key Interrupt Function

The number of key interrupt input channels differs, depending on the product.

	20-, 24-, 25-, 30-, and 32-pin Products	40- and 44-pin Products	48-pin Products	52- and 64-pin Products
Number of the key interrupt input channels	—	4	6	8

30.1 Functions of the Key Interrupt

A key interrupt (INTKR) can be generated by inputting a rising edge/falling edge to the key interrupt input pins (KR0 to KR7).

Table 30 - 1 Assignment of the Key Interrupt Detection Pins

Key Interrupt Pins	Key Return Mode Register 0 (KRM0)
KR0	KRM00
KR1	KRM01
KR2	KRM02
KR3	KRM03
KR4	KRM04
KR5	KRM05
KR6	KRM06
KR7	KRM07

Remark Pins KR0 to KR3 are only present in the 40- and 44-pin products.
Pins KR0 to KR5 are only present in the 48-pin products.
Pins KR0 to KR7 are only present in the 52- and 64-pin products.

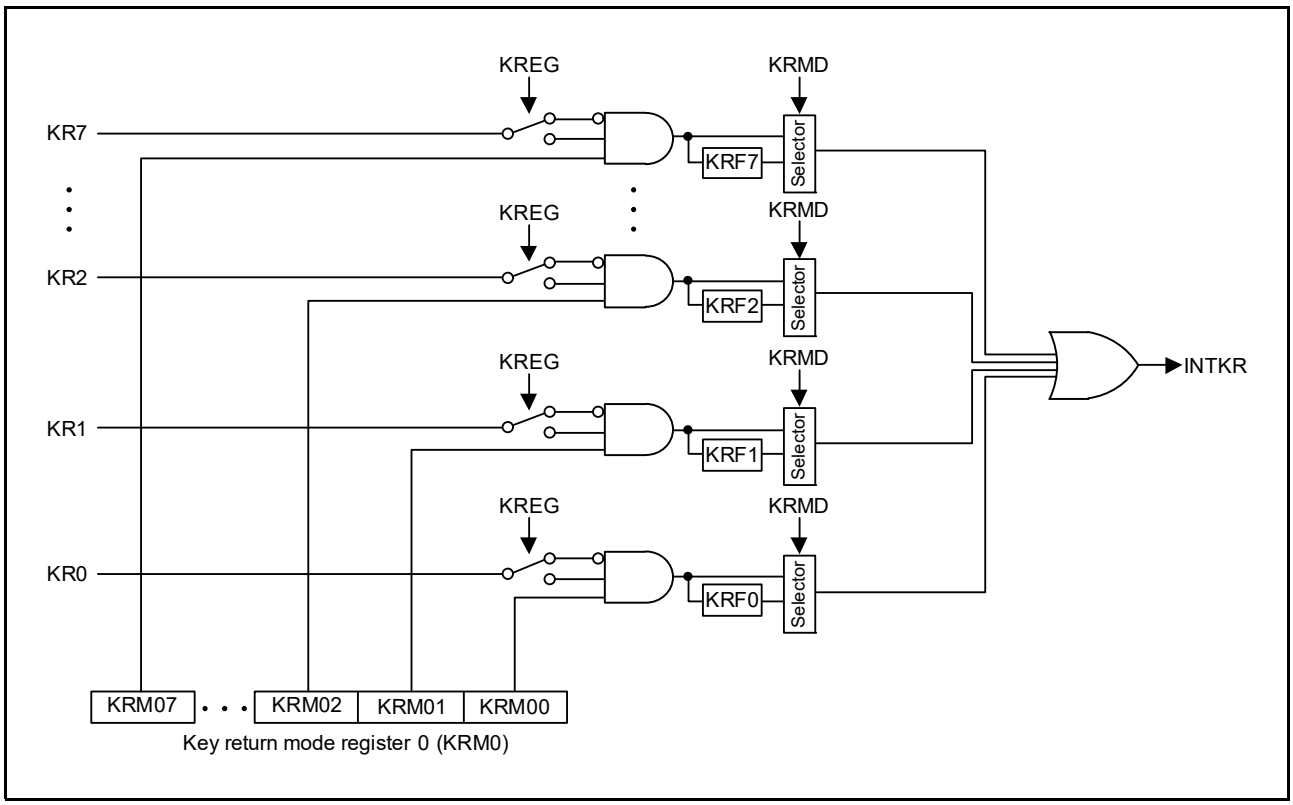
30.2 Configuration of the Key Interrupt

The key interrupt includes the following hardware blocks.

Table 30 - 2 Configuration of the Key Interrupt

Item	Configuration
Control registers	• Key return control register (KRCTL) • Key return mode register 0 (KRM0) • Key return flag register (KRF) • Port mode registers xx (PMxx) (xx = 7)

Figure 30 - 1 Block Diagram of the Key Interrupt



Remark Pins KR0 to KR3 are only present in the 40- and 44-pin products.
Pins KR0 to KR5 are only present in the 48-pin products.
Pins KR0 to KR7 are only present in the 52- and 64-pin products.

30.3 Registers to Control the Key Interrupt

The following registers are used to control the key interrupt.

- Key return control register (KRCTL)
- Key return mode register 0 (KRM0)
- Key return flag register (KRF)
- Port mode registers xx (PMxx) (xx = 7)

30.3.1 Key return control register (KRCTL)

This register controls the usage of the key return flags (KRF0 to KRF7) and sets the detection edge. The KRCTL register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 30 - 2 Format of Key Return Control Register (KRCTL)

Address: FFF34H
After reset: 00H
R/W: R/W

Symbol	<7>	6	5	4	3	2	1	<0>
KRCTL	KRMD	0	0	0	0	0	0	KREG

KRMD	Usage of key return flags (KRF0 to KRF7)
0	Does not use key return flags
1	Uses key return flags

KREG	Selection of detection edge (KR0 to KR7)
0	Falling edge
1	Rising edge

30.3.2 Key return mode register 0 (KRM0)

This register controls the KR0 to KR7 signals. The KRM0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 30 - 3 Format of Key Return Mode Register 0 (KRM0)

Address: FFF37H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
KRM0	KRM07	KRM06	KRM05	KRM04	KRM03	KRM02	KRM01	KRM00

KRM0n	Key interrupt mode control
0	Does not detect key interrupt signal
1	Detects key interrupt signal

Caution 1. The on-chip pull-up resistor can be applied by setting the corresponding key interrupt input pins (bits) in pull-up resistor register 7 (PU7) to 1.

Caution 2. An interrupt will be generated if the target bit of the KRM0 register is set while a low level (the KREG bit is set to 0) or a high level (the KREG bit is set to 1) is being input to the key interrupt input pin. To ignore this interrupt, set the KRM0 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 high-level and low-level widths (see 43.4 AC Characteristics or 44.4 AC Characteristics).

Caution 3. The pins not used in the key interrupt mode can be used as normal port pins.

Caution 4. Be sure to set the following bits to 0.
Bits 7 to 0 in the 20-, 24-, 25-, 30-, and 32-pin products
Bits 7 to 4 in the 40- and 44-pin products
Bits 7 and 6 in the 48-pin products

Remark n = 0 to 7

30.3.3 Key return flag register (KRF)

This register controls the key interrupt flags (KRF0 to KRF7). The KRF register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 30 - 4 Format of Key Return Flag Register (KRF)

Address: FFF35H
After reset: 00H
R/W: R/W**Note**

Symbol	7	6	5	4	3	2	1	0
KRF	KRF7	KRF6	KRF5	KRF4	KRF3	KRF2	KRF1	KRF0

KRFn	Key interrupt flag
0	No key interrupt signal has been detected.
1	A key interrupt signal has been detected.

Note Writing 1 has no effect. To clear the KRFn bit, write 0 to the corresponding bit and 1 to the other bits using an 8-bit memory manipulation instruction.

Remark n = 0 to 7

30.3.4 Registers for controlling the port functions multiplexed with the key interrupt inputs

Set the following register to control the port functions multiplexed with the key interrupt inputs.

- Port mode registers xx (PMxx)

For details, see **7.3.1 Port mode registers xx (PMxx)**.

When the pins multiplexed with KR0 to KR7 are to be used for key interrupt inputs, set the corresponding bits in port mode register 7 (PM7) to 1.
The on-chip pull-up resistor can be applied by setting the corresponding key interrupt input pins (bits) in the pull-up resistor register 7 (PU7) to 1.

Remark xx = 7

Section 31 Standby Function

31.1 Standby Function

The standby function reduces the operating current of the system, and the following three modes are available.

1. HALT mode
   Executing a HALT instruction places this LSI chip in the HALT mode. In the HALT mode, the CPU operating clock is stopped. If the high-speed system clock oscillator, high-speed on-chip oscillator, middle-speed on-chip oscillator, or subsystem clock 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
   Executing a STOP instruction places this LSI chip in the STOP mode. In the STOP mode, the high-speed system clock oscillator, high-speed on-chip oscillator, and middle-speed on-chip oscillator stop, stopping the whole system, thereby considerably reducing the CPU operating current.
   Because this mode can be released 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 SNOOZE mode, the RL78/G24 is released from STOP mode and the following peripheral modules can operate without CPU intervention.
   For details, see the sections on the individual modules.
   - **Section 20 A/D Converter (ADC)**
   - **Section 24 Serial Array Unit (SAU)**
   - **Section 27 Data Transfer Controller (DTC)**

   This can only be specified when the high-speed on-chip oscillator or middle-speed on-chip oscillator is selected for the CPU/peripheral hardware clock (f_{CLK}).

   In any of these modes, registers, flags and data memory retain all values they had immediately before the transition to the standby mode. The states of the output latches and the output buffers for I/O port pins are also retained.

Caution 1. Shifting to the STOP mode is only enabled when the CPU is operating with the main system clock. Do not execute the STOP instruction while the CPU operates with the subsystem clock. Shifting to the HALT mode is enabled whether the CPU is operating with the main system clock or the subsystem clock.

Caution 2. When shifting to the STOP mode, be sure to stop the peripheral hardware operating with main system clock before executing the STOP instruction (excluding the peripheral modules which use the SNOOZE mode function.

Caution 3. It can be selected by the WDTON bit of the option byte or the WUTMMCK0 bit of the subsystem clock supply mode control register (OSMC) whether the low-speed on-chip oscillator continues oscillating or stops in the HALT or STOP mode. For details, see 9.1 Functions of Clock Generator, 2. Subsystem clock, <2> Low-speed on-chip oscillator.

31.2 Registers to Control the Standby Function

The following registers are used to control the standby function.

- Subsystem clock supply mode control register (OSMC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- Standby mode release setting register (WKUPMD)

Remark For details of the OSMC, OSTC, and OSTS registers, see **Section 9 Clock Generator**. For registers which control the SNOOZE mode function, see the following sections.

- **Section 20 A/D Converter (ADC)**
- **Section 24 Serial Array Unit (SAU)**
- **Section 27 Data Transfer Controller (DTC)**

31.2.1 Standby mode release setting register (WKUPMD)

The WKUPMD register is used to set the operation at the time of release from the standby mode. The WKUPMD register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 31 - 1 Format of Standby Mode Release Setting Register (WKUPMD)

Address: F0215H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	<0>
WKUPMD	0	0	0	0	0	0	0	FWKUP

FWKUP	Setting for starting the high-speed on-chip oscillator at the times of release from STOP mode and of transitions to SNOOZE mode**Notes 1, 2**
0	Starting of the high-speed on-chip oscillator is at normal speed.**Note 3**
1	Starting of the high-speed on-chip oscillator is at high speed.**Notes 3, 4**

Note 1. This setting is only available when the high-speed on-chip oscillator is selected for the CPU clock.

Note 2. This register is initialized when the RL78/G24 is released from STOP mode in response to the generation of a reset signal, so starting of the high-speed on-chip oscillator is at normal speed.

Note 3. For the activation time, see **31.3.2 STOP mode**.
The accuracy of the high-speed on-chip oscillator's frequency depends on whether starting of the high-speed on-chip oscillator is at normal speed or at high speed. See **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** or **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.

Note 4. This is only specifiable when the setting of the FRQSEL4 bit of the option byte (000C2H) is 0.

31.3 Standby Function Operation

31.3.1 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, middle-speed on-chip oscillator clock, or subsystem clock.
   Tables 31 - 1 and **31 - 2** show the operating states in the HALT mode.

   Caution The interrupt request signal is used for release from the HALT mode, so if the interrupt mask flag is 0 (enabling the interrupt processing) and the interrupt request flag is 1 (the interrupt request signal is being generated), the HALT mode is not entered if these are the settings even when a HALT instruction is executed.

Table 31 - 1 Operating States in HALT Mode (1) (1/2)

Item \ HALT Mode Setting			When HALT Instruction Is Executed While CPU Is Operating with Main System Clock				
			When CPU Is Operating with High-speed On-chip Oscillator Clock (f_{IH})	When CPU Is Operating with Middle-speed On-chip Oscillator Clock (f_{IM})	When CPU Is Operating with X1 Clock (f_X)	When CPU Is Operating with External Main System Clock (f_{EX})	When CPU Is Operating with PLL Clock (f_{PLL})
System clock			Clock supply to the CPU is stopped				
	Main system clock	f_{IH}	Operation continues (stopping this clock signal is not possible)	Operation disabled	Operation disabled		Stopping this clock signal is not possible when it is being supplied to the PLL
		f_{IM}	Operation disabled	Operation continues (stopping this clock signal is not possible)	Operation disabled		Operation disabled
		f_X	Operation disabled		Operation continues (stopping this clock signal is not possible)	Cannot operate	Stopping this clock signal is not possible when it is being supplied to the PLL
		f_{EX}			Cannot operate	Operation continues (stopping this clock signal is not possible)	Operation disabled
		f_{PLL}	Operation disabled				Operation continues (stopping this clock signal is not possible)
	Subsystem clock	f_{XT}	Retains the state before the transition to HALT mode				
		f_{EXS}					
	Low-speed on-chip oscillator clock	f_{IL}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply mode control register (OSMC) WUTMMCK0 = 1 or SELLOSC = 1: Oscillates (Setting of WUTMMCK0 = 1 and SELLOSC = 1 is prohibited when the subsystem clock oscillator clocks (f_{SX} and f_{SXR}) are operating.) WUTMMCK0 = 0, SELLOSC = 0, and WDTON = 0: Stopped WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 1: Oscillates WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 0: Stopped				
CPU			Operation stopped				
Code flash memory							
Data flash memory							
RAM			Operation stopped (operation in response to access by the DTC is possible)				
Port (latch)			Retains the state before the transition to HALT mode (capable of operations in response to access by the DTC)				

Table 31 - 1 Operating States in HALT Mode (1) (2/2)

HALT Mode Setting / Item		When HALT Instruction Is Executed While CPU Is Operating with Main System Clock				
		When CPU Is Operating with High-speed On-chip Oscillator Clock (f_{IH})	When CPU Is Operating with Middle-speed On-chip Oscillator Clock (f_{IM})	When CPU Is Operating with X1 Clock (f_X)	When CPU Is Operating with External Main System Clock (f_{EX})	When CPU Is Operating with PLL Clock (f_{PLL})
FAA		Operation enabled				
Data shared memory		Operation stopped (operation in response to access by the FAA or DTC is possible)				
Timer array unit		Operation enabled				
RTC		Operation enabled				
32-bit interval timer		Operation enabled				
Watchdog timer		See **Section 19 Watchdog Timer (WDT)**.				
Timer RJ		Operation enabled				
Timer RD2, PWMOPA						
Timer RG2						
Timer RX						
Timer KB3						
Clock output/buzzer output						
A/D converter						
D/A converter						
Comparator						
Programmable gain amplifier						
Serial array unit						
Serial interface IICA						
Digital addressable lighting interface (DALI)						
Data transfer controller (DTC)						
Event link controller (ELC)		Functional blocks with operation enabled can be linked.				
Power-on-reset function		Operation enabled				
Voltage detection function						
External interrupt						
Key interrupt function						
CRC operation function	High-speed CRC					
	General-purpose CRC	Capable of operation in response to access by the DTC to obtain data for calculations from the RAM area				
Illegal-memory access detection function		Capable of operation in response to access by the DTC				
RAM parity error detection function						
RAM guard function						
SFR guard function						
True random number generator		Operation enabled				

Remark Operation stopped: Operation stopped: Operation is automatically stopped at the time of 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_{IM}: Middle-speed on-chip oscillator clock f_X: X1 clock
f_{EX}: External main system clock f_{XT}: XT1 clock
f_{EXS}: External subsystem clock f_{PLL}: PLL clock
f_{SX}: Subsystem clock oscillator clock f_{SXR}: Subsystem clock oscillator, clock for the RTC and other functions

Table 31 - 2 Operating States in HALT Mode (2) (1/2)

Item \ HALT Mode Setting			When HALT Instruction Is Executed While CPU Is Operating with Subsystem Clock		
			When CPU Is Operating with XT1 Clock (f_{XT})	When CPU Is Operating with External Subsystem Clock (f_{EXS})	When CPU Is Operating with Low-speed On-chip Oscillator Clock (f_{IL})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{IH}	Operation disabled		
		f_{IM}			
		f_X			
		f_{EX}			
		f_{PLL}			
	Subsystem clock	f_{XT}	Operation continues (stopping this clock signal is not possible)	Cannot operate	Operation disabled
		f_{EXS}	Cannot operate	Operation continues (stopping this clock signal is not possible)	Operation disabled
	Low-speed on-chip oscillator clock	f_{IL}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply mode control register (OSMC) WUTMMCK0 = 1: Oscillates (Setting of WUTMMCK0 = 1 and SELLOSC = 1 is prohibited when the subsystem clock oscillator clocks (f_{SX} and f_{SXR}) are operating.) WUTMMCK0 = 0, SELLOSC = 0, and WDTON = 0: Stopped WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 1: Oscillates WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 0: Stopped		Operation continues (stopping this clock signal is not possible)
CPU			Operation stopped		
Code flash memory					
Data flash memory					
RAM			Operation stopped (operation in response to access by the DTC is possible)		
Port (latch)			Retains the state before the transition to HALT mode (operation in response to access by the DTC is possible)		
FAA			Operation disabled		
Data shared memory			Operation disabled (operation in response to access by the DTC is possible)		
Timer array unit			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
RTC			Operation enabled		
32-bit interval timer			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
Watchdog timer			See **Section 19 Watchdog Timer (WDT)**.		
Timer RJ			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
Timer RD2, PWMOPA			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
Timer RG2			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
Timer RX			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
Timer KB3			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
Clock output/buzzer output			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
A/D converter			Operation disabled		
D/A converter			Retains the state before the transition to HALT mode (operation in response to access by the DTC when RTCLPC = 0 is possible)		

Table 31 - 2 Operating States in HALT Mode (2) (2/2)

HALT Mode Setting / Item		When HALT Instruction Is Executed While CPU Is Operating with Subsystem Clock		
		When CPU Is Operating with XT1 Clock (f_{XT})	When CPU Is Operating with External Subsystem Clock (f_{EXS})	When CPU Is Operating with Low-speed On-chip Oscillator Clock (f_{IL})
Comparator		Operation enabled (when the RTCLPC bit is 0, or when the digital filters are not in use)		Operation enabled
Programmable gain amplifier		Operation enabled		
Serial array unit		Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
Serial interface IICA		Operation disabled		
Digital addressable lighting interface (DALI)		Operation disabled		
Data transfer controller (DTC)		Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
Event link controller (ELC)		Functional blocks with operation enabled can be linked.		
Power-on-reset function		Operation enabled		
Voltage detection function		Operation enabled		
External interrupt		Operation enabled (the INTP0, INTP20, and INTP21 interrupts are only enabled when the digital filters are not in use)		
Key interrupt function		Operation enabled		
CRC operation function	High-speed CRC	Operation disabled		
	General-purpose CRC	Capable of operation in response to access by the DTC to obtain data for calculations from the RAM area		
Illegal-memory access detection function		Capable of operation in response to access by the DTC		
RAM parity error detection function				
RAM guard function				
SFR guard function				
True random number generator		Operation enabled		

Remark Operation stopped: Operation stopped: Operation is automatically stopped at the time of 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_{IM}: Middle-speed on-chip oscillator clock f_X: X1 clock
f_{EX}: External main system clock f_{XT}: XT1 clock
f_{EXS}: External subsystem clock f_{PLL}: PLL clock
f_{SX}: Subsystem clock oscillator clock f_{SXR}: Subsystem clock oscillator, clock for the RTC and other functions

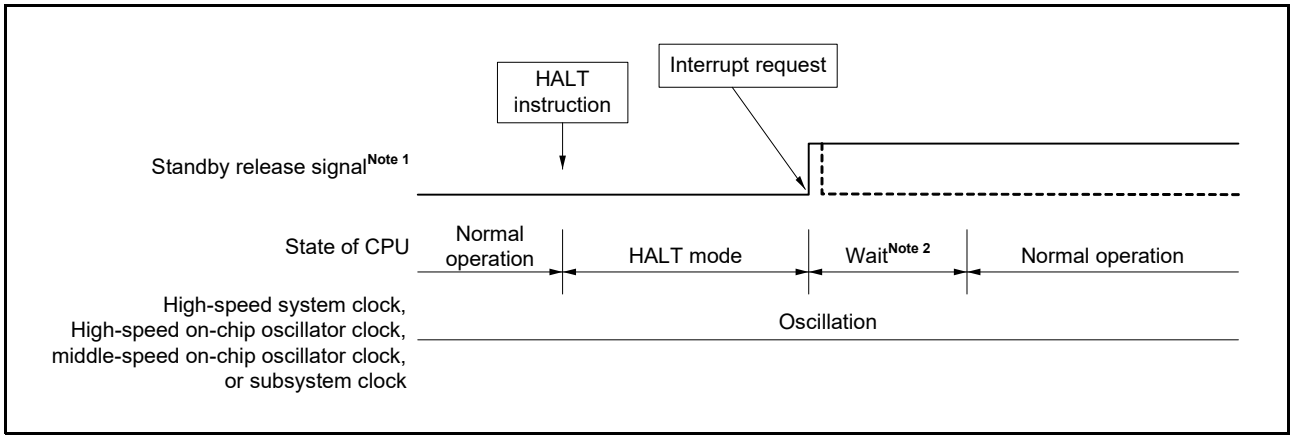
2. HALT mode release

The HALT mode can be released by the following two sources.

a) Release by a non-masked interrupt request

When a non-masked 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 instruction at the next address is executed.

Figure 31 - 2 HALT Mode Release by Interrupt Request Generation



Note 1. For details of the standby release signal, see **Figure 29 - 1 Basic Configuration of Interrupt Function**.

Note 2. Wait time for HALT mode release

- When vectored interrupt servicing is carried out
  Main system clock (with prefetching off): 15 to 16 clock cycles
  Main system clock (with prefetching on): 20 to 21 clock cycles
  Subsystem clock (RTCLPC = 0): 10 to 11 clock cycles
  Subsystem clock (RTCLPC = 1): 11 to 12 clock cycles
- When vectored interrupt servicing is not carried out
  Main system clock (with prefetching off): 9 to 10 clock cycles
  Main system clock (with prefetching on): 14 to 15 clock cycles
  Subsystem clock (RTCLPC = 0): 4 to 5 clock cycles
  Subsystem clock (RTCLPC = 1): 5 to 6 clock cycles

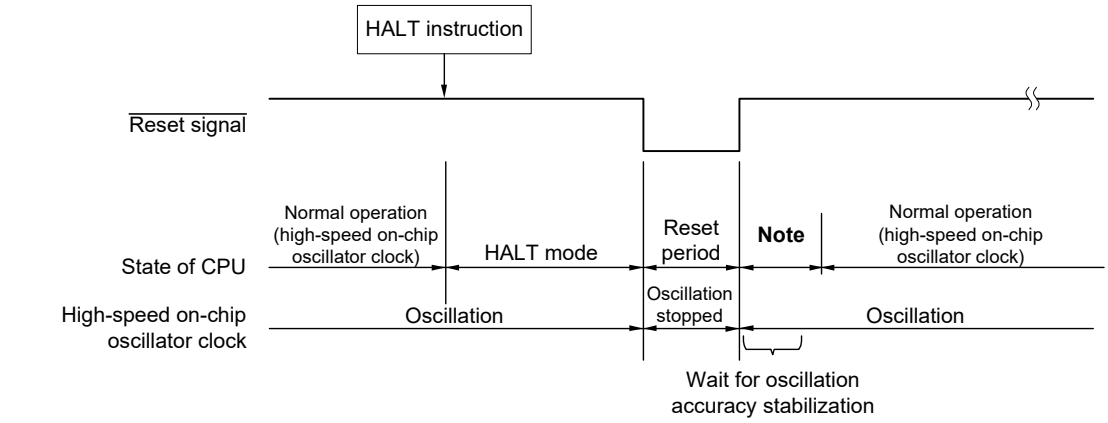
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

b) Release by reset signal generation

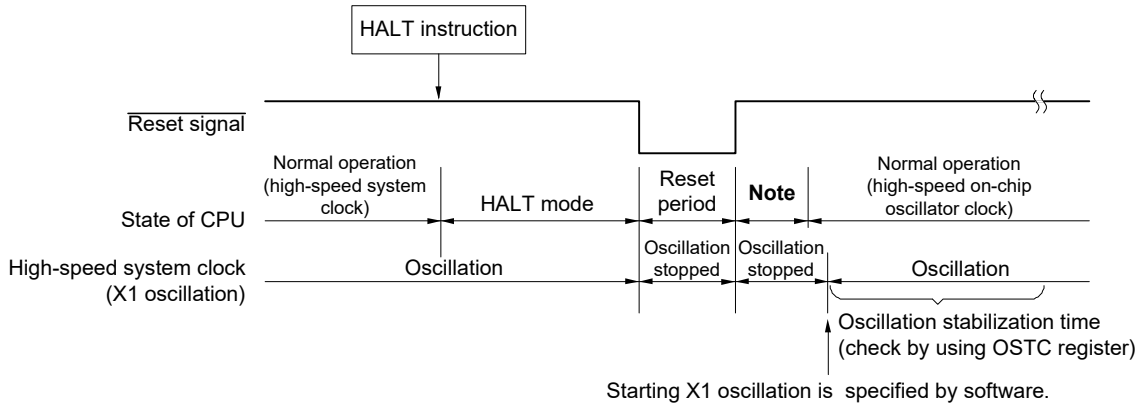
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 31 - 3 HALT Mode Release by Reset (1/2)

1. When high-speed on-chip oscillator clock is used as CPU clock



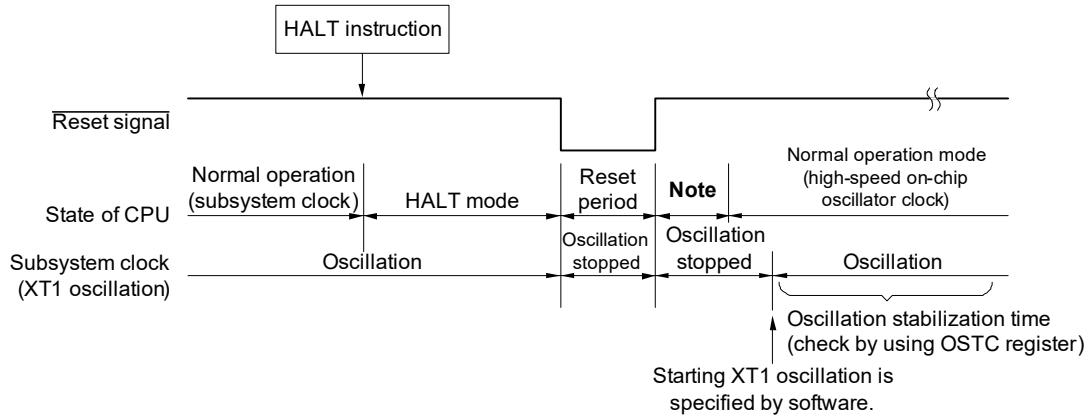
2. When high-speed system clock is used as CPU clock



Note For the reset processing time, see **Section 32 Reset Function**.
For the reset processing time of the power-on-reset circuit (POR) and voltage detectors (LVD0 and LVD1), see **Section 33 Power-on-reset Circuit (POR)**.

Figure 31 - 3 HALT Mode Release by Reset (2/2)

3. When subsystem clock is used as CPU clock



Note For the reset processing time, see **Section 32 Reset Function**.
For the reset processing time of the power-on-reset circuit (POR) and voltage detectors (LVD0 and LVD1), see **Section 33 Power-on-reset Circuit (POR)**.

31.3.2 STOP mode

1. STOP mode setting and operating states
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 main system clock.
Table 31 - 3 shows the operating states in STOP mode.

Caution 1. The interrupt request signal is used for release from the STOP mode, so if the interrupt mask flag is 0 (enabling the interrupt processing) and the interrupt request flag is 1 (the interrupt request signal is being generated), release from the STOP mode immediately proceeds after the transition to the STOP mode if these are the settings when a STOP instruction is executed. 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. Before entry to STOP mode, use the DSCON bit (bit 0 of the DSCCTL register) to stop PLL operation before executing the STOP instruction.

Table 31 - 3 Operating States in STOP Mode (1/2)

Item \ STOP Mode Setting			When STOP Instruction Is Executed While CPU Is Operating with Main System Clock			
			When CPU Is Operating with High-speed On-chip Oscillator Clock (f_{IH})	When CPU Is Operating with Middle-speed On-chip Oscillator Clock (f_{IM})	When CPU Is Operating with X1 Clock (fx)	When CPU Is Operating with External Main System Clock (f_{EX})
System clock			Clock supply to the CPU is stopped			
	Main system clock	f_{IH}	Stopped			
		f_{IM}	Stopped	Stopped	Stopped	
		f_X	Stopped			
		f_{EX}				
		f_{PLL}				
	Subsystem clock	f_{XT}	Retains the state before the transition to STOP mode			
		f_{EXS}				
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply mode control register (OSMC) WUTMMCK0 = 1 or SELLOSC = 1: Oscillates (Setting of WUTMMCK0 = 1 and SELLOSC = 1 is prohibited when the subsystem clock oscillator clocks (f_{SX} and f_{SXR}) are operating.) WUTMMCK0 = 0, SELLOSC = 0, and WDTON = 0: Stopped WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 1: Oscillates WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 0: Stopped			
CPU			Operation stopped			
Code flash memory						
Data flash memory						
RAM						
Port (latch)			Retains the state before the transition to STOP mode			
FAA			Operation disabled			
Data shared memory			Operation stopped			
Timer array unit			Operation disabled			
RTC			Operation enabled			
32-bit interval timer			Capable of operation when f_{SXL} is selected and RTCLPC = 0			
Watchdog timer			See **Section 19 Watchdog Timer (WDT)**.			
Timer RJ			• Capable of operation in event counter mode when the filter for the TRJIO input is not selected • Capable of operation when f_{SXL} is selected and RTCLPC = 0 Operation is disabled other than under the above conditions.			
Timer RD2, PWMOPA			Operation disabled			
Timer RG2			Operation disabled			
Timer RX			Operation disabled			
Timer KB3			Operation disabled			
Clock output/buzzer output			Capable of operation when f_{SXL} is selected and RTCLPC = 0			
A/D converter			Wakeup operation is enabled (switching to SNOOZE mode)			
D/A converter			Operation enabled (retains the state before the transition to STOP mode)			
Comparator			Operation enabled (the comparators are only enabled when the digital filters are not in use)			
Programmable gain amplifier			Operation enabled			
Serial array unit			Wakeup operation is enabled only for CSI00 and UART0 (switching to SNOOZE mode) Operation is disabled other than for CSI00 and UART0			
Serial interface IICA			Capable of waking up in response to address matching			
Digital addressable lighting interface (DALI)			Wakeup operation is enabled			
Data transfer controller (DTC)			DTC activation source receiving operation enabled (switching to SNOOZE mode)			

Table 31 - 3 Operating States in STOP Mode (2/2)

Item \ STOP Mode Setting		When STOP Instruction Is Executed While CPU Is Operating with Main System Clock			
		When CPU Is Operating with High-speed On-chip Oscillator Clock (f_{IH})	When CPU Is Operating with Middle-speed On-chip Oscillator Clock (f_{IM})	When CPU Is Operating with X1 Clock (f_X)	When CPU Is Operating with External Main System Clock (f_{EX})
Event link controller (ELC)		Functional blocks with operation enabled can be linked.			
Power-on-reset function		Operation enabled			
Voltage detection function		Operation enabled			
External interrupt		Operation enabled (the INTP0, INTP20, and INTP21 interrupts are only enabled when the digital filters are not in use)			
Key interrupt function		Operation enabled			
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					
True random number generator		Operation stopped			

Remark 1. Operation stopped: Operation is automatically stopped at the time of 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_{IM}: Middle-speed on-chip oscillator clock
f_X: X1 clock
f_{EX}: External main system clock
f_{XT}: XT1 clock
f_{EXS}: External subsystem clock
f_{PLL}: PLL clock
f_{SX}: Subsystem clock oscillator clock
f_{SXR}: Subsystem clock oscillator, clock for the RTC and other functions
f_{SXL}: Low-speed peripheral clock

Remark 2. 20- to 64-pin products: p = 00; q = 0

2. STOP mode release

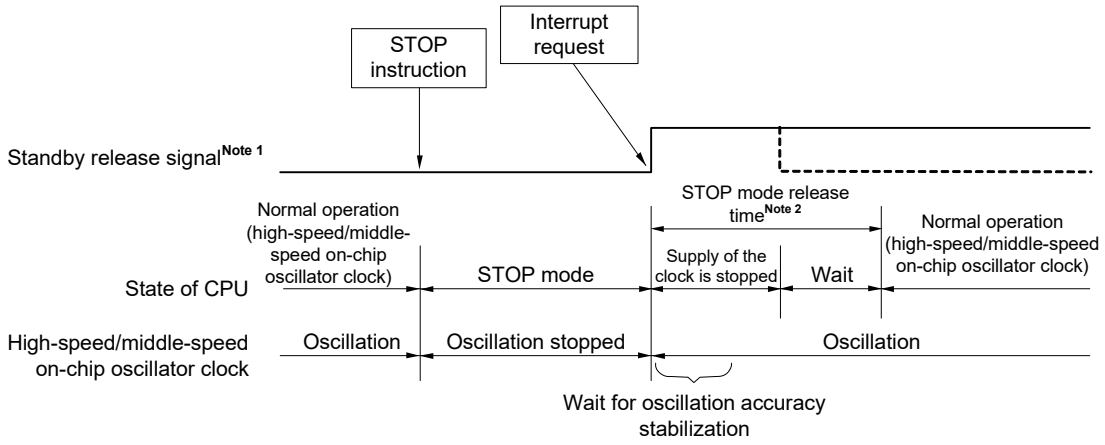
The STOP mode can be released by the following two sources.

a) Release by a non-masked interrupt request

When a non-masked 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 instruction at the next address is executed.

Figure 31 - 4 STOP Mode Release by Interrupt Request Generation (1/3)

1. When high-speed/middle-speed on-chip oscillator clock is used as CPU clock



Note 1. For details of the standby release signal, see **Figure 29 - 1 Basic Configuration of Interrupt Function**.

Note 2. STOP mode release time

Supply of the clock is stopped:

For the high-speed on-chip oscillator clock: 3.9 to 5.2 μs + 3 to 4 clock cycles (FWKUP = 0: Starting of the high-speed on-chip oscillator is at normal speed.)
0.6 to 0.8 μs + 3 to 4 clock cycles (FWKUP = 1: Starting of the high-speed on-chip oscillator is at high speed.)
The accuracy of the high-speed on-chip oscillator's frequency depends on whether starting of the high-speed on-chip oscillator is at normal speed or at high speed. See **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** or **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.

For the middle-speed on-chip oscillator clock: 1.5 to 2.5 μs + 3 to 4 clock cycles

Wait:

(common to the high-speed/middle-speed on-chip oscillator clock)

- When vectored interrupt servicing is carried out: 7 clock cycles
- When vectored interrupt servicing is not carried out: 1 clock cycle

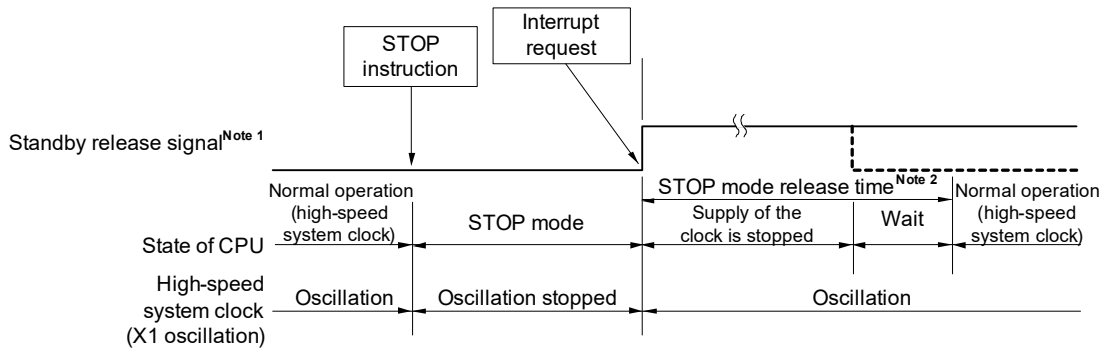
Caution To shorten oscillation stabilization time after the STOP mode is released when the CPU operates with the high-speed system clock (X1 oscillation), temporarily switch the CPU clock to the high-speed on-chip oscillator clock before the execution of the STOP instruction.

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 31 - 4 STOP Mode Release by Interrupt Request Generation (2/3)

2. When high-speed system clock (X1 oscillation) is used as CPU clock



Note 1. For details of the standby release signal, see **Figure 29 - 1 Basic Configuration of Interrupt Function**.

Note 2. STOP mode release time

Supply of the clock is stopped:

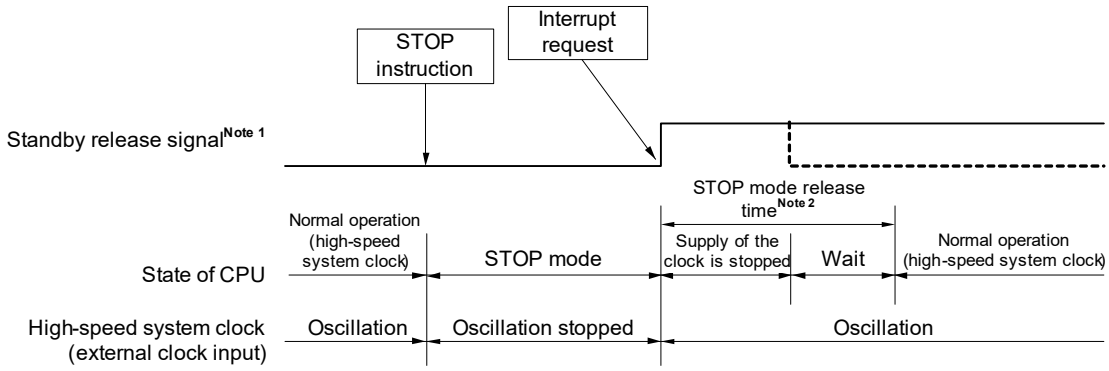
Oscillation stabilization time (set by the OSTS register) + 3 to 4 clock cycles

Wait:

- When vectored interrupt servicing is carried out: 7 clock cycles
- When vectored interrupt servicing is not carried out: 1 clock cycle

Figure 31 - 4 STOP Mode Release by Interrupt Request Generation (3/3)

3. When high-speed system clock (external clock input) is used as CPU clock



Note 1. For details of the standby release signal, see **Figure 29 - 1 Basic Configuration of Interrupt Function**.

Note 2. STOP mode release time

Supply of the clock is stopped:

50 to 51 cycles of the external clock

Wait:

- When vectored interrupt servicing is carried out: 7 clock cycles
- When vectored interrupt servicing is not carried out: 1 clock cycle

Caution **To reduce the oscillation stabilization time after release from the STOP mode while CPU operates with the high-speed system clock (X1 oscillation), switch the CPU 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.

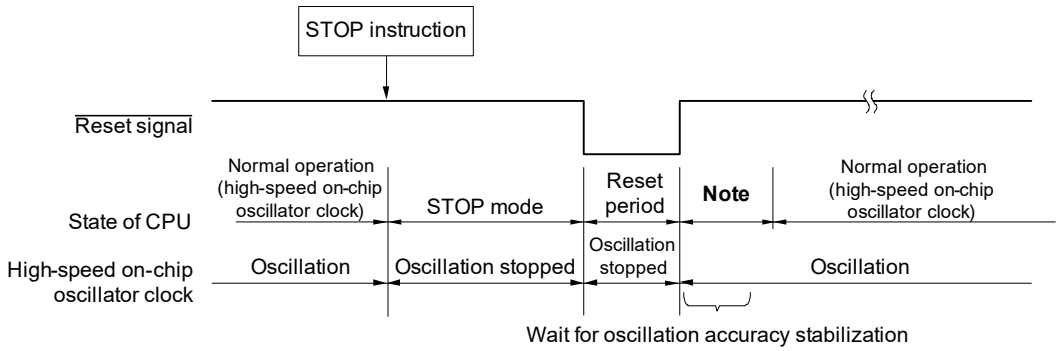
Remark 2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

b) Release by reset signal generation

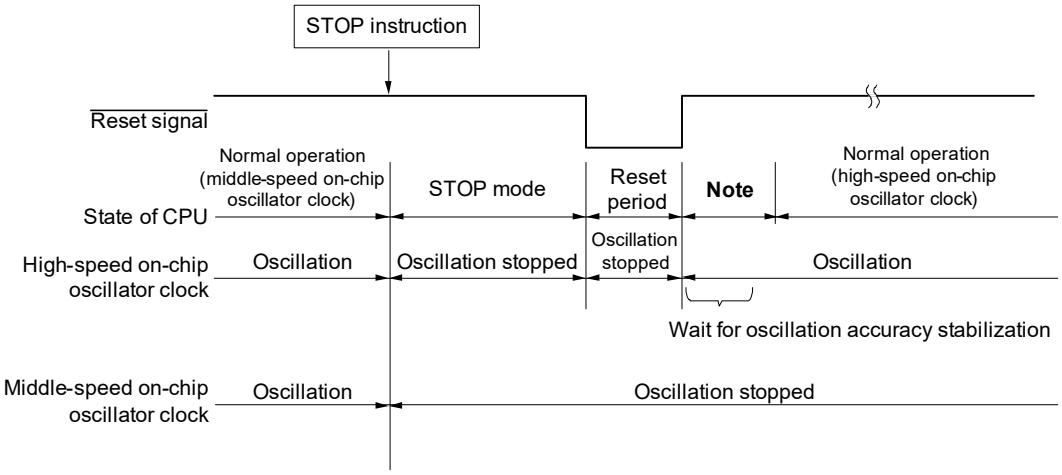
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 31 - 5 STOP Mode Release by Reset

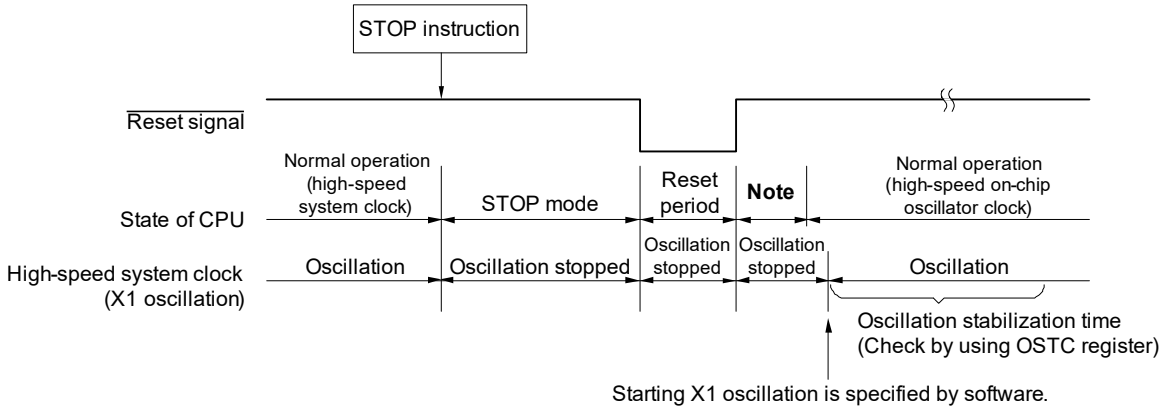
1. When high-speed on-chip oscillator clock is used as CPU clock



2. When middle-speed on-chip oscillator clock is used as CPU clock



3. When high-speed system clock is used as CPU clock



Note For the reset processing time, see **Section 32 Reset Function**. For the reset processing time of the power-on-reset circuit (POR) and voltage detectors (LVD0 and LVD1), see **Section 33 Power-on-reset Circuit (POR)**.

31.3.3 SNOOZE mode

1. SNOOZE mode setting and operating states
   The RL78/G24 can be placed in SNOOZE mode, in which operation of the following peripheral modules is selectable.
   For details, see the sections on the individual modules.
   - **Section 20 A/D Converter (ADC)**
   - **Section 24 Serial Array Unit (SAU)**
   - **Section 27 Data Transfer Controller (DTC)**

   Also, the RL78/G24 can be placed in SNOOZE mode if the CPU clock before entry to SNOOZE mode is the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock.

   For transitions to SNOOZE mode, the following intervals of waiting are inserted.
   Transition time from STOP mode to SNOOZE mode:
   For the high-speed on-chip oscillator clock: 3.9 to 5.2 μs
   (FWKUP = 0: Starting of the high-speed on-chip oscillator is at normal speed.)
   0.6 to 0.8 μs
   (FWKUP = 1: Starting of the high-speed on-chip oscillator is at high speed.)
   The accuracy of the high-speed on-chip oscillator's frequency depends on whether starting of the high-speed on-chip oscillator is at normal speed or at high speed. See **Section 43 Electrical Characteristics (T_A = –40 to +105°C)** or **Section 44 Electrical Characteristics (T_A = –40 to +125°C)**.
   For the middle-speed on-chip oscillator clock**Note**: 1.3 to 2.5 μs

   Remark The 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:
   For the high-speed on-chip oscillator clock:
   - When vectored interrupt servicing is carried out:
     "0.3 to 0.4 μs" + 10 to 11 clock cycles
   - When vectored interrupt servicing is not carried out:
     "0.3 to 0.4 μs" + 4 to 5 clock cycles

   For the middle-speed on-chip oscillator clock**Note**:
   - When vectored interrupt servicing is carried out:
     "0.6 to 1.2 μs" + 10 to 11 clock cycles
   - When vectored interrupt servicing is not carried out:
     "0.6 to 1.2 μs" + 4 to 5 clock cycles

   Note This is selected when the setting of the middle-speed on-chip oscillator trimming register (MIOTRM) is its initial value.

Table 31 - 4 shows the operating states in SNOOZE mode.

Table 31 - 4 Operating States in SNOOZE Mode (1/2)

Item \ STOP Mode Setting			Generation of Source Conditions which Lead to Transitions to SNOOZE Mode during STOP Mode	
			When CPU Is Operating with High-speed On-chip Oscillator Clock (f_{IH})	When CPU Is Operating with Middle-speed On-chip Oscillator Clock (f_{IM})
System clock			Clock supply to the CPU is stopped	
	Main system clock	f_{IH}	Operation started	Stopped
		f_{IM}	Stopped	Operation started
		f_X	Stopped	
		f_{EX}		
		f_{PLL}		
	Subsystem clock	f_{XT}	Operation enabled	
		f_{EXS}		
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply mode control register (OSMC) WUTMMCK0 = 1 or SELLOSC = 1: Oscillates (Setting of WUTMMCK0 = 1 and SELLOSC = 1 is prohibited when the subsystem clock oscillator clocks (f_{SX} and f_{SXR}) are operating.) WUTMMCK0 = 0, SELLOSC = 0, and WDTON = 0: Stopped WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 1: Oscillates WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 0: Stopped	
CPU			Operation stopped	
Code flash memory				
Data flash memory				
RAM			Operation stopped (capable of operation in response to access by the DTC)	
Port (latch)			Retains the state before the transition to SNOOZE mode (capable of operation in response to access by the DTC)	
FAA			Operation disabled	
Data shared memory			Operation disabled (capable of operation in response to access by the DTC)	
Timer array unit			Capable of operation in response to access by the DTC	
RTC			Operation enabled	
32-bit interval timer			Capable of operation when f_{SXL} is selected and RTCLPC = 0	
Watchdog timer			See **Section 19 Watchdog Timer (WDT)**.	
Timer RJ			• Capable of operation in event counter mode when the filter for the TRJIO input is not selected • Capable of operation when f_{SXL} is selected and RTCLPC = 0 Operation is disabled other than under the above conditions.	
Timer RD2, PWMOPA			Operation disabled	
Timer RG2			Operation disabled	
Timer RX			Operation disabled	
Timer KB3			Operation disabled	
Clock output/buzzer output			Capable of operation when f_{SXL} is selected and RTCLPC = 0	
A/D converter			Operation enabled	
D/A converter			Retains the state before the transition to SNOOZE mode (operation in response to access by the DTC when RTCLPC = 0 is possible)	
Comparator			Operation enabled (the comparators are only enabled when the digital filters are not in use)	
Programmable gain amplifier			Operation enabled	
Serial array unit			Only CSI00 and UART0 are capable of operation. Operation is disabled other than for CSI00 and UART0.	
Serial interface IICA			Capable of waking up in response to address matching	
Digital addressable lighting interface (DALI)			Operation disabled	
Data transfer controller (DTC)			Operation enabled	

Table 31 - 4 Operating States in SNOOZE Mode (2/2)

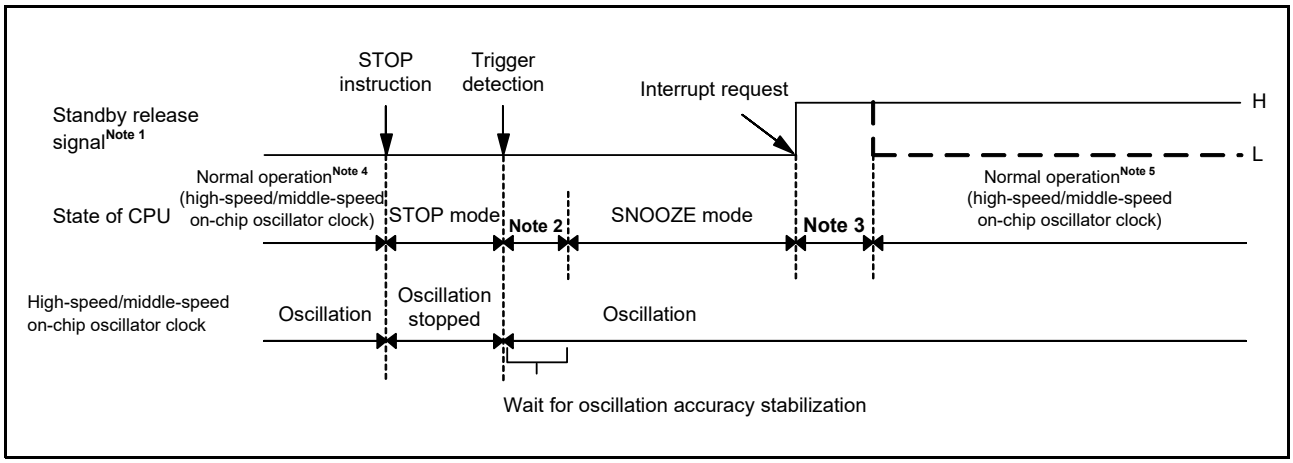
Item \ STOP Mode Setting		Generation of Source Conditions which Lead to Transitions to SNOOZE Mode during STOP Mode	
		When CPU Is Operating with High-speed On-chip Oscillator Clock (f_{IH})	When CPU Is Operating with Middle-speed On-chip Oscillator Clock (f_{IM})
Event link controller (ELC)		Functional blocks with operation enabled can be linked. Note that this excludes the A/D converter.	
Power-on-reset function		Operation enabled	
Voltage detection function		Operation enabled	
External interrupt		Operation enabled (the INTP0, INTP20, and INTP21 interrupts are only enabled when the digital filters are not in use)	
Key interrupt function		Operation enabled	
CRC operation function	High-speed CRC	Operation stopped	
	General-purpose CRC	Capable of operation in response to access by the DTC to obtain data for calculations from the RAM area	
Illegal-memory access detection function		Capable of operation in response to access by the DTC	
RAM parity error detection function			
RAM guard function			
SFR guard function			

Remark Operation stopped: Operation is automatically stopped at the time of 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_{IM}: Middle-speed on-chip oscillator clock
f_X: X1 clock
f_{EX}: External main system clock
f_{XT}: XT1 clock
f_{EXS}: External subsystem clock
f_{PLL}: PLL clock
f_{SX}: Subsystem clock oscillator clock
f_{SXR}: Subsystem clock oscillator, clock for the RTC and other functions
f_{SXL}: Low-speed peripheral clock

2. Timing diagram when the interrupt request signal is generated in the SNOOZE mode

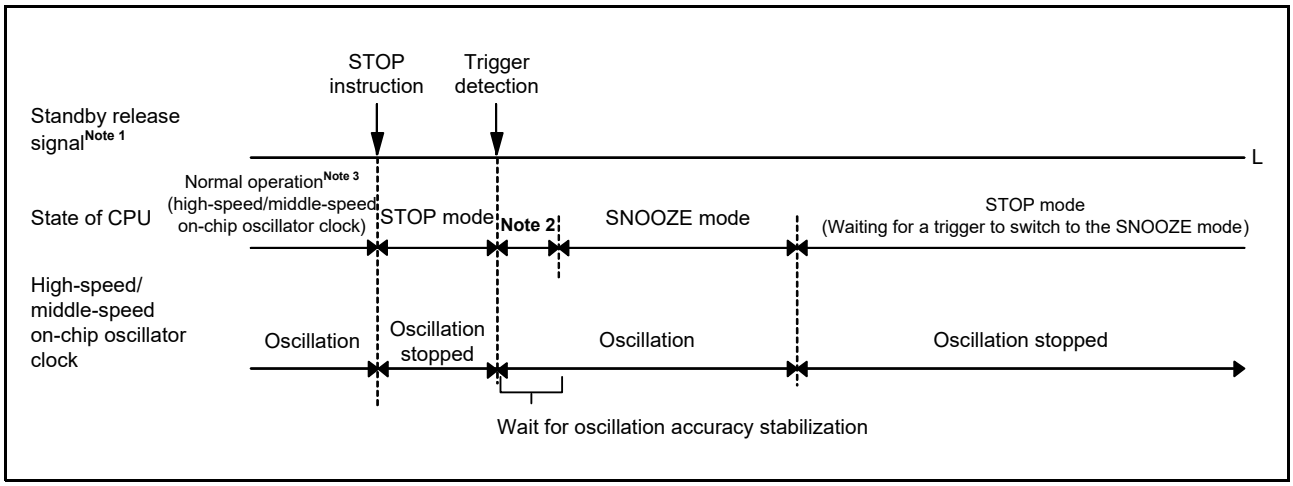
Figure 31 - 6 When the Interrupt Request Signal Is Generated in the SNOOZE Mode



Note 1. For details of the standby release signal, see **Figure 29 - 1 Basic Configuration of Interrupt Function**.
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 immediately before switching to the STOP mode.
Note 5. Be sure to release the SNOOZE mode immediately after return to the normal operation.

3. Timing diagram when the interrupt request signal is not generated in the SNOOZE mode

Figure 31 - 7 When the Interrupt Request Signal Is Not Generated in the SNOOZE Mode



Note 1. For details of the standby release signal, see **Figure 29 - 1 Basic Configuration of Interrupt Function**.
Note 2. Transition time from STOP mode to SNOOZE mode
Note 3. Enable the SNOOZE mode immediately before switching to the STOP mode.

Remark For details on the SNOOZE mode function, see the following sections.
- **Section 20 A/D Converter (ADC)**
- **Section 24 Serial Array Unit (SAU)**
- **Section 27 Data Transfer Controller (DTC)**

Section 32 Reset Function

A reset is triggered by any of the following events.

1. External reset input via the $\overline{\text{RESET}}$ pin
2. Internal reset due to detection of a program malfunction by the watchdog timer
3. Internal reset by comparison of supply voltage and detection voltage of the power-on-reset (POR) circuit
4. Internal reset by comparison of supply voltage and detection voltage of the voltage detectors (LVD0 and LVD1)
5. Internal reset due to execution of an illegal instruction**Note**
6. Internal reset due to a RAM parity error
7. Internal reset due to illegal-memory access

External and internal resets start program execution from the address at 00000H and 00001H when the reset signal is generated. A reset is applied when a low level is input to the $\overline{\text{RESET}}$ pin, the watchdog timer detects the program entering runaway execution, on voltage detection by the POR, LVD0, or LVD1 circuit, execution of an illegal instruction**Note**, generation of a RAM parity error, or illegal-memory access, and each module is set to the state shown in **Table 32 - 1**.

Note In normal operation, executing the instruction code FFH triggers an internal reset, but this is not the case during emulation by the in-circuit emulator or on-chip debugging emulator.

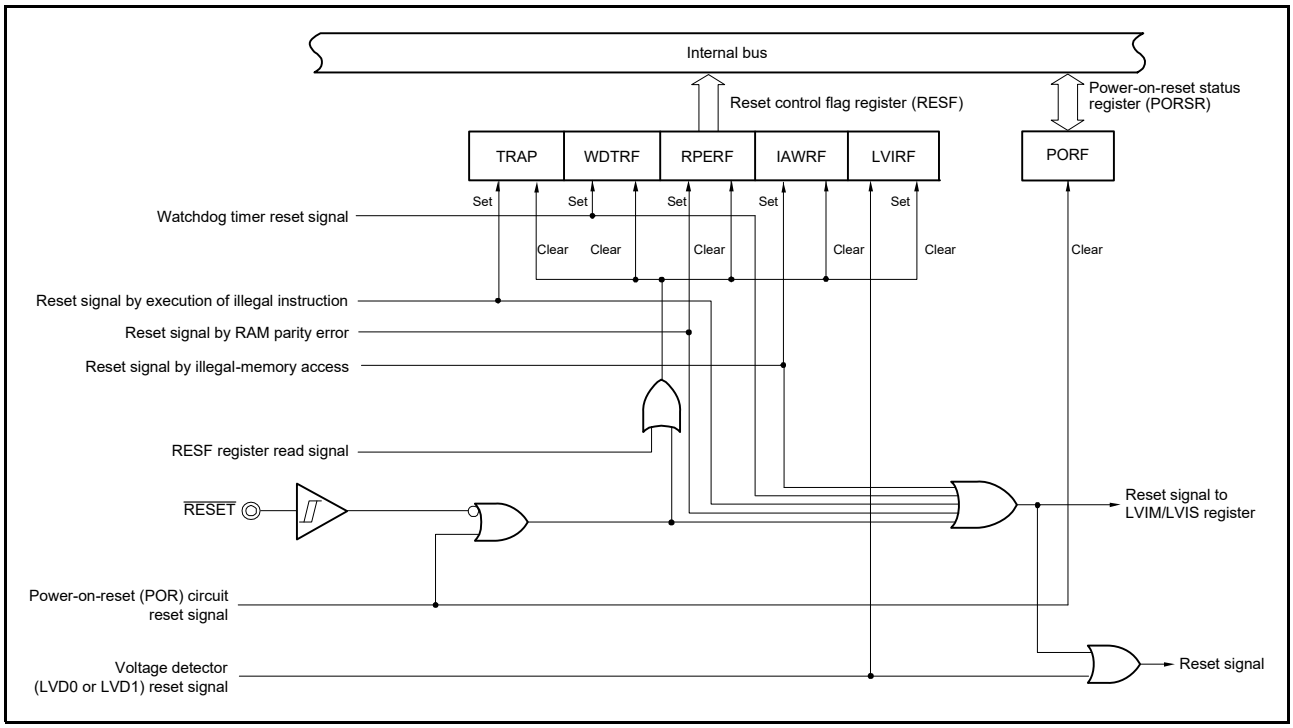
Caution 1. For an external reset, input a low level for at least 10 µs to the $\overline{\text{RESET}}$ pin.
To perform an external reset upon power application, input a low level to the $\overline{\text{RESET}}$ pin, turn power on, continue to input a low level to the pin for at least 10 µs within the operating voltage range shown in 43.4 AC Characteristics or 44.4 AC Characteristics, and then input a high level to the pin.

Caution 2. During generation of a reset signal, the X1 clock, high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, and low-speed on-chip oscillator clock stop oscillating. External main system clock input becomes invalid.
The XT1 clock and external subsystem clock only stop oscillating or their inputs become invalid in the POR state.

Caution 3. The port pins become the following state because SFRs and 2nd SFRs are initialized after a reset.
- **P40: High-impedance during the external reset period or reset period by the POR. High level during other types of reset and after the reset is released (connected to the on-chip pull-up resistor).**
- **P130: Low-level output during a reset and after the reset is released.**
- **Ports other than P40 and P130: High-impedance during a reset and after the reset is released.**

The following registers are only initialized by a POR reset.
- **RTC-related registers**
- **EXCLKS, OSCSELS, XTSEL, and AMPHS[1:0] bits of the CMC register**

Figure 32 - 1 Block Diagram of Reset Function

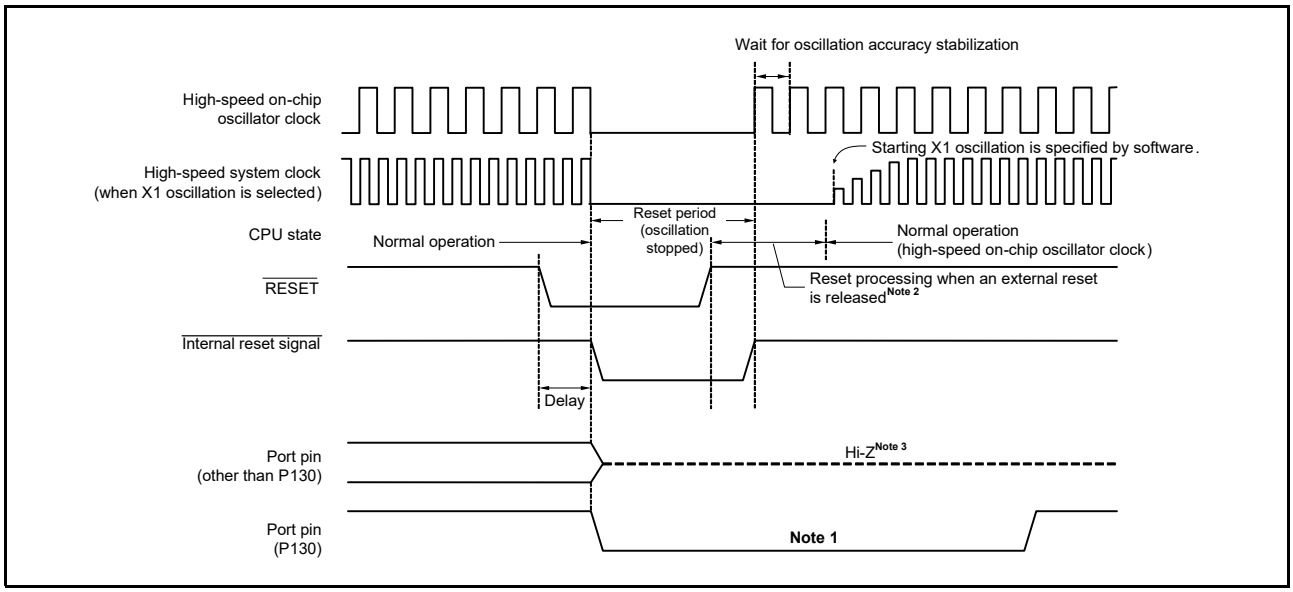


Caution An LVD0 circuit internal reset does not reset the LVD0 circuit.

32.1 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 32 - 2 Timing of Reset by $\overline{\text{RESET}}$ Input



Note 1. The P130 pin outputs a low-level signal following the application of a reset. If this pin was in use as a high-level output before the application of a reset, the output signal from the P130 pin can effectively be used as the reset signal for an external device that has an active-low reset signal. To de-assert the reset signal to the external device, set the P130 pin for high-level output by software.

Note 2. Reset times (times for release from the external reset state)

The first external reset following release from the POR state:

- When the LVD is in use: 0.506 ms (typ.), 0.694 ms (max.)
- When the LVD is not in use: 0.201 ms (typ.), 0.335 ms (max.)

The second or subsequent external reset following release from the POR state:

- When the LVD is in use: 0.476 ms (typ.), 0.616 ms (max.)
- When the LVD is not in use: 0.170 ms (typ.), 0.257 ms (max.)

After power is supplied, the following voltage stabilization waiting time is required before reset processing starts after release from the external reset state.

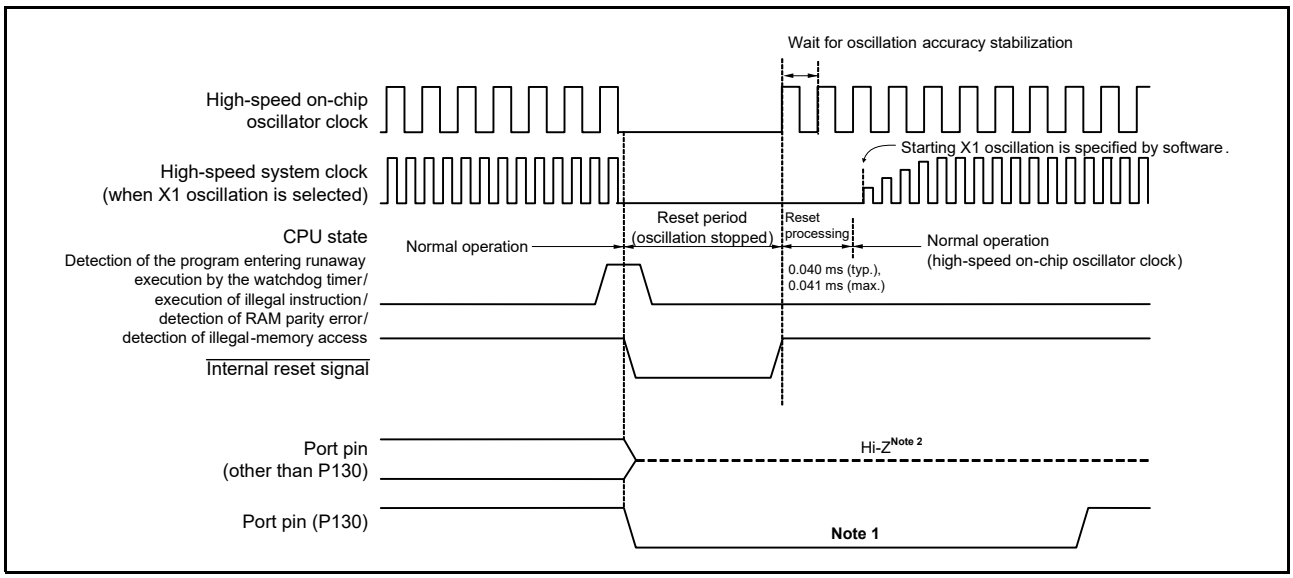
- 4.0 ms (typ.), 9.9 ms (max.)

Note 3. The state of the P40 pin is as follows.

- High-impedance during the external reset period or reset period by the POR.
- High level during other types of reset and after the reset is released (connected to the on-chip pull-up resistor).

Release from the reset state proceeds automatically in the case of a reset due to detection of the program entering runaway execution by the watchdog timer, 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 32 - 3 Timing of Reset Due to Detection of the Program Entering Runaway Execution by the Watchdog Timer, Execution of Illegal Instruction, Detection of RAM Parity Error, or Detection of Illegal-memory Access



Note 1. The P130 pin outputs a low-level signal following the application of a reset. If this pin was in use as a high-level output before the application of a reset, the output signal from the P130 pin can effectively be used as the reset signal for an external device that has an active-low reset signal. To de-assert the reset signal to the external device, set the P130 pin for high-level output by software.

Note 2. The state of the P40 pin is as follows.

- High-impedance during the external reset period or reset period by the POR.
- High level during other types of reset and after the reset is released (connected to the on-chip pull-up resistor).

A reset from the POR circuit or by LVD0 voltage detection is released when $V_{DD} \geq V_{POR}$ or $V_{DD} \geq V_{LVD0}$ after the reset. After reset processing, execution of the program starts with the high-speed on-chip oscillator clock as the operating clock. For details, see **Section 33 Power-on-reset Circuit (POR)** or **Section 34 Voltage Detector (LVD)**.

Remark V_{POR}: POR power supply rise detection voltage
V_{LVD0}: LVD0 detection voltage

Table 32 - 1 Operating States during a Reset (1/2)

Item		Operating State during a Reset
System clock		Clock supply to the CPU is stopped.
Main system clock	f_{IH}	Operation stopped
	f_{IM}	
	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_{PLL}	Operation stopped
Subsystem clock	f_{XT}	Operable (operation stops in the POR reset state, the XT1 and XT2 pins are input port mode)
	f_{EXS}	Operable (operation stops in the POR reset state, the EXCLKS pin is input port mode)
	f_{IL}	Operation stopped
CPU		Operation stopped
FAA		Operation stopped
Code flash memory		Operation stopped
Data flash memory		Operation stopped
RAM		Operation stopped
Data shared memory		Operation stopped
Port (latch)		High-impedance**Note**
Timer array unit		Operation stopped
Timer RJ		
Timer RD2, PWMOPA		
Timer RG2		
Timer RX		
16-bit timers KB30, KB31, and KB32		
Realtime clock		Resets other than the POR reset: Operable POR reset: Only the values of calendar-related registers are retained.
32-bit interval timer		Operation stopped
Watchdog timer		
Clock output/buzzer output		
A/D converter		
D/A converter		
PGA, comparator		
Serial array unit		
Serial interface IICA		
Digital addressable lighting interface		
Data transfer controller		
Power-on-reset function		Detection operation possible
Voltage detection function		LVD0: LVD0 operation is possible following an LVD0 reset but is stopped following other types of reset. LVD1: Operation stopped

Table 32 - 1 Operating States during a Reset (2/2)

Item		Operating State during a 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		

Note P40 and P130 become the following states.

- P40: High-impedance during the external reset period or reset period by the POR. High level during other types of reset (connected to the on-chip pull-up resistor).
- P130: Low-level output during the reset period

Remark f_{IH}: High-speed on-chip oscillator clock
f_{IM}: Middle-speed on-chip oscillator clock
f_{XT}: XT1 oscillation clock
f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 oscillation clock
f_{EX}: External main system clock
f_{EXS}: External subsystem clock
f_{PLL}: PLL clock

Table 32 - 2 States of the Hardware Blocks after a Reset Is Released

Hardware		State after a Reset Is Released
Program counter (PC)		The contents of the reset vector table (00000H, 00001H) are set.**Note**
Stack pointer (SP)		Undefined
Program status word (PSW)		06H
RAM	Data memory	Undefined
	General-purpose registers	Undefined

Note The contents of the PC are undefined during a reset and until the clock oscillation becomes stable after the reset is released.

Remark For the states of the special function registers (SFRs) after a reset is released, see **3.1.4 Special function register (SFR) area** and **3.1.5 Extended special function register (2nd SFR: 2nd special function register) area**.

32.2 Registers to Control the Reset Function

The following registers are used to control the reset function.

- Reset control flag register (RESF)
- Power-on-reset status register (PORSR)
- Peripheral reset control register 0 (PRR0)
- Peripheral reset control register 1 (PRR1)
- Peripheral reset control register 2 (PRR2)

32.2.1 Reset control flag register (RESF)

Many internal reset generation sources exist in the RL78 microcontroller. The RESF register is used to indicate which source has generated the reset request. The RESF register can be read by an 8-bit memory manipulation instruction. The flags TRAP, WDTRF, RPERF, IAWRF, and LVIRF are automatically cleared by any of the following event.

- Reset input via the $\overline{\text{RESET}}$ pin
- Reset by the power-on-reset (POR) circuit
- The RESF register is accessed.

Figure 32 - 4 Format of Reset Control Flag Register (RESF) (1/2)

Address: FFFA8H
After reset: Undefined**Note 1**
R/W: 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 by RAM parity error
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

IAWRF	Internal reset request by illegal-memory access
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

Figure 32 - 4 Format of Reset Control Flag Register (RESF) (2/2)

LVIRF	Internal reset request by voltage detector (LVD0 or LVD1)
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 reset source. See **Table 32 - 3 State of the RESF Register When Reset Request Is Generated**.

Note 2. In normal operation, executing the instruction code FFH triggers an internal reset, but this is not the case during emulation by the in-circuit emulator or on-chip debugging emulator.

Caution 1. Do not read data by a 1-bit memory manipulation instruction.

Caution 2. While parity error resets are enabled (RPECTL.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 generation enables RAM parity error resets (RPERDIS = 0). For details, see 35.3.4 RAM parity error detection.

The state of the RESF register when a reset request is generated is shown in **Table 32 - 3**.

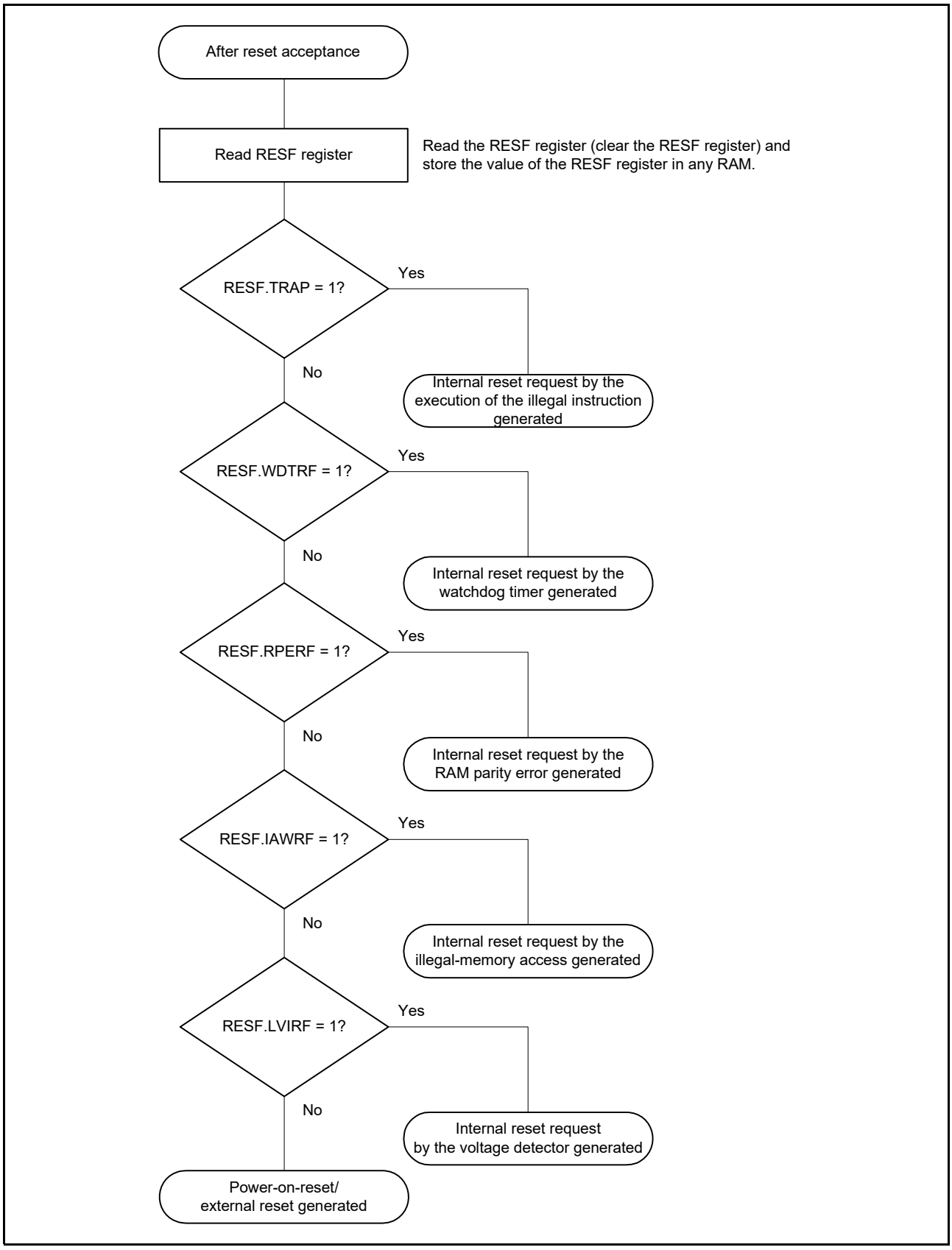
Table 32 - 3 State of the RESF Register When Reset Request Is Generated

Reset Source / Flag	$\overline{\text{RESET}}$ Input	Reset by the Power-on-reset (POR) Circuit	Reset by Execution of Illegal Instruction	Reset by the Watchdog Timer	Reset by RAM Parity Error	Reset by Illegal-memory Access	Reset by LVD0 or LVD1
TRAP	Cleared to 0	Cleared to 0	Set to 1	Retained	Retained	Retained	Retained
WDTRF			Retained	Set to 1			
RPERF				Retained	Set to 1		
IAWRF					Retained	Set to 1	
LVIRF						Retained	Set to 1

The RESF register is automatically cleared after it is read by using an 8-bit memory manipulation instruction.

Figure 32 - 5 shows the procedure for checking a reset source.

Figure 32 - 5 Example of Procedure for Checking Reset Source



The above flow is an example of the procedure for checking a reset source.

32.2.2 Power-on-reset status register (PORSR)

The PORSR register is used to check the occurrence of a power-on reset. Only writing 1 to bit 0 (PORF) of the PORSR register enables this function. Writing 0 to the bit has no effect. Write 1 to the PORF bit in advance to enable checking of the occurrence of a power-on reset. The PORSR register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register is 00H following a power-on reset.

Caution 1. The PORSR register is reset only by a power-on reset; it retains the value when a reset caused by another source occurs.

Caution 2. If the PORF flag is set to 1, it guarantees that no power-on reset has occurred, but it does not guarantee that the RAM value is retained.

Figure 32 - 6 Format of Power-on-reset Status Register (PORSR)

Address: F00F9H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	<0>
PORSR	0	0	0	0	0	0	0	PORF

PORF	Checking occurrence of power-on reset
0	A value 1 has not been written, or a power-on reset has occurred.
1	No power-on reset has occurred.

32.2.3 Peripheral reset control register 0 (PRR0)

The PRR0 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. The PRR0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 32 - 7 Format of Peripheral Reset Control Register 0 (PRR0)

Address: F00F1H
After reset: 00H
R/W: R/W

Symbol	7	6	<5>	<4>	<3>	<2>	1	<0>
PRR0	0	0	ADCRES	IICA0RES	SAU1RES	SAU0RES	0	TAU0RES

Bit n	Control resetting of the on-chip peripheral modules
0	The corresponding on-chip peripheral module is released from the reset state.
1	The corresponding on-chip peripheral module is in the reset state. • The SFRs for use with the corresponding on-chip peripheral module are initialized.

Remark n = 5 to 2, 0

Table 32 - 4 lists the on-chip peripheral modules controlled by individual bits.

Table 32 - 4 On-chip Peripheral Modules Controlled by Individual Bits in the PRR0 Register

Bit	Bit Name	Controlled On-chip Peripheral Modules
5	ADCRES	A/D converter
4	IICA0RES	Serial interface IICA0
3	SAU1RES	Serial array unit 1
2	SAU0RES	Serial array unit 0
0	TAU0RES	Timer array unit 0

Caution Be sure to set bits 7, 6, and 1 to 0.

32.2.4 Peripheral reset control register 1 (PRR1)

The PRR1 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. The PRR1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 32 - 8 Format of Peripheral Reset Control Register 1 (PRR1)

Address: F00FBH
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	3	2	1	<0>
PRR1	DACRES	0	PGACMPRES	TML32RES	0	0	0	DALIRES

Bit n	Control resetting of the on-chip peripheral modules
0	The corresponding on-chip peripheral module is released from the reset state.
1	The corresponding on-chip peripheral module is in the reset state. • The SFRs for use with the corresponding on-chip peripheral module are initialized.

Remark n = 7, 5, 4, 0

Table 32 - 5 lists the on-chip peripheral modules controlled by individual bits.

Table 32 - 5 On-chip Peripheral Modules Controlled by Individual Bits in the PRR1 Register

Bit	Bit Name	Controlled On-chip Peripheral Modules
7	DACRES	D/A converter
5	PGACMPRES	Comparator
4	TML32RES	32-bit interval timer
0	DALIRES	Digital addressable lighting interface

Caution Be sure to set bits 6 and 3 to 1 to 0.

32.2.5 Peripheral reset control register 2 (PRR2)

The PRR2 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. The PRR2 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 32 - 9 Format of Peripheral Reset Control Register 2 (PRR2)

Address: F00FDH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PRR2	FAARES	MEMRES	TKBRES	TRGRES	TRD0RES	PWMOPRES	TRXRES	TRJ0RES

Bit n	Control resetting of the on-chip peripheral modules
0	The corresponding on-chip peripheral module is released from the reset state.
1	The corresponding on-chip peripheral module is in the reset state. • The SFRs for use with the corresponding on-chip peripheral module are initialized.

Remark n = 7 to 0

Table 32 - 6 lists the on-chip peripheral modules controlled by individual bits.

Table 32 - 6 On-chip Peripheral Modules Controlled by Individual Bits in the PRR2 Register

Bit	Bit Name	Controlled On-chip Peripheral Modules
7	FAARES	FAA, divider
6	MEMRES	Data shared memory
5	TKBRES	16-bit timers KB30, KB31, and KB32
4	TRGRES	Timer RG2
3	TRD0RES	Timer RD2
2	PWMOPRES	PWM option unit A
1	TRXRES	Timer RX
0	TRJ0RES	Timer RJ

Section 33 Power-on-reset Circuit (POR)

33.1 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 **43.4 AC Characteristics** or **44.4 AC Characteristics**.
  This is done by utilizing LVD0 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 supply is turned off, this LSI should be placed in the STOP mode, or in the reset state by utilizing LVD0 or externally input reset signal, before the operating voltage falls below the range defined in **43.4 AC Characteristics** or **44.4 AC Characteristics**. When restarting the operation, make sure that the operating voltage has returned within the range of operation.

Caution **If an internal reset signal is generated by the power-on-reset circuit, the reset control flag register (RESF) and power-on-reset status register (PORSR) are cleared to 00H.**

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 **Section 32 Reset Function**.

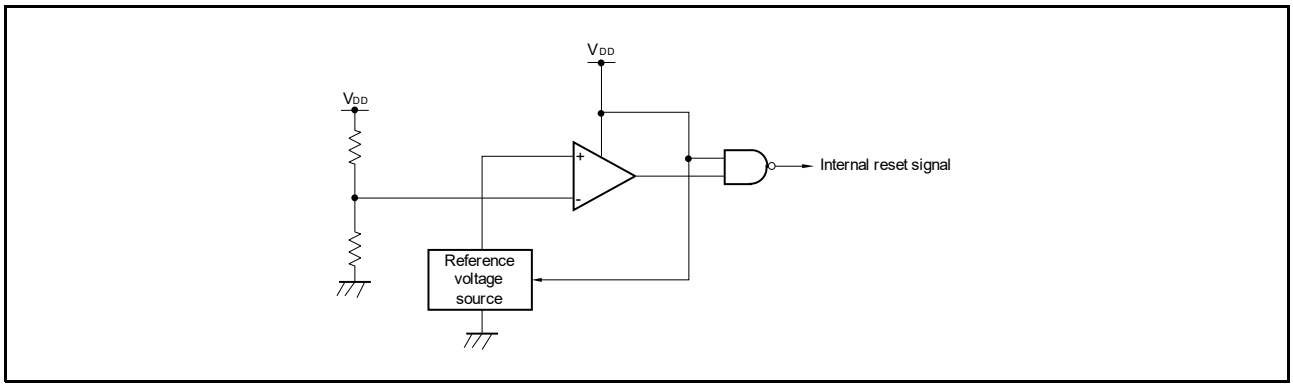
Remark 2. The power-on-reset status register (PORSR) is used to check the occurrence of an internal reset from the power-on-reset circuit. For details of the PORSR register, see **Section 32 Reset Function**.

Remark 3. V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage
For details, see **43.6.6 POR circuit characteristics** or **44.6.6 POR circuit characteristics**.

33.2 Configuration of Power-on-reset Circuit

The block diagram of the power-on-reset circuit is shown in **Figure 33 - 1**.

Figure 33 - 1 Block Diagram of Power-on-reset Circuit

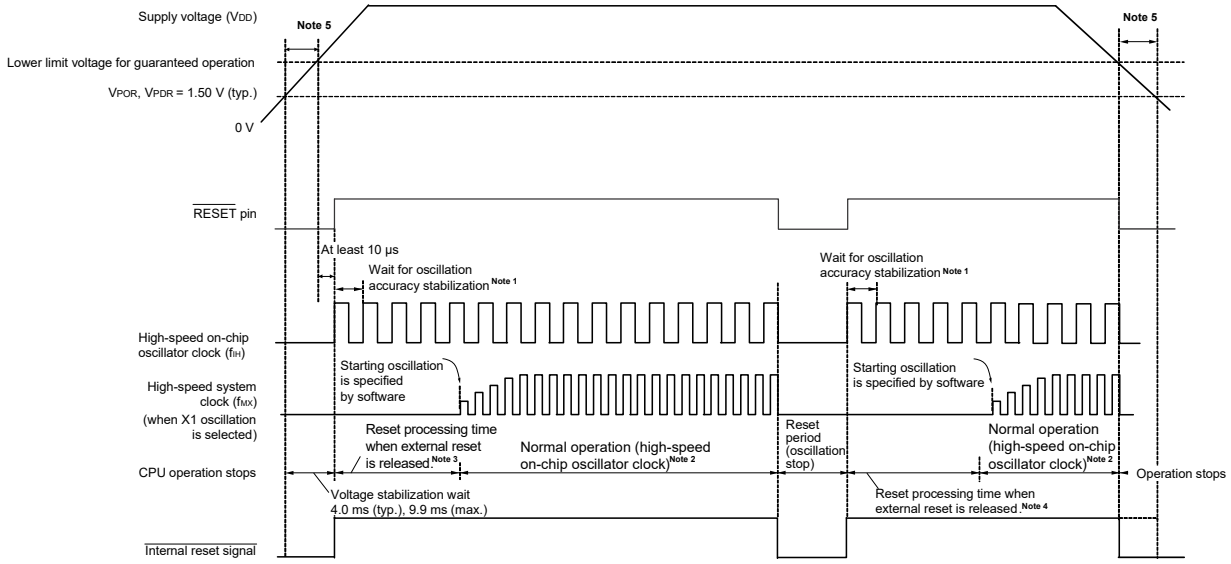


33.3 Operation of Power-on-reset Circuit

The timing of generation of the internal reset signal by the power-on-reset circuit and voltage detector is shown on the following pages.

Figure 33 - 2 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (1/2)

1. When the externally input reset signal on the $\overline{\text{RESET}}$ pin is used



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 (release from the first external reset following release from the POR state) after the $\overline{\text{RESET}}$ signal is driven high as well as the voltage stabilization wait time after VPOR (1.50 V, typ.) is reached.
With the LVD circuit in use: 0.506 ms (typ.), 0.694 ms (max.)
With the LVD circuit not in use: 0.201 ms (typ.), 0.335 ms (max.)

Note 4. The reset processing times in the case of the second or subsequent external reset following release from the POR state are listed below.
With the LVD circuit in use: 0.476 ms (typ.), 0.616 ms (max.)
With the LVD circuit not in use: 0.170 ms (typ.), 0.257 ms (max.)

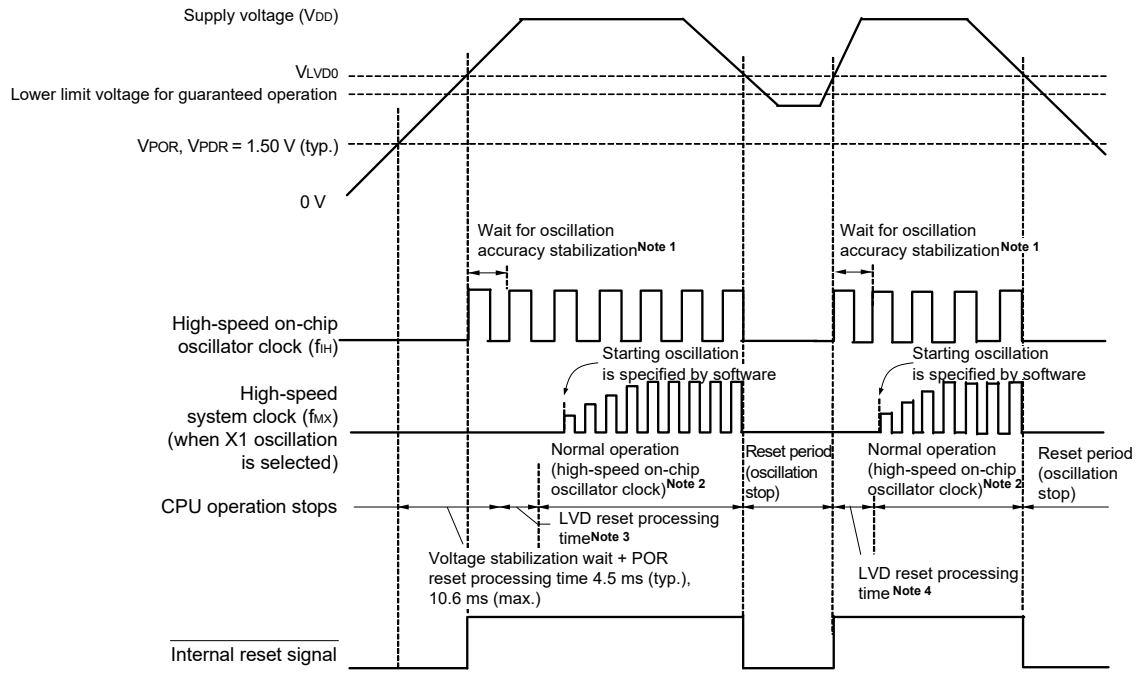
Note 5. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **43.4 AC Characteristics** or **44.4 AC Characteristics**. This is done by controlling the externally input reset signal.
After power supply is turned off, this LSI should be 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 range of operation. When restarting the operation, make sure that the operating voltage has returned within the range of operation.

Caution **For power-on reset, be sure to use the externally input reset signal on the $\overline{\text{RESET}}$ pin when LVD0 is off. For details, see Section 34 Voltage Detector (LVD).**

Remark VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

Figure 33 - 2 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (2/2)

2. LVD0 reset mode (LVD0SEL of the user option byte (000C1H) = 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 LVD0 detection level (V_{LVD0}) is reached as well as the voltage stabilization wait + POR reset processing time after the V_{POR} (1.50 V, typ.) is reached.
LVD reset processing time: 0 to 0.041 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 (LVD0), the following LVD reset processing time is required after the LVD0 detection level (V_{LVD0}) is reached.
LVD reset processing time: 0.040 ms (typ.), 0.041 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 LVD0 interrupt mode is selected (LVD0SEL of the user option byte (000C1H) = 0), the time until normal operation starts after power is turned on is the same as the time specified in **Note 3 of 2. LVD0 reset mode (LVD0SEL of the user option byte (000C1H) = 1) in Figure 33 - 2**.

Remark 3. Operation of LVD1 is stopped when power is initially supplied. LVD1 is also stopped by an internal reset.

Section 34 Voltage Detector (LVD)

34.1 Functions of Voltage Detector

Enabling, selecting the operation mode, and setting the detection voltage (V_{LVD0}) for voltage detector 0 (LVD0) are done by using a user option byte (000C1H). On the other hand, enabling, selecting the operation mode, and setting the detection voltage (V_{LVD1}) for voltage detector 1 (LVD1) are done by using the voltage detection level register.
The voltage detectors have the following functions.

- LVD0 and LVD1 compare the supply voltage (V_{DD}) with the detection voltage (V_{LVD0}, V_{LVD1}), and generate an internal reset or internal interrupt signal.
- The user option byte is used to select the detection voltage (V_{LVD0}) for LVD0 from among 6 voltages (for details, see **Section 38 Option Bytes**).
- The voltage detection level register is used to select the detection voltage (V_{LVD1}) for LVD1 from among 18 voltages.
- Operable in STOP mode.
- After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **43.4 AC Characteristics** or **44.4 AC Characteristics**. This is done by utilizing LVD0 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 LVD0 or controlling the externally input reset signal before the voltage falls below the operating range.

The internal reset and internal interrupt signals are generated in each mode as follows.

Reset Mode LVD0	Reset Mode LVD1	Interrupt Mode LVD0	Interrupt Mode LVD1
Deasserts an internal reset signal on detecting $V_{DD} \geq V_{LVD0}$. Generates an internal reset on detecting $V_{DD} < V_{LVD0}$ and retains the reset state until $V_{DD} \geq V_{LVD0}$ is detected.	Generates an internal reset on detecting $V_{DD} < V_{LVD1}$ after LVD1 operation has been enabled.	Retains the state of an internal reset by the LVD immediately after a reset until $V_{DD} \geq V_{LVD0}$. Releases the LVD internal reset by detecting $V_{DD} \geq V_{LVD0}$. Generates an interrupt request signal (INTLVI) by detecting $V_{DD} < V_{LVD0}$ or $V_{DD} \geq V_{LVD0}$ after the LVD internal reset is released.	Generates an interrupt request signal (INTLVI) on detecting $V_{DD} < V_{LVD1}$ after LVD1 operation has been enabled. After the first detection, generates an interrupt request signal (INTLVI) on detecting $V_{DD} < V_{LVD1}$ or $V_{DD} \geq V_{LVD1}$.

While LVD0 or LVD1 is operating, whether the supply voltage is no less than or less than the detection level can be checked by reading the voltage detection flag (LVDnF: bits 0 and 1 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 **Section 32 Reset Function**.

34.2 Configuration of Voltage Detector

Figures 34 - 1 and **34 - 2** show the block diagrams of the voltage detectors.

Figure 34 - 1 Block Diagram of LVD0

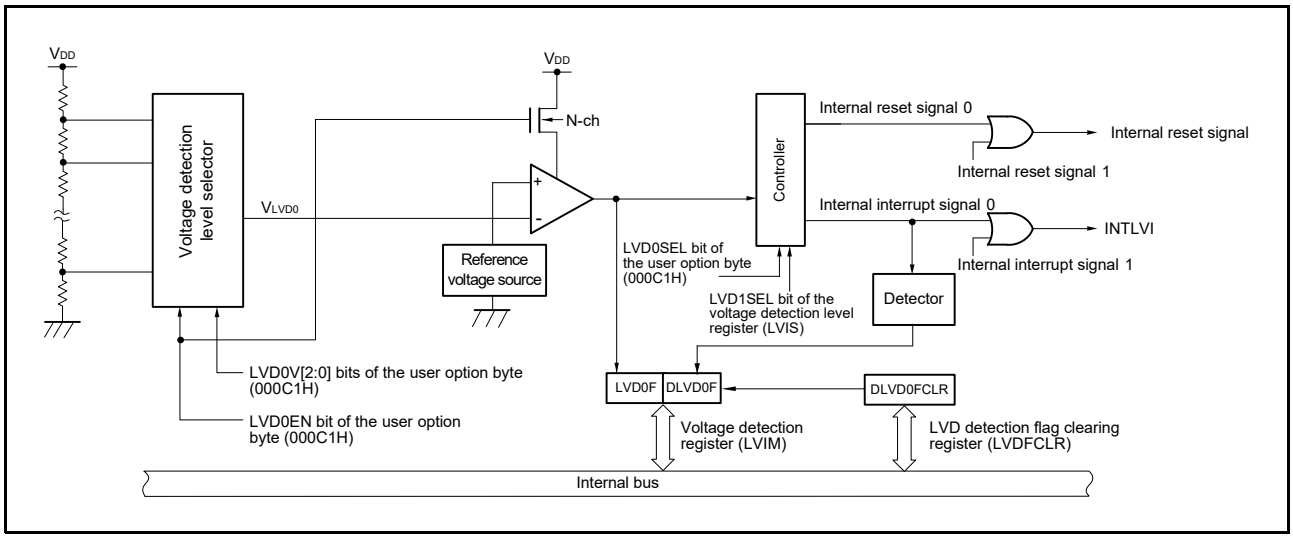
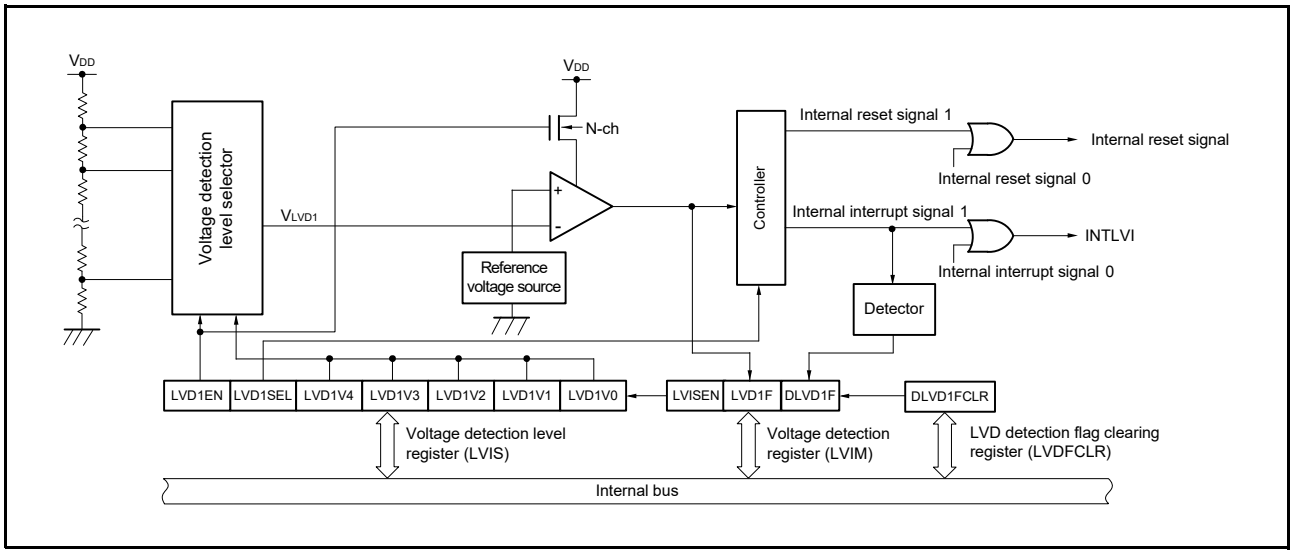


Figure 34 - 2 Block Diagram of LVD1



34.3 Registers to Control the Voltage Detector

The following registers are used to control the voltage detector.

- User option byte (000C1H/040C1H): See **Section 38 Option Bytes**.
- Voltage detection register (LVIM)
- LVD detection flag clearing register (LVDFCLR)
- Voltage detection level register (LVIS)

34.3.1 Voltage detection register (LVIM)

The LVIM register is used to specify whether to enable or disable rewriting the voltage detection level register (LVIS), as well as to check the states of LVD0 and LVD1. This register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 34 - 3 Format of Voltage Detection Register (LVIM)

Address: FFFA9H
After reset: 00H**Note 1**
R/W: R/W**Notes 2, 3**

Symbol	<7>	6	5	4	<3>	<2>	<1>	<0>
LVIM	LVISEN	0	0	0	DLVD1F	DLVD0F	LVD1F	LVD0F

LVISEN	Specification of whether to enable or disable rewriting the voltage detection level register (LVIS)
0	Rewriting of the LVIS register is disabled.
1**Note 4**	Rewriting of the LVIS register is enabled (reset and interrupt generation by LVD1 are masked).

DLVDnF	LVDn detection interrupt flag
0	The given LVDn interrupt has not been detected.
1	The given LVDn interrupt has been detected.

LVDnF	Voltage detection flag
0	Supply voltage (V_{DD}) ≥ detection voltage (V_{LVDn}), or when LVD is off
1	Supply voltage (V_{DD}) < detection voltage (V_{LVDn})

Note 1. The value after a reset is 01H when LVD0 operation is enabled and the power supply voltage (V_{DD}) is less than the detection voltage (V_{LVD0}).

Note 2. Bits 0 and 1 are read-only.

Note 3. Bits 2 and 3 are read-only. These bits are cleared by using the LVD detection flag clearing register (LVDFCLR).

Note 4. While the LVISEN bit is 1, the reset and interrupt generation by LVD1 are masked. Therefore, clear the LVISEN bit to 0 after having written a new value to the LVIS register.

Remark n = 0, 1

34.3.2 LVD detection flag clearing register (LVDFCLR)

The LVDFCLR register is used to clear the interrupt detection flags (DLVD0F and DLVD1F) of the voltage detection register (LVIM). The LVDFCLR register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 34 - 4 Format of LVD Detection Flag Clearing Register (LVDFCLR)

Address: F0218H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	<3>	<2>	1	0
LVDFCLR	0	0	0	0	DLVD1FCLR	DLVD0FCLR	0	0

DLVD1FCLR[Note]	Clearing DLVD1F
0	No effect
1	Writing 1 to this bit clears the DLVD1F flag.

DLVD0FCLR[Note]	Clearing DLVD0F
0	No effect
1	Writing 1 to this bit clears the DLVD0F flag.

Note Only 1 can be written to this bit. Writing 0 has no effect.
The bit is read as 0 even after 1 has been written to it.

34.3.3 Voltage detection level register (LVIS)

The LVIS register is used to select the voltage detection level for LVD1. This register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 19H.

Figure 34 - 5 Format of Voltage Detection Level Register (LVIS)

Address: FFFAAH
After reset: 19H
R/W: R/W

Symbol	<7>	<6>	5	4	3	2	1	0
LVIS	LVD1EN	LVD1SEL	0	LVD1V4	LVD1V3	LVD1V2	LVD1V1	LVD1V0

LVD1EN	Enabling operation of LVD1
0	Operation stopped
1	Operation enabled

LVD1SEL	Operation mode of LVD1
0	Interrupt mode
1	Reset mode

LVD1V4	LVD1V3	LVD1V2	LVD1V1	LVD1V0	Detection voltages for LVD1 Notes 1, 3, 4		
						Rising edge	Falling edge
1	1	1	1	1	V_{LVD117}	1.67 V Note 2	1.63 V Note 2
1	1	1	1	0	V_{LVD116}	1.78 V Note 2	1.74 V Note 2
1	1	1	0	1	V_{LVD115}	1.88 V Note 2	1.84 V Note 2
1	1	1	0	0	V_{LVD114}	1.98 V	1.94 V
1	1	0	1	1	V_{LVD113}	2.09 V	2.04 V
1	1	0	1	0	V_{LVD112}	2.20 V	2.15 V
1	1	0	0	1	V_{LVD111}	2.30 V	2.25 V
1	1	0	0	0	V_{LVD110}	2.40 V	2.35 V
1	0	1	1	1	V_{LVD19}	2.50 V	2.45 V
1	0	1	1	0	V_{LVD18}	2.66 V	2.60 V
1	0	1	0	1	V_{LVD17}	2.82 V	2.76 V
1	0	1	0	0	V_{LVD16}	2.97 V	2.91 V
1	0	0	1	1	V_{LVD15}	3.13 V	3.06 V
1	0	0	1	0	V_{LVD14}	3.35 V	3.27 V
1	0	0	0	1	V_{LVD13}	3.55 V	3.47 V
1	0	0	0	0	V_{LVD12}	3.75 V	3.67 V
0	1	1	1	1	V_{LVD11}	3.96 V	3.88 V
0	1	1	1	0	V_{LVD10}	4.16 V	4.08 V

Note 1. The LVD1V4 to LVD1V0 bits can only be rewritten once after release from the reset state.
Note 2. This setting can only be used when LVD0 is off.
Note 3. When setting LVD0 to reset mode, set the detection voltage of LVD1 higher than the detection voltage of LVD0.

(Note and Caution are listed on the next page.)

Note 4. If LVD0 is set to interrupt mode and the LVD0 detection voltage is greater than the LVD1 detection voltage, LVD0 becomes undefined after the LVD1 setting following release from the reset state.

Caution When the values in the LVIS register are to be changed, do so according to the procedure described in Figures 34 - 10 and 34 - 11.

34.4 Operation of Voltage Detector

34.4.1 When used as reset mode

Enabling, selecting the operation mode (reset mode: LVD0SEL = 1), and setting the detection voltage (V_{LVD0}) for LVD0 are done by using a user option byte (000C1H).
On the other hand, enabling, selecting the operation mode (reset mode: LVD1SEL = 1), and setting the detection voltage (V_{LVD1}) for LVD1 are done by using the voltage detection level register (LVIS).

- Operation in LVD reset mode
  When LVD0 is set for the reset mode (the value of the LVD0SEL bit in the user option byte is 1), the state of the internal reset being applied by LVD0 is retained until the power supply voltage (V_{DD}) exceeds the rising voltage detection level (V_{LVD0}) after power has been supplied. The internal reset is released when the supply voltage (V_{DD}) exceeds the rising voltage detection level (V_{LVD0}).
  At the fall of the operating voltage, an internal reset by LVD0 is generated when the power supply voltage (V_{DD}) falls below the falling voltage detection level (V_{LVD0}).
  Operation of LVD1 is stopped when power is initially supplied. When LVD1 operation is enabled, it generates an internal reset when the power supply voltage (V_{DD}) falls below the voltage detection level (V_{LVD1}). If operation of LVD1 is enabled while the power supply voltage (V_{DD}) is lower than the voltage detection level (V_{LVD1}), it generates an internal reset at the time its operation is enabled. If LVD1 is set for reset mode, LVD0 is placed in interrupt mode. In addition, the generation of an internal reset by LVD1 places LVD0 in reset mode.
  LVD1 detection voltage can only be set once after release from the reset state.

Figure 34 - 6 shows the timing of generation of the internal reset signal from LVD0 and **Figure 34 - 7** shows that of the signal from LVD1.

Figure 34 - 6 Timing of Generation of the Internal Reset Signal from LVD0

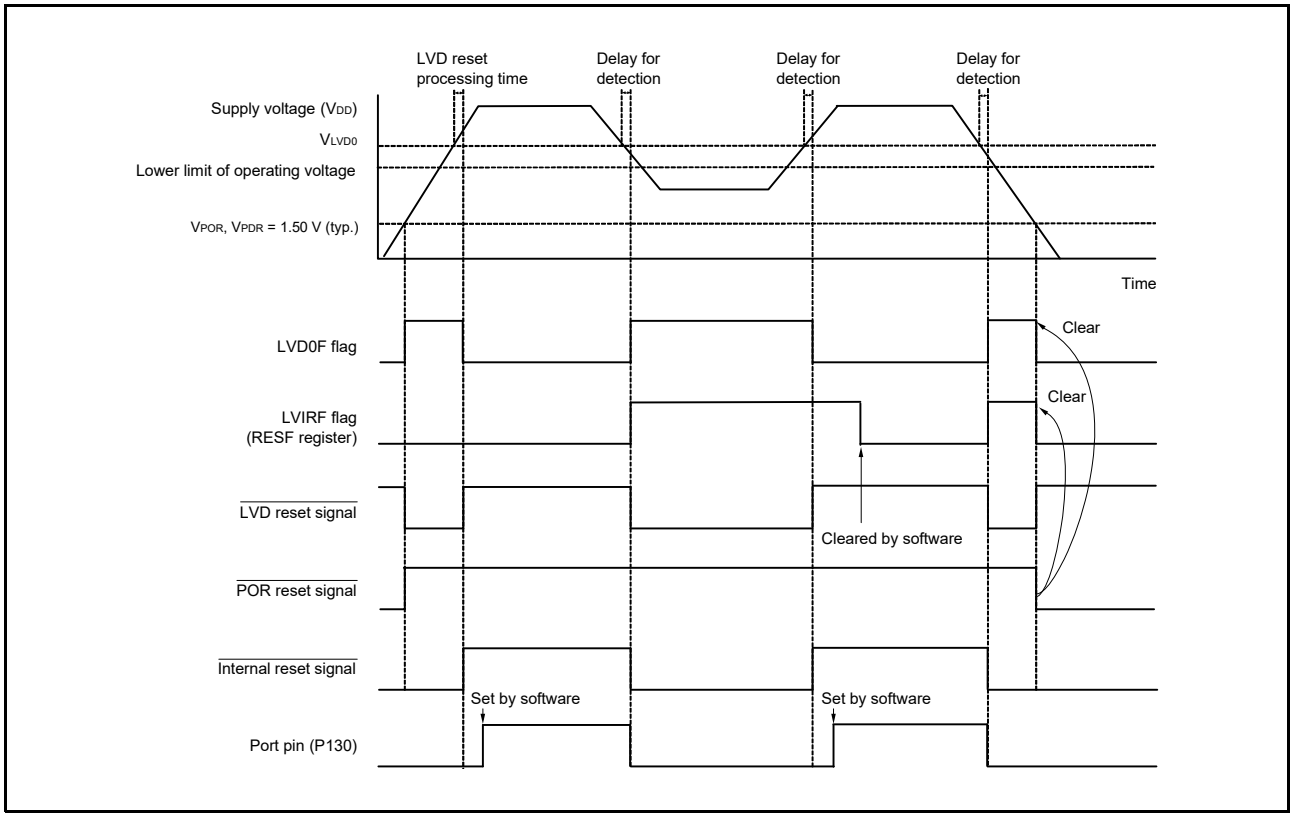
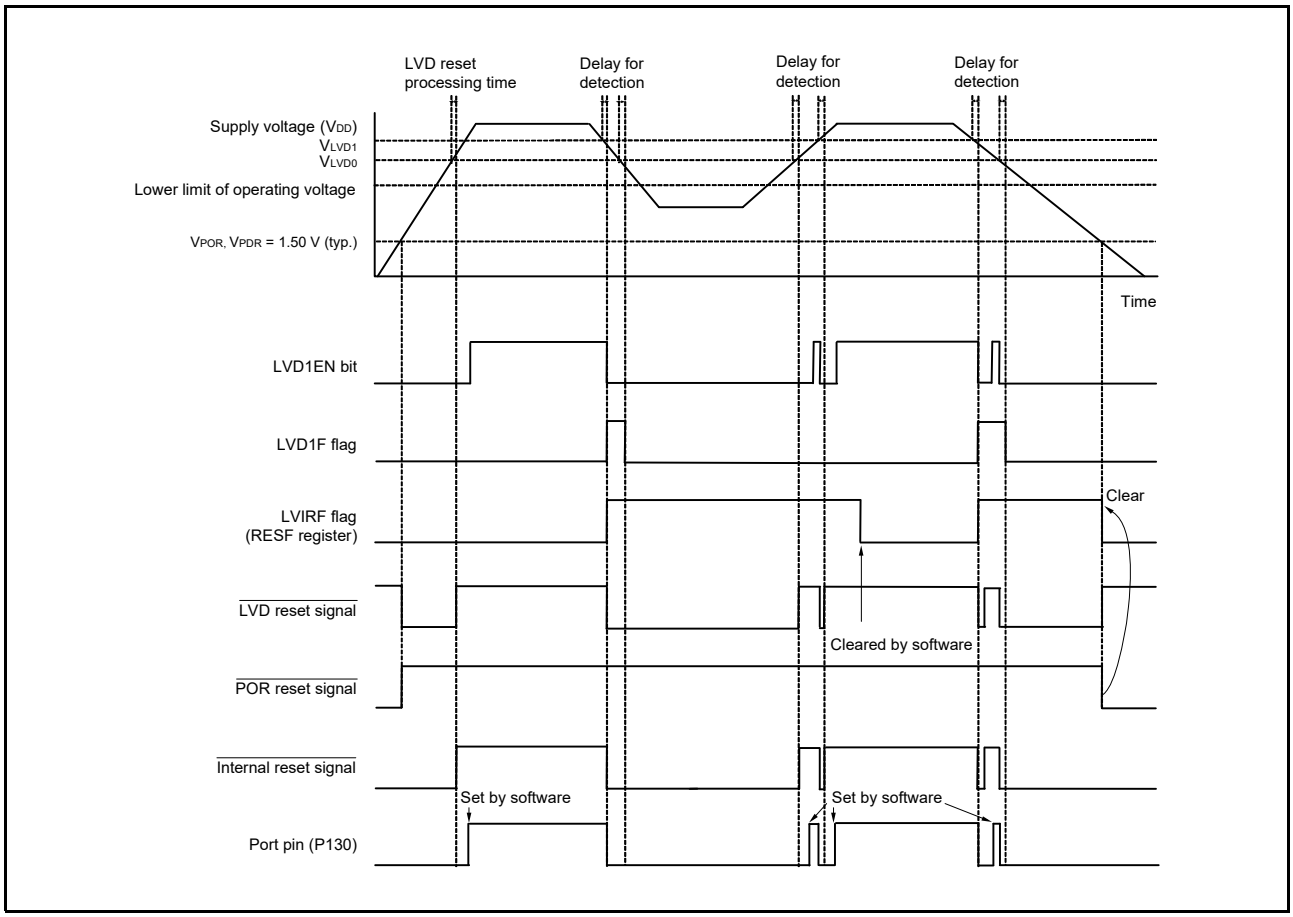


Figure 34 - 7 Timing of Generation of the Internal Reset Signal from LVD1



Remark LVD0: Reset mode

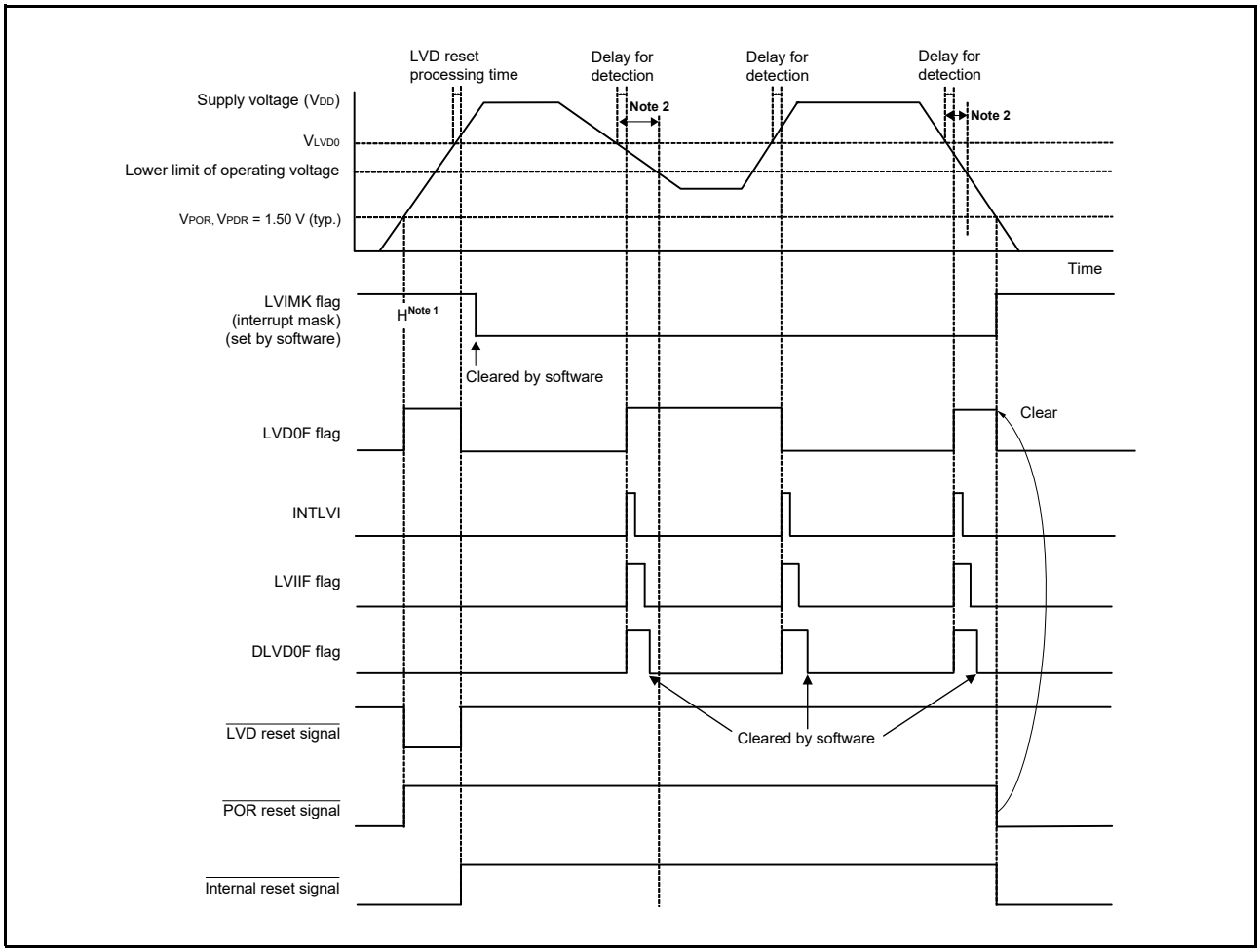
34.4.2 When used as interrupt mode

Enabling, selecting the operation mode (interrupt mode: LVD0SEL = 0), and setting the detection voltage (V_{LVD0}) for LVD0 are done by using a user option byte (000C1H).
On the other hand, enabling, selecting the operation mode (interrupt mode: LVD1SEL = 0), and setting the detection voltage (V_{LVD1}) for LVD1 are done by using the voltage detection level register (LVIS).

- Operation in LVD interrupt mode
  When LVD0 is set for the interrupt mode (the value of the LVD0SEL bit in the user option byte is 0), the state of the internal reset being applied by LVD0 is retained until the power supply voltage (V_{DD}) exceeds the rising voltage detection level (V_{LVD0}) immediately after a reset has been generated. The internal reset is released when the power supply voltage (V_{DD}) exceeds the rising voltage detection level (V_{LVD0}).
  After the internal reset signal has been deasserted, LVD0 generates an interrupt request signal (INTLVI) if the power supply voltage (V_{DD}) falls below the voltage detection level (V_{LVD0}). Similarly, when the power supply voltage (V_{DD}) rises above the voltage detection level (V_{LVD0}), LVD0 also generates an interrupt request signal (INTLVI). 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 **43.4 AC Characteristics** or **44.4 AC Characteristics**. When restarting the operation, make sure that the power supply voltage has returned within the operating voltage range.
  Operation of LVD1 is stopped when power is initially supplied. When LVD1 operation is enabled, it generates an interrupt request signal (INTLVI) when the power supply voltage (V_{DD}) falls below the voltage detection level (V_{LVD1}). Similarly, when the power supply voltage (V_{DD}) rises above the voltage detection level (V_{LVD1}), LVD1 also generates an interrupt request signal (INTLVI). Note that if operation of LVD1 is enabled while the power supply voltage (V_{DD}) is lower than the voltage detection level (V_{LVD1}), it generates an interrupt request signal (INTLVI) at the time its operation is enabled.
  LVD1 detection voltage can only be set once after release from the reset state.

Figure 34 - 8 shows the timing of generation of the interrupt signal from LVD0 and **Figure 34 - 9** shows that of the signal from LVD1.

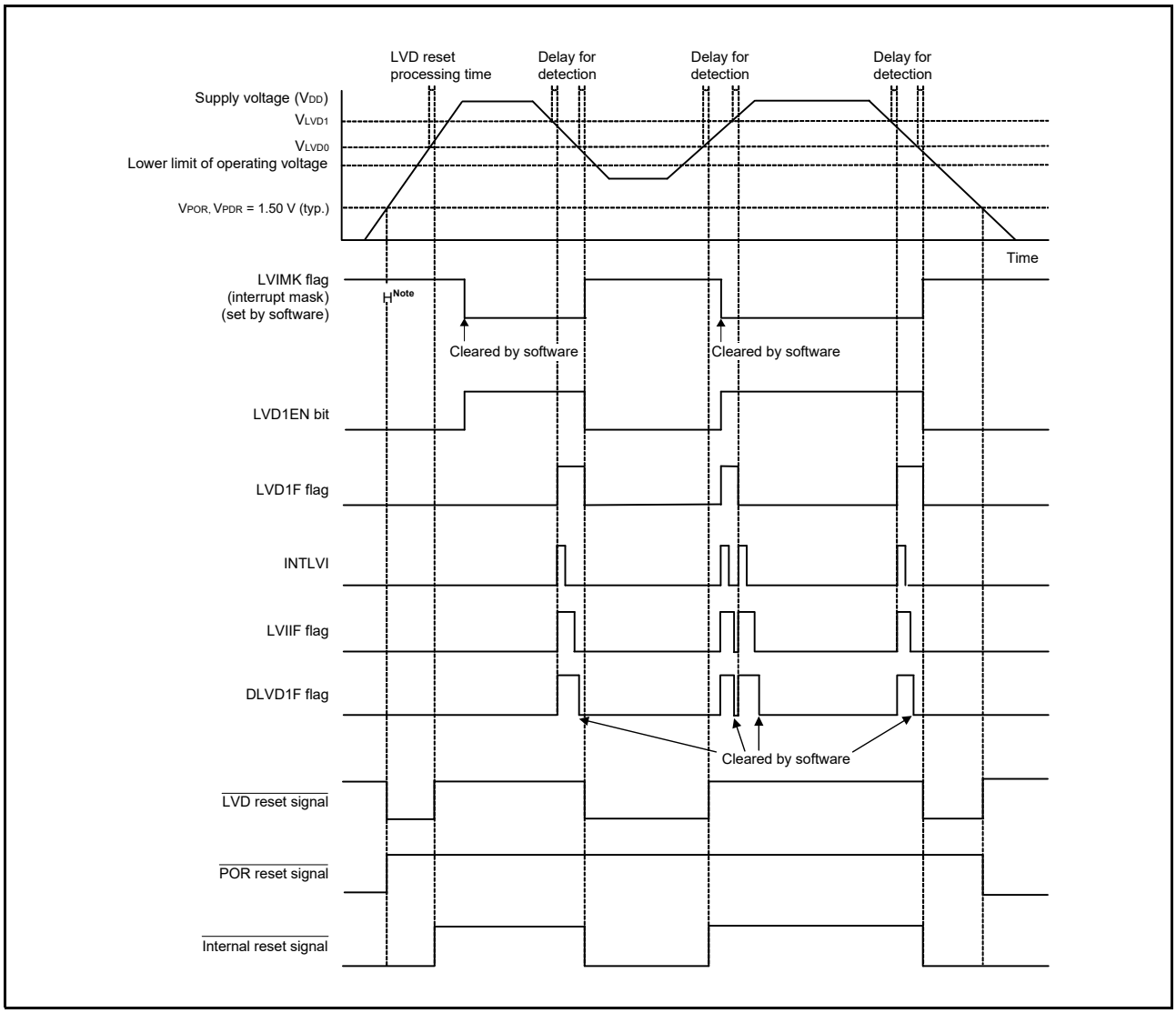
Figure 34 - 8 Timing of Generation of the Interrupt Signal from LVD0



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 **43.4 AC Characteristics** or **44.4 AC Characteristics**. When restarting the operation, make sure that the operating voltage has returned within the operating voltage range.

Figure 34 - 9 Timing of Generation of the Interrupt Signal from LVD1



Note The LVIMK flag is set to 1 by reset signal generation.

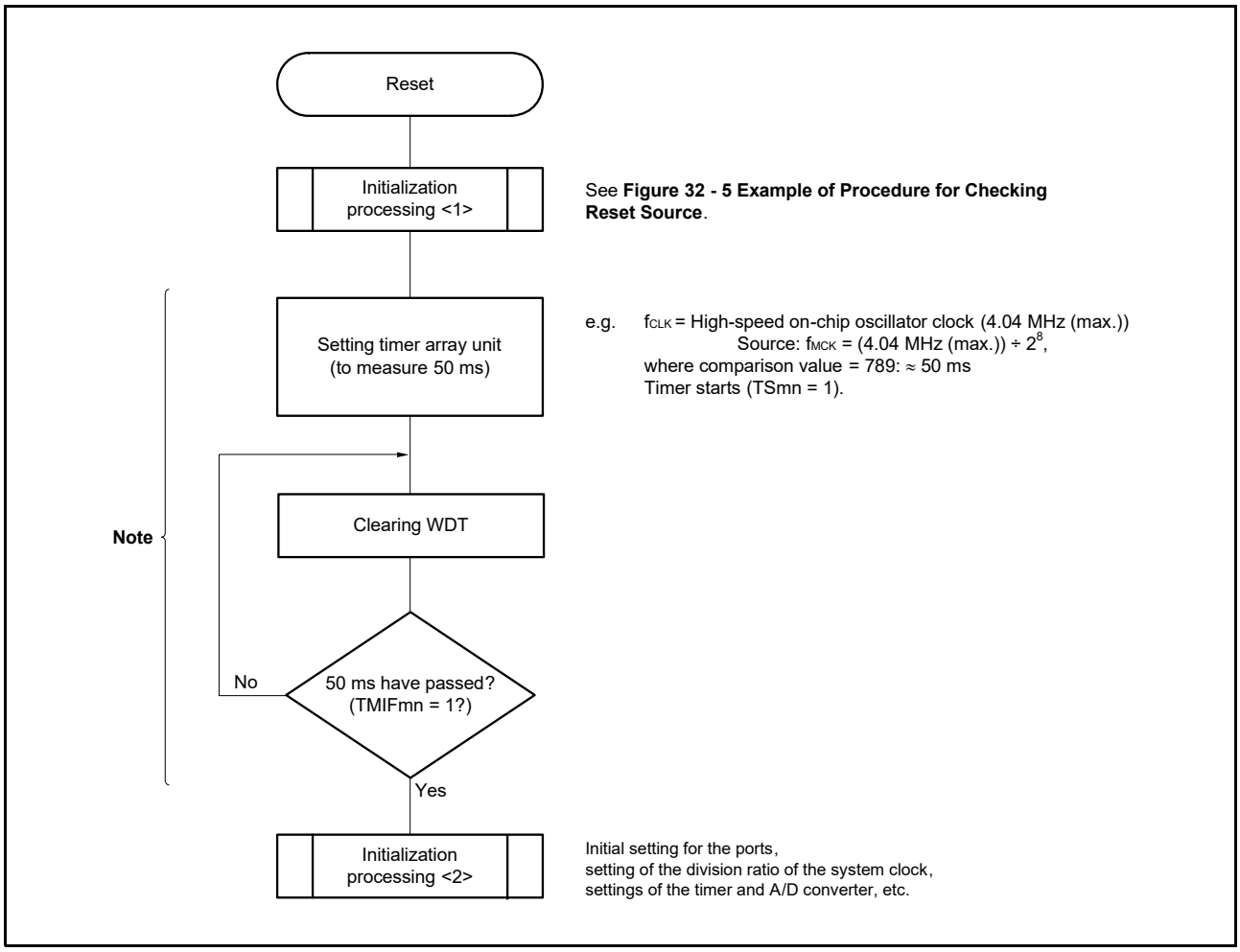
Remark LVD0: Reset mode

34.5 Points for Caution When the Voltage Detector Is to Be Used

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 detection voltages for LVD0 or LVD1, the system may be repeatedly reset and released from the reset state. 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 34 - 10 Example of Software Processing If Supply Voltage Fluctuation Is 50 ms or Less in Vicinity of the Detection Voltages for LVD0 or LVD1

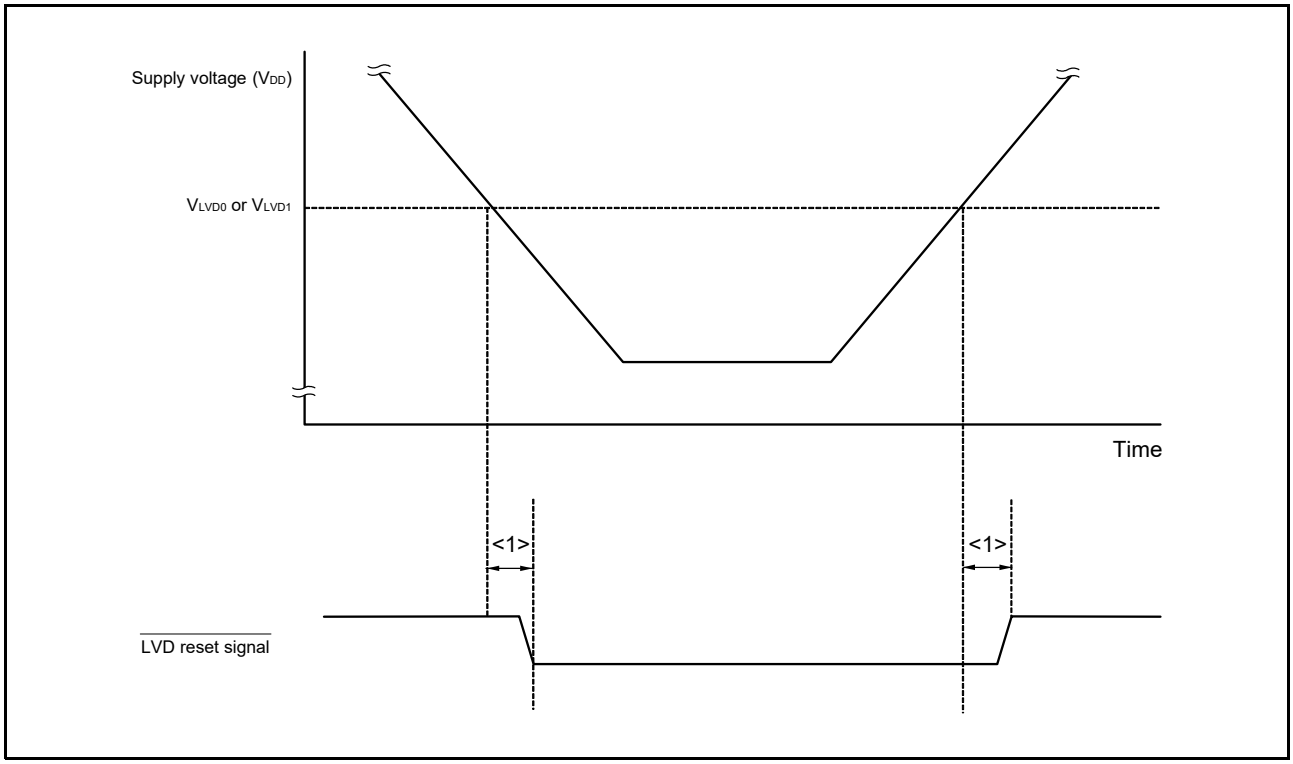


Note If reset is generated again during this period, initialization processing <2> is not started.

Remark m = 0, 1; n = 0 to 7

2. Delays from the time an LVD0 or LVD1 reset source condition is satisfied until an LVD0 or LVD1 reset has been generated and deasserted
   The delay is from the time the power supply voltage (V_{DD}) becomes less than the LVD0 or LVD1 falling detection voltage (V_{LVD0} or V_{LVD1}) until the LVD0 or LVD1 reset is generated. In the same way, the delay is from the time the LVD0 or LVD1 rising detection voltage (V_{LVD0} or V_{LVD1}) becomes no greater than the power supply voltage (V_{DD}) until the LVD0 or LVD1 reset is deasserted. See **Figure 34 - 11**.

Figure 34 - 11 Delays from the Time an LVD0 or LVD1 Reset Source Condition Is Satisfied until an LVD0 or LVD1 Reset has been Generated and Deasserted



Remark <1>: Delay for detection (300 μs (max.))

3. Turning power on when LVD0 is off
   Use the external rest input via the $\overline{\text{RESET}}$ pin when LVD0 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 **43.4 AC Characteristics** or **44.4 AC Characteristics**, and then input a high level to the pin.

4. Operating voltage fall when LVD0 is off or the interrupt mode is selected
   When the operating voltage falls with LVD0 off or with the interrupt mode 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 **43.4 AC Characteristics** or **44.4 AC Characteristics**. When restarting the operation, make sure that the operating voltage has returned within the operating voltage range.

5. Procedure for setting the LVD1 detection voltage
   Follow the procedure below to set the LVD1 detection voltage. After step <3>, LVD1 is enabled after the stabilization waiting time (at least 500 μs) has elapsed.
   <1> Set the LVISEN bit of the LVIM register to 1.
   <2> Set the LVD1EN bit of the LVIS register to 1 and change the setting of the LVD1V[4:0] bits.
   <3> Set the LVISEN bit of the LVIM register to 0.

Section 35 Safety Functions

35.1 Overview of Safety Functions

The RL78/G24 provides the following safety functions to comply with the IEC60730 and IEC61508 safety standards. These safety functions enable the microcontroller to self-diagnose abnormalities and safely stop operating if an abnormality is detected.

1. Flash memory CRC operation (high-speed CRC, general-purpose CRC)
   This detects data errors in the flash memory by performing CRC operations.
   The following two CRC functions are provided in the RL78/G24 and they 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-purpose CRC: This can be used for checking various data in addition to the code flash memory area while the CPU is running.
2. Flash memory guard function
   This prevents rewriting of data in the flash memory due to incorrect CPU operations.
3. RAM parity error detection
   This detects parity errors when reading RAM data.
4. RAM guard function
   This prevents rewriting of data in RAM due to incorrect CPU operations.
5. SFR guard function
   This prevents rewriting of data in the SFRs due to incorrect CPU operations.
6. Illicit memory access detection
   This detects illicit accesses to invalid memory areas (such as areas where no memory is allocated and areas to which access is restricted).
7. Guard function of invalid memory access detection control register (IAWCTL)
   This prevents rewriting of the invalid memory access detection control register due to incorrect CPU operations.
8. Frequency detection
   This function allows a self-check of the CPU/peripheral hardware clock frequencies using the timer array unit.
9. Testing of the A/D converter
   This test checks whether or not the A/D converter is operating normally by converting the A/D converter's positive and negative reference voltages, analog input channels (ANIxx), temperature sensor output voltage, and internal reference voltage.
10. Detection of the digital output signal level of the I/O pins
    This is used to read the output level of an I/O pin when the pin is in the output mode.
11. UART loopback
    This is used to confirm that the transmit data is output normally by shutting off the TXDn and RXDn pins of UARTn from the outside and connecting them within the MCU to loop back the output from the transmission shift register to the reception shift register.

    Remark n = 0 to 2

Remark For usage examples of the safety functions complying with the IEC60730 and IEC61508 safety standards, refer to the **application notes IEC60730/60335 Self Test Library for RL78 MCU**.

35.2 Registers to Control the Safety Functions

The following registers are used to control the safety functions.

- Flash memory CRC control register (CRC0CTL)
- Flash memory CRC operation result register (PGCRCL)
- CRC input register (CRCIN)
- CRC data register (CRCD)
- Code flash memory guard register (GFLASH0)
- Data flash memory guard register (GFLASH1)
- Flash security area guard register (GFLASH2)
- RAM parity error control register (RPECTL)
- RAM parity error control register 2 (RPECTL2)
- Invalid memory access detection control register (IAWCTL)
  - RAM guard function
  - SFR guard function
  - Illicit memory access detection
- Guard register of IAWCTL register (GIAWCTL)
- Timer I/O select register 0 (TIOS0)
- A/D test register (ADTES)
- Analog input channel specification register (ADS)
- Port mode select register (PMS)
- UART loopback select register (ULBS)

The content of each register is described in **35.3 Operation of Safety Functions**.

35.3 Operation of Safety Functions

35.3.1 Flash memory CRC operation function (high-speed CRC)

The IEC60730 standard mandates the checking of data in the flash memory, and recommends using a CRC to do it. The high-speed CRC provided in the RL78/G24 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 performs an operation by reading 32-bit data per clock from the flash memory while stopping the CPU. This function therefore can finish a check in a shorter time (for example, 512 µs@32 MHz with 64-Kbyte flash memory).
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-purpose CRC, because the general-purpose CRC operates in LSB first order.

35.3.1.1 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. The value of this register following a reset is 00H.

Figure 35 - 1 Format of Flash Memory CRC Control Register (CRC0CTL)

Address: F02F0H
After reset: 00H
R/W: R/W

Symbol	<7>	6	5	4	3	2	1	0
CRC0CTL	CRC0EN	0	FEA5	FEA4	FEA3	FEA2	FEA1	FEA0

CRC0EN	Control of CRC ALU operation
0	Stops the operation.
1	Starts the operation according to HALT instruction execution.

FEA5	FEA4	FEA3	FEA2	FEA1	FEA0	High-speed CRC operation range
0	0	0	0	0	0	0000H to 03FFBH (16 Kbytes - 4 bytes)
0	0	0	0	0	1	0000H to 07FFBH (32 Kbytes - 4 bytes)
0	0	0	0	1	0	0000H to 0BFFBH (48 Kbytes - 4 bytes)
0	0	0	0	1	1	0000H to 0FFFBH (64 Kbytes - 4 bytes)
0	0	0	1	0	0	0000H to 13FFBH (80 Kbytes - 4 bytes)
0	0	0	1	0	1	0000H to 17FFBH (96 Kbytes - 4 bytes)
0	0	0	1	1	0	0000H to 1BFFBH (112 Kbytes - 4 bytes)
0	0	0	1	1	1	0000H to 1FFFBH (128 Kbytes - 4 bytes)
Other than the above						Setting prohibited

Caution **Be sure to clear bit 6 to 0.**

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.

35.3.1.2 Flash memory CRC operation result register (PGCRCL)

This register holds the high-speed CRC operation results. The PGCRCL register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 35 - 2 Format of Flash Memory CRC Operation Result Register (PGCRCL)

Address: F02F2H
After reset: 0000H
R/W: 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

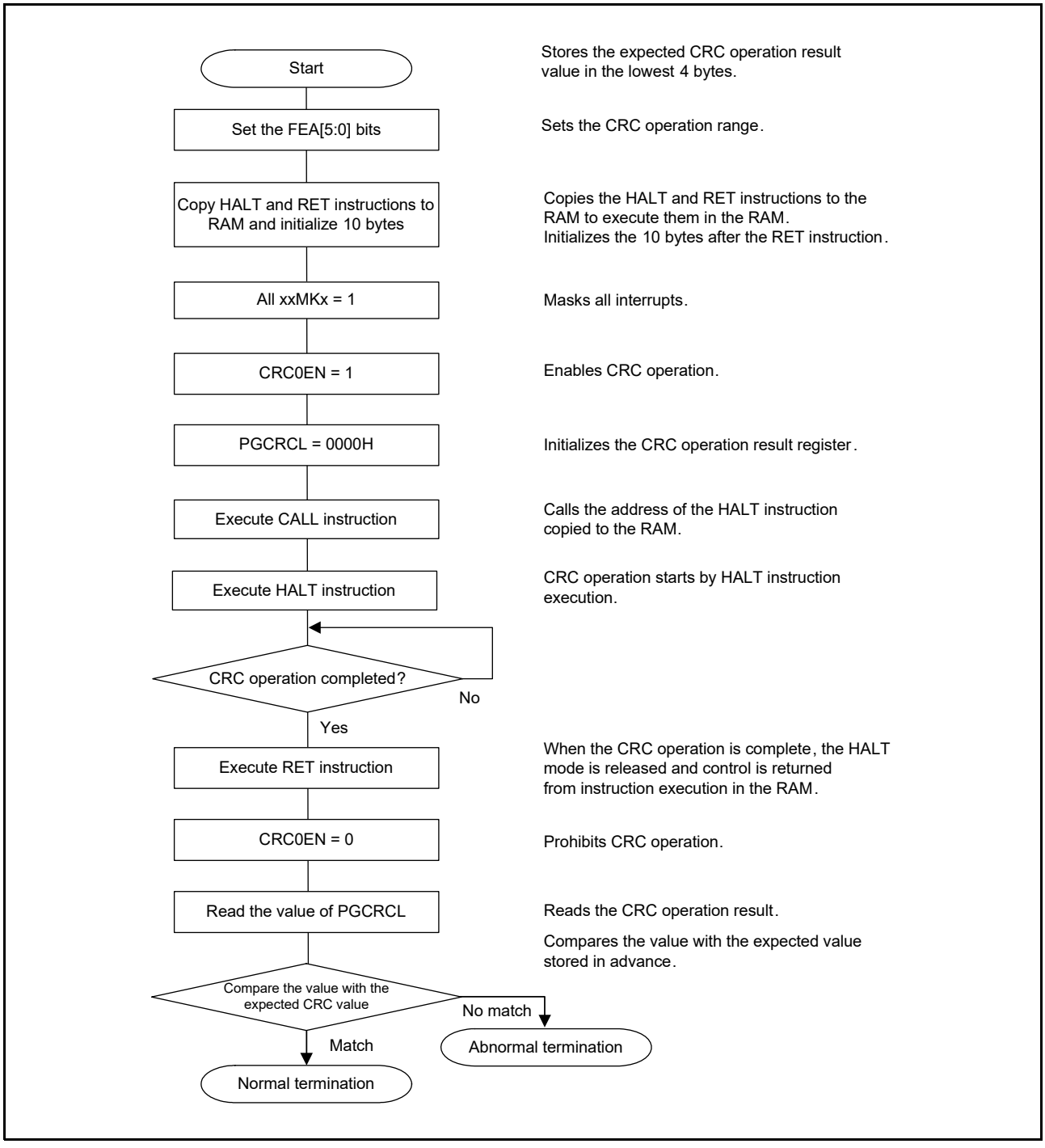
PGCRC15 to PGCRC0	High-speed CRC operation results
—	Holds the high-speed CRC operation results.

Caution The PGCRCL register can only be written if CRC0EN (bit 7 of the CRC0CTL register) = 1.

Figure 35 - 3 shows the flowchart of flash memory CRC operation function (high-speed CRC).

<Operation flow>

Figure 35 - 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 the RAM area.

The expected CRC value can be calculated by using the CS+ integrated development environment. See the **CS+ Integrated Development Environment user's manual** for details.

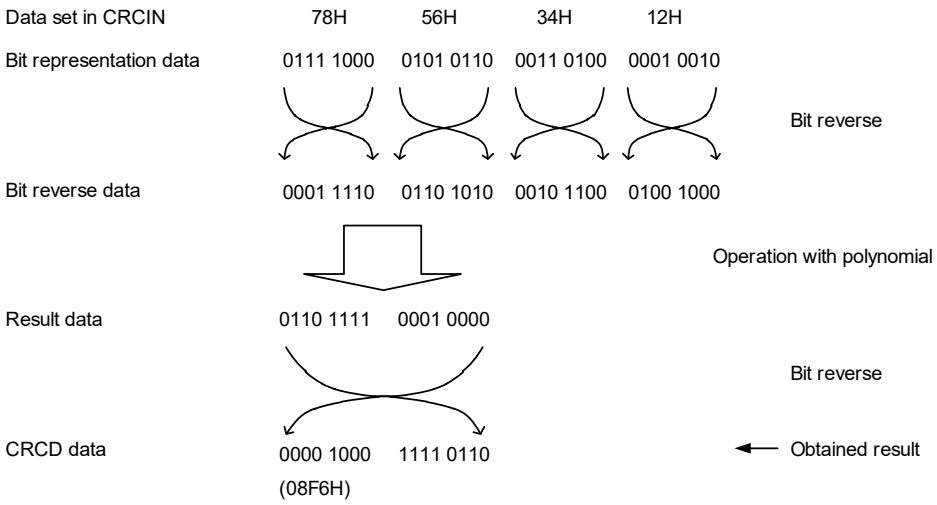
35.3.2 CRC operation (general-purpose CRC)

To guarantee safety during operation, the IEC61508 standard mandates the checking of data even while the CPU is operating.

The general-purpose CRC handles CRC operation as a peripheral module while the CPU is operating. The general-purpose 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 calculation function in the HALT mode can be used only during DTC transfer.

The general-purpose CRC can operate either in the main system clock operation mode or in the subsystem clock operation mode.

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.

35.3.2.1 CRC input register (CRCIN)

This is an 8-bit register to set the CRC operation data of the general-purpose CRC. The possible setting range is 00H to FFH. The CRCIN register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 35 - 4 Format of CRC Input Register (CRCIN)

Address: FFFACH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
CRCIN								

Bits 7 to 0	Function
00H to FFH	Data input

35.3.2.2 CRC data register (CRCD)

This register holds the CRC operation result of the general-purpose CRC. The setting range is 0000H to FFFFH. After one clock cycle of the CPU/peripheral hardware clock (f_{CLK}) has elapsed from the time CRCIN register was written to, the CRC operation result is stored in the CRCD register. The CRCD register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 35 - 5 Format of CRC Data Register (CRCD)

Address: F02FAH
After reset: 0000H
R/W: R/W

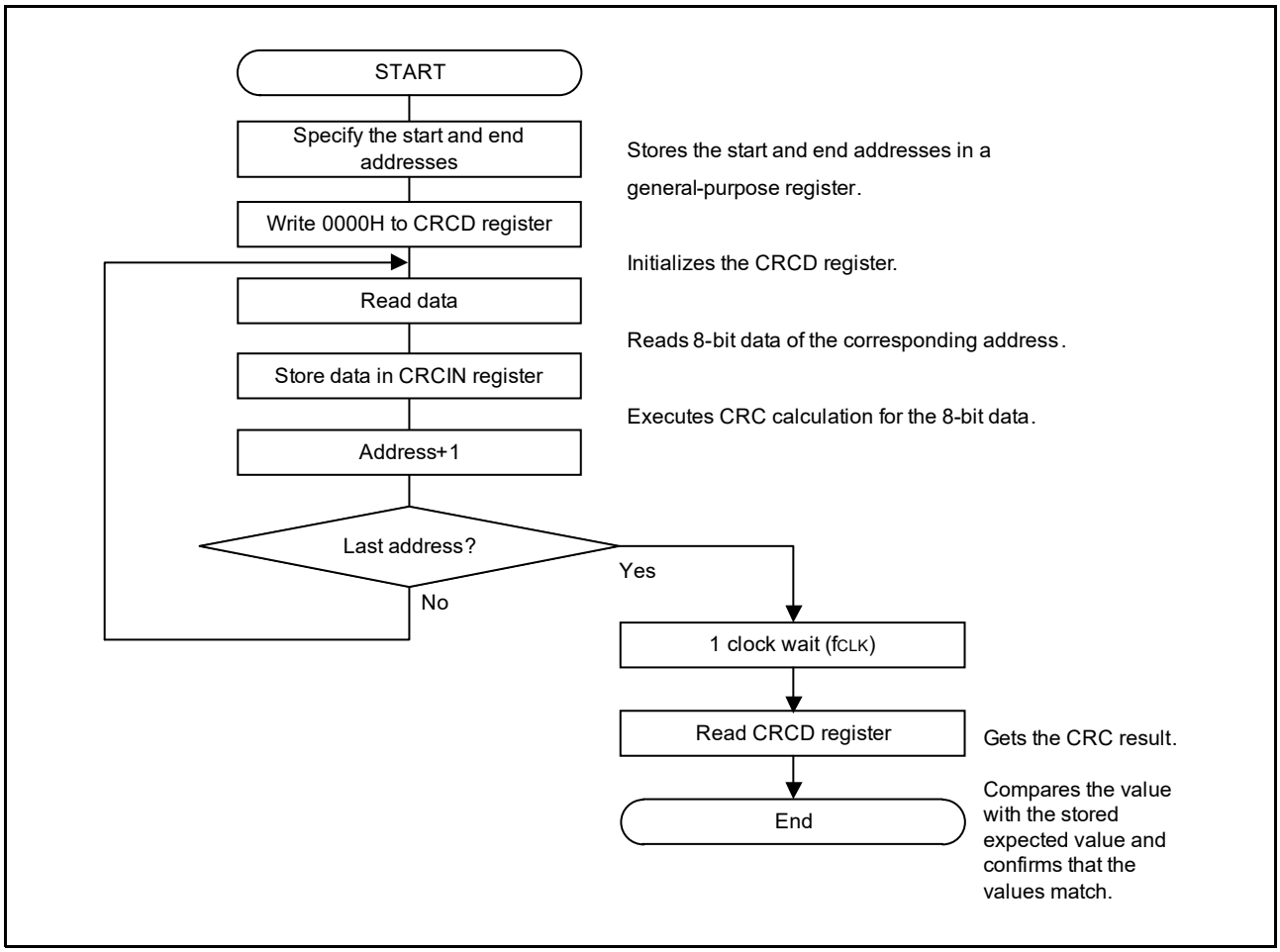
Symbol	15	14	13	12	11	10	9	8
CRCD								

7	6	5	4	3	2	1	0

Caution 1. Read the value written to the CRCD register before writing to the CRCIN register.
Caution 2. If conflict between writing a value and storing the operation result in the CRCD register occurs, the writing is ignored.

<Operation flow>

Figure 35 - 6 CRC Operation Function (General-purpose CRC)



35.3.3 Flash memory guard function

To ensure safe operation, the IEC60730 standard requires protecting flash memory from rewriting of its data due to incorrect CPU operations. The RL78/G24 has functionality to protect data in the code flash memory, data flash memory, and the security area in the flash memory. Enabling this function disables writing to the protected area of the flash memory. Reading from the protected area is possible.

35.3.3.1 Code flash memory guard register (GFLASH0)

This register is used to protect the code flash memory against being rewritten. To allow rewriting of the code flash memory, set the GFLASH0.GFL0 bit to 0. To protect the code flash memory, set the GFLASH0.GFL0 bit to 1. Be sure to confirm that the value has been set to 1 before reading from the code flash memory. The GFLASH0 register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 35 - 7 Format of Code Flash Memory Guard Register (GFLASH0)

Address: F0488H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
GFLASH0	RKEY[7:0]							

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	GFL0

RKEY[7:0]	Function
Bits 7 to 0	The RKEY[7:0] bits contain the key code to control rewriting of the GFLASH0 register. When rewriting the GFL0 bit, set RKEY[7:0] to 30H and then write to all 16 bits of this register at once. The RKEY[7:0] bits return 00H when read.

GFL0	Control of rewriting the code flash memory
0	Disables protection of the code flash memory (rewriting is allowed).
1	Enables protection of the code flash memory (rewriting is not allowed).

35.3.3.2 Data flash memory guard register (GFLASH1)

This register is used to protect the data flash memory against being rewritten. To allow rewriting of the data flash memory, set the GFLASH1.GFL1 bit to 0. To protect the data flash memory, set the GFLASH1.GFL1 bit to 1. Be sure to confirm that the value has been set to 1 before reading from the data flash memory. The GFLASH1 register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 35 - 8 Format of Data Flash Memory Guard Register (GFLASH1)

Address: F048AH
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
GFLASH1	RKEY[7:0]							

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	GFL1

RKEY[7:0]	Function
Bits 7 to 0	The RKEY[7:0] bits contain the key code to control rewriting of the GFLASH1 register. When rewriting the GFL1 bit, set RKEY[7:0] to C5H and then write to all 16 bits of this register at once. The RKEY[7:0] bits return 00H when read.

GFL1	Control of rewriting the data flash memory
0	Disables protection of the data flash memory (rewriting is allowed).
1	Enables protection of the data flash memory (rewriting is not allowed).

35.3.3.3 Flash security area guard register (GFLASH2)

This register is used to protect the flash memory's security area, which holds the security settings. To allow rewriting of the flash memory's security area, set the GFLASH2.GFL2 bit to 0. The GFLASH2 register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 35 - 9 Format of Flash Security Area Guard Register (GFLASH2)

Address: F048CH
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
GFLASH2	RKEY[7:0]							

7	6	5	4	3	2	1	0
							GFL2

RKEY[7:0]	Function
Bits 7 to 0	The RKEY[7:0] bits contain the key code to control rewriting of the GFLASH2 register. When rewriting the GFL2 bit, set RKEY[7:0] to 9AH and then write to all 16 bits of this register at once. The RKEY[7:0] bits return 00H when read.

GFL2	Control of rewriting the flash memory's security area
0	Disables protection of the security area in the flash memory (rewriting is allowed).
1	Enables protection of the security area in the flash memory (rewriting is not allowed).

35.3.4 RAM parity error detection

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/G24's RAM. By using this RAM parity error detection, 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.

35.3.4.1 RAM parity error control register (RPECTL)

This register is used to control the 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. The value of this register following a reset is 00H.

Figure 35 - 10 Format of RAM Parity Error Control Register (RPECTL)

Address: F00F5H
After reset: 00H
R/W: R/W

Symbol	<7>	6	5	4	3	2	1	<0>
RPECTL	RPERDIS	0	0	0	0	0	0	RPEF

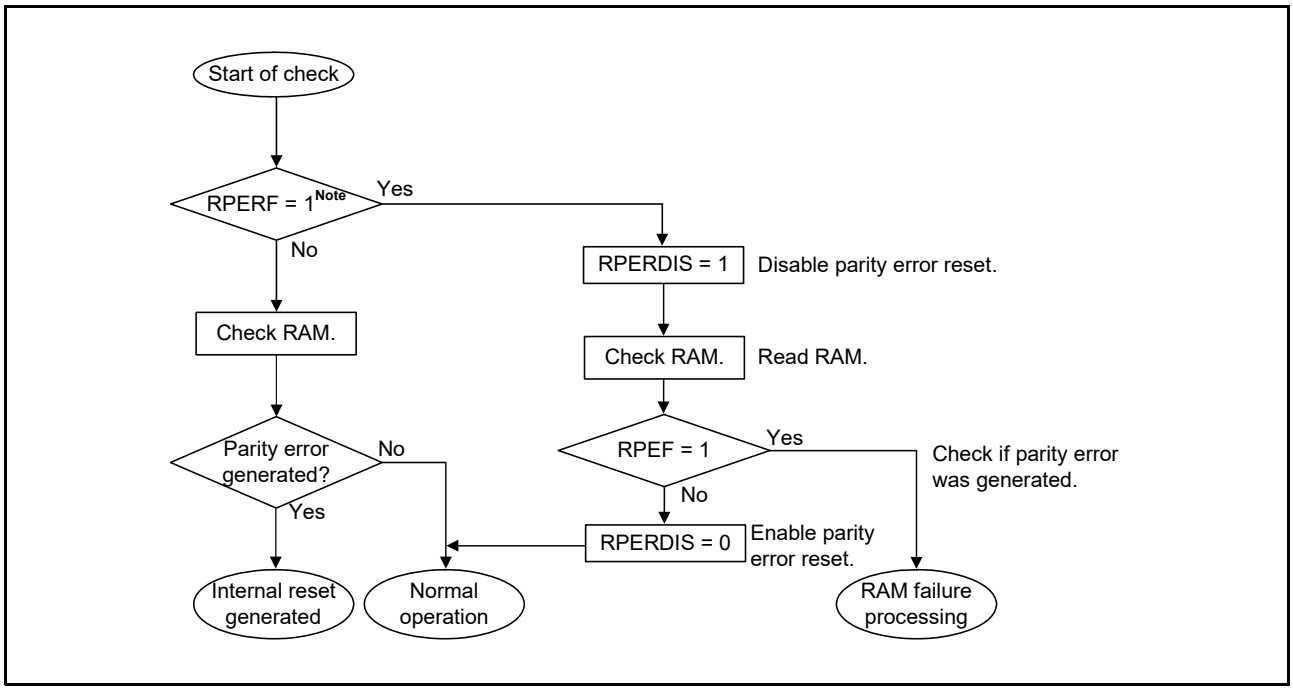
RPERDIS	Parity error reset mask flag
0	Enables parity error resets.
1	Disables parity error resets.

RPEF	Parity error status flag
0	No parity error has occurred.
1	A parity error has occurred.

Caution **The parity bit is appended when data are written, and the parity is checked when data are read. Therefore, if RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed before reading data.**
The RL78's CPU executes look-ahead 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, if RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the RAM area + 10 bytes when instructions are to be fetched from the RAM area.

Remark 1. The parity error reset is enabled by default (RPERDIS = 0).
Remark 2. Even if the parity error reset is disabled (RPERDIS = 1), the RPEF flag will be set to 1 if a parity error occurs. If parity error resets are enabled (RPERDIS = 0) with RPEF set to 1, a parity error reset is generated when the RPERDIS bit is cleared to 0.
Remark 3. The RPEF flag in the RPECTL register is set to 1 when a parity error occurs and cleared to 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.
Remark 4. The general registers are not included for RAM parity error detection.

Figure 35 - 11 Flowchart of RAM Parity Check



Note To check the state following an internal reset triggered by a RAM parity error, see **Section 32 Reset Function**.

35.3.4.2 RAM parity error control register 2 (RPECTL2)

This register is used to control the bits for confirming whether parity errors occur in access to the instruction code memory and data memory for the FAA and to control reset generation due to parity errors. The RPECTL2 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 35 - 12 Format of RAM Parity Error Control Register 2 (RPECTL2)

Address: F04B2H
After reset: 00H
R/W: R/W

Symbol	<7>	6	5	4	3	2	1	<0>
RPECTL2	FAARPEREN	0	0	0	0	0	0	FAARPEF

FAARPEREN	FAA parity error reset enable flag
0	Disables the FAA parity error reset.
1	Enables the FAA parity error reset.

FAARPEF	Parity error status flag
0	No parity error has occurred.
1	A parity error has occurred.

Caution The parity bit is appended when data are written, and the parity is checked when data are read. Therefore, if RAM parity error resets are enabled (FAARPEREN = 1), be sure to initialize RAM areas where data access is to proceed before reading data.
Execution by the FAA is look-ahead due to its pipelined operation, so it might read from an uninitialized RAM area in the instruction code memory or data memory that is allocated beyond the RAM that is actually in use, which will cause a RAM parity error. Therefore, if RAM parity error resets are enabled (FAARPEREN =1), be sure to initialize the RAM area + 10 bytes when instructions are to be fetched from the RAM area.

Remark 1. The parity error reset is disabled by default (FAARPEREN =0).

Remark 2. Even if the parity error reset is disabled (FAARPEREN =0), the FAARPEF flag will be set to 1 if a parity error occurs. If parity error resets are enabled (FAARPEREN = 1) with FAARPEF set to 1, a parity error reset is generated when the FAARPEREN bit is set to 1.

Remark 3. The FAARPEF flag in the RPECTL2 register is set to 1 when a parity error occurs and cleared to 0 by writing 0 to it or by any reset source. When FAARPEF = 1, the value is retained even if RAM for which no parity error has occurred is read.

Remark 4. When the setting of the RPECTL register is for disabling the parity error reset (RPERDIS = 1), parity errors do not occur regardless of the setting of FAARPEREN.

35.3.5 RAM guard function

To guarantee safe operation, the IEC61508 standard requires important data stored in the RAM to be protected even if a CPU malfunction occurs. The RL78/G24 has functionality to protect data in the specified memory space. Enabling this function disables writing to the specified area of the RAM. Reading from the specified area is possible.

35.3.5.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of illicit memory accesses, and protection of the RAM and SFRs. Use the GRAM1 and GRAM0 bits to protect the RAM. The IAWCTL register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 35 - 13 Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0078H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GRAM1	GRAM0	Protected area in the RAM[Note]
0	0	Disabled. Writing to the RAM is allowed.
0	1	128 bytes from the base address of the RAM
1	0	256 bytes from the base address of the RAM
1	1	512 bytes from the base address of the RAM

Note The base address of the RAM differs depending on the size of the RAM in the product.

Remark The protection of the RAM by this register is only effective against accesses from the CPU and data transfer controller (DTC).

35.3.6 SFR guard function

To guarantee safe operation, the IEC61508 standard requires important data stored in the SFRs to be protected even if a CPU malfunction occurs. The RL78/G24 provides functionality to protect the data in the control registers for use with the ports, interrupts, clock control, voltage detection, and RAM parity error detection. Enabling this function disables writing to the protected area of the SFRs. Reading from the protected area is possible.

35.3.6.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of illicit memory accesses and the protection of the RAM and SFRs. Use the GPORT, GINT, and GCSC bits to protect the SFRs. The IAWCTL register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 35 - 14 Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0078H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GPORT	Protection of the port control registers
0	Disabled. Reading from and writing to the port control registers are allowed.
1	Enabled. Writing to the port control registers is not allowed. Reading from them is allowed. [Protected SFRs] PMxx, PUxx, PIMxx, POMxx, PMCAxx, PDIDISxx, CCDE, CCSx**Note**

GINT	Protection of the interrupt control registers
0	Disabled. Reading from and writing to the interrupt control registers are allowed.
1	Enabled. Writing to the interrupt control registers is not allowed. Reading from them is allowed. [Protected SFRs] IFxx, MKxx, PRxx, EGPx, EGNx

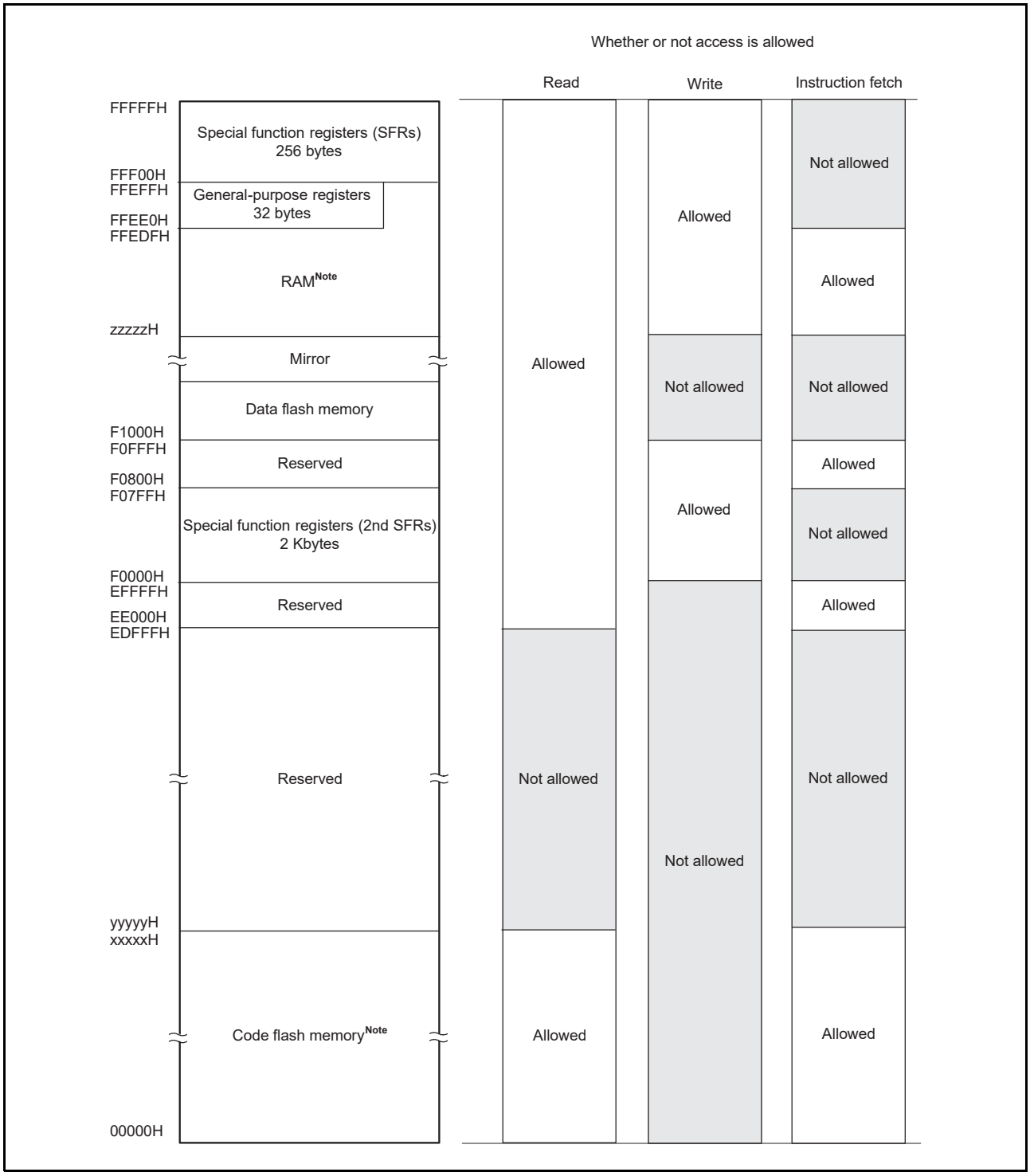
GCSC	Protection of the clock, voltage detector, and RAM parity error detection control registers
0	Disabled. Reading from and writing to the clock, voltage detector, and RAM parity error detection control registers are allowed.
1	Enabled. Writing to the clock, voltage detector, and RAM parity error detection control registers is not allowed. Reading from them is allowed. [Protected SFRs] CMC, CSC, OSTS, CKC, PERx, OSMC, LVIM, LVIS, RPECTL, CKSEL, PRRx, MOCODIV, WKUPMD, DSCCTL, MCKC, HSCLKSEL, PFBER

Note The port registers xx (Pxx) are not protected.

35.3.7 Illicit memory access detection

The IEC60730 standard mandates checking that the CPU and interrupts are operating correctly.
The RL78/G24 provides functionality to trigger a reset when an invalid memory area is accessed. Access to the areas indicated as “Not allowed” in **Figure 35 - 15** is detected as illicit.

Figure 35 - 15 Illicit Access Areas



Note The following table lists the capacity and address of the code flash memory and RAM, and the lowest address of the area to be detected as illicit when accessed of each product.

Products	Code Flash Memory (00000H to xxxxxH)	RAM (zzzzzH to FFEFFH)	Lowest Address of the Area to Be Detected as Illicit When Accessed for Reading or Instruction Fetching (yyyyyH)
R7F101GxE (x = 6, 7, 8, A, B, E, F, G, J, L)	65536 × 8 bits (00000H to 0FFFFH)	12288 × 8 bits (FCF00H to FFEFFH)	10000H
R7F101GxG (x = 6, 7, 8, A, B, E, F, G, J, L)	131072 × 8 bits (00000H to 1FFFFH)	12288 × 8 bits (FCF00H to FFEFFH)	20000H

35.3.7.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of illicit memory accesses and the protection of the RAM and SFRs. Use the IAWEN bit to protect the SFRs. The IAWCTL register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 35 - 16 Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0078H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

IAWEN[Note]	Control of illicit memory access detection
0	Disables the detection of illicit memory accesses.
1	Enables the detection of illicit memory accesses.

Note Only writing 1 to the IAWEN bit has an effect. Writing 0 to it has no effect after the IAWEN bit has been set to 1.

Remark When WDTON = 1 (watchdog timer operation enabled) is set in the option byte (000C0H), the illicit memory access detection is enabled even if IAWEN = 0.

35.3.8 Guard function of invalid memory access detection control register

To ensure safe operation, the IEC60730 standard requires the setting for enabling or disabling illicit memory access detection to be protected from being rewritten even if a CPU malfunction occurs. The RL78/G24 provides functionality to protect the invalid memory access detection control register (IAWCTL) against being rewritten. Enabling the protection of the invalid memory access detection control register disables writing to the given register. Reading from the protected register is possible.

35.3.8.1 Guard register of IAWCTL register (GIAWCTL)

This register is used to protect the setting for enabling or disabling the illicit memory access detection. To allow rewriting of the invalid memory access detection control register (IAWCTL), set the GIAWCTL.GIA bit to 0 to disable protection of the IAWCTL register. The GIAWCTL register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 35 - 17 Format of Guard Register of IAWCTL Register (GIAWCTL)

Address: F048EH
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
GIAWCTL	RKEY[7:0]							

7	6	5	4	3	2	1	0
							GIA

RKEY[7:0]	Function
Bits 7 to 0	The RKEY[7:0] bits contain the key code to control rewriting of the GIAWCTL register. When rewriting the GIA bit, set RKEY[7:0] to A4H and then write to all 16 bits of this register at once. The RKEY[7:0] bits return 00H when read.

GIA	Control of rewriting the IAWCTL register
0	Disables protection of the IAWCTL register (rewriting is allowed).
1	Enables protection of the IAWCTL register (rewriting is not allowed).

35.3.9 Frequency detection

The IEC60730 standard mandates checking that the oscillation frequency is correct.
By using the CPU/peripheral hardware clock frequency (f_{CLK}) to measure the pulse width of the input signal to channel 1 of 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.

<Clock frequencies to be compared>

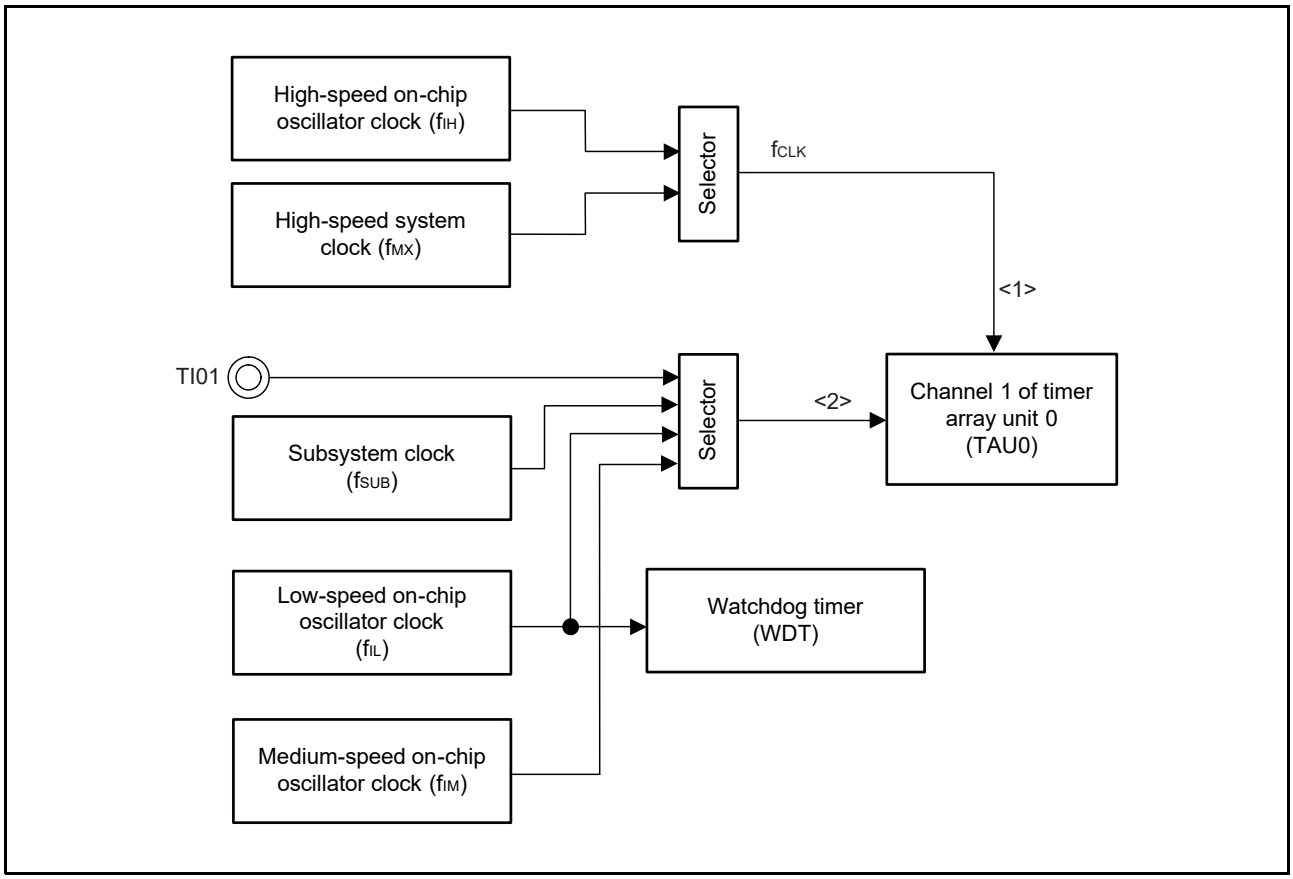
<1> CPU/peripheral hardware clock frequency (f_{CLK}):

- High-speed on-chip oscillator clock (f_{IH})
- High-speed system clock (f_{MX})

<2> Input to channel 1 of timer array unit 0

- Timer input to channel 1 (TI01)
- Low-speed on-chip oscillator clock (f_{IL})
- Subsystem clock (f_{SUB})
- Medium-speed on-chip oscillator clock (f_{IM})

Figure 35 - 18 Configuration of Frequency Detection Function



If the results of input pulse interval measurement are abnormal, the clock frequency is considered to be abnormal. For details on the input pulse interval measurement, see **10.8.4 Operation for input pulse interval measurement**.

35.3.9.1 Timer I/O select register 0 (TIOS0)

This register is used to select the timer input of channels 0 and 1 and the timer output of channel 2 of the timer array unit. The TIOS0 register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 35 - 19 Format of Timer I/O Select Register 0 (TIOS0)

Address: F0074H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TIOS0	TIS07	TIS06	TIS05	TIS04	TIS03	TIS02	TIS01	TIS00

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 the ELC
0	1	0	Input signal of timer input pin (TI01)
0	1	1	Medium-speed on-chip oscillator clock (f_{IM})
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
1	0	1	Subsystem clock (f_{SUB})
Other than the above			Setting prohibited

35.3.10 Testing of the A/D converter

The IEC60730 standard mandates testing of the A/D converter. This test checks whether or not the A/D converter is operating normally by converting the A/D converter's positive and negative reference voltages, analog input channels (ANIxx), temperature sensor output voltage, and internal reference voltage. For details on the method of checking, refer to the **application note (R01AN5607) Safety Function (A/D test)**.

Use the following procedure to check the analog multiplexer.

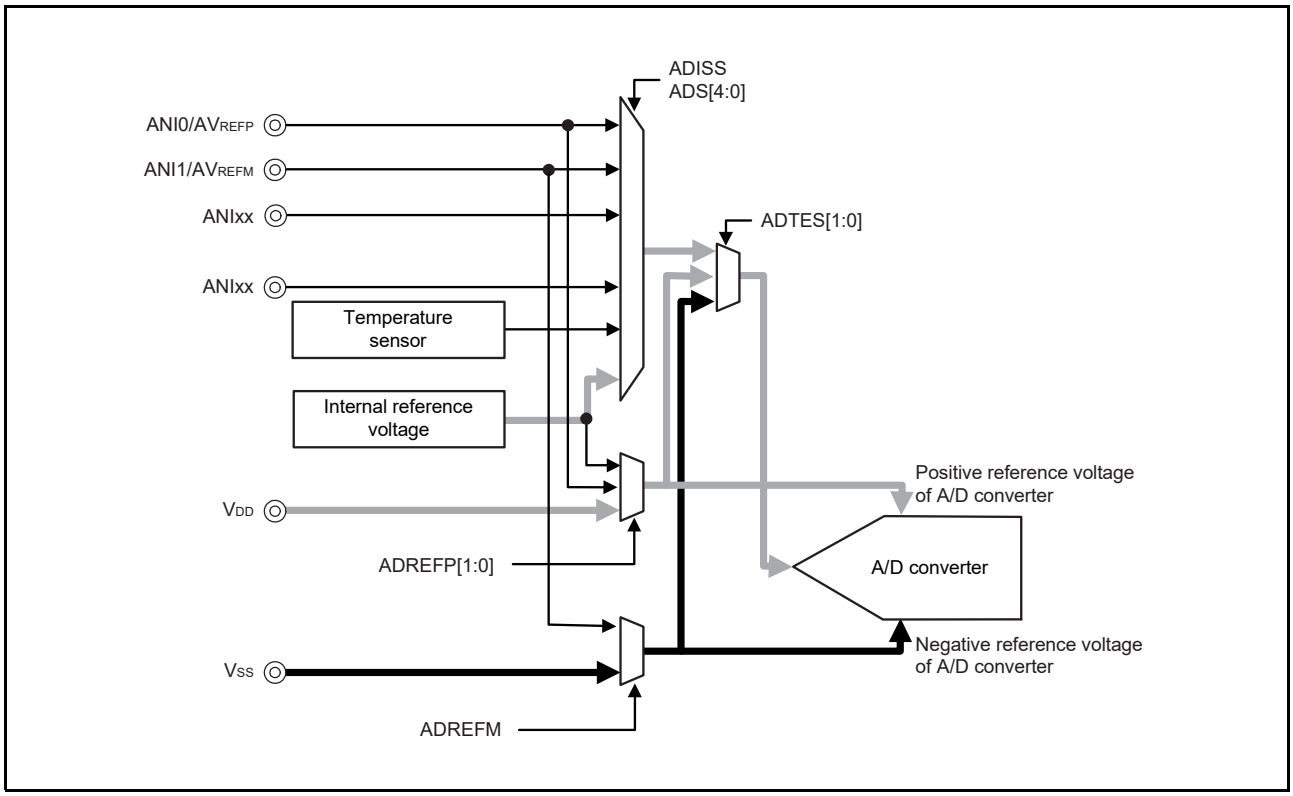
<1> Select the ANIxx pin for A/D conversion using the ADTES register (ADTES[1:0] = 00B).
<2> Perform A/D conversion for the ANIxx pin (conversion result 1-1).
<3> Select the A/D converter's negative reference voltage for A/D conversion using the ADTES register (ADTES[1:0] = 10B)
<4> Perform A/D conversion of the negative reference voltage of the A/D converter (conversion result 2-1).
<5> Select the ANIxx pin for A/D conversion using the ADTES register (ADTES[1:0] = 00B).
<6> Perform A/D conversion for the ANIxx pin (conversion result 1-2).
<7> Select the A/D converter's positive reference voltage for A/D conversion using the ADTES register (ADTES[1:0] = 11B)
<8> Perform A/D conversion of the positive reference voltage of the A/D converter (conversion result 2-2).
<9> Select the ANIxx pin for A/D conversion using the ADTES register (ADTES[1:0] = 00B).
<10> Perform A/D conversion for the ANIxx pin (conversion result 1-3).
<11> Check that the conversion results 1-1, 1-2, and 1-3 are equal.
<12> Check that the A/D conversion result 2-1 is all zero and the A/D conversion result 2-2 is all one.

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 A/D conversion in steps <1> to <10> above, use another method to check the analog multiplexer.

Remark 2. The results of conversion might include an error. Consider an appropriate level of error in comparison of the results of conversion.

Figure 35 - 20 Configuration of Testing of the A/D Converter



35.3.10.1 A/D test register (ADTES)

This register is used to select the A/D converter's positive reference voltage, A/D converter's negative reference voltage, analog input channels (ANIxx), temperature sensor output voltage, or internal reference voltage as the target of A/D conversion. When testing the A/D converter, specify the following settings:

- Select the negative reference voltage as the target of A/D conversion for zero-scale measurement.
- Select the positive reference voltage as the target of A/D conversion for full-scale measurement.

The ADTES register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 35 - 21 Format of A/D Test Register (ADTES)

Address: F0013H
After reset: 00H
R/W: 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/internal reference voltage (This is specified using the analog input channel specification register (ADS).)
1	0	Negative reference voltage (selected with the ADREFM bit in the ADM2 register)
1	1	Positive reference voltage (selected with the ADREFP[1:0] bits in the ADM2 register)
Other than the above		Setting prohibited

Caution Be sure to clear bits 7 to 2 to 0.

35.3.10.2 Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted. Set the A/D test register (ADTES) to 00H when measuring ANIxx, temperature sensor output voltage, or internal reference voltage. The ADS register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 35 - 22 Format of Analog Input Channel Specification Register (ADS)

Address: FFF31H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

<Select mode (ADM0.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**Note 4**
0	0	0	1	0	1	ANI5	P25/ANI5 pin**Note 4**
0	0	0	1	1	0	ANI6	P26/ANI6 pin**Note 4**
0	0	0	1	1	1	ANI7	P27/ANI7 pin**Note 3**
0	1	0	0	0	0	ANI16	P03/ANI16 pin**Note 5**
0	1	0	0	0	1	ANI17	P02/ANI17 pin**Note 5**
0	1	0	0	1	0	ANI18	P147/ANI18 pin**Note 1**
0	1	0	0	1	1	ANI19	P120/ANI19 pin
0	1	0	1	0	0	ANI20	P10/ANI20 pin
0	1	0	1	0	1	ANI21	P11/ANI21 pin
0	1	0	1	1	0	ANI22	P12/ANI22 pin
0	1	0	1	1	1	ANI23	P13/ANI23 pin
0	1	1	0	0	0	ANI24	P14/ANI24 pin**Note 1**
0	1	1	0	0	1	ANI25	P15/ANI25 pin**Note 2**
0	1	1	0	1	0	ANI26	P16/ANI26 pin**Note 6**
0	1	1	0	1	1	ANI27	P17/ANI27 pin**Note 2**
0	1	1	1	0	0	ANI28	P146/ANI28 pin**Note 3**
0	1	1	1	0	1	ANI29	P00/ANI29 pin
0	1	1	1	1	0	ANI30	P01/ANI30 pin
1	0	0	0	0	0	—	Temperature sensor output voltage
1	0	0	0	0	1	—	Internal reference voltage
Other than the above						Setting prohibited	

Note 1. This is not present in the 24-pin products.
Note 2. This is not present in the 20- and 24-pin products.
Note 3. This is not present in the 20-, 24-, 25-, 30-, 32-, and 40-pin products.
Note 4. This is not present in the 20-, 24-, 25-, 30-, and 32-pin products.
Note 5. This is not present in the 20-, 24-, 25-, 30-, 32-, 40-, 44-, and 48-pin products.
Note 6. This is not present in the 20-pin products.

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

(Cautions are listed on the next page.)

Caution 2. Select input mode for the ports which are set to analog input with the PMCA registers, using the port mode registers 0 to 2, 12, and 14 (PM0 to PM2, PM12, and PM14).

Caution 3. Do not use the ADS register to set the pins which should be set as digital I/O with the port mode control A registers 0 to 2, 12, and 14 (PMCA0 to PMCA2, PMCA12, and PMCA14).

Caution 4. Only rewrite the value of the ADISS bit while the conversion operation is stopped (ADCS = 0, ADCE = 0).

Caution 5. If using AV_{REFP} as the positive reference voltage source of the A/D converter, do not select ANI0 as an A/D conversion channel.

Caution 6. If using AV_{REFM} as the negative reference voltage source of the A/D converter, do not select ANI1 as an A/D conversion channel.

Caution 7. When ADISS is 1, the internal reference voltage cannot be used for the positive reference voltage. In addition, the first conversion result obtained after setting ADISS to 1 is not available. For a detailed setting flow, see 20.7.6 Settings when temperature sensor output voltage or internal reference voltage is selected (example for software trigger no-wait mode and one-shot conversion mode).

Caution 8. If a transition is made to STOP mode or a transition is made to HALT mode during CPU operation with subsystem clock, do not set ADISS to 1. When ADISS is 1, the A/D converter reference voltage current (I_{ADREF}) shown in 43.3.2 Supply current characteristics or 44.3.2 Supply current characteristics is added.

Caution 9. When the ADISS bit is set to 1, the hardware trigger wait mode and one-shot conversion mode are not available.

Caution 10. Run the self-check of the A/D converter while the advanced mode is disabled (ADM3.ADVMOD = 0).

35.3.11 Detection of the digital output signal level of the I/O pins

The IEC60730 standard mandates checking that the I/O function is operating correctly.
The RL78/G24 provides functionality to read the output level of an I/O pin when the pin is in the output mode.

35.3.11.1 Port mode select register (PMS)

This register is used to select whether to read the output latch value of a port or read the output level of a pin when the pin is in the output mode (PMmn bit of port mode register m (PMm) is 0). The PMS register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 35 - 23 Format of Port Mode Select Register (PMS)

Address: F007BH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PMS	0	0	0	0	0	0	0	PMS0

PMS0	Selection of data to be read when pin is in output mode
0	Reads the value of the Pmn register.
1	Reads the digital output level of the pin.

Remark m = 0 to 7, 12, 14
n = 0 to 7

Caution 1. Do not rewrite port registers xx (Pxx) with 1-bit memory manipulation instructions when the setting of the PMS0 bit in the PMS register is 1. Use 8-bit memory manipulation instructions to rewrite port registers xx (Pxx).

Caution 2. The setting of this register has no effect on the input port pins (P123, P124, and P137) and output port pin (P130).

Caution 3. Be sure to clear bits 7 to 1 to 0.

Caution 4. The setting of this register has no effect when the setting of the PMCAmn bit is 1. To read the digital output level of a pin with the PMCAmn bit set to 1, first clear the PMCAmn bit to 0 and then set the PMS0 bit to 1.

Remark m = 0 to 3, 5 to 7, 10 to 12, 14, 15
n = 0 to 7

Caution 5. The setting of this register has no effect when the setting of the PDIDISmn bit is 1. To read the digital output level of a pin with the PDIDISmn bit set to 1, first clear the PDIDISmn bit to 0 and then set the PMS0 bit to 1.

Remark m = 0, 1, 3, 5, 7
n = 0 to 7

35.3.12 UART loopback

The IEC60730 standard recommends to diagnose abnormalities of external interfaces (communications). The UART loopback is used to confirm the normal output of the UART transmit data by shutting off the RxDn pin from the outside and connecting it within the MCU to loop back the output from the transmission shift register to the reception shift register.

When the UART loopback is selected, the transmit data from the TxDn pin is controlled by the port functions so that it does not affect its communication partner.

- Communication with negative logic
  The port that also uses the TxDn pin is set to the input mode (PMxx = 1) and an on-chip pull-up resistor is connected (PUxx = 1) to retain the setting of 1.
- Communication with positive logic
  0 (PMxx = 0, Pxx = 0) is output from the port that also uses the TxDn pin.

Remark n = 0 to 2

35.3.12.1 UART loopback select register (ULBS)

This register is used to enable the UART loopback. This register has respective bits for independently controlling each UART channel. Setting the bit corresponding to each channel to 1 will select the UART loopback and loop back the output of the transmission shift register to the reception shift register. The ULBS register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 35 - 24 Format of UART Loopback Select Register (ULBS)

Address: F0079H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
ULBS	0	0	0	0	0	ULBS2	ULBS1	ULBS0

ULBS2	UART2 loopback selection
0	Inputs the state of the RxD2 pin of serial array unit UART2 to the reception shift register.
1	Loops back the output of the transmission shift register to the reception shift register.

ULBS1	UART1 loopback selection
0	Inputs the state of the RxD1 pin of serial array unit UART1 to the reception shift register.
1	Loops back the output of the transmission shift register to the reception shift register.

ULBS0	UART0 loopback selection
0	Inputs the state of the RxD0 pin of serial array unit UART0 to the reception shift register.
1	Loops back the output of the transmission shift register to the reception shift register.

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

Section 36 Security Functions

36.1 True Random Number Generator

36.1.1 Function of the true random number generator

The true random number generator generates 32-bit random number seeds (which are true random numbers).

36.1.2 Registers to control the true random number generator

The following registers are used to control the true random number generator.

- Random number seed command register 0 (TRNGSCR0)
- Random number seed data register (TRNGSDR)

36.1.2.1 Random number seed command register 0 (TRNGSCR0)

The TRNGSCR0 register controls operation of the true random number generator. Setting the TRNGST bit to 1 after having set the TRNGEN bit to 1 starts the generation of a random number seed. When the true random number generator finishes generating the random number seed, the TRNGRDY flag is set to 1. Since the TRNGST bit serves as the trigger for starting the generation of a random number seed, it is cleared to 0 immediately after 1 having been written to it. The TRNGSCR0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 36 - 1 Format of Random Number Seed Command Register 0 (TRNGSCR0)

Address: F0542H
After reset: 00H
R/W: R/W

Symbol	<7>	6	5	4	<3>	<2>	1	0
TRNGSCR0	TRNGRDY	0	0	0	TRNGEN	TRNGST	0	0

TRNGRDY	Random number seed generation status flag
0	A random number seed has not been generated or four rounds of reading from the TRNGSDR register have been completed.
1	A random number seed has been generated.

TRNGEN	Control over operation of the true random number generator
0	Stops the true random number generator.
1	Enables the true random number generator.

TRNGST	Trigger to start generating a random number seed
0	The trigger is inactive.
1	Starts generation of a random number seed.

36.1.2.2 Random number seed data register (TRNGSDR)

The TRNGSDR register is an 8-bit register that holds the bytes of random number seeds generated by the true random number generator. The random number seed can be read from this register after the TRNGRDY flag is set to 1. As a random number seed consists of 32 bits, four rounds of access to the register are required for each seed. Bit 7 (TRNGRDY) of the TRNGSCR0 register is cleared to 0 following the four rounds of access. The TRNGSDR register can be read by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 36 - 2 Format of Random Number Seed Data Register (TRNGSDR)

Address: F0540H
After reset: 00H
R/W: R

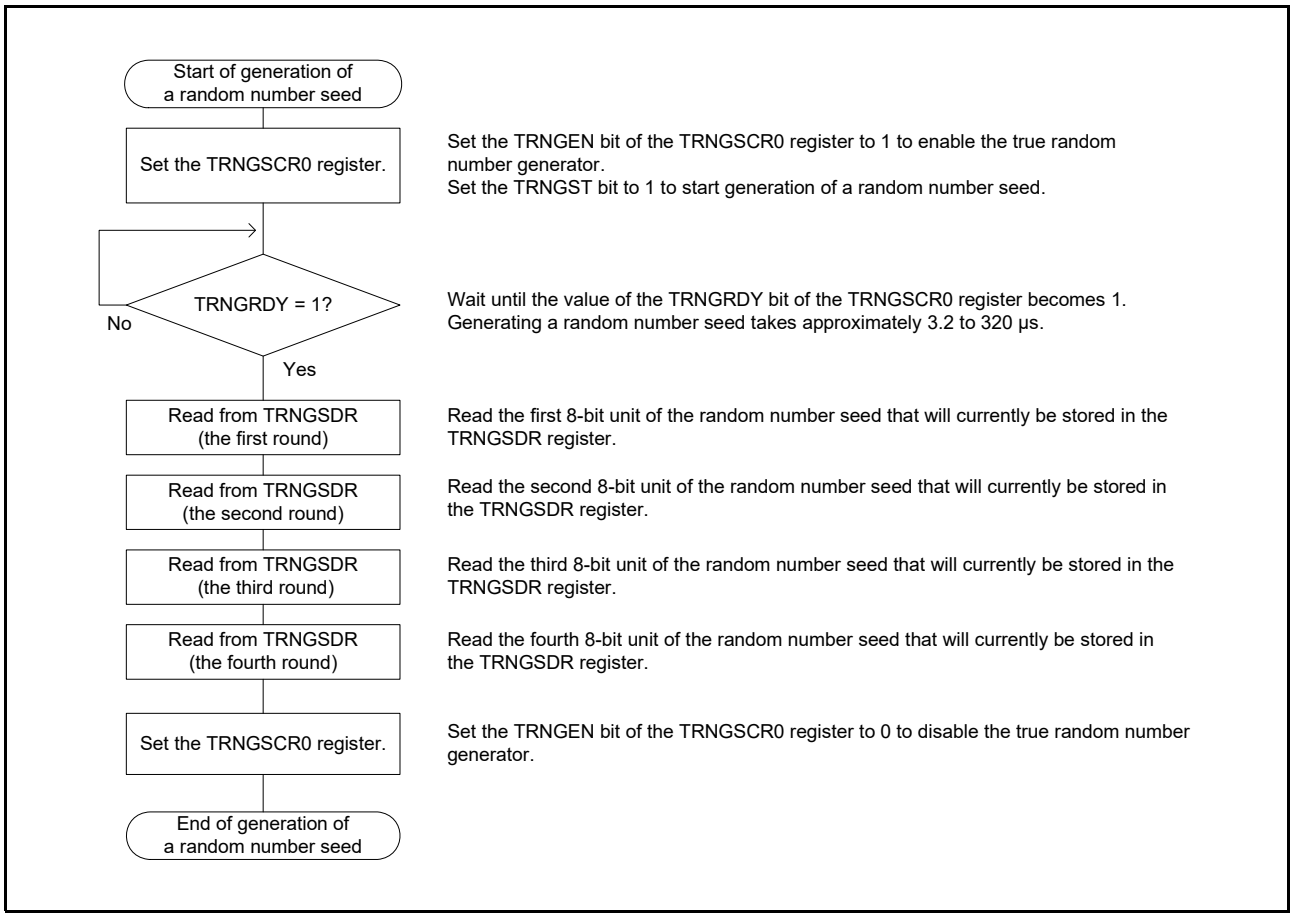
Symbol	7	6	5	4	3	2	1	0
TRNGSDR								

Caution When the value of the TRNGRDY bit is 0, the value in the TRNGSDR register is 00H.

36.1.3 Operations of the true random number generator

Figure 36 - 3 shows the procedure for using the true random number generator to generate a random number seed.

Figure 36 - 3 Procedure for Using the True Random Number Generator to Generate a Random Number Seed



36.2 Flash Read Protection

36.2.1 Function of flash read protection

The flash read protection function can be used to protect a specified range of the code flash memory area against read access by the CPU or DTC. Fetching of instructions in the specified range by the CPU is still possible. Note that verifying the code flash memory area protected against read access by the flash read protection is possible in the serial programming mode. This function can only be used when f_{CLK} is no greater than 32 MHz and the prefetch buffer is disabled.

36.2.2 Setting of flash read protection

The settings for flash read protection are made through serial programming by using a flash memory programmer or through self-programming in the extra area. Read access to the whole range of the code flash memory is enabled by the default setting at the time of shipment. Set a range in the code flash memory to be protected against read access by specifying the flash read protection start block and end block, and disable changing of the flash read protection settings. This makes read access to addresses in the specified range of blocks in the code flash memory impossible. In addition, changing of the flash read protection settings can be disabled, which makes the settings for the start and end blocks fixed, thus making changes to the blocks where flash read protection starts and ends impossible. However, disabling changing of the flash read protection settings can be released by using a flash memory programmer. Note that release from disabling of changes to the flash read protection settings requires deletion of the contents of the full range of the code flash memory. Accordingly, reading the data written in the range protected against read access is impossible after release from disabling of changes to the flash read protection settings. **Table 36 - 1** describes the settings for flash read protection and their functions. **Table 36 - 2** describes the method of setting flash read protection.

Table 36 - 1 Settings for Flash Read Protection and Their Functions

Item to Be Set	Function
Block where flash read protection starts	Specifies the number of the block where the read-access disabled area starts. The specifiable values are in the range from the block numbered 001H to the number of the block at the highest-order address in the code flash memory. The block for which the number is set as the block where protection starts is part of the read-access disabled area. Setting 000H as the block where protection starts is prohibited. The initial setting is 1FFH.
Block where flash read protection ends	Specifies the number of the block where the read-access disabled area ends. The specifiable values are in the range from the number of the block where the protection starts to the number of the block at the highest-order address in the code flash memory. The block for which the number is set as the block where protection ends is part of the read-access disabled area. The initial setting is 1FFH.
Fixing the flash read protection settings	Fixes the settings of the blocks where flash read protection starts and ends. When the “enabled” setting is made, changes to the blocks where protection starts and ends are not possible.

For details on the relationship between the addresses and block numbers, refer to **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**.

Table 36 - 2 Method of Setting Flash Read Protection

Item to Be Set	Method of Setting	Method of Changing
Block where flash read protection starts	Using a flash memory programmer or self-programming.	Using a flash memory programmer or self-programming. Note that the block where protection starts is not adjustable while fixing of the flash read protection settings is enabled.
Block where flash read protection ends	Using a flash memory programmer or self-programming.	Using a flash memory programmer or self-programming. Note that the block where protection ends is not adjustable while fixing of the flash read protection settings is enabled.
Fixing the flash read protection settings	Using a flash memory programmer or self-programming.	Fixing of the flash read protection settings can be released by using a flash memory programmer.**Note** If you do so, the values for the start and end blocks are initialized.

Note Release from the fixed setting is only possible when erasure of blocks is not prohibited, rewriting of boot cluster 0 is not prohibited, and the code and data flash memory areas are blank.

Caution 1. The settings for flash read protection in the extra area are not readable. To confirm that the settings for flash read protection are in place, read from the read-access disabled area and confirm that FFH is returned.

Caution 2. To specify the read-access disabled area for flash read protection, be sure to specify the numbers of both the block where protection starts and the block where it ends.

Caution 3. Reading from the read-access disabled area by using an on-chip debugger is also impossible. This means that program code allocated to the read access-disabled area cannot be debugged by using the on-chip debugger. Therefore, only make the settings for flash read protection after having debugged the program code in the protected areas.

Caution 4. When a part of boot cluster 0 or boot cluster 1 is to be set as a part of the read-access disabled area, boot swapping may cause data in the read-access disabled area to be swapped with data in the read access-enabled area. To prevent this, when setting a part of boot cluster 0 or boot cluster 1 as part of the read-access disabled area, make the setting for prohibiting the rewriting of boot cluster 0 so as to prohibit boot swapping itself.

Caution 5. When the prefetch buffer is enabled, flash read protection is not guaranteed.

Caution 6. When settings for flash read protection have been made through self-programming, the settings become enabled after the MCU is reset and then released from the reset state.

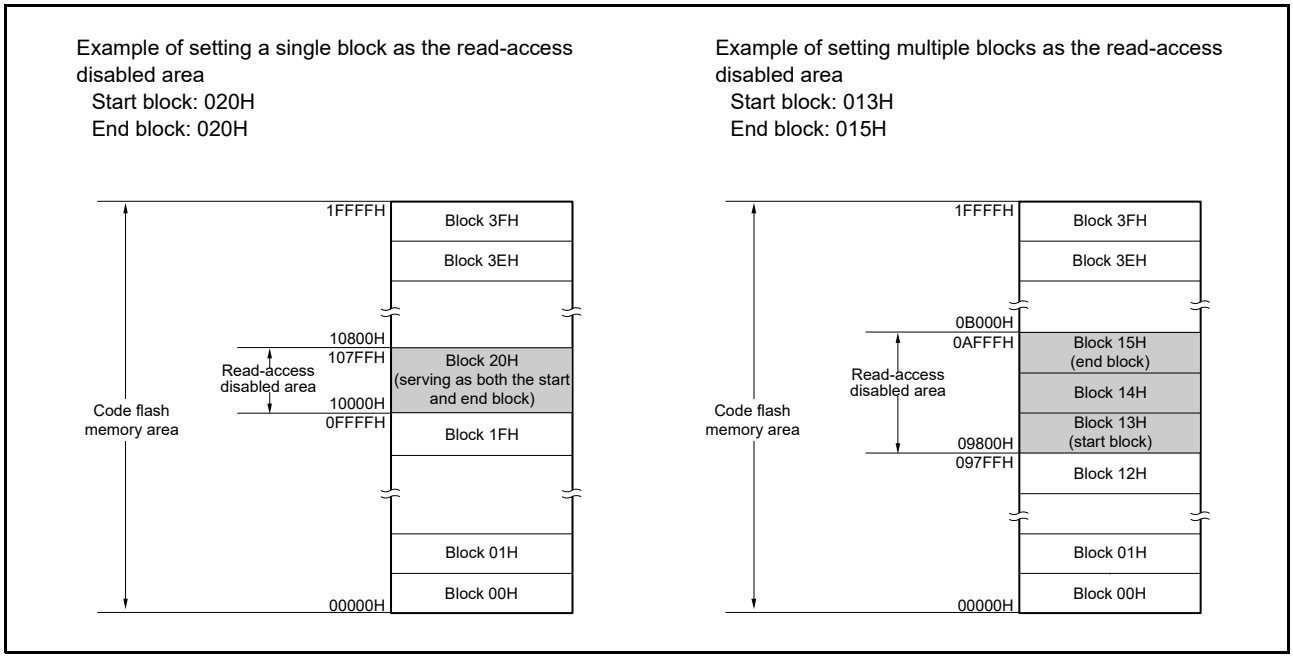
36.2.3 Operation

Reading by the CPU or DTC from the area to which read access has been disabled with the use of flash read protection always returns FFH.

Fetching of instructions in the read-access disabled area by the CPU is still possible. Note that even if program code is to be executed from the read-access disabled area, it is unable to read data that have been placed in the read-access disabled area. Place data for use with code to be executed from the read-access disabled area in areas that are not protected.

Figure 36 - 4 shows examples of setting read-access disabled areas for the application of flash read protection.

Figure 36 - 4 Examples of Setting Read-access Disabled Areas for the Application of Flash Read Protection



To enable read access to the full range of the code flash memory again after having set a range to be protected against read access, set the flash read protection start block and end block as 1FFH while prohibition of changing the flash read protection settings is disabled.

36.3 Unique ID

36.3.1 Function of a unique ID

A unique ID is a unique value that is allocated to an individual product and stored in the extra area.
A unique ID is entered for each product at the time of manufacturing the MCUs. Changing the entered ID is not possible.
The data length is 16 bytes (128 bits).

Caution The value of a unique ID is not a random number.

36.3.2 ASCII codes representing the product name

Product names are also stored as strings of ASCII codes in the extra area.
Since the product names include indicators of the number of pins and capacity of the flash memory for the given product, conditional branch processing on the basis of the product name is possible.

Table 36 - 3 shows the places of the unique ID, basic product name, and indicators of the number of pins and capacity of the flash memory in the memory map.

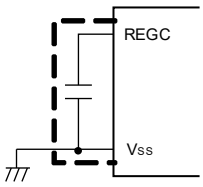
Table 36 - 3 Memory Map of the Unique ID and Product Name

Address	Item Name	Value to Be Entered
EFFC0H to EFFCFH	Unique ID	The unique value allocated to an individual product
EFFD5H	ASCII codes representing the product name	52H: “R”
EFFD6H		37H: “7”
EFFD7H		46H: “F”
EFFD8H		31H: “1”
EFFD9H		30H: “0”
EFFDAH		31H: “1”
EFFDBH		47H: “G”
EFFDCH		36H: “7” (20 pins) 37H: “8” (24 pins) 38H: “9” (25 pins) 41H: “A” (30 pins) 42H: “B” (32 pins) 45H: “E” (40 pins) 46H: “F” (44 pins) 47H: “G” (48 pins) 4AH: “J” (52 pins) 4CH: “L” (64 pins)
EFFDDH		45H: “E” (64-Kbyte flash memory) 47H: “G” (128-Kbyte flash memory)

Section 37 Regulator

37.1 Overview

The RL78/G24 incorporates a circuit for constant voltage operation. To stabilize the output voltage from the regulator, connect the REGC pin to Vss via a capacitor (0.47 to 1 μF). Use a capacitor with good characteristics, since it is for stabilizing the internal voltage.
The REGC pin can be used to provide a reference voltage for an external circuit. The input impedance of the external circuit to be connected to the REGC pin should be no less than 1.5 MΩ. The voltage on the REGC pin is specified as 1.5 V (typ.) and the voltage range is from 1.38 to 1.60 V.



Caution The length of the wiring within the broken lines in the above figure should be as short as possible.

The regulator outputs a 1.5-V voltage.

Section 38 Option Bytes

38.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the RL78/G24 form an option byte area.
Option bytes consist of user option bytes (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.
To use the boot swap operation during self-programming, 000C0H to 000C3H are replaced by 040C0H to 040C3H. Therefore, set the same values as 000C0H to 000C3H to 040C0H to 040C3H.

Caution The option bytes should always be set regardless of whether each function is used.

38.1.1 User option bytes (000C0H to 000C2H or 040C0H to 040C2H)

1. 000C0H or 040C0H
   - Setting of watchdog timer operation
     - Enabling or disabling of counter operation
     - Enabling or stopping of counter operation 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
     - Interval interrupt is used or not used

   Caution When boot swapping is to be used, 000C0H is replaced with 040C0H. Therefore, set the same value as the setting in 000C0H in 040C0H.

2. 000C1H or 040C1H
   - Setting of LVD0 operation mode
     - Reset mode
     - Interrupt mode
     - LVD0 off (by controlling the externally input reset signal on the $\overline{\text{RESET}}$ pin)
   - Setting of LVD0 detection level (V_{LVD0})

   Caution 1. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 43.4 AC Characteristics or 44.4 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 or 040C2H).
   Caution 2. When boot swapping is to be used, 000C1H is replaced with 040C1H. Therefore, set the same value as the setting in 000C1H in 040C1H.

3. 000C2H or 040C2H
   - 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
     – LP (low-power main) mode
   - Setting of the frequency of the high-speed on-chip oscillator
     – Select from 1 MHz to 32 MHz, 48 MHz, and 64 MHz.

Caution When boot swapping is to be used, 000C2H is replaced with 040C2H. Therefore, set the same value as the setting in 000C2H in 040C2H.

38.1.2 On-chip debug option byte (000C3H or 040C3H)

- 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.

Caution When boot swapping is to be used, 000C3H is replaced with 040C3H. Therefore, set the same value as the setting in 000C3H in 040C3H.

38.2 Format of User Option Bytes

Figure 38 - 1 Format of User Option Byte (000C0H or 040C0H)

Address: 000C0H or 040C0H[Note 1]

Symbol	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% of the overflow time + 1/4 f_{IL} is reached.

WINDOW1	WINDOW0	Watchdog timer window open period[Note 2]
0	0	Setting prohibited
0	1	50%
1	0	Setting prohibited
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} = 37.683 kHz (max.))
0	0	0	$2^7/f_{IL}$ (3.39 ms)
0	0	1	$2^8/f_{IL}$ (6.79 ms)
0	1	0	$2^9/f_{IL}$ (13.58 ms)
0	1	1	$2^{10}/f_{IL}$ (27.17 ms)
1	0	0	$2^{12}/f_{IL}$ (108.69 ms)
1	0	1	$2^{14}/f_{IL}$ (434.78 ms)
1	1	0	$2^{15}/f_{IL}$ (869.56 ms)
1	1	1	$2^{17}/f_{IL}$ (3478.26 ms)

WDSTBYON	Operation control of watchdog timer counter (HALT/STOP mode)
0	Counter operation stopped in HALT/STOP mode[Note 2]
1	Counter operation enabled in HALT/STOP mode

Note 1. When boot swapping is to be used, 000C0H is replaced with 040C0H. Therefore, set the same value as the setting in 000C0H in 040C0H.

Note 2. The window open period is 100% when WDSTBYON = 0, regardless of the value of the WINDOW1 and WINDOW0 bits.

Figure 38 - 2 Format of User Option Byte (000C1H or 040C1H)

Address: 000C1H or 040C1H[Note]

Symbol	7	6	5	4	3	2	1	0
	LVD0EN	LVD0SEL	1	1	1	LVD0V2	LVD0V1	LVD0V0

LVD0 setting (reset mode)

Detection voltage		Option byte setting value				
V_{LVD0}		LVD0EN	Mode setting	LVD0V2	LVD0V1	LVD0V0
Rising edge	Falling edge		LVD0SEL			
1.69 V	1.65 V	1	1	1	1	1
1.90 V	1.86 V			1	1	0
2.38 V	2.33 V			1	0	1
2.67 V	2.62 V			1	0	0
2.97 V	2.91 V			0	1	1
3.96 V	3.88 V			0	1	0
—		Settings other than the above are prohibited.				

LVD0 setting (interrupt mode)

Detection voltage		Option byte setting value				
V_{LVD0}		LVD0EN	Mode setting	LVD0V2	LVD0V1	LVD0V0
Rising edge	Falling edge		LVD0SEL			
1.69 V	1.65 V	1	0	1	1	1
1.90 V	1.86 V			1	1	0
2.38 V	2.33 V			1	0	1
2.67 V	2.62 V			1	0	0
2.97 V	2.91 V			0	1	1
3.96 V	3.88 V			0	1	0
—		Settings other than the above are prohibited.				

LVD0 off setting (external reset input from the $\overline{\text{RESET}}$ pin is used)

Detection voltage		Option byte setting value				
V_{LVD0}		LVD0EN	Mode setting	LVD0V2	LVD0V1	LVD0V0
Rising edge	Falling edge		LVD0SEL			
—	—	0	×	0	1	0
—		Settings other than the above are prohibited.				

Note When boot swapping is to be used, 000C1H is replaced with 040C1H. Therefore, set the same value as the setting in 000C1H in 040C1H.

Caution 1. Be sure to set bits 5 to 3 to 1.

(Caution and Remarks are listed on the next page.)

Caution 2. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 43.4 AC Characteristics or 44.4 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 or 040C2H).

Remark 1. ×: Don't care.

Remark 2. For details on the LVD0 circuit, see **Section 34 Voltage Detector (LVD)**.

Remark 3. The detection voltage is a typical value. For details, see **43.6.7 LVD circuit characteristics** or **44.6.7 LVD circuit characteristics**.

Figure 38 - 3 Format of User Option Byte (000C2H or 040C2H)

Address: 000C2H or 040C2H[Note 1]

Symbol	7	6	5	4	3	2	1	0
	CMODE1	CMODE0	1	FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

CMODE1	CMODE0	Flash operation mode	Operating frequency range	Operating voltage range
0	1	LP (low-power main) mode	1 to 2 MHz (Rewriting of flash memory is not possible.)	1.6 to 5.5 V
1	0	LS (low-speed main) mode	1 to 4 MHz (Rewriting of flash memory is not possible.)	1.6 to 5.5 V
			1 to 24 MHz	1.8 to 5.5 V
1	1	HS (high-speed main) mode (prefetching is off)	1 to 4 MHz (Rewriting of flash memory is not possible.)	1.6 to 5.5 V
			1 to 32 MHz	1.8 to 5.5 V
		HS (high-speed main) mode (prefetching is on)	48 MHz	2.4 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}[Note 2]
1	1	0	0	0	64 MHz	32 MHz
1	0	0	0	0	48 MHz	24 MHz
0	1	0	0	0	32 MHz	32 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	0	0	1	0	6 MHz	6 MHz
0	1	0	1	1	4 MHz	4 MHz
0	0	0	1	1	3 MHz	3 MHz
0	1	1	0	0	2 MHz	2 MHz
0	1	1	0	1	1 MHz	1 MHz
Other than above					Setting prohibited	

Note 1. When boot swapping is to be used, 000C2H is replaced with 040C2H. Therefore, set the same value as the setting in 000C2H in 040C2H.

Note 2. This option byte cannot be used to select 48 MHz as the frequency of the high-speed on-chip oscillator (f_{IH}). Regarding the procedure for selecting 48 MHz, see **Section 9 Clock Generator**.

Caution 1. Be sure to set bit 5 to 1.

Caution 2. The operating frequency range and operating voltage range vary depending on the flash operation mode. For details, see 43.4 AC Characteristics or 44.4 AC Characteristics.

38.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 38 - 4 Format of On-chip Debug Option Byte (000C3H or 040C3H)

Address: 000C3H or 040C3H[Note]

Symbol	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 debugging.
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 erase data of flash memory in case of failures in authenticating on-chip debug security ID.

Note When boot swapping is to be used, 000C3H is replaced with 040C3H. Therefore, set the same value as the setting in 000C3H in 040C3H.

Caution **Only bits 7 and 0 (OCDENSET and OCDERSD) are specifiable.**
Be sure to set bits 6 to 1 to 000010B.

Remark The use of on-chip debugging changes the values of bits 3 to 1. Accordingly, the values after such settings are made become undefined. However, note that when setting the option byte, be sure to set bits 3 to 1 to their default values 010B.

38.4 Setting of Option Bytes

The user option bytes and on-chip debug option byte can also be set by using linker options instead of statements in the source code. In such cases, the settings made by using linker options are given priority over the statements in the source code as shown below.
An example of the statements in relation to the option byte settings in software is shown below.

```
        .CSEG    OPT_BYTE
        .DB      0x36     ; Does not use the interval interrupt of the watchdog timer.
                          ; Enables watchdog timer operation.
                          ; The window open period of the watchdog timer is 50%.
                          ; The overflow time of the watchdog timer is 2^10/fIL.
                          ; Stops watchdog timer operation during HALT/STOP mode.
        .DB      0xBF     ; Selects VLVD0 as 1.69 V on rising edges and 1.65 V on falling edges.
                          ; Select the interrupt mode as the LVD0 operation mode.
        .DB      0x6D     ; Select the LP (low-power main) mode as the flash operation mode and 1 MHz
                          ; as the frequency of the high-speed on-chip oscillator clock.
        .DB      0x85     ; Enables on-chip debug operation, does not erase flash memory data when
                          ; security ID authentication fails.
```

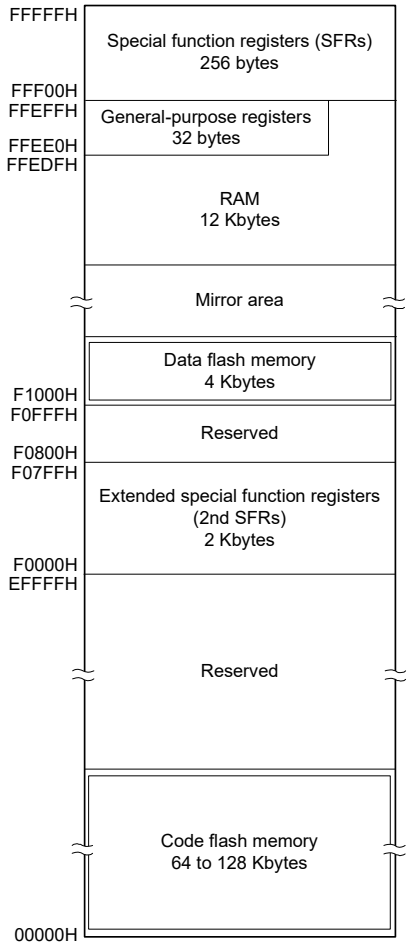
When boot swapping is to be used during self-programming, 000C0H to 000C3H are replaced with 040C0H to 040C3H. Therefore, set the same values as the settings in 000C0H to 000C3H in 040C0H to 040C3H as shown below.

```
OPT2    .CSEG    AT       0x040C0
        .DB               0x36     ; Does not use the interval interrupt of the watchdog timer.
                                   ; Enables watchdog timer operation.
                                   ; The window open period of the watchdog timer is 50%.
                                   ; The overflow time of the watchdog timer is 2^10/fIL.
                                   ; Stops watchdog timer operation during HALT/STOP mode.
        .DB               0xBF     ; Selects VLVD0 as 1.69 V on rising edges and 1.65 V on falling edges.
                                   ; Select the interrupt mode as the LVD0 operation mode.
        .DB               0x6D     ; Select the LP (low-power main) mode as the flash operation mode and 1 MHz
                                   ; as the frequency of the high-speed on-chip oscillator clock.
        .DB               0x85     ; Enables on-chip debug operation, does not erase flash memory data when
                                   ; security ID authentication fails.
```

Caution To specify the option byte by using assembly language, use OPT_BYTE as the relocation attribute name of the CSEG pseudo instruction. To specify 040C0H to 040C3H for use as the option bytes in the case of boot swapping, use the relocation attribute AT to specify absolute addresses.

Section 39 Flash Memory

The RL78 microcontroller incorporates flash memory that allows the writing, erasure, and overwriting of programs. The flash memory consists of a code flash memory area, from which programs can be executed, a data flash memory area, which is suitable for storing data, and an extra area, which is for storing flash memory operation and security settings.



The following methods for programming the flash memory are available.

The contents of code flash memory can be rewritten through serial programming by using a flash memory programmer or other external device (via UART communications), or through self-programming.

- Serial Programming Using Flash Memory Programmer (see **39.1**.)
  Data can be written to the flash memory on-board or off-board by using a dedicated flash memory programmer.
- Serial Programming Using External Device (that Incorporates UART) (see **39.2**.)
  Data can be written to the flash memory on-board through UART communications with an external device (a microcontroller or ASIC).
- Self-programming (see **39.6**.)
  The user application can execute self-programming of the code flash memory.

Caution When rewriting the flash memory, stop the middle-speed on-chip oscillator (MIOEN = 0) and select the high-speed on-chip oscillator (MCM1 = 0) as the main on-chip oscillator clock (fOCO). Do not change the flash operating mode select register (FLMODE).

The data flash memory can be rewritten to by using self-programming during user program execution (background operation). For access and writing to the data flash memory, see **39.6 Self-programming** and **39.10 Data Flash Memory**.
The code flash memory and the data flash memory have a function to protect them from rewriting. For details, see **35.3.3 Flash memory guard function**.

39.1 Serial Programming Using Flash Memory Programmer

The following dedicated flash memory programmers can be used to write data to the internal flash memory of the RL78 microcontroller.

- PG-FP6
- E2 or E2 Lite 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 has been mounted on the target system. Mount a connector for the dedicated flash memory programmer on the target system.
2. Off-board programming
   Data can be written to the flash memory with a dedicated program adapter before the RL78 microcontroller is mounted on the target system.

Table 39 - 1 Wiring between RL78/G24 and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin No.				
					20-pin	24-pin	25-pin	30-pin	32-pin
Signal Name		I/O	Pin Function		LSSOP	HWQFN (4 × 4)	WFLGA (3 × 3)	LSSOP (300 mil)	HWQFN (5 × 5) LQFP (7 × 7)
PG-FP6	E2 or E2 Lite On-chip Debugging Emulator								
—	TOOL0	I/O	Transmit/ receive signal	TOOL0/P40	4	1	A5	5	1
SI/RxD	—	I/O							
—	$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	5	2	B5	6	2
/RESET	—	Output							
V_{CC}	V_{DD}	I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	10	8	B3	12	8
GND		—	Ground	V_{SS}	9	7	B2	11	7
				REGC Note	8	6	A2	10	6
FLMD1	EMV_{DD}	—	Driving power for TOOL0 pin	V_{DD}	10	8	B3	12	8

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin No.				
					40-pin	44-pin	48-pin	52-pin	64-pin
Signal Name		I/O	Pin Function		HWQFN (6 × 6)	LQFP (10 × 10)	LFQFP (7 × 7) HWQFN (7 × 7)	LQFP (10 × 10)	LQFP (12 × 12) LFQFP (10 × 10)
PG-FP6	E2 or E2 Lite On-chip Debugging Emulator								
—	TOOL0	I/O	Transmit/ receive signal	TOOL0/P40	1	2	39	4	5
SI/RxD	—	I/O							
—	$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	2	3	40	5	6
/RESET	—	Output							
V_{CC}	V_{DD}	I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	10	11	48	13	15
GND		—	Ground	V_{SS}	9	10	47	12	13
				EV_{SS}	—	—	—	—	14
				REGC Note	8	9	46	11	12
FLMD1	EMV_{DD}	—	Driving power for the TOOL0 pin	EV_{DD}	—	—	—	—	16

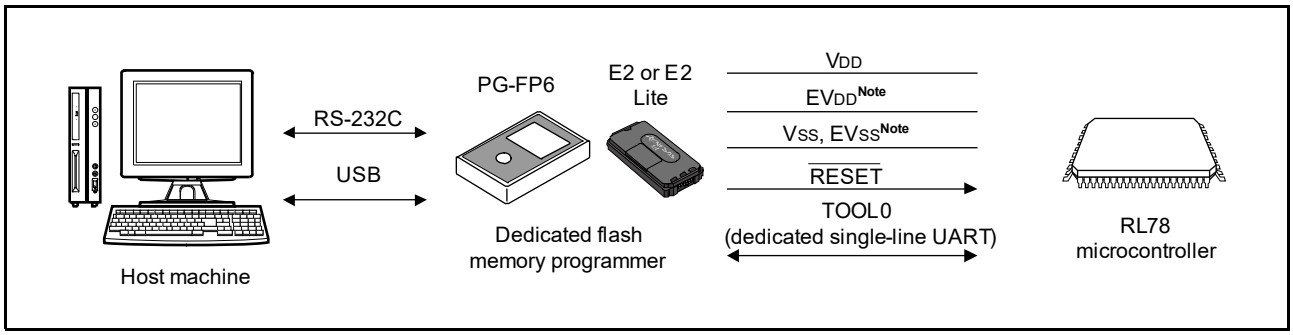
Note Connect the REGC pin to ground via a capacitor (0.47 to 1 μF).

Remark Pins that are not indicated in the above table can be left open-circuit when using the flash memory programmer for flash memory programming.

39.1.1 Programming environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 39 - 1 Environment for Writing Program to Flash Memory



Note Only present in 64-pin products.

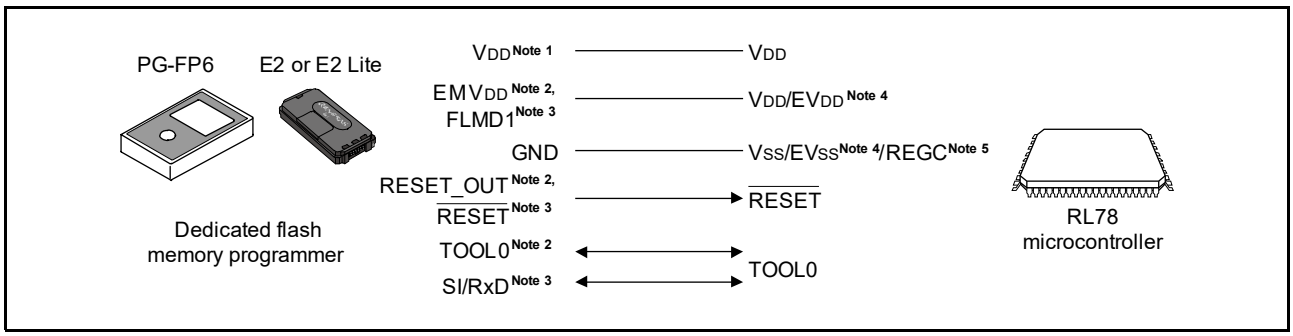
A host machine that controls the dedicated flash memory programmer is necessary.
For the interface between the dedicated flash memory programmer and RL78 microcontroller, the TOOL0 pin of the RL78 microcontroller is used for writing and erasure via the dedicated single-line UART.

39.1.2 Communications mode

The TOOL0 pin of the RL78 microcontroller is used for communications between the dedicated flash memory programmer and RL78 microcontroller through serial communications by using the dedicated single-line UART.

Transfer rate: 1 Mbps, 500 kbps, 250 kbps, 115.2 kbps

Figure 39 - 2 Communications with Dedicated Flash Memory Programmer



Note 1. The signal name for the PG-FP6 is VCC.
Note 2. When using the E2 or E2 Lite on-chip debugging emulator.
Note 3. When using the PG-FP6.
Note 4. Only present in 64-pin products.
Note 5. Connect the 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 for the PG-FP6 or, E2 or E2 Lite on-chip debugging emulator for details.

Table 39 - 2 Pin Connection

Dedicated Flash Memory Programmer				RL78 Microcontroller
Signal Name		I/O	Pin Function	Pin Name[Note 1]
PG-FP6	E2 or E2 Lite On-chip Debugging Emulator			
VCC	VDD	I/O	VDD voltage generation/power monitoring	VDD
GND		—	Ground	VSS, EVSS, REGC[Note 2]
FLMD1	EMVDD	—	Driving power for the TOOL0 pin	VDD, EVDD
$\overline{\text{RESET}}$	—	Output	Reset signal	$\overline{\text{RESET}}$
—	RESET_OUT	Output		
—	TOOL0	I/O	Transmit/receive signal	TOOL0
SI/RxD	—	I/O	Transmit/receive signal	

Note 1. Pins to be connected differ with the product. For details, see **Table 39 - 1**.

Note 2. Connect the REGC pin to ground via a capacitor (0.47 to 1 μF).

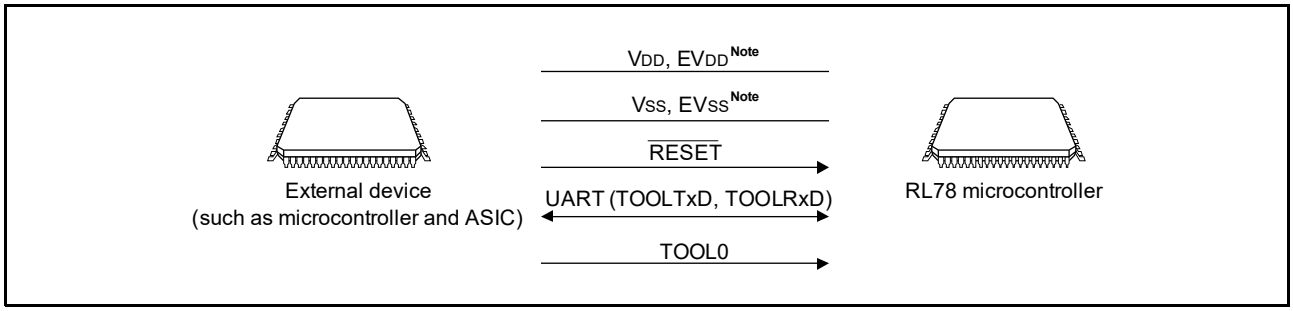
39.2 Serial Programming Using External Device (that Incorporates UART)

On-board data writing to the flash memory of the RL78 microcontroller is possible by using an external device (a microcontroller or ASIC) that is connected to the microcontroller through a UART.
On the development of flash memory programmer by user, refer to the application note, **RL78 Microcontroller (RL78 Protocol C) Serial Programming Guide (R01AN5756)**.

39.2.1 Programming environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 39 - 3 Environment for Writing Program to Flash Memory



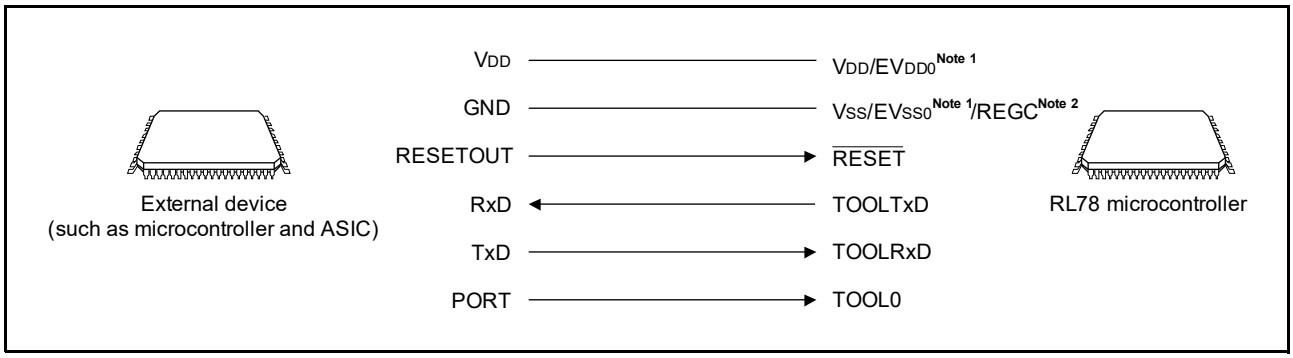
Note Only present in 64-pin products.

39.2.2 Communications mode

The TOOLTxD and TOOLRxD pins of the RL78 microcontroller are used for communications between the external device and RL78 microcontroller through serial communications by using the dedicated UART.

Transfer rate: 1 Mbps, 500 kbps, 250 kbps, 115.2 kbps

Figure 39 - 4 Communications with External Device



Note 1. Only present in 64-pin products.

Note 2. Connect the REGC pin to ground via a capacitor (0.47 to 1 µF).

The external device generates the following signals for the RL78 microcontroller.

Table 39 - 3 Pin Connection

External Device			RL78 Microcontroller
Signal Name	I/O	Pin Function	Pin Name
VDD	I/O	VDD voltage generation/power monitoring	VDD, EVDDNote 1
GND	—	Ground	VSS, REGCNote 2, EVSSNote 1
RESETOUT	Output	Reset signal output	$\overline{\text{RESET}}$
RxD	Input	Receive signal	TOOLTxD
TxD	Output	Transmit signal	TOOLRxD
PORT	Output	Mode signal	TOOL0

Note 1. Only present in 64-pin products.

Note 2. Connect the REGC pin to ground via a capacitor (0.47 to 1 µF).

39.3 Handling of Pins on the Board

To write to the flash memory on board by using a flash memory programmer, mount a connector for the dedicated flash memory programmer on the target system. In addition, include a function for mode switching from normal operation to flash memory programming in the board design.
Entry to flash memory programming mode places all pins that are not to be used in programming the flash memory in the same states as those immediately after a reset. If the given states do not suit the operation of external devices, apply appropriate handling to the pins.

Remark For details on flash memory programming mode, refer to **39.4.2 Flash memory programming mode**.

39.3.1 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 the low level is prohibited for t_{HD} period after external reset release. However, when this pin is used via a pull-down resistor, use a resistor with a value of 500 kΩ or more.
When used as an output pin: When this pin is used via a pull-down resistor, use a resistor with a value of 500 kΩ or more.

Remark 1. t_{HD}: This is the time over which the TOOL0 pin must be kept at the low level following the end of the external and internal resets for setting of the flash memory programming mode. See **43.10 Timing of Entry to Flash Memory Programming Modes** or **44.10 Timing of Entry to Flash Memory Programming Modes**.

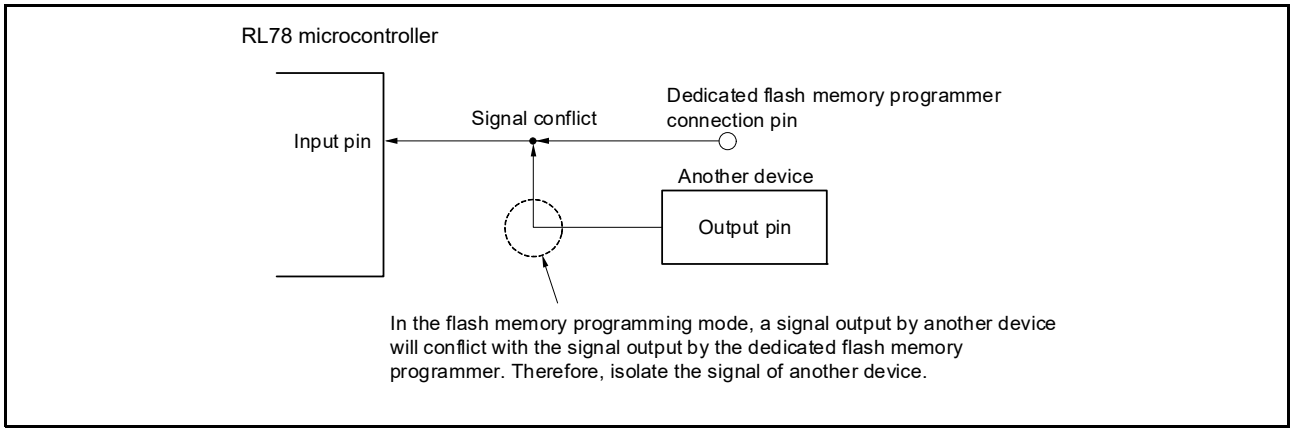
Remark 2. The SAU and IICA pins are not used for communications between the RL78 microcontroller and dedicated flash memory programmer, because the single-line UART (TOOL0 pin) is used.

39.3.2 $\overline{\text{RESET}}$ pin

Signal conflict will occur if the reset signal of the dedicated flash memory programmer and external device is 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 39 - 5 Signal Conflict ($\overline{\text{RESET}}$ Pin)



39.3.3 Port pins

Entry to flash memory programming mode places all pins that are not to be used in programming the flash memory in the same states as those immediately after a reset. If the given states of the port pins do not suit the operation of external devices that are connected to individual port pins, apply appropriate handling to the pins such as connection to V_{DD} or V_{SS} via resistors.

39.3.4 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.

39.3.5 Power supply

To use the supply voltage output of the flash memory programmer, connect the V_{DD} pin to V_{DD}**Note** 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 the power supply in the same manner as during normal operation. 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}**Note** and GND of the flash memory programmer to use the power monitor function by the flash memory programmer.

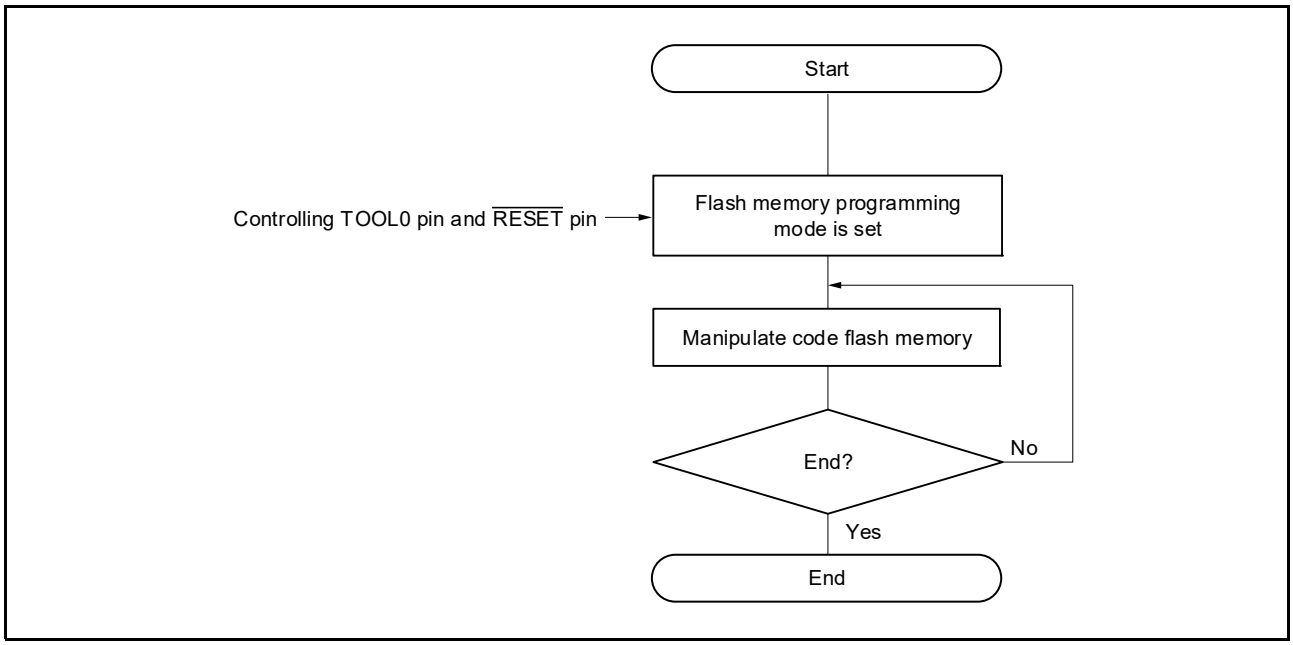
Note The signal name for the PG-FP6 is V_{CC}.

39.4 Programming Method

39.4.1 Serial programming procedure

The following figure illustrates a flow for rewriting the code flash memory through serial programming.

Figure 39 - 6 Code Flash Memory Manipulation Procedure



39.4.2 Flash memory programming mode

To rewrite the contents of the code flash memory through serial programming, specify the flash memory programming mode. The following describes how to enter the flash memory programming mode.

<For serial programming by using the dedicated flash memory programmer>

Connect the RL78 microcontroller to a dedicated flash memory programmer. Starting communications with the dedicated flash memory programmer automatically places the RL78 LSI chip in the flash memory programming mode.

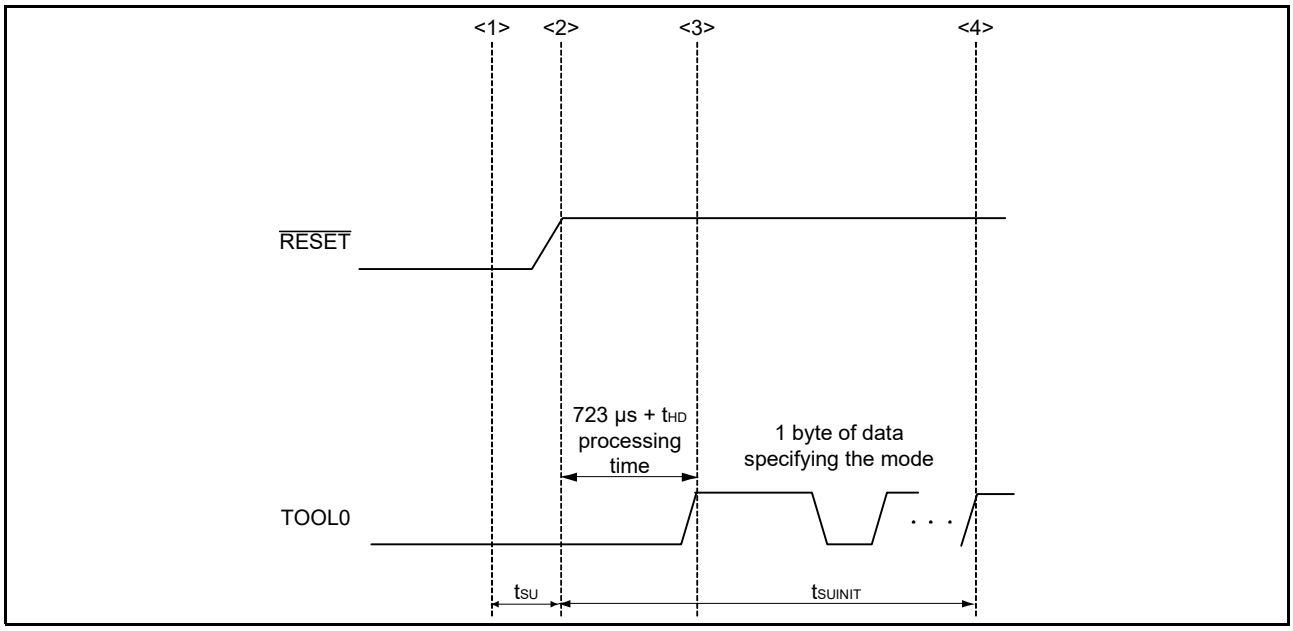
<For serial programming by using an external device (UART communications)>

Set the TOOL0 pin to the low level, and then cancel the reset (see **Table 39 - 4**). After that, enter flash memory programming mode according to steps <1> to <4> shown in **Figure 39 - 7**. For details, refer to the application note, **RL78 Microcontroller (RL78 Protocol C) Serial Programming Guide (R01AN5756)**.

Table 39 - 4 Relationship between the Voltage Applied to the TOOL0 Pin and the Operating Mode after Release from the Reset State

TOOL0	Operating Mode
EV_{DD}	Normal operation mode
0 V	Flash memory programming mode

Figure 39 - 7 Entry to Flash Memory Programming Mode



<1> The low level is input to the TOOL0 pin.
<2> The external reset is de-asserted (release from a POR or LVD reset must precede this).
<3> The TOOL0 pin is set to the high level.
<4> The bit rate setting is based on the UART reception.

Remark t_{SUINIT}: The section is the up to 100 ms from the end of the external reset within which specifying the initial communications settings must be finished.

t_{SU}: This is the time from the TOOL0 pin being placed at the low level until the end of the external reset.

t_{HD}: This is the time over which the TOOL0 pin must be kept at the low level following the end of an external reset (excluding the processing time of the firmware to control the flash memory).

For details, see **43.10 Timing of Entry to Flash Memory Programming Modes** or **44.10 Timing of Entry to Flash Memory Programming Modes**.

39.4.3 Selecting communications mode

The communications modes of the RL78 microcontroller are shown in the table below.

Table 39 - 5 Communications Modes

Communications Mode	Standard Setting[Note 1]				Pins Used
	Port	Speed[Note 2]	Frequency	Multiply Rate	
Single-line UART (when a flash memory programmer is used, or when an external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	—	—	TOOL0
Dedicated UART (when an 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. As factors other than bit rate errors, such as dullness of the signal waveform, may also affect UART communications, conduct extensive evaluation with a selected bit rate before attempting to use it.

39.4.4 Communications commands

The RL78 microcontroller executes serial programming through the commands listed in **Table 39 - 6**.
The signals sent from the dedicated flash memory programmer or external device to the RL78 microcontroller are called commands, and programming functions corresponding to the commands are executed. For details, refer to the application note, **RL78 Microcontroller (RL78 Protocol C) Serial Programming Guide (R01AN5756)**.

Table 39 - 6 Flash Memory Control Commands

Classification	Command Name	Function
Verification	Verify	Compares the contents of a specified area of the flash memory with data transmitted from the programmer.
Erasure	Block Erase	Erases a specified area in the flash memory.
Blank checking	Block Blank Check	Checks if a specified block in the flash memory has been correctly erased.
Writing	Programming	Writes data to a specified area in the flash memory**Note**.
Getting information	Silicon Signature	Gets the RL78 microcontroller information (such as the device name, 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	Releases the setting to prohibit writing.
Others	Reset	Used to detect the state of synchronization during communications.
	Baud Rate Set	Sets the bit rate when UART communications mode is selected.

Note Confirm that no data have been written to the write area. If data in the area has not been erased, do not attempt further writing of data because data cannot be erased after the setting to prohibit block erasure has been made.

The product information (such as device name and firmware version) can be obtained by executing the Silicon Signature command.

Tables 39 - 7 and **39 - 8** show the signature data list and an example of signature data, respectively.

Table 39 - 7 Signature Data List

Field Name	Description	Number of Bytes 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 end address	End address of the code flash memory area (Sent from the lower-order byte of the address. Example: 00000H to 1FFFFH (128 Kbytes) → FFH, FFH, 01H)	3 bytes
Data flash memory area end address	End address of the data flash memory area (Sent from the lower-order byte of the address. Example: F1000H to F1FFFH (4 Kbytes) → FFH, 1FH, 0FH)	3 bytes
Firmware version	Version information of firmware for programming (Sent from the byte equivalent to the highest-order digit of the version number. Example: Ver. 1.02 → 01H, 00H, 02H)	3 bytes

Table 39 - 8 Example of Signature Data

Field Name	Description	Number of Bytes of Transmit Data	Data (Hexadecimal)
Device code	RL78 protocol C	3 bytes	10 00 0A
Device name	R7F101GLG	10 bytes	52 = “R” 37 = “7” 46 = “F” 31 = “1” 30 = “0” 31 = “1” 47 = “G” 4C = “L” 47 = “G” 20 = “ ”
Code flash memory area end address	Code flash memory area 00000H to 1FFFFH (128 Kbytes)	3 bytes	FF FF 1F
Data flash memory area end address	Data flash memory area F1000H to F1FFFH (4 Kbytes)	3 bytes	FF 1F 0F
Firmware version	Ver. 1.02	3 bytes	01 00 02

39.5 Processing Times for Commands When the Dedicated Flash Memory Programmer Is in Use (Reference Values)

The following shows the processing times for each command (reference value) when the PG-FP6 is used as a dedicated flash memory programmer.

Table 39 - 9 Processing Times for Commands When the PG-FP6 Is in Use (Reference Values)

PG-FP6 Command	Code Flash Memory	
	64 Kbytes	128 Kbytes
Erasure	1.0 s	1.3 s
Writing	1.6 s	2.9 s
Verification	1.2 s	2.2 s
Writing after erasure	2.3 s	4.0 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

39.6 Self-programming

The RL78 microcontroller supports 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, it can be used to update the program in the field. For details, refer to the **RL78 Family Renesas Flash Driver RL78 Type 01 User's Manual (R20UT4830)**.

Caution 1. The self-programming function cannot be used when the CPU operates with the subsystem clock (f_{SUB}).

Caution 2. To prohibit an interrupt during self-programming, in the same way as in the normal operation mode, execute self-programming in the state where the IE flag is cleared to 0 by the DI instruction. To enable an interrupt, clear the interrupt mask flag for an interrupt to be accepted to 0 in the state where the IE flag is set to 1 by the EI instruction, and then execute self-programming.

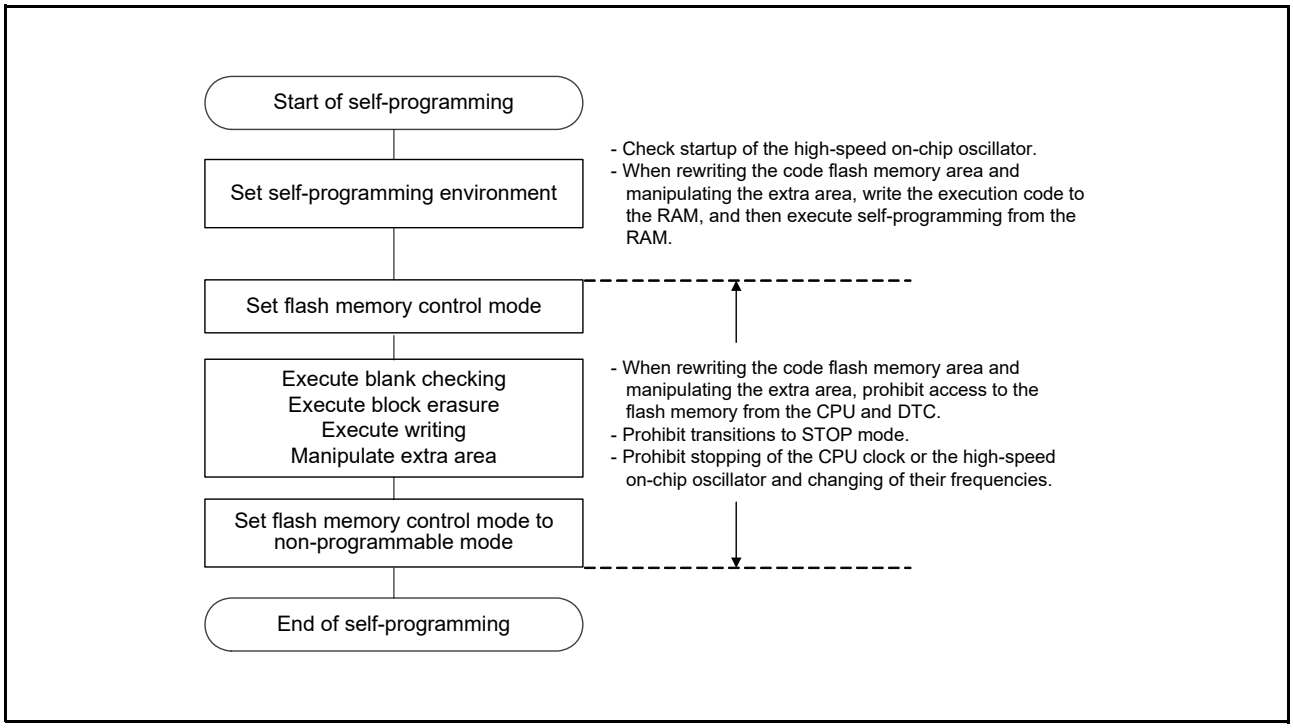
Caution 3. The high-speed on-chip oscillator should be kept operating during self-programming. If it is stopped, it should be made to operate again (HIOSTOP = 0), and self-programming should be re-executed after 5 μs have elapsed. Stop the middle-speed on-chip oscillator (MIOEN = 0) and select the high-speed on-chip oscillator (MCM1 = 0) as the main on-chip oscillator clock (f_{OCO}).

Caution 4. Do not change the flash operating mode select register (FLMODE) while the flash memory is being rewritten.

39.6.1 Self-programming procedure

The following figure illustrates a flow for rewriting the flash memory by using self-programming.
For details on registers for use in self-programming, see **39.6.2 Registers to control the flash memory**.

Figure 39 - 8 Flow of Self-programming (Rewriting the Flash Memory)



39.6.2 Registers to control the flash memory

The following registers are used to control the flash memory.

- Flash address pointer registers H and L (FLAPH, FLAPL)
- Flash end address pointer registers H and L (FLSEDH, FLSEDL)
- Flash write buffer registers H and L (FLWH, FLWL)
- Flash protect command register (PFCMD)
- Flash status register (PFS)
- Flash programming mode control register (FLPMC)
- Flash area selection register (FLARS)
- Flash memory sequencer initial setting register (FSSET)
- Flash memory sequencer control register (FSSQ)
- Flash extra area sequencer control register (FSSE)
- Flash registers initialization register (FLRST)
- Flash memory sequencer status registers H and L (FSASTH, FSASTL)
- Flash security flag monitoring register (FLSEC)
- Flash FSW monitoring register E (FLFSWE)
- Flash FSW monitoring register S (FLFSWS)
- Data flash control register (DFLCTL)
- Interrupt vector jump enable register (VECTCTRL)
- Interrupt vector change registers 0 and 1 (FLSIVC0, FLSIVC1)

39.6.2.1 Flash address pointer registers H and L (FLAPH, FLAPL)

The FLAPH and FLAPL registers specify the address where programming the flash memory is to start. The FLAPH and FLAPL registers can be set by 8-bit and 16-bit memory manipulation instructions, respectively. The values of the FLAPH and FLAPL registers are 00H and 0000H, respectively, under either of the following conditions.

- Following a reset
- The value of the FLRST bit of the FLRST register is 1.

Figure 39 - 9 Format of Flash Address Pointer Registers H and L (FLAPH, FLAPL)

Address: F02C4H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
FLAPH	0	0	0	0	FLAP19	FLAP18	FLAP17	FLAP16

Address: F02C2H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
FLAPL	FLAP15	FLAP14	FLAP13	FLAP12	FLAP11	FLAP10	FLAP9	FLAP8

7	6	5	4	3	2	1	0
FLAP7	FLAP6	FLAP5	FLAP4	FLAP3	FLAP2	FLAP1	FLAP0

Caution 1. The FLAPH and FLAPL registers can be rewritten under either of the following conditions.
- **The FLSPM bit in the FLPMC register is 1 (code flash memory area: programming mode).**
- **The EEEMD bit in the FLPMC register is 1 (data flash memory area: programming mode).**

Caution 2. Rewrite or read these registers when the extra area sequencer and the code/data flash memory area sequencer are stopped (SQEND = 0, ESQEND = 0 in the FSASTH register, SQST = 0 in the FSSQ register, and ESQST = 0 in the FSSE register).

39.6.2.2 Flash end address pointer registers H and L (FLSEDH, FLSEDL)

The FLSEDH and FLSEDL registers specify the address where programming of the flash memory is to end. The FLSEDH and FLSEDL registers can be set by 8-bit and 16-bit memory manipulation instructions, respectively. The values of the FLSEDH and FLSEDL registers are 00H and 0000H, respectively, under either of the following conditions.

- Following a reset
- The value of the FLRST bit of the FLRST register is 1.

Figure 39 - 10 Format of Flash End Address Pointer Registers H and L (FLSEDH, FLSEDL)

Address: F02C8H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
FLSEDH	0	0	0	0	EWA19	EWA18	EWA17	EWA16

Address: F02C6H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
FLSEDL	EWA15	EWA14	EWA13	EWA12	EWA11	EWA10	EWA9	EWA8

7	6	5	4	3	2	1	0
EWA7	EWA6	EWA5	EWA4	EWA3	EWA2	EWA1	EWA0

Caution 1. The FLSEDH and FLSEDL registers can be rewritten under either of the following conditions.
- **The FLSPM bit in the FLPMC register is 1 (code flash memory area: programming mode).**
- **The EEEMD bit in the FLPMC register is 1 (data flash memory area: programming mode).**

Caution 2. Rewrite or read these registers when the extra area sequencer and the code/data flash memory area sequencer are stopped (SQEND = 0, ESQEND = 0 in the FSASTH register, SQST = 0 in the FSSQ register, and ESQST = 0 in the FSSE register).

Caution 3. The settings of the EWA1 and EWA0 bits are meaningless during programming of the code flash memory.

Table 39 - 10 Method of Setting the FLAPH, FLAPL, FLSEDH, and FLSEDL Registers

Commands Exclusively for Use with the Code/Data Flash Memory Area Sequencer		Settings of the FLAP and FLSED Registers	
FSSQ	Writing	FLAPH, FLAPL: Address from which writing is to proceed FLSEDH, FLSEDL: All 0s	
	Blank checking	For one word: Settings of FLAPH and FLAPL = settings of FLSEDH and FLSEDL For two or more words: Settings of FLAPH and FLAPL < settings of FLSEDH and FLSEDL	
	Block erasure**Note**	Code flash memory	FLAPH, FLAPL: Set the start address and 0s in the FLAP[19:11] bits and the FLAP[10:2] bits, respectively.
			FLSEDH, FLSEDL: Set the end address and 1s in the EWA[19:11] bits and the EWA[10:2] bits, respectively.
		Data flash memory	FLAPH, FLAPL: Set the start address and 0s in the FLAP[19:8] bits and the FLAP[7:0] bits, respectively.
			FLSEDH, FLSEDL: Set the end address and 1s in the EWA[19:8] bits and the EWA[7:0] bits, respectively.
FSSE	All commands	Settings of FLAPH and FLAPL: All 0s Settings of FLSEDH and FLSEDL: All 0s	

Note Set the FLAPH, FLAPL, FLSEDH, and FLSEDL registers so that the following condition is met.
Combined settings of FLAPH and FLAPL ≤ combined settings of FLSEDH and FLSEDL

39.6.2.3 Flash write buffer registers H and L (FLWH, FLWL)

The FLWH and FLWL registers hold data to be written during programming of the flash memory. The FLWH and FLWL registers can be set by a 16-bit memory manipulation instruction. The value of each of the FLWH and FLWL registers is 0000H under any of the following conditions.

- Following a reset
- The value of the FLRST bit of the FLRST register is 1.
- The flash memory sequencer has finished operating.

Set data to be written to the data flash memory in the 8 lower-order bits of the FLWL register.

Figure 39 - 11 Format of Flash Write Buffer Registers H and L (FLWH, FLWL)

Address: F02CEH
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
FLWH	FLW31	FLW30	FLW29	FLW28	FLW27	FLW26	FLW25	FLW24

7	6	5	4	3	2	1	0
FLW23	FLW22	FLW21	FLW20	FLW19	FLW18	FLW17	FLW16

Address: F02CCH
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
FLWL	FLW15	FLW14	FLW13	FLW12	FLW11	FLW10	FLW9	FLW8

7	6	5	4	3	2	1	0
FLW7	FLW6	FLW5	FLW4	FLW3	FLW2	FLW1	FLW0

Caution 1. The FLWH and FLWL registers can be rewritten under either of the following conditions.
• The FLSPM bit in the FLPMC register is 1 (code flash memory area: programming mode).
• The EEEMD bit in the FLPMC register is 1 (data flash memory area: programming mode).

Caution 2. Rewrite or read these registers when the extra area sequencer and the code/data flash memory area sequencer are stopped (SQEND = 0, ESQEND = 0 in the FSASTH register, SQST = 0 in the FSSQ register, and ESQST = 0 in the FSSE register).

Caution 3. When writing to the data flash memory, set write data in the 8 lower-order bits of the FLWL register. Set other bits to 0.

39.6.2.4 Flash protect command register (PFCMD)

The PFCMD register protects the flash programming mode control register (FLPMC) against write access. To enable write access to the FLPMC register, write A5H according to the specific sequence. For details on the procedure for handling the specific sequence, see **39.6.3 Setting the flash memory control mode**. The PFCMD register can be set by an 8-bit memory manipulation instruction.

Figure 39 - 12 Format of Flash Protect Command Register (PFCMD)

Address: F00C0H
After reset: Undefined
R/W: W

Symbol	7	6	5	4	3	2	1	0
PFCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0

39.6.2.5 Flash status register (PFS)

The PFS register indicates whether or not a protection error has occurred during write access to the flash programming mode control register (FLPMC). For details on the conditions for setting and clearing the FPRERR bit, see **39.6.3.1 Procedure for executing the specific sequence**. The PFS register can be read by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 39 - 13 Format of Flash Status Register (PFS)

Address: F00C1H
After reset: 00H
R/W: R

Symbol	7	6	5	4	3	2	1	0
PFS	0	0	0	0	0	0	0	FPRERR

FPRERR	Protection error flag
0	No error has occurred.
1	An error has occurred.

39.6.2.6 Flash programming mode control register (FLPMC)

The FLPMC register disables or enables writing to the flash memory and selects the programming mode. Enabling write access to the FLPMC register requires the execution of a specific sequence. For details on the procedure for handling the specific sequence, see **39.6.3 Setting the flash memory control mode**. The FLPMC register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 08H.

Figure 39 - 14 Format of Flash Programming Mode Control Register (FLPMC)

Address: F02C0H
After reset: 08H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
FLPMC	0	0	0	EEEMD	FWEDIS	0	FLSPM	0

EEEMD	Selection of the programming mode for the data flash memory
0	Non-programmable mode
1	Programming mode

FWEDIS	Software control over enabling or disabling erasure and programming of the code flash memory[Note]
0	Enables erasure and programming.
1	Disables erasure and programming.

FLSPM	Selection of the programming mode for the code flash memory
0	Non-programmable mode
1	Programming mode

Note Be sure to keep the value of this bit at 0 until erasure or programming of the code flash memory is completed.

Caution **When the extra area sequencer and the code/data flash memory area sequencer are stopped (SQEND = 0, ESQEND = 0 in the FSASTH register, SQST = 0 in the FSSQ register, and ESQST = 0 in the FSSE register), rewriting to the FLPMC register is enabled.**

39.6.2.7 Flash area selection register (FLARS)

The FLARS register selects the area of the flash memory for self-programming. The FLARS register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of the FLARS register is 00H under either of the following conditions.

- Following a reset
- The value of the FLRST bit of the FLRST register is 1.

Figure 39 - 15 Format of Flash Area Selection Register (FLARS)

Address: F02C1H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
FLARS	0	0	0	0	0	0	0	EXA

EXA	Selection of the area of the flash memory for self-programming
0	Code/data flash memory areas
1	Extra area

Caution The FLARS register can be rewritten under either of the following conditions.
• The FLSPM bit in the FLPMC register is 1 (code flash memory area: programming mode).
• The EEEMD bit in the FLPMC register is 1 (data flash memory area: programming mode).

39.6.2.8 Flash memory sequencer initial setting register (FSSET)

The FSSET register sets the operating frequency of the flash memory sequencer and makes the initial settings for boot swapping. The FSSET register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 39 - 16 Format of Flash Memory Sequencer Initial Setting Register (FSSET)

Address: F00B6H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
FSSET	TMSPMD	TMBTSEL	FSET5	FSET4	FSET3	FSET2	FSET1	FSET0

TMSPMD	Selection of boot area setting[Note]
0	Specifies the boot area according to the setting of EX bit 8 (BTFLG) in the security flag and boot swap function setting area of the extra area. BTFLG = 0: Boot cluster 1 as the boot area BTFLG = 1: Boot cluster 0 as the boot area (default)
1	Specifies the boot area according to the setting of the TMBTSEL bit.

TMBTSEL	Specification of the boot area when TMSPMD = 1
0	Specifies boot cluster 0 as the boot area.
1	Specifies boot cluster 1 as the boot area.

FSET[5:0]	Setting of the operating frequency of the flash memory sequencer
—	Sets the operating frequency of the flash memory sequencer. For the correspondence between the operating frequency of the flash memory sequencer and the setting of the FSET[5:0] bits, see **Table 39 - 11**.

Note Setting the TMSPMD and TMBTSEL bits is not possible while the BTPR bit in FLSEC is 0 (rewriting of the boot area is disabled).

Caution 1. The FSSET register can be rewritten under either of the following conditions.
- **The FLSPM bit in the FLPMC register is 1 (code flash memory area: programming mode).**
- **The EEEMD bit in the FLPMC register is 1 (data flash memory area: programming mode).**

Caution 2. The set values of the boot area are immediately reflected. To change the boot area after a reset is released, read the MBTSEL bit in FSASTL while the TMSPMD bit is set to 0 and set the same value to the TMBTSEL bit. After that, set the TMSPMD bit to 1, and then specify the boot cluster to be started as the boot area at release from the reset state in the BTFLG bit using the extra area sequencer. The boot cluster set by the BTFLG bit is activated as the boot area at the next reset release.

Table 39 - 11 Correspondence between the Operating Frequency of the Flash Memory Sequencer and the Setting of the FSET[5:0] Bits

Operating Frequency (MHz)	Setting of the FSET[5:0] Bits	Operating Frequency (MHz)	Setting of the FSET[5:0] Bits	Operating Frequency (MHz)	Setting of the FSET[5:0] Bits
48	100111B	—	—	—	—
32	011111B	31	011110B	30	011101B
29	011100B	28	011011B	27	011010B
26	011001B	25	011000B	24	010111B
23	010110B	22	010101B	21	010100B
20	010011B	19	010010B	18	010001B
17	010000B	16	001111B	15	001110B
14	001101B	13	001100B	12	001011B
11	001010B	10	001001B	9	001000B
8	000111B	7	000110B	6	000101B
5	000100B	4	000011B	3	000010B
2	000001B	1	000000B	—	—

Caution Set the value corresponding to that obtained by rounding the CPU operating frequency up to the nearest whole number in the FSET[4:0] bits.
(For example, when the CPU operating frequency is 4.5 MHz, set the bits for 5 MHz.)
Note that frequencies that are not whole numbers, such as 1.5 MHz, are not available as CPU operating frequencies below 4 MHz.

39.6.2.9 Flash memory sequencer control register (FSSQ)

The FSSQ register specifies operation control and commands for use with the code/data flash memory area sequencer. When the SQST bit in this register is set to 1, the code/data flash memory area sequencer executes the command set in the MDCH and SQMD[2:0] bits. The FSSQ register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of the FSSQ register is 00H under either of the following conditions.

- Following a reset
- The value of the FLRST bit of the FLRST register is 1.

Figure 39 - 17 Format of Flash Memory Sequencer Control Register (FSSQ) (1/2)

Address: F02C5H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	5	4	3	2	1	0
FSSQ	SQST	FSSTP	0	0	MDCH	SQMD2	SQMD1	SQMD0

SQST	Operation control of the code/data flash memory area sequencer
0	The code/data flash memory area sequencer is stopped.**Note 1**
1	The code/data flash memory area sequencer is started.

FSSTP	Forcible termination control of the code/data flash memory area sequencer
0	The code/data flash memory area sequencer is not forcibly terminated.
1	The code/data flash memory area sequencer is forcibly terminated.

Figure 39 - 17 Format of Flash Memory Sequencer Control Register (FSSQ) (2/2)

MDCH	SQMD2	SQMD1	SQMD0	Commands for use with the code/data flash memory area sequencer
0	0	0	1	• Writing Writes data stored in the FLWH and FLWL registers to the address specified in the FLAPH and FLAPL registers.Note 2 Writes 4-byte data when the code flash memory area address is specified. Writes the 1-byte data stored in the eight lower-order bits (FLW[7:0]) in FLWL to the specified address when the data flash memory area address is specified.
0	0	1	1	• Blank checking of the code flash memory area Checks whether the value of the code flash memory area from the address specified in the FLAPH and FLAPL registers to the address specified in the FLSEDH and FLSEDL registers is 1.Note 3
1	0	1	1	• Blank checking of the data flash memory area Checks whether the value of the data flash memory area from the address specified in the FLAPH and FLAPL registers to the address specified in the FLSEDH and FLSEDL registers is 1.
0	1	0	0	• Block erasure Erases blocks in the range from the block start address specified in the FLAPH and FLAPL registers to the block end address specified in the FLSEDH and FLSEDL registers.Note 4
Other than above				Setting prohibited

Note 1. Check that the SQEND bit in the FSASTH register is 1 (the code/data flash memory area sequencer being stopped), and then set the SQST bit to 0 to stop the code/data flash memory area sequencer.

Note 2. Four-byte data can be written to the code flash memory area. Set the two lower-order bits of the FLSEDL register to 00B to be a multiple of 4. For details, see **39.6.6.4 Operations for rewriting the code flash memory area**.

Note 3. Specify a start address (at intervals of four bytes) for blank checking of the code flash memory area. Set the two lower-order bits of the FLSEDL register to 00B to be a multiple of 4. For details, see **39.6.6.4 Operations for rewriting the code flash memory area**.

Note 4. The code flash memory area blocks can be erased in units of 2 Kbytes. The data flash memory blocks can be erased in units of 256 bytes. Specify the erase addresses (start address and end address) so that all blocks to be erased are included. For details, see **39.6.6.4 Operations for rewriting the code flash memory area** and **39.6.6.5 Operations for rewriting the data flash memory area**. For the relationship between the address and block number, see **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**.

Caution The FSSQ register can be rewritten under either of the following conditions.

- **The FLSPM bit in the FLPMC register is 1 (code flash memory area: programming mode) and the FWEDIS bit is 0 (enabling erasure and programming of the code flash memory).**
- **The EEEMD bit in the FLPMC register is 1 (data flash memory area: programming mode).**

39.6.2.10 Flash extra area sequencer control register (FSSE)

The FSSE register specifies operation control and commands for use with the extra area sequencer. When the ESQST bit of the FSSE register is set to 1, the extra area sequencer executes the command set by the ESQMD[3:0] bits. The FSSE register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of the FSSE register is 00H under either of the following conditions.

- Following a reset
- The value of the FLRST bit of the FLRST register is 1.

Figure 39 - 18 Format of Flash Extra Area Sequencer Control Register (FSSE) (1/2)

Address: F00B7H
After reset: 00H
R/W: R/W

Symbol	<7>	6	5	4	3	2	1	0
FSSE	ESQST	0	0	0	ESQMD3	ESQMD2	ESQMD1	ESQMD0

ESQST	Operation control of the extra area sequencer
0	The extra area sequencer is stopped.**Note**
1	The extra area sequencer is started.

Figure 39 - 18 Format of Flash Extra Area Sequencer Control Register (FSSE) (2/2)

ESQMD3	ESQMD2	ESQMD1	ESQMD0	Commands for Use with the extra area sequencer
0	0	0	1	• Write to the flash shield window setting area Writes the 4-byte data specified in the FLWH and FLWL registers to the flash shield window setting area of the extra area to control flash shield window mode and set the start block and end block. Also, if the setting of EX bit 15 (FSPR) in the flash shield window setting area is 0, no value can be written to the area. Attempted writing leads to setting of the extra area sequencer error flag (ESEQER) to 1.
0	1	1	0	• Write to the flash read protection setting area Writes the 4-byte data specified in the FLWH and FLWL registers to the flash read protection setting area of the extra area to disable changing of the flash read protection setting and set the start block and end block. Also, if the setting of EX bit 31 (SWPR) in the flash read protection setting area is 0, no value can be written to the area. Attempted writing leads to setting of the extra area sequencer error flag (ESEQER) to 1.
0	1	1	1	• Write to the security flag and boot swap function setting area Writes the 4-byte data specified in the FLWH and FLWL registers to the flash memory security flag and boot swap function setting area of the extra area to disable block erasure, writing, and rewriting of the boot area and set selection of the boot area. Also, if the setting of EX bit 9 (BTPR) in the security flag and boot swap function setting area is 0, no value can be written to the area. Attempted writing leads to setting of the extra area sequencer error flag (ESEQER) to 1.
Other than above				Setting prohibited

Note Check that the ESQEND bit in the FSASTH register is 1 (the extra area sequencer being stopped), and then set the ESQST bit to 0 to stop the extra area sequencer.

Caution 1. The FSSE register can be rewritten when the following condition is met.
The FLSPM bit in the FLPMC register is 1 (code flash memory area: programming mode) and the FWEDIS bit is 0 (enabling erasure and programming of the code flash memory).

Caution 2. To write to the extra area, set the EXA bit of the FLARS register to 1 and set the data to be written in the FLWH and FLWL registers before activating the extra area sequencer.

Caution 3. Rewrite the ESQMD[3:0] bits while the extra area sequencer and the code/data flash memory area sequencer are stopped (SQEND = 0, ESQEND = 0 in the FSASTH register, SQST = 0 in the FSSQ register, and ESQST = 0 in the FSSE register).

39.6.2.11 Flash registers initialization register (FLRST)

The FLRST register initializes all registers related to the sequencer whether it is to be used with the extra area or code/data flash memory areas. The FLRST register can be set by a 1-bit or 8-bit memory manipulation instruction. For details about how to manipulate the FLRST register, see **39.6.4 Initializing the registers for use with the flash memory sequencer**. The value of this register following a reset is 00H.

Figure 39 - 19 Format of Flash Registers Initialization Register (FLRST)

Address: F02C9H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
FLRST	0	0	0	0	0	0	0	FLRST

FLRST	Control of initializing the registers
0	The registers are not reset.
1	The FLAPH, FLAPL, FLSEDH, FLSEDL, FLWH, FLWL, FLARS, FSSQ, and FSSE registers are reset.

Caution 1. Registers can be initialized by setting the FLRST bit to 1 only when the extra area sequencer and the code/data flash memory area sequencer are stopped (SQEND = 0, ESQEND = 0 in the FSASTH register, SQST = 0 in the FSSQ register, and ESQST = 0 in the FSSE register).

Caution 2. When using the sequencer, be sure to set the FLRST bit to 0 before setting the FLAPH, FLAPL, FLSEDH, FLSEDL, FLWH, FLWL, FLARS, FSSQ, and FSSE registers. Do not set the FLRST bit to 1 during operation of the sequencer.

39.6.2.12 Flash memory sequencer status registers H and L (FSASTH, FSASTL)

The FSASTH and FSASTL registers indicate the results of the respective operations of the flash memory sequencer when it has been used with the extra area or code/data flash memory areas. The FSASTH and FSASTL registers can be read by a 1-bit or 8-bit memory manipulation instruction.

Figure 39 - 20 Format of Flash Memory Sequencer Status Registers H and L (FSASTH, FSASTL) (1/2)

Address: F02CBH
After reset: 00H/04H
R/W: R

Symbol	7	6	5	4	3	2	1	0
FSASTH	ESQEND	SQEND	0	0	×	0	0	0

Address: F02CAH
After reset: Undefined[Note 1]
R/W: R

Symbol	7	6	5	4	3	2	1	0
FSASTL	MBTSEL	MOPEN	ESEQER	SEQER	BLER	0	WRER	ERER

ESQEND	Extra area sequencer operation end status flag
0	The extra area sequencer is operating or stopped by setting the ESQST bit to 0.
1	The extra area sequencer is stopped.

SQEND	Code/data flash memory area sequencer operation end status flag
0	The code/data flash memory area sequencer is operating or stopped by setting the SQST bit to 0.
1	The code/data flash memory area sequencer is stopped.

MBTSEL	Boot flag monitoring bit[Note 2]
0	BTFLG = 1 (boot area: boot cluster 0)
1	BTFLG = 0 (boot area: boot cluster 1)

MOPEN	Code/data flash memory area sequencer operation status flag
0	The code/data flash memory area sequencer is stopped.
1	The code/data flash memory area sequencer is operating.

ESEQER	Error flag of the extra area sequencer
0	No error has occurred.
1	An error has occurred.
This flag is cleared to 0 when the extra area sequencer is activated.	

Figure 39 - 20 Format of Flash Memory Sequencer Status Registers H and L (FSASTH, FSASTL) (2/2)

SEQER	Error flag of the flash memory sequencer
0	No error has occurred.
1	An error has occurred.
This flag is cleared to 0 when the extra area sequencer or the code/data flash memory area sequencer is activated.	

BLER	Error flag for the blank check command
0	No error has occurred.
1	An error has occurred.
This flag is cleared to 0 when the extra area sequencer or the code/data flash memory area sequencer is activated.	

WRER	Error flag for the write command
0	No error has occurred.
1	An error has occurred.
This flag is cleared to 0 when the extra area sequencer or the code/data flash memory area sequencer is activated. The read value of this bit becomes undefined if the command is forcibly terminated during writing.	

ERER	Error flag for the block erase command
0	No error has occurred.
1	An error has occurred.
This flag is cleared to 0 when the extra area sequencer or the code/data flash memory area sequencer is activated. The read value of this bit becomes undefined if the command is forcibly terminated during block erasure.	

Note 1. The initial value of the MBTSEL bit is undefined because it depends on the value of the BTFLG bit (boot area switching flag) stored in the extra area.

Note 2. This bit indicates the inverse of the value of the BTFLG bit (boot area switching flag) stored in the extra area.

39.6.2.13 Flash security flag monitoring register (FLSEC)

The FLSEC register monitors the information on the security flag and boot swap function settings in the extra area. The FLSEC register can be read by a 16-bit memory manipulation instruction.

Figure 39 - 21 Format of Flash Security Flag Monitoring Register (FLSEC)

Address: F00B0H
After reset: Undefined
R/W: R

Symbol	15	14	13	12	11	10	9	8
FLSEC	0	0	0	WRPR	0	SEPR	BTPR	BTFLG

7	6	5	4	3	2	1	0
0	0	0	0	SWPR	0	IFPR	IDEN

WRPR	Write-disabled flag
0	Writing is disabled.
1	Writing is enabled.

SEPR	Block erase-disabled flag
0	Block erasure is disabled.
1	Block erasure is enabled.

BTPR	Boot area rewrite-disabled flag
0	Rewriting of the boot area is disabled.
1	Rewriting of the boot area is enabled.

BTFLG	Boot area switching flag
0	The boot area is boot cluster 1.
1	The boot area is boot cluster 0.

SWPR	Changing of the flash memory read protection setting disabled flag
0	Changing of the flash memory read protection setting is disabled.
1	Changing of the flash memory read protection setting is enabled.

IFPR	Connection to the programmer and on-chip debugger disabled flag
0	Connection to the programmer and on-chip debugger is disabled.
1	Connection to the programmer and on-chip debugger is enabled.

IDEN	Programmer connection ID authentication enabled flag
0	ID authentication is enabled.
1	ID authentication is disabled.

39.6.2.14 Flash FSW monitoring register E (FLFSWE)

The FLFSWE register monitors the end block number specified for the flash memory shield area and whether the shield area is inside or outside the window range. After a reset or writing to the extra area, the values in the extra area are reflected in the FLFSWE register. For details on the flash shield window function, see **39.8 Flash Shield Window Function**. The FLFSWE register can be read by a 16-bit memory manipulation instruction.

Figure 39 - 22 Format of Flash FSW Monitoring Register E (FLFSWE)

Address: F00B4H
After reset: Undefined
R/W: R

Symbol	15	14	13	12	11	10	9	8
FLFSWE	FSWC	0	0	0	0	0	0	FSWE8

7	6	5	4	3	2	1	0
FSWE7	FSWE6	FSWE5	FSWE4	FSWE3	FSWE2	FSWE1	FSWE0

FSWC	Setting of the flash memory shield area
0	Inside shield mode The flash memory shield area is set inside the window range.
1	Outside shield mode The flash memory shield area is set outside the window range.

FSWE[8:0]	End block number of the flash memory shield area
—	End block number + 1[Note]

Note These bits show the value set in the extra area. The actual end block number is (the value of the FSWE[8:0] bits - 1). Though the end block number is specified for serial programming, (end block number + 1) is set in the extra area. For details, see **Table 39 - 12**.

39.6.2.15 Flash FSW monitoring register S (FLFSWS)

The FLFSWS register monitors the start block number of the flash memory shield area and whether rewriting of the flash shield window setting is disabled or enabled. After a reset or writing to the extra area, the values in the extra area are reflected in the FLFSWS register. For details on the flash shield window function, see **39.8 Flash Shield Window Function**. The FLFSWS register can be read by a 16-bit memory manipulation instruction.

Figure 39 - 23 Format of Flash FSW Monitoring Register S (FLFSWS)

Address: F00B2H
After reset: Undefined
R/W: R

Symbol	15	14	13	12	11	10	9	8
FLFSWS	FSPR	0	0	0	0	0	0	FSWS8

7	6	5	4	3	2	1	0
FSWS7	FSWS6	FSWS5	FSWS4	FSWS3	FSWS2	FSWS1	FSWS0

FSPR	Changing of the flash shield window setting disabled flag
0	Changing of the flash shield window setting is disabled.
1	Changing of the flash shield window setting is enabled.

FSWS[8:0]	Start block number of the flash memory shield area
—	Start block number

39.6.2.16 Data flash control register (DFLCTL)

The DFLCTL register enables or disables access to the data flash memory area. The DFLCTL register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 39 - 24 Format of Data Flash Control Register (DFLCTL)

Address: F0090H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	<0>
DFLCTL	0	0	0	0	0	0	0	DFLEN

DFLEN	Data flash memory area access control
0	Access to the data flash memory area is disabled.
1	Access to the data flash memory area is enabled.

39.6.2.17 Interrupt vector jump enable register (VECTCTRL)

The VECTCTRL register specifies the destination address of the jump in response to any interrupt which occurs during self-programming. The VECTCTRL register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 39 - 25 Format of Interrupt Vector Jump Enable Register (VECTCTRL)

Address: F00FFH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
VECTCTRL	0	0	0	0	0	0	0	VECTCTRL

VECTCTRL	Setting the interrupt branch destinations
0	Interrupt vector addresses in the ROM
1	Specified addresses in the RAM[Note]

Note A destination address in the RAM is specified by the FLSIVC1 and FLSIVC0 registers.
For details, see **39.6.2.18 Interrupt vector change registers 0 and 1 (FLSIVC0, FLSIVC1)**.

39.6.2.18 Interrupt vector change registers 0 and 1 (FLSIVC0, FLSIVC1)

The FLSIVC0 and FLSIVC1 registers specify the destination address of the jump in response to any interrupt which occurs during self-programming.

For details on how to handle an interrupt during self-programming, see **39.6.7 Interrupts in code flash memory programming mode**.

The FLSIVC0 and FLSIVC1 registers can be set by a 16-bit memory manipulation instruction.

The values of the FLSIVC0 and FLSIVC1 registers following a reset are 0000H and 000FH, respectively.

Figure 39 - 26 Format of Interrupt Vector Changer Registers 0 and 1 (FLSIVC0, FLSIVC1)

Address: F0480H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
FLSIVC0	FLSIV15	FLSIV14	FLSIV13	FLSIV12	FLSIV11	FLSIV10	FLSIV9	FLSIV8

7	6	5	4	3	2	1	0
FLSIV7	FLSIV6	FLSIV5	FLSIV4	FLSIV3	FLSIV2	FLSIV1	FLSIV0

Address: F0482H
After reset: 000FH
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
FLSIVC1	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	FLSIV19	FLSIV18	FLSIV17	FLSIV16

Caution Set the values of the 4 higher-order bits and 16 lower-order bits of the branch destination address in the FLSIVC1 and FLSIVC0 registers, respectively.

39.6.3 Setting the flash memory control mode

The flash memory has the following flash memory control modes.

– Code flash memory programming mode
  The code flash memory area and the extra area can be rewritten.
– Data flash memory programming mode
  The data flash memory area and the extra area can be rewritten.
– Non-programmable mode
  The flash memory (code flash memory area, data flash memory area, and extra area) cannot be rewritten.

To rewrite the flash memory, set the flash memory control mode to code flash memory programming mode or data flash memory programming mode. Setting each of the flash memory control modes requires executing the specific sequence for setting the flash protect command register (PFCMD) and flash programming mode control register (FLPMC).

Caution For handling of the data flash memory area, follow the procedure while access to the data flash memory is enabled (the value of the DFLEN bit of the DFLCTL register is 1).

39.6.3.1 Procedure for executing the specific sequence

Writing the required values to the flash protect command register (PFCMD) and the flash programming mode control register (FLPMC) by following steps <1> to <4> below enables the transitions to each of the flash memory control modes.

<1> Write A5H to the PFCMD register.
<2> Write the value to be set to the FLPMC register.
<3> Write the inverse of the value to be set to the FLPMC register.
<4> Write the value to be set to the FLPMC register.

- The specific sequence can only be executed while the value of the FLRST bit of the FLRST register is 0 and the flash memory sequencer is stopped.
- If writing to any other memory area or register is attempted in the intervals between steps <1> to <4> during execution of the specific sequence, a protection error occurs, writing to the specified register does not proceed, and the FPRERR flag of the flash status register (PFS) is set to 1. The FPRERR flag is cleared following a reset or when execution of the specific sequence is re-started.

39.6.3.2 Procedure for entry to the code flash memory programming mode

The procedure for entry to the code flash memory programming mode is given below.

<1> Write A5H to the PFCMD register.
<2> Write 02H to the FLPMC register (EEEMD = 0, FWEDIS = 0, FLSPM = 1).
<3> Write FDH to the FLPMC register (inverse of 02H).
<4> Write 02H to the FLPMC register (EEEMD = 0, FWEDIS = 0, FLSPM = 1).

39.6.3.3 Procedure for entry to the data flash memory programming mode

The procedure for entry to the data flash memory programming mode is given below.

<1> Write A5H to the PFCMD register.
<2> Write 10H to the FLPMC register (EEEMD = 1, FWEDIS = 0, FLSPM = 0).
<3> Write EFH to the FLPMC register (inverse of 10H).
<4> Write 10H to the FLPMC register (EEEMD = 1, FWEDIS = 0, FLSPM = 0).

39.6.3.4 Procedure for entry to the non-programmable mode

After executing the procedure for entry to the non-programmable mode from the code flash memory programming mode or data flash memory programming mode and waiting**Note**, reading from the target flash memory for the programming mode before the mode transition becomes possible.

Note Wait time: 10 μs

<When the interrupt vector has not been changed to addresses in the RAM>

The procedure for the transition in cases where branch destinations in response to interrupts are interrupt vector addresses in the ROM is given below.

<1> Write A5H to the PFCMD register.
<2> Write 08H to the FLPMC register (EEEMD = 0, FWEDIS = 1, FLSPM = 0).
<3> Write F7H to the FLPMC register (inverse of 08H).
<4> Write 08H to the FLPMC register (EEEMD = 0, FWEDIS = 1, FLSPM = 0).
<5> Reading from the target flash memory area becomes possible after waiting for 10 μs.

<When the interrupt vectors have been changed to addresses in the RAM>

The procedure for the transition in cases where branch destinations in response to interrupts are changed to specified addresses in the RAM is given below.

<1> Write A5H to the PFCMD register.
<2> Write 00H to the FLPMC register (EEEMD = 0, FWEDIS = 0, FLSPM = 0).
<3> Write FFH to the FLPMC register (inverse of 00H).
<4> Write 00H to the FLPMC register (EEEMD = 0, FWEDIS = 0, FLSPM = 0).
<5> Reading from the target flash memory area becomes possible after waiting for 10 μs.

39.6.4 Initializing the registers for use with the flash memory sequencer

The following registers can be initialized by setting the FLRST bit of the flash registers initialization register (FLRST) to 1.

Target registers: FLAPH, FLAPL, FLSEDH, FLSEDL, FLWH, FLWL, FLARS, FSSQ, FSSE

The procedure for clearing the target registers is given below.

<1> Set the FLRST bit.
<2> Use software code to wait for at least one cycle of the CPU clock.
<3> Clear the FLRST bit.

39.6.5 Setting the operating frequency of the flash memory sequencer

Set the value corresponding to the operating frequency of the CPU (1 MHz to 48 MHz) in the FSET[5:0] bits of the flash memory sequencer initial setting register (FSSET). When making the setting, round the CPU operating frequency value up to the nearest whole number.
(Example: When the CPU operating frequency is 4.5 MHz, set the bits to 05H.)

How to set the operating frequency of the flash memory sequencer is described below.

<1> Enter the code flash memory programming mode or data flash memory programming mode.
For the procedures for entry to each of these flash memory programming modes, see **39.6.3.1 Procedure for executing the specific sequence**, **39.6.3.2 Procedure for entry to the code flash memory programming mode**, and **39.6.3.3 Procedure for entry to the data flash memory programming mode**.
<2> After reading from the flash memory sequencer initial setting register (FSSET), set the TMSPMD and TMBTSEL bits to the same values as those read from the FSSET register and the FSET[5:0] bits to the value corresponding to the CPU operating frequency, respectively.

Caution When using the code/data flash memory area sequencer and the extra area sequencer to rewrite the code or data flash memory or the extra area, set the value corresponding to the CPU operating frequency in the FSET[5:0] bits of the FSSET register before doing so.
Note that if rewriting of any of these areas is attempted while the value corresponding to the CPU operating frequency is not correct, operation is undefined and written data are not guaranteed. Even if the values in the flash memory are as expected immediately after rewriting, retaining the values for any specified period is not guaranteed.

39.6.6 Rewriting the flash memory

39.6.6.1 Overview

The flash memory sequencer serves as a code/data flash memory area sequencer or an extra area sequencer. In the former role, it is used to rewrite the code flash memory area or data flash memory area, while in the latter role it is used to rewrite the extra area. To rewrite a given area, execute the corresponding commands for use with the sequencer.

39.6.6.2 Selecting the area to be rewritten

Select the code flash memory area, data flash memory area, or extra area for rewriting by using the flash area selection register (FLARS).

39.6.6.3 Commands for use with the code/data flash memory area sequencer

Use the commands exclusively used for the code/data flash memory area sequencer to rewrite the code or data flash memory area. To execute a command, set the target command in the SQMD[2:0] bits of the flash memory sequencer control register (FSSQ) and set the SQST bit to 1. The SQMD[2:0] bits and the SQST bit can be set simultaneously. For the commands exclusively for use with the code/data flash memory area sequencer, see **39.6.2.9 Flash memory sequencer control register (FSSQ)**.

39.6.6.4 Operations for rewriting the code flash memory area

To rewrite the code flash memory area, execute commands for use with the code/data flash memory area sequencer. Before starting to execute a command, set the data required for execution, such as specifying an address, addresses, or data, in the corresponding registers. Allocate the processing software to rewrite the code flash memory area in the RAM and execute it from the RAM.

Units for block erasure and for writing in rewriting of the code flash memory area

- Unit for blocks to be erased: 2 Kbytes
- Unit for writing: 4 bytes

<Handling the commands>

<1> Enter the code flash memory programming mode. For the procedure for entry to the code flash memory programming mode, see **39.6.3.1 Procedure for executing the specific sequence** and **39.6.3.2 Procedure for entry to the code flash memory programming mode**.

<2> Set the EXA bit of the FLARS register to 0 (code/data flash memory areas).

<3> Before executing each command, set the address data, write data, and command in the corresponding registers.

- Block erasure

  Set the block start address[Note 1] (example: 002000H) of the code flash memory to be erased in the FLAPH and FLAPL registers.

  Set the block end address[Note 1] (example: 0027FFH) of the code flash memory to be erased in the FLSEDH and FLSEDL registers.
- Writing

  Set the start address[Note 2] (example: 002000H) of the flash memory to be written in the FLAPH and FLAPL registers.

  Set 4-byte write data in the FLWH and FLWL registers.
- Blank checking

  Set the start address[Note 2] (example: 002000H) of the flash memory to be blank-checked in the FLAPH and FLAPL registers.

  Set the end address (example: 0027FFH) of the flash memory to be blank-checked in the FLSEDH and FLSEDL registers.

  When blank checking is only to be applied to one word (four bytes), set the FLSEDH and FLSEDL registers to the same values as those in the FLAPH and FLAPL registers.

<4> When the value of the command to be executed is set in the MDCH and SQMD[2:0] bits of the FSSQ register and the SQST bit is set to 1, the code/data flash memory area sequencer executes the specified command. The MDCH, SQST, and SQMD[2:0] bits can be set simultaneously. If these bits are set simultaneously, the FSSQ register is set to the following values.

- Block erasure: 84H
- Writing: 81H
- Blank checking of the code flash memory area: 83H

<5> Wait until the command for use with the code/data flash memory area sequencer is complete. For details on the procedure for waiting for the completion of command execution, see the section titled “**Procedure for checking the completion of commands for use with the code/data flash memory area sequencer**” in **39.6.6.9 Procedures for checking completion of the commands for use with the flash memory sequencer in the respective areas**.

<6> Processing after executing a command

<Continuing command processing>

The same command or another command can be executed by continuously updating the address data and write data in step <3> with the state remaining in code flash memory programming mode.

<Completing command processing>

Switch to the non-programmable mode. For the procedure for switching to the non-programmable mode, see **39.6.3.1 Procedure for executing the specific sequence** and **39.6.3.4 Procedure for entry to the non-programmable mode**.

Note 1. The code flash memory area blocks can be erased in units of 2 Kbytes. Specify the erase addresses (start address and end address) so that all blocks to be erased are included. For the relationship between addresses and block numbers, see **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**.

Note 2. The code flash memory area can be written and blank-checked in units of 4 bytes. Therefore, set the two lower-order bits of the FLAPL register for specifying the address to 00B (an integer of 4).

39.6.6.5 Operations for rewriting the data flash memory area

To rewrite the data flash memory area, execute commands for use with the code/data flash memory area sequencer. Before starting to execute a command, set the data required for execution, such as specifying an address, addresses, or data, in the corresponding registers.

Units for block erasure and for writing in rewriting of the data flash memory area

– Unit for blocks to be erased: 256 bytes
– Unit for writing: 1 byte

<Handling the commands>

<1> Enter the data flash memory programming mode. For the procedure for entry to the data flash memory programming mode, see **39.6.3.1 Procedure for executing the specific sequence** and **39.6.3.3 Procedure for entry to the data flash memory programming mode**.

<2> Set the EXA bit of the FLARS register to 0 (code/data flash memory areas).

<3> Before executing each command, set the address data, write data, and command in the corresponding registers.

- Block erasure
  Set the block start address[Note] (example: 0F1100H) of the data flash memory to be erased in the FLAPH and FLAPL registers.
  Set the block end address[Note] (example: 0F11FFH) of the data flash memory to be erased in the FLSEDH and FLSEDL registers.
- Writing
  Set the start address (example: 0F1101H) of the flash memory to be written in the FLAPH and FLAPL registers.
  Set the write data in the 8 lower-order bits of the FLWL register.
- Blank checking:
  Set the start address (example: 0F1100H) of the flash memory to be blank-checked in the FLAPH and FLAPL registers.
  Set the end address (example: 0F11FFH) of the flash memory to be blank-checked in the FLSEDH and FLSEDL registers.
  When blank checking is only to be applied to one byte, set the FLSEDH and FLSEDL registers to the same values as those in the FLAPH and FLAPL registers.

<4> When the value of the command to be executed is set in the MDCH and SQMD[2:0] bits of the FSSQ register and the SQST bit is set to 1, the code/data flash memory area sequencer executes the specified command. The MDCH, SQST, and SQMD[2:0] bits can be set simultaneously. If these bits are set simultaneously, the FSSQ register is set to the following values.

- Block erasure: 84H
- Writing: 81H
- Blank checking of the data flash memory area: 8BH

<5> Wait until the command for use with the code/data flash memory area sequencer is complete. For details on the procedure for waiting for the completion of command execution, see the section titled “**Procedure for checking the completion of commands for use with the code/data flash memory area sequencer**” in **39.6.6.9 Procedures for checking completion of the commands for use with the flash memory sequencer in the respective areas**.

<6> Processing after executing a command

<Continuing command processing>

The same command or another command can be executed by continuously updating the address data and write data in step <3> with the state remaining in data flash memory programming mode.

<Completing command processing>

Switch to the non-programmable mode. For the procedure for switching to the non-programmable mode, see **39.6.3.1 Procedure for executing the specific sequence** and **39.6.3.4 Procedure for entry to the non-programmable mode**.

Note The data flash memory area blocks can be erased in units of 256 bytes. Therefore, set the 8 lower-order bits of the FLAPL register for specifying the start address to 0000 0000B (an integer of 256). Also set the 8 lower-order bits of the FLSEDL register for specifying the end address to 1111 1111B.

39.6.6.6 Commands for use with the extra area sequencer

Use the commands for use with the extra area sequencer to rewrite the settings of the flash shield window, flash read protection, flash security, and boot swap function in the extra area. To execute a command, set the command to be executed in the ESQMD[3:0] bits of the flash extra area sequencer control register (FSSE), and then set the ESQST bit to 1. The ESQMD[3:0] bits and the ESQST bit can be set simultaneously.
Allocate the extra area sequencer command processing software in the RAM and execute it from the RAM.

39.6.6.7 Operations for rewriting the extra area

To rewrite the extra area, enter the code flash memory programming mode and then execute commands for use with the extra area sequencer. Before starting to execute a command, set the data required for executing each command in the corresponding registers.

<Handling the commands>

<1> Enter the code flash memory programming mode. For the procedure for entry to the code flash memory programming mode, see **39.6.3.1 Procedure for executing the specific sequence** and **39.6.3.2 Procedure for entry to the code flash memory programming mode**.

<2> Set the EXA bit of the FLARS register to 1 (extra area).

<3> Before executing a command, set 4-byte data in the FLWH and FLWL registers. Each bit of the combined FLW[31:0] bits of the FLWH and FLWL registers corresponds to EX bits 31 to 0 of the target extra area data. For details on data to be set for each command, see **39.6.6.8 Data to be set for the commands for use with the extra area sequencer**.

<4> When the value of the command to be executed is set in the ESQMD[3:0] bits of the FSSE register and the ESQST bit is set to 1, the extra area sequencer executes the specified command. The ESQMD[3:0] bits and the ESQST bit can be set simultaneously. If these bits are set simultaneously, the FSSE register is set to the following values.

- Write data to the flash shield window setting area: 81H
- Write data to the flash read protection setting area: 86H
- Write data to the security flag and boot swap function setting area: 87H

<5> Wait until the command for use with the extra area sequencer is complete. For details on the procedure for waiting for the completion of command execution, see the section titled "**Procedure for checking the completion of commands for use with the extra area sequencer**" in **39.6.6.9 Procedures for checking completion of the commands for use with the flash memory sequencer in the respective areas**.

<6> Processing after executing a command

<Continuing command processing>

The same command or another command can be executed by continuously updating the FLWH and FLWL register data to be set in the extra area in step <3> with the state remaining in code flash memory programming mode.

<Completing command processing>

Switch to the non-programmable mode. For the procedure for switching to the non-programmable mode, see **39.6.3.1 Procedure for executing the specific sequence** and **39.6.3.4 Procedure for entry to the non-programmable mode**.

39.6.6.8 Data to be set for the commands for use with the extra area sequencer

Writing to the extra area proceeds per 4 bytes.

Each command for use with the extra area sequencer writes the data set in the combined FLW[31:0] bits of the FLWH and FLWL registers to EX bits 31 to 0 in the extra area corresponding to the given command.

1. Write to the flash shield window setting area

   Set the data in the FLWH and FLWL registers to the flash shield window setting area.

EX bit 31	EX bit 30	EX bit 29	EX bit 28	EX bit 27	EX bit 26	EX bit 25	EX bit 24
FSWC	1	1	1	1	1	1	FSWE8

EX bit 23	EX bit 22	EX bit 21	EX bit 20	EX bit 19	EX bit 18	EX bit 17	EX bit 16
FSWE7	FSWE6	FSWE5	FSWE4	FSWE3	FSWE2	FSWE1	FSWE0

EX bit 15	EX bit 14	EX bit 13	EX bit 12	EX bit 11	EX bit 10	EX bit 9	EX bit 8
FSPR	1	1	1	1	1	1	FSWS8

EX bit 7	EX bit 6	EX bit 5	EX bit 4	EX bit 3	EX bit 2	EX bit 1	EX bit 0
FSWS7	FSWS6	FSWS5	FSWS4	FSWS3	FSWS2	FSWS1	FSWS0

Bit Name	Setting
FSWC	Specifies the range of the flash memory shield area. 0: Flash memory shield area: Inside the window range 1: Flash memory shield area: Outside the window range (default)
FSPR	Specifies whether to enable or disable changing of the flash shield window setting area. 0: Changing of the flash shield window setting area is disabled. 1: Changing of the flash shield window setting area is enabled (default).
FSWE[8:0]	Flash shield window end block setting area Specify the block number (end block number + 1).**Note**
FSWS[8:0]	Flash shield window start block setting area Specify the start block number.**Note**

Note For the relationship between addresses and block numbers, see **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**.

Caution **The value of the FSPR bit can be changed from 0 (disabling) to 1 (enabling) by executing the chip erase command of the dedicated flash memory programmer or the Security Release command for use in serial programming mode with the entire flash memory erased.**
Note that if either of the disabling settings listed below is made, executing the chip erase command of the dedicated flash memory programmer or the Security Release command for use in serial programming mode is not possible.
- **SEPR = 0 (block erasure is disabled)**
- **BTPR = 0 (rewriting of the boot area is disabled)**

Moreover, setting the FSPR bit to 1 (enabling) is not possible because a command cannot be transmitted when connection in serial programming mode is not available due to the setting for disabling connection to the programmer and on-chip debugger or for enabling programmer connection ID authentication having been made.

2. Write to the flash read protection setting area
   Set the data in the FLWH and FLWL registers to the flash read protection setting area.

EX bit 31	EX bit 30	EX bit 29	EX bit 28	EX bit 27	EX bit 26	EX bit 25	EX bit 24
SWPR	1	1	1	1	1	1	UPAddr8

EX bit 23	EX bit 22	EX bit 21	EX bit 20	EX bit 19	EX bit 18	EX bit 17	EX bit 16
UPAddr7	UPAddr6	UPAddr5	UPAddr4	UPAddr3	UPAddr2	UPAddr1	UPAddr0

EX bit 15	EX bit 14	EX bit 13	EX bit 12	EX bit 11	EX bit 10	EX bit 9	EX bit 8
1	1	1	1	1	1	1	LOWAddr8

EX bit 7	EX bit 6	EX bit 5	EX bit 4	EX bit 3	EX bit 2	EX bit 1	EX bit 0
LOWAddr7	LOWAddr6	LOWAddr5	LOWAddr4	LOWAddr3	LOWAddr2	LOWAddr1	LOWAddr0

Bit Name	Setting
SWPR	Specifies whether to enable or disable changing of the flash read protection setting area. 0: Changing of the flash read protection setting area is disabled. 1: Changing of the flash read protection setting area is enabled (default).
UPAddr [8:0]	Flash read protection end block setting area Specify the end block number.**Note**
LOWAddr [8:0]	Flash read protection start block setting area Specify the start block number.**Note**

Note For the relationship between addresses and block numbers, see **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**. The flash read protection setting area cannot be read after a reset is released.

Caution **The value of the SWPR bit can be changed from 0 (disabling) to 1 (enabling) by executing the chip erase command of the dedicated flash memory programmer or the Security Release command for use in serial programming mode with the entire flash memory erased.**
Note that if either of the disabling settings listed below is made, executing the chip erase command of the dedicated flash memory programmer or the Security Release command for use in serial programming mode is not possible.
- **SEPR = 0 (block erasure is disabled)**
- **BTPR = 0 (rewriting of the boot area is disabled)**

Moreover, setting the SWPR bit to 1 (enabling) is not possible because a command cannot be transmitted when connection in serial programming mode is not available due to the setting for disabling connection to the programmer and on-chip debugger or for enabling programmer connection ID authentication having been made.

3. Write to the security flag and boot swap function setting area
   Set the data in the FLWH and FLWL registers to the security flag and boot swap function setting area. For details of the security settings, see **39.9 Security Settings**.

EX bit 31	EX bit 30	EX bit 29	EX bit 28	EX bit 27	EX bit 26	EX bit 25	EX bit 24
1	1	1	1	1	1	1	1

EX bit 23	EX bit 22	EX bit 21	EX bit 20	EX bit 19	EX bit 18	EX bit 17	EX bit 16
1	1	1	1	1	1	1	1

EX bit 15	EX bit 14	EX bit 13	EX bit 12	EX bit 11	EX bit 10	EX bit 9	EX bit 8
1	1	1	WRPR	1	SEPR	BTPR	BTFLG

EX bit 7	EX bit 6	EX bit 5	EX bit 4	EX bit 3	EX bit 2	EX bit 1	EX bit 0
1	1	1	1	1	IFPR	1	IDEN

Bit Name	Setting
WRPR	Specifies whether to enable or disable writing in serial programming mode. Writing in serial programming mode is disabled. 0: Writing in serial programming mode is disabled. 1: Writing in serial programming mode is enabled (default).
SEPR	Specifies whether to enable or disable block erasure. Block erasure in serial programming mode is disabled. 0: Block erasure in serial programming mode is disabled. 1: Block erasure in serial programming mode is enabled (default).
BTPR	Specifies whether to enable or disable rewriting of the boot area. Rewriting of the boot area and boot swapping are disabled. 0: Rewriting of the boot area and boot swapping are disabled. 1: Rewriting of the boot area and boot swapping are enabled (default).
BTFLG	Specifies the boot area when the TMSPMD bit in the FSSET register is 0. 0: Boot area: Boot cluster 1 1: Boot area: Boot cluster 0 (default)
IFPR	Specifies whether to enable or disable connection to the programmer and on-chip debugger. The serial programming mode and connection to the on-chip debugger are disabled. 0: Serial programming mode and connection to the on-chip debugger are disabled. 1: Serial programming mode and connection to the on-chip debugger are enabled (default).
IDEN	Specifies whether to enable or disable programmer connection ID authentication. ID authentication for connection in serial programming mode proceeds. 0: ID authentication for connection in serial programming mode is enabled. 1: ID authentication for connection in serial programming mode is disabled (default).

Caution 1. When changing the value of the BTFLG bit, set all other bits to 1.

Caution 2. When changing the values of security flags other than the BTFLG bit to 0 (disabling), read the register first and set the BTFLG bit to the same value as was read, and set the other bits to 1.

(Cautions are listed on the next page.)

Caution 3. **The value of the WRPR bit can be changed from 0 (disabling) to 1 (enabling) by executing the chip erase command of the dedicated flash memory programmer or the Security Release command for use in serial programming mode with the entire flash memory erased.**
Note that if either of the disabling settings listed below is made, executing the chip erase command of the dedicated flash memory programmer or the Security Release command for use in serial programming mode is not possible.
- **SEPR = 0 (block erasure is disabled)**
- **BTPR = 0 (rewriting of the boot area is disabled)**

Moreover, setting the WRPR bit to 1 (enabling) is not possible because a command cannot be transmitted when connection in serial programming mode is not available due to the setting for disabling connection to the programmer and on-chip debugger or for enabling programmer connection ID authentication having been made.

Caution 4. **Restoring the value of any among the SEPR, BTPR, IFPR, and IDEN bits to 1 after having set it to 0 is not possible.**

39.6.6.9 Procedures for checking completion of the commands for use with the flash memory sequencer in the respective areas

Perform the following procedure for checking the completion of commands when terminating an activated command for use with the code/data flash memory area sequencer and the extra area sequencer.

- Procedure for checking the completion of commands for use with the code/data flash memory area sequencer
  1. Activate a command for use with the code/data flash memory area sequencer, and then wait until the SQEND flag of the FSASTH register is set to 1.
  2. Confirm that the SQEND flag of the FSASTH register is set to 1, and then clear the SQST bit of the FSSQ register.
  3. Wait until the SQEND flag of the FSASTH register is cleared. When the SQEND flag is cleared, the command is completed and the sequencer stops.

- Procedure for checking the completion of commands for use with the extra area sequencer
  1. Activate a command for use with the extra area sequencer, and then wait until the ESQEND flag of the FSASTH register is set to 1.
  2. Confirm that the ESQEND flag of the FSASTH register is set to 1, and then clear the ESQST bit of the FSSE register.
  3. Wait until the ESQEND flag of the FSASTH register is cleared. When the ESQEND flag is cleared, the command is completed and the sequencer stops.

39.6.6.10 Procedure for forcibly terminating a command for use with the code/data flash memory area sequencer

A command for use with the code/data flash memory area sequencer can be forcibly terminated while it is in progress. Note that no command for use with the extra area sequencer can be forcibly terminated while it is in progress.

<Procedure for forcible termination>

1. Set the FSSTP bit of the FSSQ register to 1 after a command for use with the code/data flash memory area sequencer is activated until the SQST bit of the FSSQ register in step <2> is cleared.
2. Confirm that the SQEND flag of the FSASTH register is set to 1, and then clear the SQST and FSSTP bits of the FSSQ register.
3. Wait until the SQEND flag of the FSASTH register is automatically cleared. When the SQEND flag is cleared, the forcible termination is completed.

39.6.7 Interrupts in code flash memory programming mode

39.6.7.1 Overview

When an interrupt occurs, reference to the interrupt vector in the ROM proceeds, and execution branches to the interrupt processing that is allocated in the up to 64-Kbyte ROM space in accord with the address in the interrupt vector (16 bits), from which the interrupt processing is executed. However, in code flash memory programming mode, where the code flash memory and extra area can be rewritten, referring to the ROM is not possible. As a result, the interrupt processing cannot be executed in this way.
Note that even when referring to the ROM is not possible, the interrupt processing can be executed by changing the branch destinations of the interrupts. Specifically, changing the branch destinations of all interrupts to specified addresses in the RAM enables the execution of interrupt processing from the RAM instead of by using the interrupt vector and interrupt processing in the ROM.

39.6.7.2 Operation to change the branch destinations of the interrupts

To change the branch destinations of the interrupts, set the interrupt vector change registers 0, 1 (FLSIVC1, FLSIVC0) and interrupt vector jump enable register (VECTCTRL) so that execution branches to an address in the RAM following an interrupt. This enables execution of the interrupt processing from the RAM without reference to the interrupt vector in the ROM, even when an interrupt occurs in code flash memory programming mode.
The FLSIVC1 and FLSIVC0 registers are used to specify the branch destination address in response to interrupts which occur during rewriting of the code flash memory or extra area. Set the values of the 16 lower-order bits and 4 higher-order bits of the branch destination address in the FLSIVC0 and FLSIVC1 registers, respectively.

Settings for control of the branch destination of interrupts that occur during self-programming are as follows.

- For branching to vector addresses in the ROM: VECTCTRL = 0 or FWEDIS of FLPMC = 1
- For branching to a RAM address: VECTCTRL = 1 (with FWEDIS of FLPMC = 0)

Caution 1. The user must determine the type of interrupt by checking the interrupt flags. Therefore, the interrupt flags are not automatically cleared after the VECTCTRL register has been set.
Caution 2. A changed interrupt branch destination in the ROM is not specifiable.
Caution 3. The change to the interrupt branch destination made by the VECTCTRL register is only effective during self-programming.
Caution 4. Disable interrupts while changing the interrupt branch destination to an address in the RAM.

39.6.7.3 Operation to change the interrupt branch destination

To specify interrupt processing from the RAM, update the FLSIVC1, FLSIVC0, and VECTCTRL registers while the value of the FWEDIS bit of the flash programming mode control register (FLPMC) is 0. Execute the specific sequence to handle the FWEDIS bit of the FLPMC register, and set the FLSIVC1, FLSIVC0, and VECTCTRL registers. The interrupt branch destination is thus changed to an address in RAM.

<When changing the interrupt branch destinations to an address in RAM>

The following describes operation for changing all interrupt branch destinations to a specified address in the RAM.

- Save the setting for enabling or disabling interrupts that was in place before starting this procedure and make the setting to disable interrupts.
- Execute the specific sequence and set the FWEDIS bit of the FLPMC register to 0.
  - <1> Write A5H to the PFCMD register.
  - <2> Write 00H to the FLPMC register (EEEMD = 0, FWEDIS = 0, FLSPM = 0).
  - <3> Write FFH to the FLPMC register (inverse of 00H).
  - <4> Write 00H to the FLPMC register (EEEMD = 0, FWEDIS = 0, FLSPM = 0).
- Specify an address in RAM in the FLSIVC1 and FLSIVC0 registers.
- Set the VECTCTRL register to 01H to set the RAM address as the branch destination for interrupts.
- Restore the saved setting for enabling or disabling interrupts.

Caution 1. Retain the value 0 in the FWEDIS bit as long as interrupt processing from the RAM remains specified.

Caution 2. Do not set the interrupt branch destination in the saddr space at addresses from FFE20H to FFEFFH.

Caution 3. When instructions are being executed from the RAM area and RAM parity error resets are enabled (RPERDIS = 0), initialize the RAM area where data access is to proceed + 10 bytes.

<When returning the interrupt branch destinations from the RAM address to the vectors in the ROM>

The following describes operation for returning the interrupt branch destinations to the addresses indicated by the interrupt vectors in the ROM (default state).

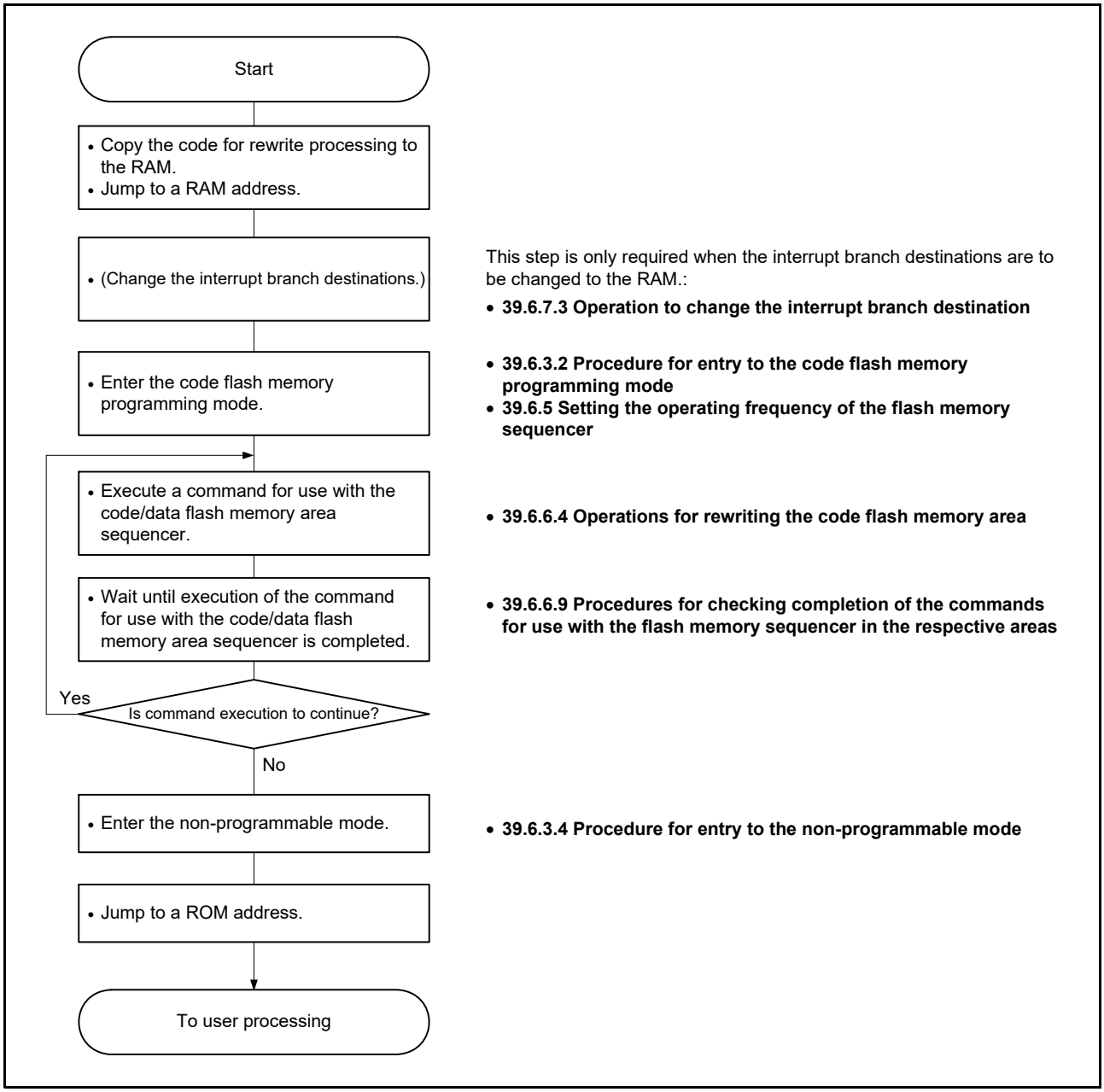
- Save the setting for enabling or disabling interrupts that was in place before starting this procedure and make the setting to disable interrupts.
- Execute the specific sequence and set the FWEDIS bit of the FLPMC register to 1.
  - <1> Write A5H to the PFCMD register.
  - <2> Write 08H to the FLPMC register (EEEMD = 0, FWEDIS = 1, FLSPM = 0).
  - <3> Write F7H to the FLPMC register (inverse of 08H).
  - <4> Write 08H to the FLPMC register (EEEMD = 0, FWEDIS = 1, FLSPM = 0).
- Set the VECTCTRL register to 00H to set the vector addresses in the ROM as the interrupt branch destinations.
- Restore the saved setting for enabling or disabling interrupts.

39.6.8 Example of executing the commands to rewrite the flash memory areas

39.6.8.1 Example of executing the commands to rewrite the code flash memory area

Figure 39 - 27 shows the flow of executing the commands to rewrite the code flash memory area.

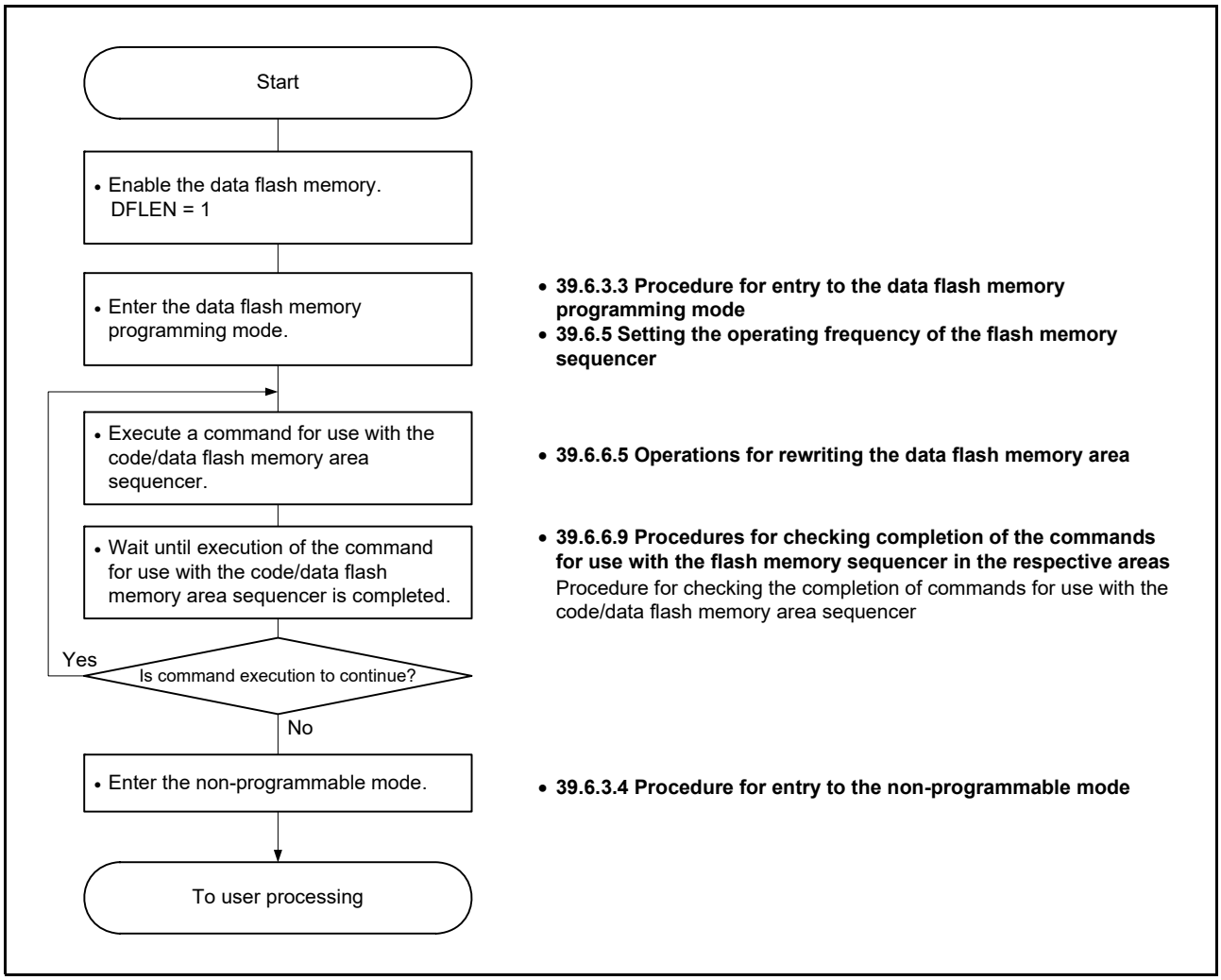
Figure 39 - 27 Flow of Executing the Commands to Rewrite the Code Flash Memory Area



39.6.8.2 Example of executing the commands to rewrite the data flash memory area

Figure 39 - 28 shows the flow of executing the commands to rewrite the data flash memory area.

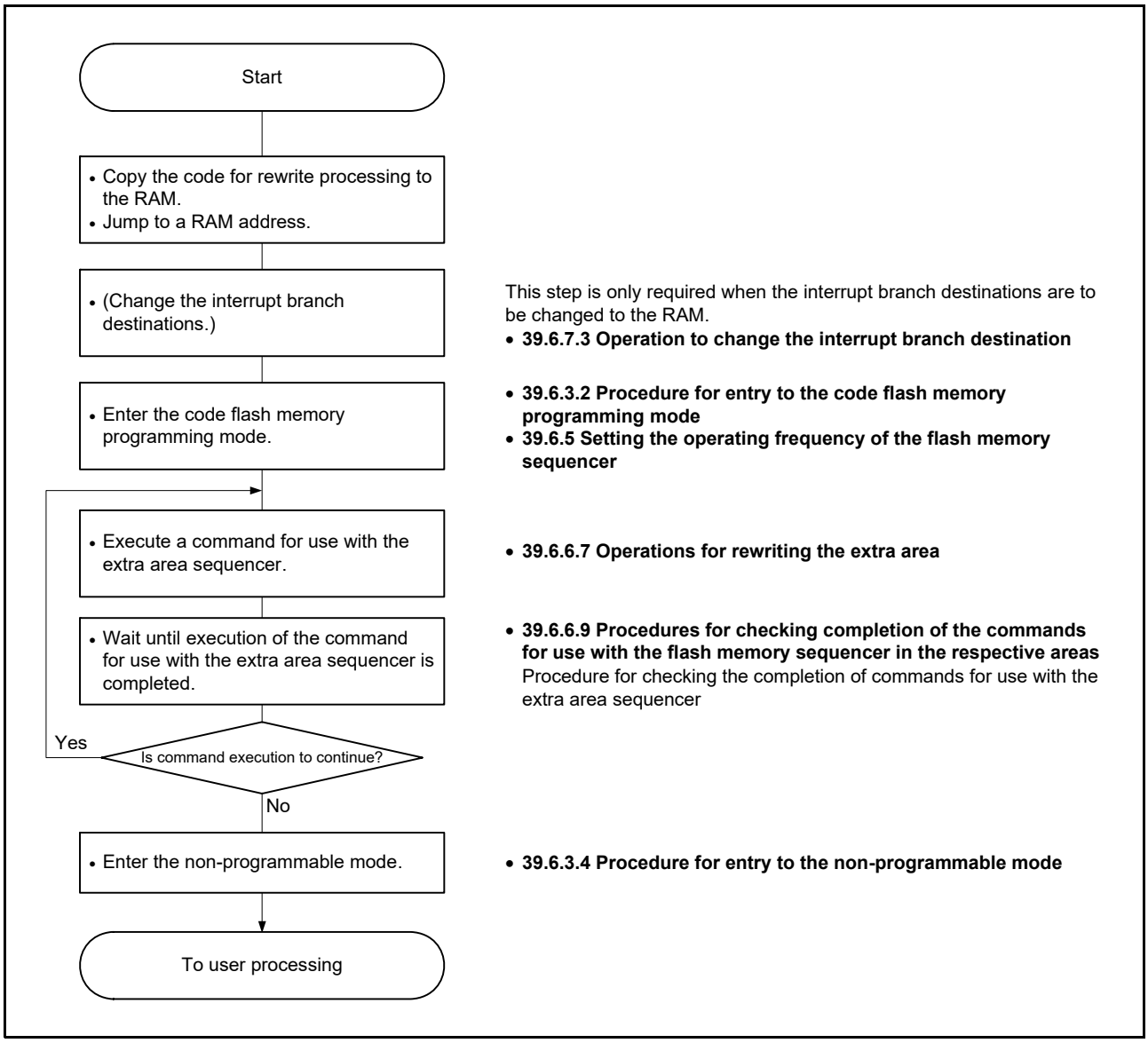
Figure 39 - 28 Flow of Executing the Commands to Rewrite the Data Flash Memory Area



39.6.8.3 Example of executing the commands to rewrite the extra area

Figure 39 - 29 shows the flow of executing the commands to rewrite the extra area.

Figure 39 - 29 Flow of Executing the Commands to Rewrite the Extra Area



39.6.9 Notes on self-programming

1. Rewriting the code flash memory or extra area
   To rewrite the code flash memory or extra area, place the code or values in the RAM.

2. Precondition for manipulating the data flash memory area
   Before manipulating the data flash memory area, set the DFLEN bit of the data flash control register (DFLCTL) to 1 (enabling access to the data flash memory).

3. Execution of programs during rewriting of the flash memory
   The flash memory sequencer is used to control rewriting of the flash memory during self-programming.
   In the flash memory control modes where rewriting of the flash memory is enabled, reference to the flash memory to be manipulated is not possible.
   - In code flash memory programming mode, reference to the code flash memory is not possible. Accordingly, in code flash memory programming mode, copy the user program that is to be executed from the ROM (code flash memory) and its data for reference to the RAM in advance so that the program can be executed and reference to the data in the RAM is possible.
   - In data flash memory programming mode, reference to the data flash memory is not possible. Accordingly, in data flash memory programming mode, copy data that are for reference to the RAM in advance so that reference to the data in the RAM is possible.

4. Specifying the range of unavailable area
   Specify the range of blank checking and block erasure within the range of code flash memory area or data flash memory area. Do not specify any unavailable area or both the code flash memory area and data flash memory area including an unavailable area.

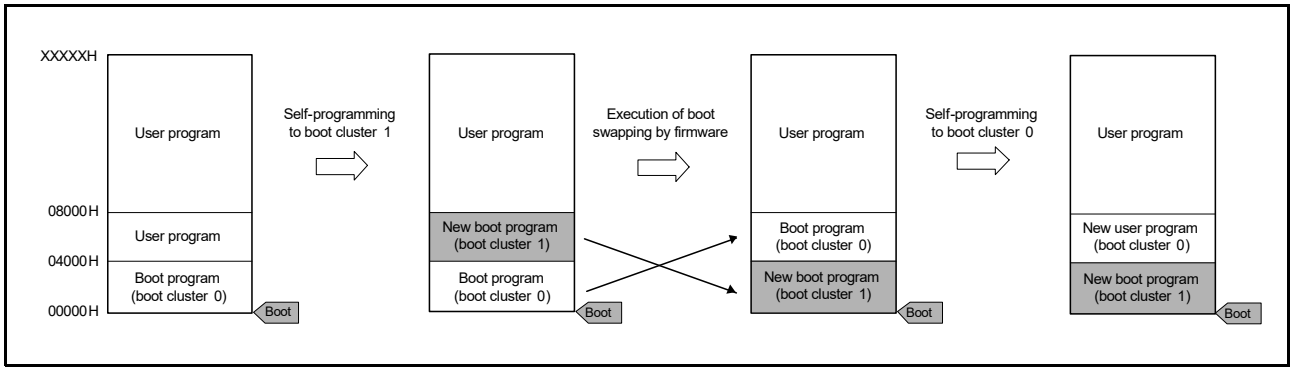
39.7 Boot Swap Function

The boot area consists of the vector table area, CALLT table area, option bytes area, setting areas for the on-chip debug security ID and programmer connection ID, and program area. The required settings and information respectively for starting the program and for connecting an on-chip debugger and programmer are stored in these areas. Thus, if rewriting in the boot area through self-programming failed by temporary power failure or other reasons, restarting a program by resetting or overwriting is disabled due to data destruction in the boot area.
The boot swap function is used to avoid this problem.
When the boot area is set to boot cluster 0, write a new boot program to boot cluster 1 through self-programming before erasing boot cluster 0**Note**. After the boot program has been successfully written to boot cluster 1, change the boot area from boot cluster 0 to boot cluster 1 by self-programming to use boot cluster 1 as the boot area. After that, erase the data in boot cluster 0, and then write data to boot cluster 0.
As a result, even if a power failure occurs while the area is being rewritten, the program is executed correctly because it is booted from boot cluster 1 when the program is reset and started next.
A boot cluster is a 16-Kbyte area.

Note Allocate the addresses in the new boot program as follows.

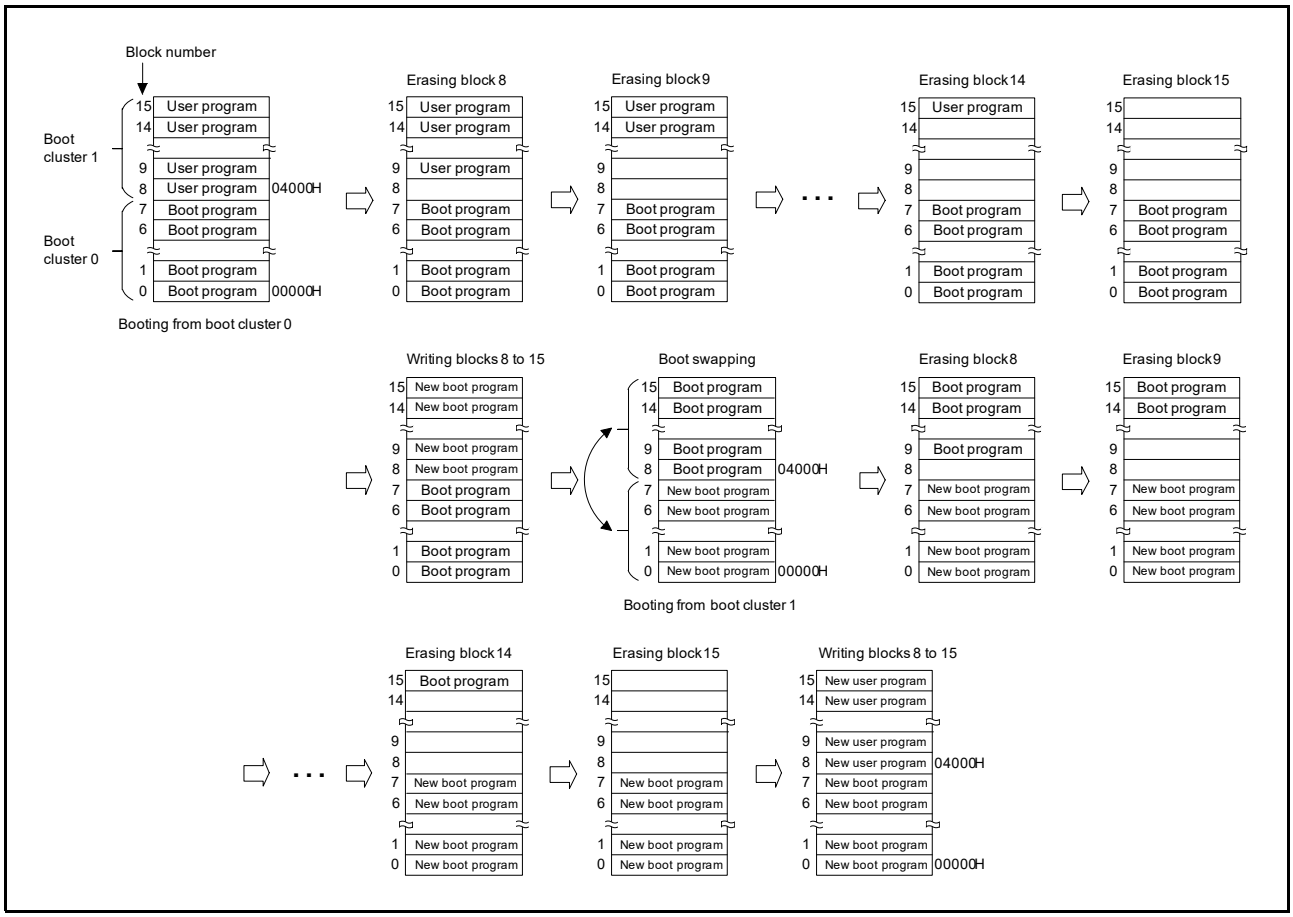
- 04000H to 0407FH (128 bytes): Vector table area
- 04080H to 040BFH (64 bytes): CALLT table area
- 040C0H to 040C3H (4 bytes): Option bytes area
- 040C4H to 040CDH (10 bytes): On-chip debug security ID setting area

Figure 39 - 30 Boot Swap Function



In the example of the above figure, the boot areas before and after boot swapping are as follows.
Boot cluster 0: Boot area before boot swapping
Boot cluster 1: Boot area after boot swapping

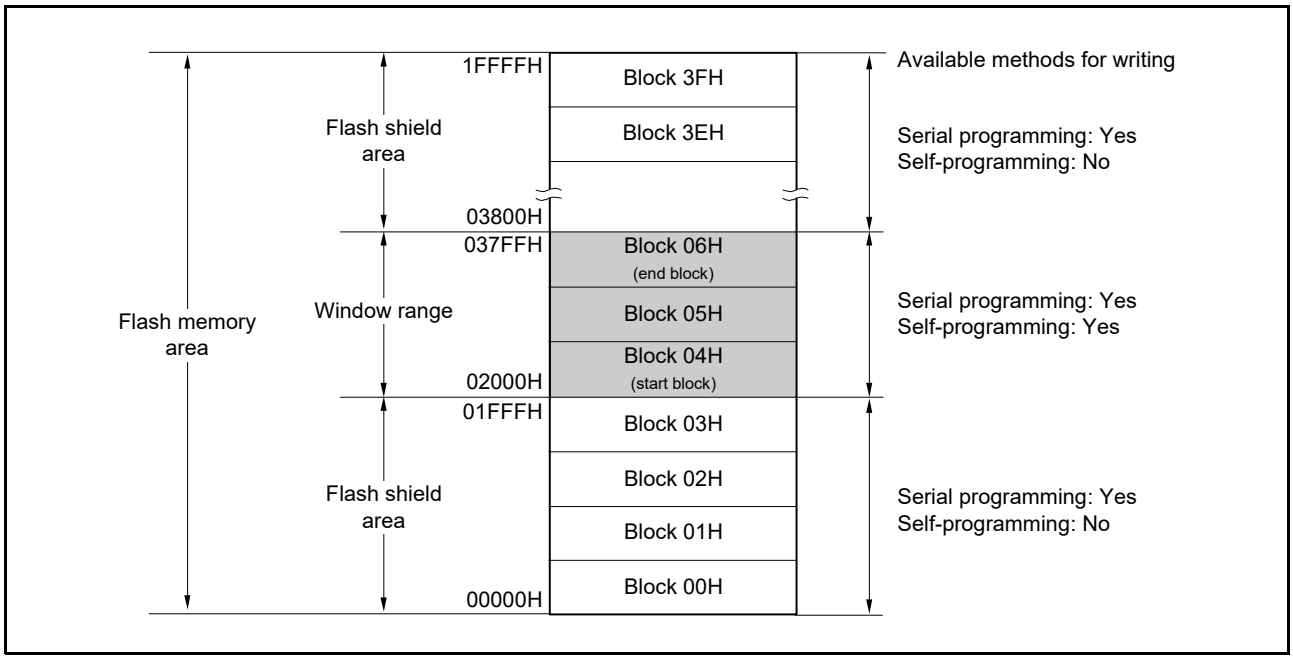
Figure 39 - 31 Example of Executing Boot Swapping



39.8 Flash Shield Window Function

The flash shield window function is provided as a security function which disables writing to and erasing of the selected flash memory shield area. This function is only effective for self-programming.
The flash memory shield area is selectable as either the area inside or areas outside the range specified as the window. The window range is set by specifying the blocks where it starts and ends. The flash memory shield area can be set or changed during both serial programming and self-programming.
Writing to and erasing of the flash memory shield area are disabled during self-programming. During serial programming, however, the flash memory shield area can also be written and erased.

Figure 39 - 32 Flash Shield Window Setting Example (Target Devices: R7F101GLG, Start Block Number: 04H, End Block Number: 06H, FSWC: 1)



Caution 1. If the non-programmable area of the boot area overlaps with the flash shield window range, disabling of rewriting the boot area takes priority.

Caution 2. The flash shield window can only be used for the code flash memory (and is not supported for the data flash memory).

Table 39 - 12 Relationship between Flash Shield Window Function Setting/Change Methods and Commands

Programming Condition	Window Range Setting/Change Method	Command to Be Executed	
		Block Erase	Write
Self-programming	Specify the window start block number and the (end block number + 1) block number (following the end block) in the flash shield window setting area using the self-programming.	Block erasure is not possible inside the flash memory shield area.	Writing is not possible inside the flash memory shield area.
Serial programming	Specify the start block number in the window range and the end block number in the window range on GUI of dedicated flash memory programmer, etc.	Block erasure is also possible inside the flash memory shield area.	Writing is also possible inside the flash memory shield area.

39.9 Security Settings

The RL78 microcontroller supports 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 by serial programming or self-programming.

- Disabling block erasure
  Execution of the block erase command for a specific block in the code flash memory is prohibited during serial programming. However, blocks can be erased by means of self-programming.
- Disabling writing
  Execution of the write command for entire blocks in the code flash memory is prohibited during serial programming. However, blocks can be written by means of self-programming.
  After the setting to prohibit writing has been made, releasing the setting by the Security Release command is enabled by a reset.
- Disabling rewriting the boot area
  Execution of the block erase command and write command for the boot area (00000H to 03FFFH) in the code flash memory is prohibited.
- Disabling connection to the programmer and on-chip debugger
  Connection to a dedicated flash memory programmer and on-chip debugger is prohibited.
  A dedicated flash memory programmer and on-chip debugger cannot be used to manipulate the flash memory.
- Enabling programmer connection ID authentication
  Authentication for an arbitrary 10-byte ID code is enabled when connecting to a dedicated flash memory programmer. The 10-byte ID area is 000C4H to 000CDH[Note]. If the ID does not match when using serial programming, the dedicated flash memory programmer cannot be used to manipulate the flash memory.

Block erasure, writing, and rewriting the boot area are enabled by the default setting at the time of shipment. Security can be set by serial programming and self-programming. Each security setting can be used in combination.
Table 39 - 13 shows the relationship between the erase and write commands when the RL78 microcontroller security function is enabled.

Note The 10-byte ID code area for the programmer connection ID is shared with the security ID code for on-chip debugging.

Caution **The security function of the dedicated flash programmer does not support self-programming.**

Remark To prohibit writing and erasure during self-programming, use the flash shield window function (see **39.8 Flash Shield Window Function** for detail).

Table 39 - 13 Relationship between Enabling the Security Function and Commands

1. During serial programming

Valid Security	Command to Be Executed	
	Block Erase	Write
Prohibition of block erasure	Blocks cannot be erased.	Data can be written.Note
Prohibition of writing	Blocks can be erased.	Data cannot be written.
Prohibition of rewriting the boot area	The boot area cannot be erased	The boot area cannot be written.
Prohibition of connection to the programmer and on-chip debugger	Blocks cannot be erased.	Data cannot be written.
Success in authentication with programmer connection ID authentication enabled	Blocks can be erased.	Data can be written.
Failure in authentication with programmer connection ID authentication enabled	Blocks cannot be erased.	Data cannot be written.

Note Confirm that no data have been written to the write area. If data in the area has not been erased, do not attempt further writing of data because data cannot be erased after the setting to prohibit block erasure has been made.

2. During self-programming

Valid Security	Command to Be Executed	
	Block Erase	Write
Prohibition of block erasure	Blocks can be erased.	Data can be written.
Prohibition of writing		
Prohibition of rewriting the boot area	The boot area cannot be erased.	The boot area cannot be written.
Prohibition of connection to the programmer and on-chip debugger	Blocks can be erased.	Data can be written.
Programmer connection ID authentication enabled	Blocks can be erased.	Data can be written.

Remark To prohibit writing and erasure during self-programming, use the flash shield window function (see **39.8 Flash Shield Window Function** for detail).

Table 39 - 14 Setting Security in Each Programming Mode

1. During serial programming

Security	Security Setting	Disabling the Security Setting
Prohibition of block erasure	Set via GUI of dedicated flash memory programmer, etc.	Disabling the setting is not possible.
Prohibition of writing		Set via GUI of dedicated flash memory programmer, etc.
Prohibition of rewriting the boot area		Disabling the setting is not possible.
Prohibition of connection to the programmer and on-chip debugger		
Enabling programmer connection ID authentication		

Caution **The setting to prohibit writing can only be released when the settings to prohibit erasing blocks and rewriting the boot area are not made and the code and data flash memory areas are blank. However, if connection for serial programming is prohibited due to the setting to prohibit connection to the programmer and on-chip debugger or to enable programmer connection ID authentication, releasing the setting to prohibit writing is not possible because serial programming cannot be executed.**

2. During self-programming

Security	Security Setting	Disabling the Security Setting
Prohibition of block erasure	Set by using self-programming.	Disabling the setting is not possible.
Prohibition of writing		Disabling the setting is not possible during self-programming. Set via GUI of dedicated flash memory programmer, etc.
Prohibition of rewriting the boot area		Disabling the setting is not possible.
Prohibition of connection to the programmer and on-chip debugger		
Enabling programmer connection ID authentication		

39.10 Data Flash Memory

39.10.1 Overview of the data flash memory

An overview of the data flash memory is provided below.

- The user program can rewrite the data flash memory by self-programming. For details, refer to the **RL78 Family Renesas Flash Driver RL78 Type 01 User's Manual (R20UT4830)**.
- The data flash memory can also be rewritten to through serial programming using a dedicated flash memory programmer or an external device.
- The data flash memory can be erased in 1-block (256-byte) units.
- The data flash memory can be accessed only in 8-bit units.
- The data flash memory can be directly read by CPU instructions.
- Instructions can be executed from the code flash memory while rewriting the data flash memory (that is, background operation (BGO) is supported).
- The data flash memory area is exclusively to be used for data, so executing instructions from the data flash memory is prohibited.
- Accessing the data flash memory is prohibited while rewriting the code flash memory (during self-programming).
- Manipulating the DFLCTL register is prohibited while rewriting the data flash memory.
- Transition to the STOP mode is prohibited while rewriting the data flash memory.

Caution 1. The data flash memory is stopped after a reset is canceled. The data flash control register (DFLCTL) must be set up in order to use the data flash memory.

Caution 2. The high-speed on-chip oscillator should be kept operating during rewriting of the data flash memory area. If it is stopped, it should be made to operate again (HIOSTOP = 0), and self-programming should be executed after 5 μs have elapsed.

Remark For rewriting the code flash memory via a user program, see **39.6 Self-programming**.

39.10.2 Procedure for accessing the data flash memory

The data flash memory is stopped after release from a reset. For access to this memory, follow the procedure below to make the required initial setting.

<1> Set 1 in bit 0 (DFLEN) of the data flash control register (DFLCTL).
<2> Wait for the setup time by using a software timer, etc.
The setup time depends on the operating mode of the flash memory in terms of the main clock selection.
<Setup time for each operating mode of the flash memory>
- HS (high-speed main) mode: 250 ns
- LS (low-speed main) mode: 250 ns
- LP (low-power main) mode: 0 ns

<3> After the wait, the data flash memory can be accessed.

Caution 1. Accessing the data flash memory is prohibited during the setup time.
Caution 2. Transition to the STOP mode is prohibited during the setup time. To enter the STOP mode during the setup time, clear DFLEN to 0 and then execute the STOP instruction.
Caution 3. The high-speed on-chip oscillator should be kept operating during rewriting of the data flash memory area. If it is stopped, it should be made to operate again (HIOSTOP = 0), and self-programming should be executed after 5 μs have elapsed.
Caution 4. When switching the CPU clock between the main system clock and sub system clock, accessing the data flash memory is prohibited until the CLS bit changes after a clock is selected by the CSS bit.

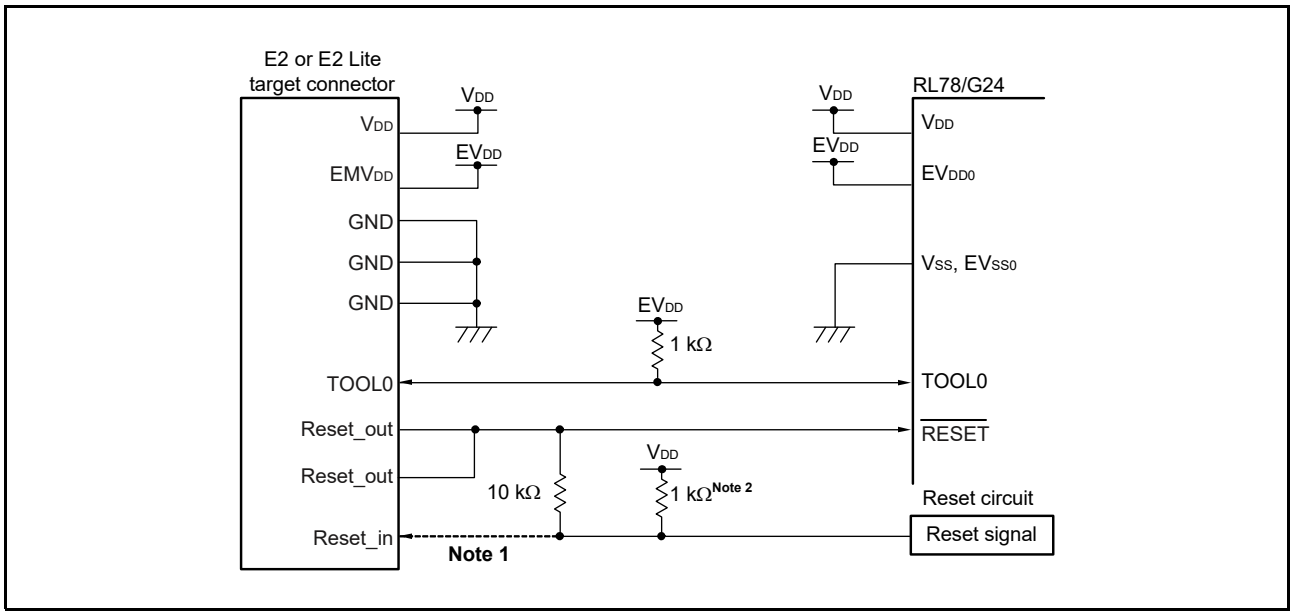
Section 40 On-chip Debugging

40.1 Connection between the E2 or E2 Lite On-chip Debugging Emulator and RL78/G24

On-chip debugging is handled by connecting the RL78 microcontroller and host machine through the E2 or E2 Lite on-chip debugging emulator. Pins V_{DD}, $\overline{RESET}$, TOOL0, and V_{SS} are used for connection to the emulator. Serial communications are handled through the TOOL0 pin as a single-line UART connection. For details and usage notes on the circuit to make the connection, refer to the **E1/E20/E2 Emulator**, **E2 Emulator Lite Additional Document for User's Manual (Notes on Connection of RL78) (R20UT1994)**.

Caution RL78 microcontrollers have on-chip debugging functionality for use in the development and evaluation of user systems. Do not use on-chip debugging with products designated as part of mass production, because using this function may cause the guaranteed number of times the flash memory is rewriting to be exceeded, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when on-chip debugging is used with products designated as part of mass production.

Figure 40 - 1 Example of Connection between the E2 or E2 Lite On-chip Debugging Emulator and RL78/G24



Note 1. The connection shown as a broken line is not required for serial programming.

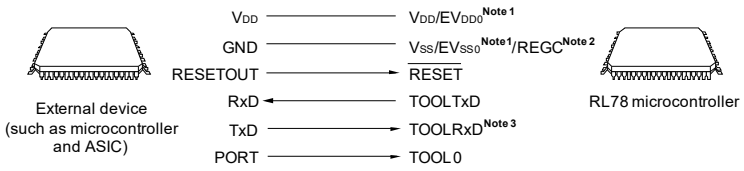
Note 2. If the reset circuit on the target system does not have a buffer so the reset signal is only generated through resistors and capacitors, this pull-up resistor is not required.

Caution This circuit diagram is an example where the reset signal is output from an N-channel open-drain buffer with an output resistance no greater than 100 Ω.

Remark Pins EV_{DD0} and EV_{SS0} are not present in some products. In such cases, use the V_{DD} and V_{SS} pins as the alternative.

40.2 Connection between the External Device that Incorporates UART and RL78/G24

On-board communications between an external device (a microcontroller or ASIC) that is connected to the RL78 microcontroller via a UART and the host machine is possible. Pins V_{DD}, $\overline{RESET}$, TOOL0, V_{SS}, TOOLTxD, and TOOLRxD are used for the communications. Communications 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. For details and usage notes on the circuit to make the connection, refer to the **RL78 Debugging Functions Using the Serial Port (R20AN0632)**.



Note 1. This pin is only present in the 64-pin products.

Note 2. Connect the REGC pin to the ground via a capacitor (0.47 to 1 µF).

Note 3. Set the port pin with which TOOLRxD is multiplexed as an input. The input to the input buffer must also be enabled by using the PDIDISx register.

40.3 Security Settings for On-chip Debugging

To protect against third parties reading the contents of memory, on-chip debugging includes the following functionality.

- Disabling of connection between the RL78 microcontroller and the programmer or on-chip debugger (see **39.9 Security Settings**).
- On-chip debugging control bits in the flash memory at 000C3H (see **Section 38 Option Bytes**)
- An area in the range from 000C4H to 000CDH to hold the security ID code for on-chip debugging.**Note**

Note The area to hold the security ID code for use in on-chip debugging is also used to hold the ID code for the programmer connection ID authentication when a programmer is to be used.

Table 40 - 1 On-chip Debug Security ID

Address	Security ID Code for On-chip Debugging
000C4H to 000CDH	Any 10-byte ID code**Note**
040C4H to 040CDH	

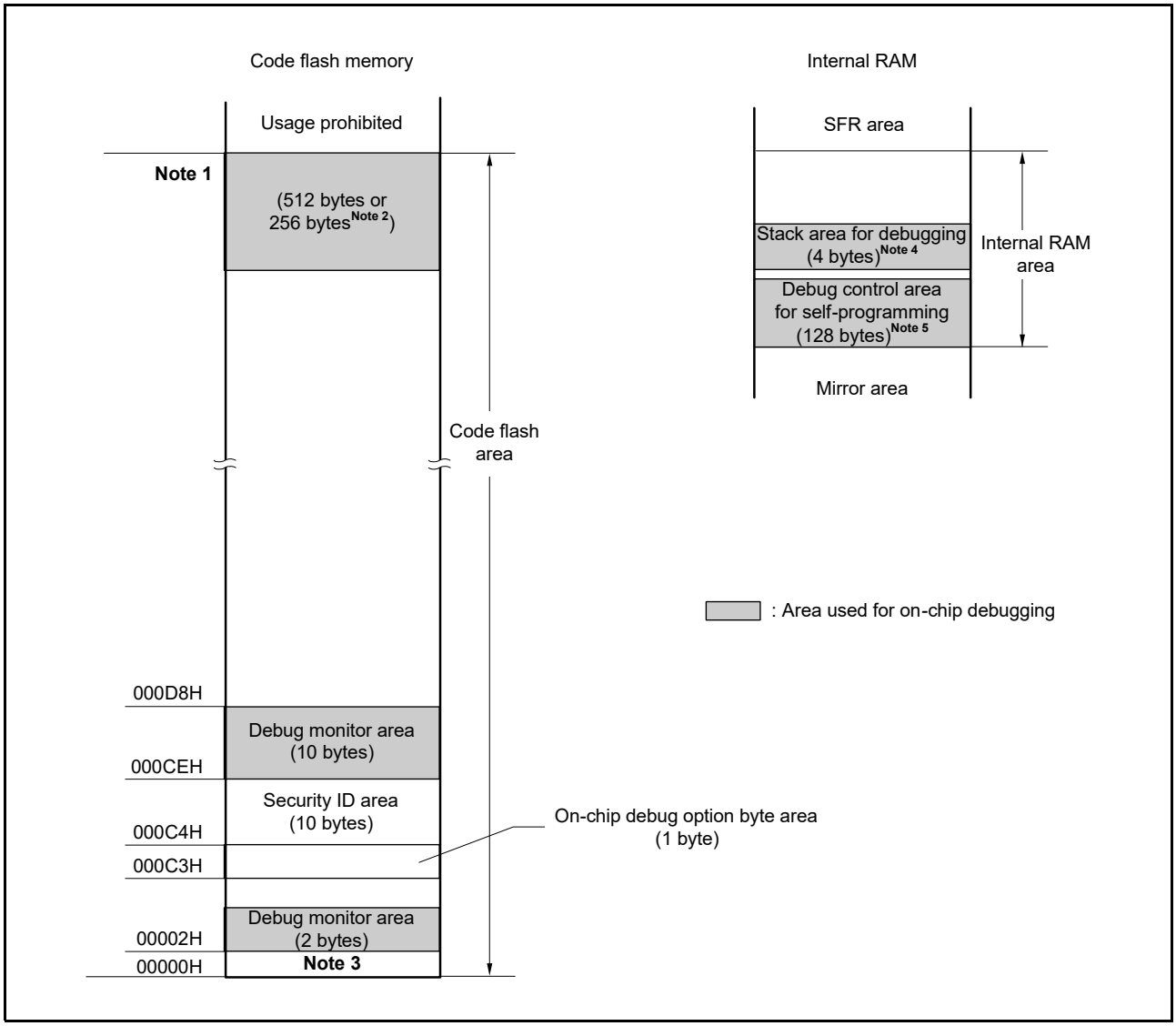
Note The setting FFFFFFFFFFFFFFFFFFFFH is not allowed.

40.4 Allocation of Memory Spaces to User Resources

Allocation of memory spaces to user resources is required before communications between the RL78 microcontroller and E2 or E2 Lite on-chip debugging emulator, and on-chip debugging, can proceed. If you are using an assembler or compiler from Renesas Electronics, you can use linker options to allocate the memory spaces.

1. Allocation of memory spaces to the user program
   The shaded areas in **Figure 40 - 2** are reserved for the monitor program for debugging, and user programs and data cannot be allocated to these areas. When using on-chip debugging, ensure that nothing is allocated to these areas so that they can be secured for on-chip debugging. Also ensure that the contents of these areas are not modified by the user program.

Figure 40 - 2 Memory Spaces Allocated for Use by the Monitor Program for Debugging



Note 1. The address depends on the products as shown below.

Products	Address of Note 1
R7F101GxE (x = 6 to 8, A, B, E to G, J, and L)	0FFFFH
R7F101GxG (x = 6 to 8, A, B, E to G, J, and L)	1FFFFH

Note 2. When the realtime RAM monitor (RRM) and dynamic memory modification (DMM) are not to be used, the size of this area is 256 bytes.

Note 3. During debugging, the reset vector is relocated to the address of the monitor program.

Note 4. Since this area is allocated immediately below the portion of the main stack area that is currently in use, the address range of this area depends on the amount of the stack in use other than for debugging. Accordingly, four additional bytes are required for the entire stack area. In the case of self-programming, this is a 12-byte area, so 12 additional bytes are required.

Note 5. The on-chip debugger uses the 128-byte RAM area stated below for breaks during self-programming.

- FCF00H to FCF7FH

If the setting to disable debugging by the on-chip debugger during self-programming has been made, the RAM area stated above is not used. For details on the settings for debugging during self-programming, see the user's manual for the integrated development environment in use.

Section 41 BCD Correction Circuit

41.1 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).

41.2 Register to Control the BCD Correction Circuit

The following register is used to control the BCD correction circuit.

- BCD correction result register (BCDADJ)

41.2.1 BCD correction result register (BCDADJ)

The BCDADJ register holds correction values for obtaining the addition/subtraction 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.
The value of this register following a reset is undefined.

Figure 41 - 1 Format of BCD Correction Result Register (BCDADJ)

Address: F00FEH
After reset: Undefined
R/W: R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

41.3 Operation of BCD Correction Circuit

The basic operation of the BCD correction circuit is as follows.

1. Addition: Obtaining the result of the binary addition of BCD values and BCD values as a BCD 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 values 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. BCD correction in the interrupt-enabled state requires saving and restoring the A register within the interrupt function. The RETI instruction restores the program status word (PSW), which contains the CY and AC flags.

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: Obtaining the result of the binary subtraction of BCD values and BCD values as a BCD 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. BCD correction in the interrupt-enabled state requires saving and restoring the A register within the interrupt function. The RETI instruction restores the program status word (PSW), which contains the CY and AC flags.

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	—

Section 42 Instruction Set

This section 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)**.

42.1 Conventions Used in Operation List

42.1.1 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 42 - 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 **Table 3 - 5 List of Special Function Registers (SFRs)** for the symbols of the special function registers. The extended special function registers can be described to operand !addr16 as symbols. See **Table 3 - 6 List of Extended Special Function Registers (2nd SFRs)** for the symbols of the extended special function registers.

42.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 42 - 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)

42.1.3 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 42 - 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

42.1.4 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 and DTC transfer are not acknowledged between a PREFIX instruction code and the instruction immediately after.

Table 42 - 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.

42.2 Operation List

Table 42 - 5 Operation List (1/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 42 - 5 Operation List (2/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 42 - 5 Operation List (3/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 42 - 5 Operation List (4/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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 (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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 42 - 5 Operation List (5/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 42 - 5 Operation List (6/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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, 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 (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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 42 - 5 Operation List (7/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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, 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	×	×	×
	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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 42 - 5 Operation List (8/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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 (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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 42 - 5 Operation List (9/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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 (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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 42 - 5 Operation List (10/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 42 - 5 Operation List (11/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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)	×	×	×

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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 42 - 5 Operation List (12/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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 (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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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. 1.01.00 and later versions of CC-RL (Renesas Electronics compiler), for both C and assembly language source code**
- **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**
- **LLVM RL78 (CyberTHOR compiler), for both C and C++ language source code**
- **GNURL78 (CyberTHOR compiler), for both C and 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 42 - 5 Operation List (13/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 42 - 5 Operation List (14/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Rotate	ROR	A, 1	2	1	—	(CY, $A_7 \leftarrow A_0$, $A_{m-1} \leftarrow A_m$) × 1			×
	ROL	A, 1	2	1	—	(CY, $A_0 \leftarrow A_7$, $A_{m+1} \leftarrow A_m$) × 1			×
	RORC	A, 1	2	1	—	(CY $\leftarrow A_0$, $A_7 \leftarrow$ CY, $A_{m-1} \leftarrow A_m$) × 1			×
	ROLC	A, 1	2	1	—	(CY $\leftarrow A_7$, $A_0 \leftarrow$ CY, $A_{m+1} \leftarrow A_m$) × 1			×
	ROLWC	AX,1	2	1	—	(CY $\leftarrow AX_{15}$, $AX_0 \leftarrow$ CY, $AX_{m+1} \leftarrow AX_m$) × 1			×
		BC,1	2	1	—	(CY $\leftarrow BC_{15}$, $BC_0 \leftarrow$ CY, $BC_{m+1} \leftarrow BC_m$) × 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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 42 - 5 Operation List (15/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipulate	XOR1	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			×
	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 ← $\overline{CY}$			×

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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 42 - 5 Operation List (16/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Call/return	CALL	rp	2	3/5Note 3	—	(SP - 2) ← (PC + 2)S, (SP - 3) ← (PC + 2)H, (SP - 4) ← (PC + 2)L, PC ← CS, rp, SP ← SP - 4			
		$!addr20	3	3/5Note 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	3/5Note 3	—	(SP - 2) ← (PC + 3)S, (SP - 3) ← (PC + 3)H, (SP - 4) ← (PC + 3)L, PC ← 0000, addr16, SP ← SP - 4			
		!!addr20	4	3/5Note 3	—	(SP - 2) ← (PC + 4)S, (SP - 3) ← (PC + 4)H, (SP - 4) ← (PC + 4)L, PC ← addr20, SP ← SP - 4			
	CALLT	[addr5]	2	5/7Note 3	—	(SP - 2) ← (PC + 2)S, (SP - 3) ← (PC + 2)H, (SP - 4) ← (PC + 2)L , PCS ← 0000, PCH ← (0000, addr5 + 1), PCL ← (0000, addr5), SP ← SP - 4			
	BRK	—	2	5/7Note 3	—	(SP - 1) ← PSW, (SP - 2) ← (PC + 2)S, (SP - 3) ← (PC + 2)H, (SP - 4) ← (PC + 2)L, PCS ← 0000, PCH ← (0007FH), PCL ← (0007EH), SP ← SP - 4, IE ← 0			
	RET	—	1	6/8Note 3	—	PCL ← (SP), PCH ← (SP + 1), PCS ← (SP + 2), SP ← SP + 4			
	RETI	—	2	6/8Note 3	—	PCL ← (SP), PCH ← (SP + 1), PCS ← (SP + 2), PSW ← (SP + 3), SP ← SP + 4	R	R	R
	RETB	—	2	6/8Note 3	—	PCL ← (SP), PCH ← (SP + 1), PCS ← (SP + 2), PSW ← (SP + 3), SP ← SP + 4	R	R	R

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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. The two numbers indicate the numbers of clock cycles when PFBE = 0 and when PFBE = 1.

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 42 - 5 Operation List (17/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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	2/4/6Note 3	—	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	$addr20	2	2/4/6Note 3	—	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	$addr20	2	2/4/6Note 3	—	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	$addr20	2	2/4/6Note 3	—	PC ← PC + 2 + jdisp8 if Z = 0			
	BH	$addr20	3	2/4/6Note 3	—	PC ← PC + 3 + jdisp8 if (Z ∨ CY) = 0			
	BNH	$addr20	3	2/4/6Note 3	—	PC ← PC + 3 + jdisp8 if (Z ∨ CY) = 1			
	BT	saddr.bit, $addr20	4	3/5/7Note 3	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 1			
		sfr.bit, $addr20	4	3/5/7Note 3	—	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
		A.bit, $addr20	3	3/5/7Note 3	—	PC ← PC + 3 + jdisp8 if A.bit = 1			
		PSW.bit, $addr20	4	3/5/7Note 3	—	PC ← PC + 4 + jdisp8 if PSW.bit = 1			
		[HL].bit, $addr20	3	3/5/7Note 3	6/7/9Note 3	PC ← PC + 3 + jdisp8 if (HL).bit = 1			
		ES:[HL].bit, $addr20	4	4/6/8Note 3	7/8/10Note 3	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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. The three numbers indicate the numbers of clock cycles when the condition is not met, when the condition is met and PFBE = 0, and when the condition is met and PFBE = 1.

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 42 - 5 Operation List (18/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Conditional branch	BF	saddr.bit, $addr20	4	3/5/7Note 3	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 0			
		sfr.bit, $addr20	4	3/5/7Note 3	—	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
		A.bit, $addr20	3	3/5/7Note 3	—	PC ← PC + 3 + jdisp8 if A.bit = 0			
		PSW.bit, $addr20	4	3/5/7Note 3	—	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
		[HL].bit, $addr20	3	3/5/7Note 3	6/7/9Note 3	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
		ES:[HL].bit, $addr20	4	4/6/8Note 3	7/8/10Note 3	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0			
	BTCLR	saddr.bit, $addr20	4	3/5/7Note 3	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit			
		sfr.bit, $addr20	4	3/5/7Note 3	—	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit			
		A.bit, $addr20	3	3/5/7Note 3	—	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit			
		PSW.bit, $addr20	4	3/5/7Note 3	—	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	×	×	×
		[HL].bit, $addr20	3	3/5/7Note 3	—	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit			
		ES:[HL].bit, $addr20	4	4/6/8Note 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	SELNote 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 (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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. The three numbers indicate the numbers of clock cycles when the condition is not met, when the condition is met and PFBE = 0, and when the condition is met and PFBE = 1.

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.

Section 43 Electrical Characteristics (T_A = –40 to +105°C)

This section describes the electrical characteristics of the following types of products.

- 2D: Consumer applications, T_A = –40 to +85°C
  R7F101Gxx2Dxx
- 3C: Industrial applications, T_A = –40 to +105°C
  R7F101Gxx3Cxx
- 4C: Industrial applications, products of T_A = –40 to +125°C, but under the condition T_A = –40 to +105°C
  R7F101Gxx4Cxx

Caution 1. RL78 microcontrollers have on-chip debugging functionality for use in the development and evaluation of user systems. Do not use on-chip debugging with products designated as part of mass production, because using this function may cause the guaranteed number of times the flash memory is rewritten to be exceeded, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when on-chip debugging is used with products designated as part of mass production.

Caution 2. For the consumer application products (2D), the ambient operating temperature of T_A = –40°C to +85°C applies.

Caution 3. For products that do not have an EV_{DD0} or EV_{SS0} pin, read EV_{DD0} as V_{DD}, and EV_{SS0} as V_{SS}.

Caution 4. The present pins differ depending on the products. For details, see 2.1 Functions of Port Pins through 2.2.1 Functions for each product.

43.1 Absolute Maximum Ratings

(1/2)

Item	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}		–0.5 to +6.5	V
	EV_{DD0}		–0.5 to +6.5	V
	EV_{SS0}		–0.5 to +0.3	V
REGC pin input voltage	V_{IREGC}	REGC	–0.3 to +2.1 and –0.3 to V_{DD} + 0.3 **Note 1**	V
Input voltage	V_{I1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63, P70 to P77, P120, P140, P141, P146, P147	–0.3 to EV_{DD0} + 0.3 and –0.3 to V_{DD} + 0.3 **Note 2**	V
	V_{I2}	P60, P61 (N-ch open drain)	–0.3 to +6.5	V
	V_{I3}	P20 to P27, P121 to P124, P137, EXCLK, EXCLKS, $\overline{RESET}$	–0.3 to V_{DD} + 0.3 **Note 2**	V
Output voltage	V_{O1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63, P70 to P77, P120, P130, P140, P141, P146, P147	–0.3 to EV_{DD0} + 0.3 and –0.3 to V_{DD} + 0.3 **Note 2**	V
	V_{O2}	P20 to P27, P121, P122	–0.3 to V_{DD} + 0.3 **Note 2**	V
Analog input voltage	V_{AI1}	ANI16 to ANI30	–0.3 to EV_{DD0} + 0.3 and –0.3 to AV_{REFP} + 0.3 **Notes 2, 3**	V
	V_{AI2}	ANI0 to ANI7	–0.3 to V_{DD} + 0.3 and –0.3 to AV_{REFP} + 0.3 **Notes 2, 3**	V

Note 1. Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). The listed value is the absolute maximum rating of the REGC pin. Do not apply a specific voltage to this pin.

Note 2. This voltage must be no higher than 6.5 V.

Note 3. The voltage on a pin in use for A/D conversion must not exceed AV_{REFP} + 0.3.

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. The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

Remark 2. AV_{REFP} refers to the positive reference voltage of the A/D converter.

Remark 3. The reference voltage is V_{SS}.

(2/2)

Item	Symbols	Conditions			Ratings	Unit
High-level output current	I_{OH1}	Per pin	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63, P70 to P77, P120, P130, P140, P141, P146, P147		–40	mA
		Total of all pins –170 mA	P00 to P04, P40 to P43, P120, P130, P140, P141		–70	mA
			P05, P06, P10 to P17, P30, P31, P50 to P55, P62, P63, P70 to P77, P146, P147		–100	mA
	I_{OH2}	Per pin	P20 to P27, P121, P122		–5	mA
		Total of all pins			–20	mA
Low-level output current	I_{OL1}	Per pin	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63, P70 to P77, P120, P130, P140, P141, P146, P147		40	mA
		Total of all pins 170 mA	P00 to P04, P40 to P43, P120, P130, P140, P141		70	mA
			P05, P06, P10 to P17, P30, P31, P50 to P55, P60 to P63, P70 to P77, P146, P147		100	mA
	I_{OL2}	Per pin	P20 to P27, P121, P122		10	mA
		Total of all pins			20	mA
Ambient operating temperature	T_A	In normal operation mode		3C: Industrial applications	–40 to +105	°C
				2D: Consumer applications	–40 to +85	°C
		In flash memory programming mode		3C: Industrial applications	–40 to +105	°C
				2D: Consumer applications	–40 to +85	°C
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 The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

43.2 Characteristics of the Oscillators

43.2.1 Characteristics of the X1 and XT1 oscillators

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V (20- to 32-pin products), 1.6 V ≤ V_{DD} ≤ 5.5 V (40- to 64-pin products), V_{SS} = 0 V)

Item	Resonator	Conditions	Min.	Typ.	Max.	Unit
X1 clock oscillation allowable input cycle time**Note**	Ceramic resonator/ crystal resonator		0.05		1	μs
XT1 clock oscillation frequency (f_{XT})**Note**	Crystal resonator			32.768		kHz

Note The listed time and frequency indicate permissible ranges of the oscillators. For actual applications, request the resonator manufacturer for evaluation of the resonators on the oscillator circuit mounted on a board so you can use appropriate values. Refer to **43.4 AC Characteristics** for instruction execution time.

Caution **Since the CPU is started by the high-speed on-chip oscillator clock after release from the reset state, the user should use the oscillation stabilization time counter status register (OSTC) to check the X1 clock oscillation stabilization time. Sufficiently evaluate the oscillation stabilization time with the resonator to be used, and then specify the values for the oscillation stabilization time in the OSTC register and the oscillation stabilization time select register (OSTS).**

43.2.2 Characteristics of the on-chip oscillators

(T_A = –40 to +105°C, 1.6 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item	Symbol	Conditions			Min.	Typ.	Max.	Unit
High-speed on-chip oscillator clock frequency	f_{IH}				1		48	MHz
High-speed on-chip oscillator clock frequency accuracy **Note 1**		HIPREC = 1	+85 to +105°C	1.8 V ≤ V_{DD} ≤ 5.5 V	–1.5		+1.5	%
				1.6 V ≤ V_{DD} ≤ 5.5 V	–6.0		+6.0	%
			–20 to +85°C	1.8 V ≤ V_{DD} ≤ 5.5 V	–1.0		+1.0	%
				1.6 V ≤ V_{DD} ≤ 5.5 V	–5.0		+5.0	%
			–40 to –20°C	1.8 V ≤ V_{DD} ≤ 5.5 V	–1.5		+1.5	%
				1.6 V ≤ V_{DD} ≤ 5.5 V	–5.5		+5.5	%
		HIPREC = 0 **Note 4**			–15		0	%
High-speed on-chip oscillator clock correction resolution						0.05		%
Middle-speed on-chip oscillator clock frequency **Note 2**	f_{IM}				1		4	MHz
Middle-speed on-chip oscillator clock frequency accuracy **Note 1**					–12		+12	%
Middle-speed on-chip oscillator clock correction resolution						0.15		%
Middle-speed on-chip oscillator frequency temperature coefficient							±0.17 **Note 3**	%/°C
Low-speed on-chip oscillator clock frequency **Note 2**	f_{IL}					32.768		kHz
Low-speed on-chip oscillator clock frequency accuracy **Note 1**					–15		+15	%
Low-speed on-chip oscillator clock correction resolution						0.3		%
Low-speed on-chip oscillator frequency temperature coefficient							±0.21 **Note 3**	%/°C

Note 1. The accuracy values were obtained in testing of this product.

Note 2. The listed values only indicate the characteristics of the oscillators. Refer to **43.4 AC Characteristics** for instruction execution time.

Note 3. These values were obtained in the evaluation.

Note 4. This condition applies when the setting of the FRQSEL3 bit is 1.

43.2.3 Characteristics of the PLL oscillator

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
PLL input frequency	f_{PLLIN}	High-speed system clock (f_{MX}) or high-speed on-chip oscillator clock (f_{IH})		8		MHz
PLL output frequency	f_{PLL}	f_{PLLIN} × 12		96		MHz
		f_{PLLIN} × 8		64		MHz
Lock-up wait time		Wait time after PLL output is enabled until the output frequency is stabilized	50			μs
Interval wait time		Wait time after PLL stop until PLL operation is set again	4			μs
Setting wait time		Required wait time after the PLL input clock is stabilized and the PLL setting is determined until startup settings are made	1			μs

43.3 DC Characteristics

43.3.1 Pin characteristics

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (1/7)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Allowable high-level output current**Note 1**	I_{OH1}	Per pin for P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63, P70-P77, P120, P130, P140, P141, P146, P147	1.6 V ≤ EV_{DD0} ≤ 5.5 V			–10.0 **Note 2**	mA
		Total of P00 to P04, P40 to P43, P120, P130, P140, P141 (when duty ≤ 70%**Note 3**)	4.0 V ≤ EV_{DD0} ≤ 5.5 V			–55.0 **Note 4**	mA
			2.7 V ≤ EV_{DD0} < 4.0 V			–10.0	mA
			1.8 V ≤ EV_{DD0} < 2.7 V			–5.0	mA
			1.6 V ≤ EV_{DD0} < 1.8 V			–2.5	mA
		Total of P05, P06, P10 to P17, P30, P31, P50 to P55, P62, P63, P70 to P77, P146, P147 (when duty ≤ 70%**Note 3**)	4.0 V ≤ EV_{DD0} ≤ 5.5 V			–80.0 **Note 5**	mA
			2.7 V ≤ EV_{DD0} < 4.0 V			–19.0 **Note 7**	mA
			1.8 V ≤ EV_{DD0} < 2.7 V			–10.0	mA
			1.6 V ≤ EV_{DD0} < 1.8 V			–5.0	mA
		Total of all pins (when duty ≤ 70%**Note 3**)	1.6 V ≤ EV_{DD0} ≤ 5.5 V			–135.0 **Note 6**	mA
	I_{OH2}	Per pin for P20 to P27, P121, P122	4.0 V ≤ V_{DD} ≤ 5.5 V			–3.0 **Note 2**	mA
			2.7 V ≤ V_{DD} < 4.0 V			–1.0 **Note 2**	mA
			1.8 V ≤ V_{DD} < 2.7 V			–1.0 **Note 2**	mA
			1.6 V ≤ V_{DD} < 1.8 V			–0.5 **Note 2**	mA
		Total of all pins (when duty ≤ 70%**Note 3**)	4.0 V ≤ V_{DD} ≤ 5.5 V			–20.0 **Note 8**	mA
			2.7 V ≤ V_{DD} < 4.0 V			–10.0 **Note 9**	mA
			1.8 V ≤ V_{DD} < 2.7 V			–5.0	mA
			1.6 V ≤ V_{DD} < 1.8 V			–5.0	mA

(Notes, Caution, and Remark are listed on the next page.)

Note 1. Device operation is guaranteed at the listed currents even if current is flowing from the EV_{DD0} or V_{DD} pin to an output pin.

Note 2. The combination of these and other pins must not exceed the total current value.

Note 3. The listed output current values apply when the duty cycle is no greater than 70%.
Use the following formula to calculate the output current when the duty cycle is greater than 70%, where n is the duty cycle (%).

- Total output current from all pins = (I_{OH} × 0.7)/(n × 0.01)
  Example when I_{OH} = –10.0 mA, n = 80%
  Total output current from all pins = (–10.0 × 0.7)/(80 × 0.01) ≈ –8.7 mA
  Note that the duty cycle has no effect on the current that is allowed to flow into a single pin.

Note 4. The maximum value is –30 mA in products for industrial applications (R7F101Gxx3Cxx) with an ambient operating temperature range of +85°C to +105°C.
The maximum value is –24 mA in products for industrial applications (R7F101Gxx4Cxx) with an ambient operating temperature range of –40°C to +105°C.

Note 5. The maximum value is –50 mA in products for industrial applications (R7F101Gxx3Cxx) with an ambient operating temperature range of +85°C to +105°C.
The maximum value is –42 mA in products for industrial applications (R7F101Gxx4Cxx) with an ambient operating temperature range of –40°C to +105°C.

Note 6. The maximum value is –60 mA with an ambient operating temperature range of +85°C to +105°C and -100 mA with an ambient operating temperature range of –40°C to +85°C in products for industrial applications (R7F101Gxx3Cxx).
The maximum value is –54 mA in products for industrial applications (R7F101Gxx4Cxx) with an ambient operating temperature range of –40°C to +105°C.

Note 7. The maximum value is –17 mA in products for industrial applications (R7F101Gxx4Cxx) with an ambient operating temperature range of –40°C to +105°C.

Note 8. The maximum value is –14 mA in products for industrial applications (R7F101Gxx4Cxx) with an ambient operating temperature range of –40°C to +105°C.

Note 9. The maximum value is –8 mA in products for industrial applications (R7F101Gxx4Cxx) with an ambient operating temperature range of –40°C to +105°C.

Caution **The following pins do not output high-level signals in the N-ch open-drain mode.**
P00, P02 to P04, P10 to P15, P17, P30, P50, P51, P55, and P71 to P74

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (2/7)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Allowable low-level output current**Note 1**	I_{OL1}	Per pin for P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63,P70 to P77, P120, P130, P140, P141, P146, P147				20.0 **Notes 2, 8**	mA
		Per pin for P60, P61				15.0 **Note 2**	mA
		Total of P00 to P04, P40 to P43, P120, P130, P140, P141 (when duty ≤ 70%**Note 3**)	4.0 V ≤ EV_{DD0} ≤ 5.5 V			70.0 **Note 4**	mA
			2.7 V ≤ EV_{DD0} < 4.0 V			15.0	mA
			1.8 V ≤ EV_{DD0} < 2.7 V			9.0	mA
			1.6 V ≤ EV_{DD0} < 1.8 V			4.5	mA
		Total of P05, P06, P10 to P17, P30, P31, P50 to P55, P62, P63, P70 to P77, P146, P147 (when duty ≤ 70%**Note 3**)	4.0 V ≤ EV_{DD0} ≤ 5.5 V			80.0 **Note 4**	mA
			2.7 V ≤ EV_{DD0} < 4.0 V			35.0 **Note 6**	mA
			1.8 V ≤ EV_{DD0} < 2.7 V			20.0	mA
			1.6 V ≤ EV_{DD0} < 1.8 V			10.0	mA
		Total of all pins (when duty ≤ 70%**Note 3**)				150.0 **Note 5**	mA
	I_{OL2}	Per pin for P20 to P27, P121, P122	4.0 V ≤ V_{DD} ≤ 5.5 V			8.5**Note 2**	mA
			2.7 V ≤ V_{DD} < 4.0 V			1.5**Note 2**	mA
			1.8 V ≤ V_{DD} < 2.7 V			0.6**Note 2**	mA
			1.6 V ≤ V_{DD} < 1.8 V			0.4**Note 2**	mA
		Total of all pins (when duty ≤ 70%**Note 3**)	4.0 V ≤ V_{DD} ≤ 5.5 V			20 **Note 7**	mA
			2.7 V ≤ V_{DD} < 4.0 V			20 **Note 7**	mA
			1.8 V ≤ V_{DD} < 2.7 V			15.0	mA
			1.6 V ≤ V_{DD} < 1.8 V			10.0	mA

(Notes and Remark are listed on the next page.)

Note 1. Device operation is guaranteed at the listed currents even if current is flowing from an output pin to the EV_{SS0} or V_{SS} pin.

Note 2. The combination of these and other pins must not exceed the total current value.

Note 3. The listed output current values apply when the duty cycle is no greater than 70%.
Use the following formula to calculate the output current when the duty cycle is greater than 70%, where n is the duty cycle (%).

- Total output current from all pins = (I_{OH} × 0.7)/(n × 0.01)
  Example when I_{OH} = –10.0 mA, n = 80%
  Total output current from all pins = (–10.0 × 0.7)/(80 × 0.01) ≈ –8.7 mA
  Note that the duty cycle has no effect on the current that is allowed to flow into a single pin. No current higher than the absolute maximum rating must not flow into a single pin.

Note 4. The maximum value is 40 mA in products for industrial applications (R7F101Gxx3Cxx) with an ambient operating temperature range of +85°C to +105°C.
The maximum value is 34 mA in products for industrial applications (R7F101Gxx4Cxx) with an ambient operating temperature range of –40°C to +105°C.

Note 5. The maximum value is 80 mA in products for industrial applications (R7F101Gxx3Cxx) with an ambient operating temperature range of +85°C to +105°C.
The maximum value is 68 mA in products for industrial applications (R7F101Gxx4Cxx) with an ambient operating temperature range of –40°C to +105°C.

Note 6. The maximum value is 15 mA in products for industrial applications (R7F101Gxx4Cxx) with an ambient operating temperature range of –40°C to +105°C.

Note 7. The maximum value is 14 mA in products for industrial applications (R7F101Gxx4Cxx) with an ambient operating temperature range of –40°C to +105°C.

Note 8. The maximum value is 17 mA in products for industrial applications (R7F101Gxx4Cxx) with an ambient operating temperature range of –40°C to +105°C.

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (3/7)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Input voltage, high	V_{IH1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63, P70 to P77, P120, P140, P141, P146, P147	Normal input buffer	0.8 EV_{DD0}		EV_{DD0}	V
	V_{IH2}	P01, P03, P04, P10, P11, P14 to P17, P30, P50, P55, P73	TTL input buffer 4.0 V ≤ EV_{DD0} ≤ 5.5 V	2.2		EV_{DD0}	V
			TTL input buffer 3.3 V ≤ EV_{DD0} < 4.0 V	2.0		EV_{DD0}	V
			TTL input buffer 1.6 V ≤ EV_{DD0} < 3.3 V	1.5		EV_{DD0}	V
	V_{IH3}	P20 to P27		0.7 V_{DD}		V_{DD}	V
	V_{IH4}	P60, P61	I/O port mode	0.7 EV_{DD0}		6.0	V
	V_{IH5}	P121 to P124, P137, EXCLK, EXCLKS, $\overline{\text{RESET}}$		0.8 V_{DD}		V_{DD}	V
	V_{IH6}	P60, P61	SMBus input mode 2.7 V ≤ EV_{DD0} ≤ 5.5 V	1.35		EV_{DD0}	V
Input voltage, low	V_{IL1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63, P70 to P77, P120, P140, P141, P146, P147	Normal input buffer	0		0.2 EV_{DD0}	V
	V_{IL2}	P01, P03, P04, P10, P11, P14 to P17, P30, P50, P55, P73	TTL input buffer 4.0 V ≤ EV_{DD0} ≤ 5.5 V	0		0.8	V
			TTL input buffer 3.3 V ≤ EV_{DD0} < 4.0 V	0		0.5	V
			TTL input buffer 1.6 V ≤ EV_{DD0} < 3.3 V	0		0.32	V
	V_{IL3}	P20 to P27		0		0.3 V_{DD}	V
	V_{IL4}	P60, P61	I/O port mode	0		0.3 EV_{DD0}	V
	V_{IL5}	P121 to P124, P137, EXCLK, EXCLKS, $\overline{\text{RESET}}$		0		0.2 V_{DD}	V
	V_{IL6}	P60, P61	SMBus input mode 2.7 V ≤ EV_{DD0} ≤ 5.5 V			0.8	V

Caution **The maximum value of V_{IH} of pins P00, P02 to P04, P10 to P15, P17, P30, P50, P51, P55, and P71 to P74 is EV_{DD0}, even in the N-ch open-drain mode.**

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (4/7)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Output voltage, high	V_{OH1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63, P70 to P77, P120, P130, P140, P141, P146, P147	4.0 V ≤ EV_{DD0} ≤ 5.5 V, I_{OH1} = –10.0 mA	EV_{DD0} – 1.5			V
			4.0 V ≤ EV_{DD0} ≤ 5.5 V, I_{OH1} = –3.0 mA	EV_{DD0} – 0.7			V
			2.7 V ≤ EV_{DD0} ≤ 5.5 V, I_{OH1} = –2.0 mA	EV_{DD0} – 0.6			V
			1.8 V ≤ EV_{DD0} ≤ 5.5 V, I_{OH1} = –1.5 mA	EV_{DD0} – 0.5			V
			1.6 V ≤ EV_{DD0} < 5.5 V, I_{OH1} = –1.0 mA	EV_{DD0} – 0.5			V
	V_{OH2}	P20 to P27, P121, P122	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH2} = –3.0 mA	V_{DD} – 0.7			V
			2.7 V ≤ V_{DD} < 4.0 V, I_{OH2} = –1.0 mA	V_{DD} – 0.5			V
			1.8 V ≤ V_{DD} < 2.7 V, I_{OH2} = –1.0 mA	V_{DD} – 0.5			V
			1.6 V ≤ V_{DD} < 1.8 V, I_{OH2} = –0.5 mA	V_{DD} – 0.5			V

Caution **Pins P00, P02 to P04, P10 to P15, P17, P30, P50, P51, P55, P71 to P74 do not output high-level signals in the N-ch open-drain mode.**

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (5/7)

Item	Symbol	Conditions			Min.	Typ.	Max.	Unit
Output voltage, low	V_{OL1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63, P70 to P77, P120, P130, P140, P141, P146, P147	4.0 V ≤ EV_{DD0} ≤ 5.5 V	I_{OL1} = 20.0 mA			1.3	V
			4.0 V ≤ EV_{DD0} ≤ 5.5 V	I_{OL1} = 8.5 mA			0.7	V
			2.7 V ≤ EV_{DD0} ≤ 5.5 V	I_{OL1} = 3.0 mA			0.6	V
			2.7 V ≤ EV_{DD0} ≤ 5.5 V	I_{OL1} = 1.5 mA			0.4	V
			1.8 V ≤ EV_{DD0} ≤ 5.5 V	I_{OL1} = 0.6 mA			0.4	V
			1.6 V ≤ EV_{DD0} ≤ 5.5 V	I_{OL1} = 0.3 mA			0.4	V
	V_{OL2}	P20 to P27, P121, P122	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL2} = 8.5 mA				0.7	V
			2.7 V ≤ V_{DD} < 4.0 V, I_{OL2} = 1.5 mA				0.5	V
			1.8 V ≤ V_{DD} < 2.7 V, I_{OL2} = 0.6 mA				0.4	V
			1.6 V ≤ V_{DD} < 1.8 V, I_{OL2} = 0.4 mA				0.4	V
	V_{OL3}	P60, P61	4.0 V ≤ EV_{DD0} ≤ 5.5 V, I_{OL3} = 15.0 mA				2.0	V
			4.0 V ≤ EV_{DD0} ≤ 5.5 V, I_{OL3} = 5.0 mA				0.4	V
			2.7 V ≤ EV_{DD0} ≤ 5.5 V, I_{OL3} = 3.0 mA				0.4	V
			1.8 V ≤ EV_{DD0} ≤ 5.5 V, I_{OL3} = 2.0 mA				0.4	V
			1.6 V ≤ EV_{DD0} ≤ 5.5 V, I_{OL3} = 1.0 mA				0.4	V

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (6/7)

Item	Symbol	Conditions			Min.	Typ.	Max.	Unit
Output current[Note]	CCDIOL	P10, P11, P16, P17, P60 to P63	CCSm = 01H	4.0 V ≤ EV_{DD0} ≤ 5.5 V	1.0	1.8	2.6	mA
				2.7 V ≤ EV_{DD0} < 4.0 V	0.8	1.5	2.3	mA
			CCSm = 02H	4.0 V ≤ EV_{DD0} ≤ 5.5 V	3.0	4.9	6.5	mA
				3.0 V ≤ EV_{DD0} < 4.0 V	2.7	4.3	5.9	mA
			CCSm = 03H	4.0 V ≤ EV_{DD0} ≤ 5.5 V	6.6	10.0	13.2	mA
				3.3 V ≤ EV_{DD0} < 4.0 V	6.0	9.1	12.1	mA
		P60, P61	CCSm = 04H	4.0 V ≤ EV_{DD0} ≤ 5.5 V	10.2	15.0	19.8	mA
				3.3 V ≤ EV_{DD0} < 4.0 V	9.4	13.8	18.2	mA

Note The listed currents apply when the output current control function is enabled.

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (7/7)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Input leakage current, high	ILIH1	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P60 to P63, P70 to P77, P120, P140, P141, P146, P147	V_I = EV_{DD0}			1	μA
	ILIH2	P20 to P27, P137, $\overline{\text{RESET}}$	V_I = V_{DD}			1	μA
	ILIH3	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	V_I = V_{DD}			1	μA
Input leakage current, low	ILIL1	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63, P70 to P77, P120, P130, P140, P141, P146, P147	V_I = EV_{SS0}			1	μA
	ILIL2	P20 to P27, P137, $\overline{\text{RESET}}$	V_I = V_{SS}			1	μA
	ILIL3	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	V_I = V_{SS}			1	μA
On-chip pull-up resistance	RU	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63, P70 to P77, P120 to P122, P140, P141, P146, P147	V_I = EV_{SS0}, input port	10	20	100	kΩ

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

43.3.2 Supply current characteristics

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (1/5)

Item	Symbol	Conditions					Min.	Typ.	Max.	Unit
Supply current **Note 1**	I_{DD1}	Operating mode	HS (high-speed main) mode	f_{PLL} = 96 MHz f_{CLK} = 48 MHz (MCM0 = 0)**Note 2**	Normal operation	V_{DD} = 5.0 V		5.5	17.6	mA
						V_{DD} = 2.4 V		5.5	17.6	
				f_{PLL} = 96 MHz f_{CLK} = 48 MHz (MCM = 1)**Note 4**	Normal operation	V_{DD} = 5.0 V		5.3	17.4	mA
						V_{DD} = 2.4 V		5.3	17.4	
				f_{IH} = 48 MHz**Note 2**	Normal operation	V_{DD} = 5.0 V		4.6	11.9	mA
						V_{DD} = 2.4 V		4.6	11.9	
				f_{PLL} = 64 MHz f_{CLK} = 32 MHz (MCM0 = 0)**Note 2**	Normal operation	V_{DD} = 5.0 V		3.9	12.1	mA
						V_{DD} = 1.8 V		3.9	12.1	
				f_{PLL} = 64 MHz f_{CLK} = 32 MHz (MCM = 1)**Note 4**	Normal operation	V_{DD} = 5.0 V		3.7	11.9	mA
						V_{DD} = 1.8 V		3.7	11.9	
				f_{IH} = 32 MHz**Note 2**	Basic operation	V_{DD} = 5.0 V		1.6	—	mA
						V_{DD} = 1.8 V		1.6	—	
					Normal operation	V_{DD} = 5.0 V		3.3	8.3	mA
						V_{DD} = 1.8 V		3.3	8.3	
			LS (low-speed main) mode	f_{IH} = 24 MHz**Note 2**	Normal operation	V_{DD} = 5.0 V		2.5	6.3	mA
						V_{DD} = 1.8 V		2.5	6.3	
				f_{IH} = 16 MHz**Note 2**	Normal operation	V_{DD} = 5.0 V		1.8	4.4	mA
						V_{DD} = 1.8 V		1.8	4.4	
				f_{IM} = 4 MHz**Note 3**	Normal operation	V_{DD} = 5.0 V		0.5	1.3	mA
						V_{DD} = 1.6 V		0.5	1.3	
			LP (low-power main) mode	f_{IM} = 2 MHz**Note 3**	Normal operation	V_{DD} = 5.0 V		215	707	µA
						V_{DD} = 1.6 V		214	706	
				f_{IM} = 1 MHz**Note 3**	Normal operation	V_{DD} = 5.0 V		120	466	µA
						V_{DD} = 1.6 V		119	464	
			HS (high-speed main) mode	f_{MX} = 20 MHz**Note 4**, Square wave input	Normal operation	V_{DD} = 5.0 V		2.0	5.2	mA
						V_{DD} = 1.8 V		2.0	5.2	
			LS (low-speed main) mode	f_{MX} = 20 MHz**Note 4**, Square wave input	Normal operation	V_{DD} = 5.0 V		1.9	5.1	mA
						V_{DD} = 1.8 V		1.9	5.0	
				f_{MX} = 20 MHz**Note 4**, Resonator connection	Normal operation	V_{DD} = 5.0 V		2.1	5.3	mA
						V_{DD} = 1.8 V		2.1	5.3	
				f_{MX} = 10 MHz**Note 4**, Square wave input	Normal operation	V_{DD} = 5.0 V		1.0	2.7	mA
						V_{DD} = 1.8 V		1.0	2.7	
				f_{MX} = 10 MHz**Note 4**, Resonator connection	Normal operation	V_{DD} = 5.0 V		1.1	2.9	mA
						V_{DD} = 1.8 V		1.1	2.9	
				f_{MX} = 8 MHz**Note 4**, Square wave input	Normal operation	V_{DD} = 5.0 V		0.8	2.2	mA
						V_{DD} = 1.8 V		0.8	2.2	

(Notes and Remarks are listed on the next page.)

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (2/5)

Item	Symbol	Conditions					Min.	Typ.	Max.	Unit
Supply current **Note 1**	I_{DD1}	Operating mode	LS (low-speed main) mode	f_{MX} = 8 MHz**Note 4**, Resonator connection	Normal operation	V_{DD} = 5.0 V		0.9	2.4	mA
						V_{DD} = 1.8 V		0.9	2.4	

Note 1. The listed currents are the total currents flowing into V_{DD} and EV_{DD0}, including the input leakage currents flowing when the level of the input pin is fixed to V_{DD}, EV_{DD0} or V_{SS}, EV_{SS0}. The currents in the Typ. column do not include the peripheral operating current in the HS (high-speed main), LS (low-speed main), or LV (low-voltage main) mode. The currents in the Max. column include the peripheral operating current, but do not include those of the FAA, A/D converter, sample & hold circuit, D/A converter, PGA, comparator, TRNG, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors, and those flowing when the data flash memory is being rewritten.

Note 2. The listed currents apply when the high-speed system clock, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock are stopped.

Note 3. The listed currents apply when the high-speed on-chip oscillator, high-speed system clock, low-speed on-chip oscillator, and subsystem clock are stopped.

Note 4. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock are stopped.

Remark 1. f_{IH}: High-speed on-chip oscillator clock frequency

Remark 2. f_{IM}: Middle-speed on-chip oscillator clock frequency

Remark 3. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

Remark 4. f_{PLL}: PLL clock frequency (up to 96 MHz)

Remark 5. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 6. The typical value for the ambient operating temperature (T_A) is 25°C unless otherwise specified.

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (3/5)

Item	Symbol	Conditions					Min.	Typ.	Max.	Unit
Supply current **Note 1**	I_{DD1}	Operating mode	Subsystem clock operation mode	f_{SUB} = 32.768 kHz **Note 2**, Low-speed on-chip oscillator operation	Normal operation	T_A = –40°C		3.9	16.8	µA
						T_A = +25°C		4.7	17.4	
						T_A = +50°C		6.3	30.9	
						T_A = +70°C		9.7	52.3	
						T_A = +85°C		15.3	83.2	
						T_A = +105°C		30.6	177.3	
				f_{SUB} = 32.768 kHz **Note 3**, Square wave input	Normal operation	T_A = –40°C		3.5	16.3	µA
						T_A = +25°C		4.9	22.0	
						T_A = +50°C		5.9	31.7	
						T_A = +70°C		9.2	53.9	
						T_A = +85°C		14.7	81.8	
						T_A = +105°C		30.3	180.4	
				f_{SUB} = 32.768 kHz **Note 3**, Resonator connection	Normal operation	T_A = –40°C		3.6	13.4	µA
						T_A = +25°C		4.3	14.1	
						T_A = +50°C		5.8	27.2	
						T_A = +70°C		9.2	50.0	
						T_A = +85°C		14.9	79.7	
						T_A = +105°C		30.0	174.3	

Note 1. The listed currents are the total currents flowing into V_{DD} and EV_{DD0}, including the input leakage currents flowing when the level of the input pin is fixed to V_{DD}, EV_{DD0} or V_{SS}, EV_{SS0}. The currents in the Typ. and Max. columns do not include the peripheral operating current when the CPU is operating with the sub-system clock.

Note 2. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and subsystem clock are stopped. They do not include the currents flowing into the RTC, 32-bit interval timer, and watchdog timer.

Note 3. The listed currents apply when the high-speed on-chip oscillator, high-speed system clock, middle-speed on-chip oscillator, and low-speed on-chip oscillator are stopped, and the low power consumption oscillation 3 is specified (AMPHS1, AMPHS0 = 11B). They do not include the currents flowing into the RTC, 32-bit interval timer, and watchdog timer.

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

Remark 2. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (4/5)

Item	Symbol	Conditions				Min.	Typ.	Max.	Unit
Supply current**Note 1**	I_{DD2} **Note 2**	HALT mode	HS (high-speed main) mode	f_{PLL} = 96 MHz f_{CLK} = 48 MHz (MCM0 = 0)**Note 2**	V_{DD} = 5.0 V		1.57	12.84	mA
					V_{DD} = 2.4 V		1.57	12.84	
				f_{PLL} = 96 MHz f_{CLK} = 48 MHz (MCM = 1)**Note 4**	V_{DD} = 5.0 V		1.39	12.62	mA
					V_{DD} = 2.4 V		1.39	12.62	
				f_{IH} = 48 MHz**Note 2**	V_{DD} = 5.0 V		0.73	7.13	mA
					V_{DD} = 2.4 V		0.73	7.12	
				f_{PLL} = 64 MHz f_{CLK} = 32 MHz (MCM0 = 0)**Note 2**	V_{DD} = 5.0 V		1.19	8.79	mA
					V_{DD} = 1.8 V		1.18	8.78	
				f_{PLL} = 64 MHz f_{CLK} = 32 MHz (MCM = 1)**Note 4**	V_{DD} = 5.0 V		1.01	8.58	mA
					V_{DD} = 1.8 V		0.99	8.56	
				f_{IH} = 32 MHz**Note 3**	V_{DD} = 5.0 V		0.62	4.98	mA
					V_{DD} = 1.8 V		0.61	4.96	
			LS (low-speed main) mode	f_{IH} = 24 MHz**Note 3**	V_{DD} = 5.0 V		0.51	3.83	mA
					V_{DD} = 1.8 V		0.50	3.82	
				f_{IH} = 16 MHz**Note 3**	V_{DD} = 5.0 V		0.48	2.79	mA
					V_{DD} = 1.8 V		0.47	2.78	
				f_{IM} = 4 MHz**Note 4**	V_{DD} = 5.0 V		0.10	0.82	mA
					V_{DD} = 1.6 V		0.09	0.81	
			LP (low-power main) mode	f_{IM} = 2 MHz**Note 4**	V_{DD} = 5.0 V		39	493	µA
					V_{DD} = 1.6 V		40	494	
				f_{IM} = 1 MHz**Note 4**	V_{DD} = 5.0 V		32	358	µA
					V_{DD} = 1.6 V		31	357	
			HS (high-speed main) mode	f_{MX} = 20 MHz**Note 5**, Square wave input	V_{DD} = 5.0 V		0.25	3.02	mA
					V_{DD} = 1.8 V		0.23	2.99	
			LS (low-speed main) mode	f_{MX} = 20 MHz**Note 5**, Square wave input	V_{DD} = 5.0 V		0.26	3.03	mA
					V_{DD} = 1.8 V		0.23	2.99	
				f_{MX} = 20 MHz**Note 5**, Resonator connection	V_{DD} = 5.0 V		0.44	3.25	mA
					V_{DD} = 1.8 V		0.43	3.23	
				f_{MX} = 10 MHz**Note 5**, Square wave input	V_{DD} = 5.0 V		0.16	1.65	mA
					V_{DD} = 1.8 V		0.13	1.62	
				f_{MX} = 10 MHz**Note 5**, Resonator connection	V_{DD} = 5.0 V		0.30	1.82	mA
					V_{DD} = 1.8 V		0.29	1.81	
				f_{MX} = 8 MHz**Note 5**, Square wave input	V_{DD} = 5.0 V		0.14	1.37	mA
					V_{DD} = 1.8 V		0.12	1.35	
				f_{MX} = 8 MHz**Note 5**, Resonator connection	V_{DD} = 5.0 V		0.23	1.49	mA
					V_{DD} = 1.8 V		0.22	1.47	

(Notes and Remarks are listed on the next page.)

Note 1. The listed currents are the total currents flowing into V_{DD} and EV_{DD0}, including the input leakage currents flowing when the level of the input pin is fixed to V_{DD}, EV_{DD0} or V_{SS}, EV_{SS0}. The currents in the Typ. column do not include the peripheral operating current when the CPU is placed in the HS (high-speed main), LS (low-speed main), or LV (low-voltage main) mode. The currents in the Max. column include the peripheral operating current, but do not include those of the FAA, A/D converter, sample & hold circuit, D/A converter, PGA, comparator, TRNG, LVD circuit, I/O port, and on-chip pull-up-/pull-down resistors, and those flowing when the data flash memory is being rewritten. The currents in the Max. column include that of the RTC when the CPU is placed in the HALT mode.

Note 2. The listed currents apply when the HALT instruction has been fetched from the flash memory for execution.

Note 3. The listed currents apply when the high-speed system clock, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock are stopped.

Note 4. The listed currents apply when the high-speed on-chip oscillator, high-speed system clock, low-speed on-chip oscillator, and subsystem clock are stopped.

Note 5. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock are stopped.

Remark 1. f_{IH}: High-speed on-chip oscillator clock frequency

Remark 2. f_{IM}: Middle-speed on-chip oscillator clock frequency

Remark 3. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

Remark 4. f_{PLL}: PLL clock frequency (up to 96 MHz)

Remark 5. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 6. The typical value for the ambient operating temperature (T_A) is 25°C unless otherwise specified.

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (5/5)

Item	Symbol	Conditions				Min.	Typ.	Max.	Unit
Supply current **Note 1**	I_{DD2} **Note 2**	HALT mode	Subsystem clock operation mode	f_{SUB} = 32.768 kHz **Note 3**, Low-speed on-chip oscillator operation	T_A = –40°C		0.97	12.31	µA
					T_A = +25°C		1.55	12.61	
					T_A = +50°C		2.80	25.50	
					T_A = +70°C		5.54	45.88	
					T_A = +85°C		10.41	75.70	
					T_A = +105°C		23.12	165.88	
				f_{SUB} = 32.768 kHz, Square wave input **Note 4**	T_A = –40°C		0.27	11.34	µA
					T_A = +25°C		1.48	16.73	
					T_A = +50°C		2.19	26.04	
					T_A = +70°C		4.93	47.32	
					T_A = +85°C		9.37	73.70	
					T_A = +105°C		22.71	168.71	
				f_{SUB} = 32.768 kHz, Resonator connection **Note 5**	T_A = –40°C		0.40	8.83	µA
					T_A = +25°C		0.94	9.53	
					T_A = +50°C		2.16	22.41	
					T_A = +70°C		4.91	43.76	
					T_A = +85°C		9.71	72.66	
					T_A = +105°C		22.43	163.33	
	I_{DD3}	STOP mode	Realtime clock stopped **Note 6**		T_A = –40°C		0.16	10.00	µA
					T_A = +25°C		0.63	10.00	
					T_A = +50°C		1.80	20.00	
					T_A = +70°C		4.30	40.00	
					T_A = +85°C		9.30	70.00	
					T_A = +105°C		22.00	160.00	
			128-Hz realtime clock operation **Note 7**		T_A = –40°C		0.24	11.00	µA
					T_A = +25°C		0.71	11.00	
					T_A = +50°C		1.95	22.00	
					T_A = +70°C		4.60	45.00	
					T_A = +85°C		9.50	80.00	
					T_A = +105°C		23.00	170.00	

(Notes and Remarks are listed on the next page.)

Note 1. The listed currents are the total currents flowing into V_{DD} and EV_{DD0}, including the input leakage currents flowing when the level of the input pin is fixed to V_{DD}, EV_{DD0} or V_{SS}, EV_{SS0}. The currents in the Typ. and Max. columns do not include the peripheral operating current when the CPU is operating with the sub-system clock, or when the CPU is placed in the STOP mode, but include that of the RTC when in the HALT mode.

Note 2. The listed currents apply when the HALT instruction has been fetched from the flash memory for execution.

Note 3. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and subsystem clock are stopped, including the current flowing into the RTC. They do not include the currents flowing into the 32-bit interval timer and watchdog timer.

Note 4. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and low-speed on-chip oscillator are stopped, including the current flowing into the RTC. They do not include the currents flowing into the 32-bit interval timer and watchdog timer.

Note 5. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and low-speed on-chip oscillator are stopped, and when RTCLPC is set to 1 and the low power consumption oscillation 3 is specified (AMPHS1, AMPHS0 = 11B), including the current flowing into the RTC. They do not include the currents flowing into the 32-bit interval timer and watchdog timer.

Note 6. The listed currents apply when the low-speed on-chip oscillator and subsystem clock oscillation are stopped. They do not include the currents flowing into the RTC, 32-bit interval timer and watchdog timer. For the current for operation of the subsystem clock in the STOP mode, refer to that in the HALT mode.

Note 7. The listed currents apply when the low-speed on-chip oscillator is stopped, and when RTCLPC is set to 1 and the low power consumption oscillation 3 is specified (AMPHS1, AMPHS0 = 11B), including the current flowing into the RTC. They do not include the currents flowing into the 32-bit interval timer and watchdog timer.

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

Remark 2. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

Peripheral Functions (Common to all products)

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (1/2)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
High-speed on-chip oscillator operating current	I_{FIH} **Note 1**	HIPREC = 0			380	—	μA
		HIPREC = 1			240	—	μA
Middle-speed on-chip oscillator operating current	I_{FIM} **Note 1**				20	—	μA
Low-speed on-chip oscillator operating current	I_{FIL} **Note 1**				0.3	—	μA
RTC operating current	I_{RTC} **Notes 1, 2, 3**	f_{RTCCLK} = 32.768 kHz			0.005	—	μA
		f_{RTCCLK} = 128 Hz			0.002	—	μA
32-bit interval timer operating current	I_{IT} **Notes 1, 2, 4**				0.04	—	μA
Watchdog timer operating current	I_{WDT} **Notes 1, 2, 5**	f_{IL} = 32.768 kHz (typ.)			0.32	—	μA
A/D converter operating current	I_{ADC} **Notes 1, 6**	Conversion at maximum speed	Normal mode, AV_{REFP} = V_{DD} = 5.0 V		0.95	1.6	mA
			Low-voltage mode, AV_{REFP} = V_{DD} = 3.0 V		0.54	0.81	mA
AV_{REFP} current	I_{ADREF} **Note 7**	AV_{REFP} = 5.0 V			60	—	μA
A/D converter internal reference voltage current	I_{ADREF} **Note 1**				114	—	μA
Temperature sensor operating current	I_{TMPS} **Note 1**				110	—	μA
D/A converter operating current	I_{DAC} **Notes 1, 8**	Per channel	10-bit DAC, V_{DD} = 5.0 V		223	—	μA
			8-bit DAC, V_{DD} = 5.0 V		120	—	μA
Comparator operating current	I_{CMP} **Notes 1, 9**	Per channel			100	—	μA
<R> PGA operating current	I_{PGA} **Notes 1, 10**				460	—	μA
Sample & hold circuit operating current	I_{SH} **Notes 1, 11**	Per channel			800	—	μA
LVD operating current	I_{LVD0} **Notes 1, 12**				0.03	—	μA
	I_{LVD1} **Notes 1, 12**				0.03	—	μA
FAA operating current	I_{FAA} **Notes 1, 13**	f_{CLK} = 48 MHz			11.0	—	mA
		f_{CLK} = 32 MHz			7.3	—	mA
True random number generator operating current	I_{TRNG}				1.6	—	mA
SMBus operating current	I_{SMBUS}				250	—	μA
Self-programming operating current	I_{FSP} **Notes 1, 14**				2.5	12.2	mA
Data flash rewrite operating current	I_{BGO} **Notes 1, 15**				2.5	12.2	mA

(Notes and Remarks are listed on the next page.)

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (2/2)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
SNOOZE operating current	I_{SNOZ} **Note 1**	ADC to be in use	The ADC is shifting to the SNOOZE mode. **Note 16**		0.7	1.2	mA
			The ADC is operating in the low-voltage mode, AV_{REFP} = V_{DD} = 3.0 V		1.2	2.0	
		Simplified SPI (CSI)/UART to be in use			0.7	1.07	
Low-speed peripheral clock supply current	I_{SXP} **Notes 1, 17**	RTCLPC = 0			0.27	—	μA
Output current control operating current	I_{CCDA} **Notes 1, 18**	The setting of the CCDE register is not 00H.			100	—	μA
	I_{CCDP} **Notes 19, 20**	Per single controlled current drive port	Low-level output current setting: Hi-Z		30	—	μA
			Low-level output current setting: 2 to 15 mA		210	—	μA

Note 1. This current flows into V_{DD}.

Note 2. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, and high-speed system clock are stopped.

Note 3. This current only flows to the realtime clock (RTC). It does not include the operating current of the low-speed on-chip oscillator or the XT1 oscillator. The supply current of the RL78 microcontrollers is the sum of I_{DD1} or I_{DD2}, and I_{RTC} when the realtime clock is operating in the operation mode or HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be included in the supply current. I_{DD2} in the subsystem clock operation mode includes the operating current of the realtime clock.

Note 4. This current only flows to the 32-bit interval timer. It does not include the operating current of the low-speed on-chip oscillator or the XT1 oscillator. The supply current of the RL78 microcontrollers is the sum of I_{DD1} or I_{DD2}, and I_{IT} when the 32-bit interval timer is operating or in the HALT mode.

Note 5. This current only flows to the watchdog timer. It includes the operating current of the low-speed on-chip oscillator. The supply current of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3}, and I_{WDT} when the watchdog timer is operating.

Note 6. This current only flows to the A/D converter. The supply current of the RL78 microcontrollers is the sum of I_{DD1} or I_{DD2}, and I_{ADC} when the A/D converter is operating or in the HALT mode.

Note 7. This current flows into AV_{REFP}.

Note 8. This current only flows to the D/A converter. The supply current of the RL78 microcontrollers is the sum of I_{DD1} or I_{DD2}, and I_{DAC} when the D/A converter is operating.

Note 9. This current only flows to the comparator circuit. The supply current of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3}, and I_{CMP} when the comparator circuit is operating.

Note 10. This current only flows to the PGA circuit. The supply current of the RL78 microcontrollers is the sum of I_{DD1} or I_{DD2}, and I_{PGA} when the PGA circuit is operating.

Note 11. This current only flows to the sample & hold circuit. The supply current of the RL78 microcontrollers is the sum of I_{DD1} or I_{DD2}, and I_{SH} when the sample & hold circuit is operating.

Note 12. This current only flows to the LVD circuit. The supply current of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3}, and I_{LVD} when the LVD circuit is operating.

Note 13. This current only flows to the FAA circuit. The supply current of the RL78 microcontrollers is the sum of I_{DD1} or I_{DD2}, and I_{FAA} when the FAA circuit is operating.

Note 14. This current only flows during self-programming.

Note 15. This current only flows while the data flash memory is being rewritten.

Note 16. For shift time to the SNOOZE mode, see **20.9 SNOOZE Mode Function**.

Note 17. This current is added to the supply current in the STOP mode when the setting of RTCLPC is 0 with the subsystem clock X (f_{SX}) oscillating, or in the HALT mode when the setting of RTCLPC is 0 with the subsystem clock X (f_{SX}) selected as the CPU clock.

Note 18. This current is added to the supply current when the controlled current drive port is set.

Note 19. This current does not include the current flowing into the I/O ports.

Note 20. This current flows into EV_{DD0} and EV_{DD1}.

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

Remark 2. f_{SX}: Subsystem clock X frequency

Remark 3. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 4. The typical value for the ambient operating temperature is 25°C.

43.4 AC Characteristics

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (1/2)

Item	Symbol	Conditions			Min.	Typ.	Max.	Unit
Instruction cycle	T_{CY}	Main system clock (f_{MAIN}) operation	HS (high-speed main) mode (Prefetch ON)	2.4 V ≤ V_{DD} ≤ 5.5 V	0.02083		1	μs
			HS (high-speed main) mode (Prefetch OFF)	1.8 V ≤ V_{DD} ≤ 5.5 V	0.03125		1	μs
				1.6 V ≤ V_{DD} ≤ 1.8 V	0.25		1	μs
			LS (low-speed main) mode	1.8 V ≤ V_{DD} ≤ 5.5 V	0.04167		1	μs
				1.6 V ≤ V_{DD} ≤ 1.8 V	0.25		1	μs
			LP (low-power main) mode	1.6 V ≤ V_{DD} ≤ 5.5 V	0.5		1	μs
		Subsystem clock (f_{SUB}) operation		1.8 V ≤ V_{DD} ≤ 5.5 V	26.041	30.5	31.3	μs
		Self-programming mode	HS (high-speed main) mode	1.8 V ≤ V_{DD} ≤ 5.5 V	0.03125		1	μs
				2.4 V ≤ V_{DD} ≤ 5.5 V	0.02083		1	μs
			LS (low-speed main) mode	1.8 V ≤ V_{DD} ≤ 5.5 V	0.04167		1	μs
External system clock frequency	f_{EX}	1.8 V ≤ V_{DD} ≤ 5.5 V			1.0		20.0	MHz
		1.6 V ≤ V_{DD} < 1.8 V			1.0		4.0	MHz
	f_{EXS}				32		38.4	kHz
External system clock input high-level width, low-level width	t_{EXH}, t_{EXL}	1.8 V ≤ V_{DD} ≤ 5.5 V			24			ns
		1.6 V ≤ V_{DD} < 1.8 V			120			ns
	t_{EXHS}, t_{EXLS}				13.7			μs
TI00 to TI03 input high-level width, low-level width	t_{TIH}, t_{TIL}				1/f_{MCK} + 10			ns **Note**
Timer RJ input cycle	t_C	TRJIO		2.7 V ≤ EV_{DD0} ≤ 4.0 V	100			ns
				1.8 V ≤ EV_{DD0} ≤ 2.7 V	300			ns
				1.6 V ≤ EV_{DD0} ≤ 1.8 V	500			ns
Timer RJ input high-level width, low-level width	t_{TJIH}, t_{TJIL}	TRJIO		2.7 V ≤ EV_{DD0} ≤ 4.0 V	40			ns
				1.8 V ≤ EV_{DD0} ≤ 2.7 V	120			ns
				1.6 V ≤ EV_{DD0} ≤ 1.8 V	200			ns
Timer RD2 input high-level width, low-level width	t_{TDIH}, t_{TDIL}	TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1, TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1			3/f_{CLK}			ns
Timer RD2 forcible shut-off signal input low-level width	t_{TDSIL}	P137/INTP0		2 MHz ≤ f_{CLK} ≤ 48 MHz	1			μs
				f_{CLK} ≤ 2 MHz	1/f_{CLK} + 1			μs

(Note and Remark are listed on the next page.)

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (2/2)

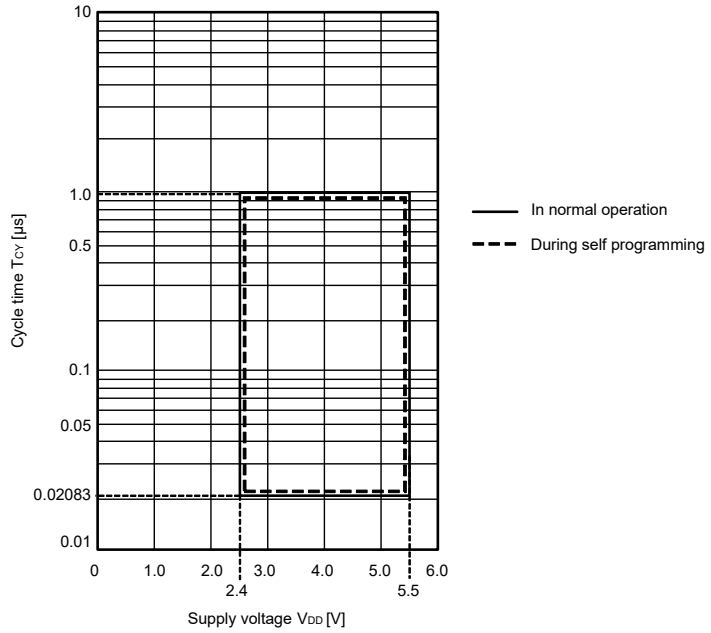
Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Timer RG2 input high-level width, low-level width	t_{TGIH}, t_{TGIL}	TRGIOA, TRGIOB, TRGIDZ, TRGTRG		2.5/f_{CLK}			ns
TO00 to TO03 TKBO00, TKBO01, TKBO10, TKBO11, TKBO20, TKBO21, TRJIO0, TRJO0, TRGIOA, TRGIOB, TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1, TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1 output frequency	f_{TO}	HS (high-speed main) mode LS (low-speed main) mode	4.0 V ≤ EV_{DD0} ≤ 5.5 V			16	MHz
			2.7 V ≤ EV_{DD0} < 4.0 V			8	MHz
			1.8 V ≤ EV_{DD0} < 2.7 V			4	MHz
			1.6 V ≤ EV_{DD0} < 1.8 V			2	MHz
		LP (low-power main) mode	1.6 V ≤ EV_{DD0} ≤ 5.5 V			2	MHz
PCLBUZ0, PCLBUZ1 output frequency	f_{PCL}	HS (high-speed main) mode LS (low-speed main) mode	4.0 V ≤ EV_{DD0} ≤ 5.5 V			16	MHz
			2.7 V ≤ EV_{DD0} < 4.0 V			8	MHz
			1.8 V ≤ EV_{DD0} < 2.7 V			4	MHz
			1.6 V ≤ EV_{DD0} < 1.8 V			2	MHz
		LP (low-power main) mode	1.6 V ≤ EV_{DD0} < 1.8 V			2	MHz
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}	INTP0, INTP20, INTP21	1.6 V ≤ V_{DD} ≤ 5.5 V	1			µs
		INTP1 to INTP11	1.6 V ≤ EV_{DD0} ≤ 5.5 V	1			µs
Key interrupt input high-level width, low-level width	t_{KRH}, t_{KRL}	KR0 to KR7	1.8 V ≤ EV_{DD0} ≤ 5.5 V	250			ns
			1.6 V ≤ EV_{DD0} < 1.8 V	1			µs
$\overline{\text{RESET}}$ low-level width	t_{RSL}			10			µs

Note The following conditions are required for low-voltage interface when EV_{DD0} < V_{DD}.
1.8 V ≤ EV_{DD0} < 2.7 V: 125 ns (min.)
1.6 V ≤ EV_{DD0} < 1.8 V: 250 ns (min.)

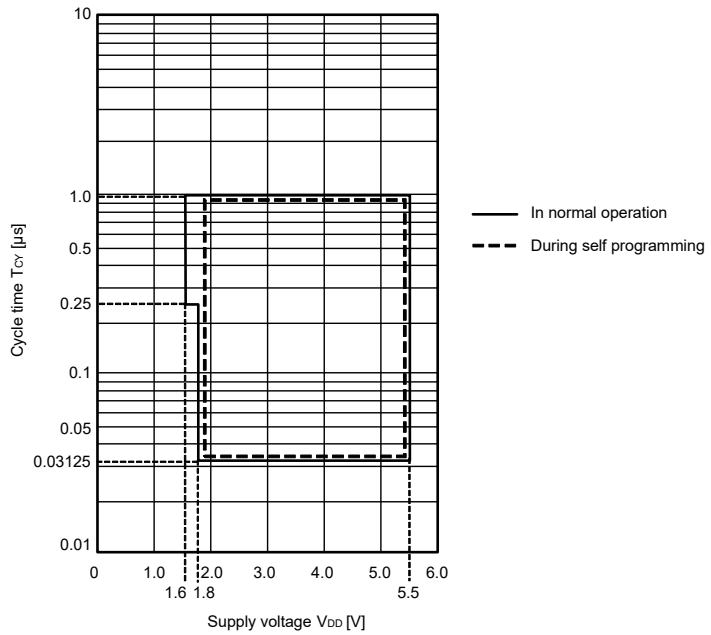
Remark f_{MCK}: Timer array unit operating clock frequency
To set this operating clock, use the CKSmn0 and CKSmn1 bits of the timer mode register mn (TMRmn).
m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Minimum Instruction Execution Time during Main System Clock Operation

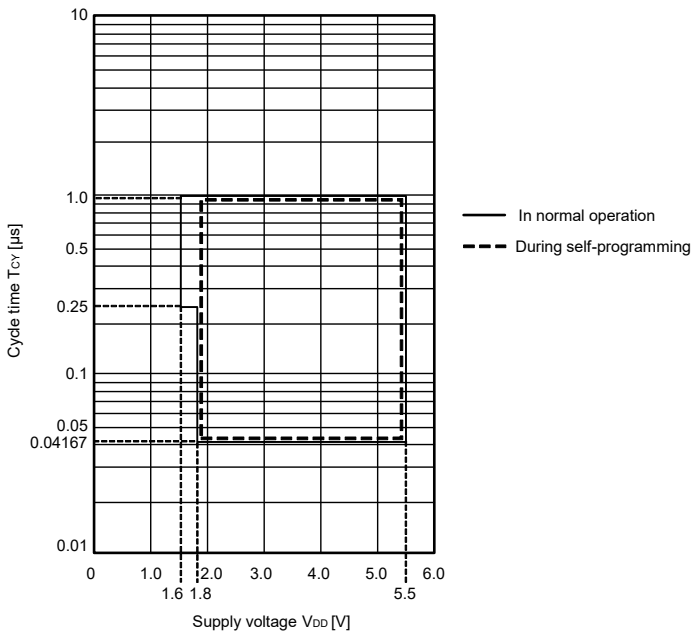
T_{CY} vs V_{DD} (HS (high-speed main) mode) prefetch ON



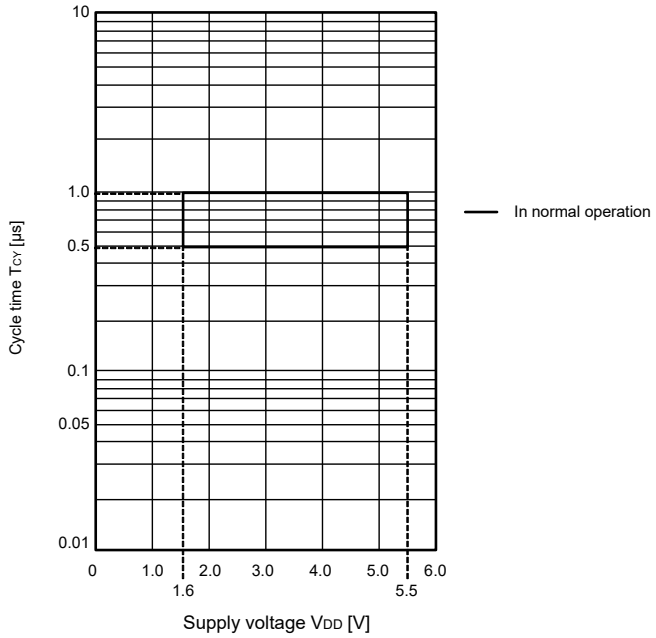
T_{CY} vs V_{DD} (HS (high-speed main) mode) prefetch OFF



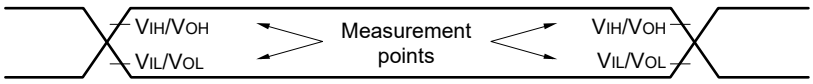
T_{CY} vs V_{DD} (LS (low-speed main) mode)



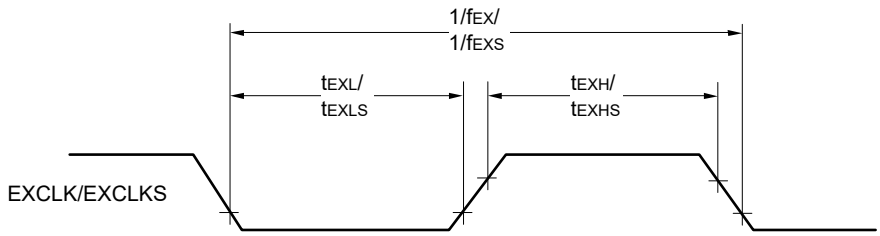
T_{CY} vs V_{DD} (LP (low-power main) mode)



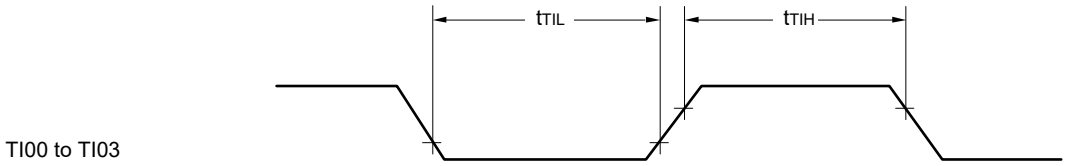
AC Timing Measurement Points

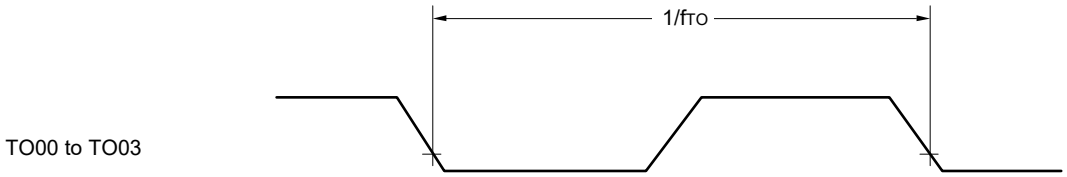


External System Clock Timing



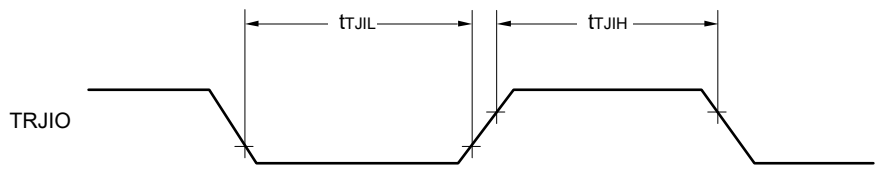
TI/TO Timing



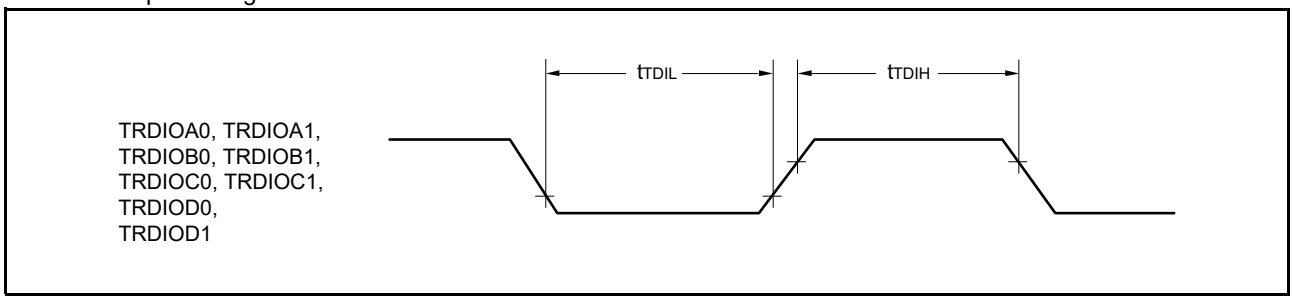


TKBO00, TKBO01, TKBO10, TKBO11,
TKBO20, TKBO21,
TRJIO0, TRJO0, TRGIOA, TRGIOB,
TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1,
TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1

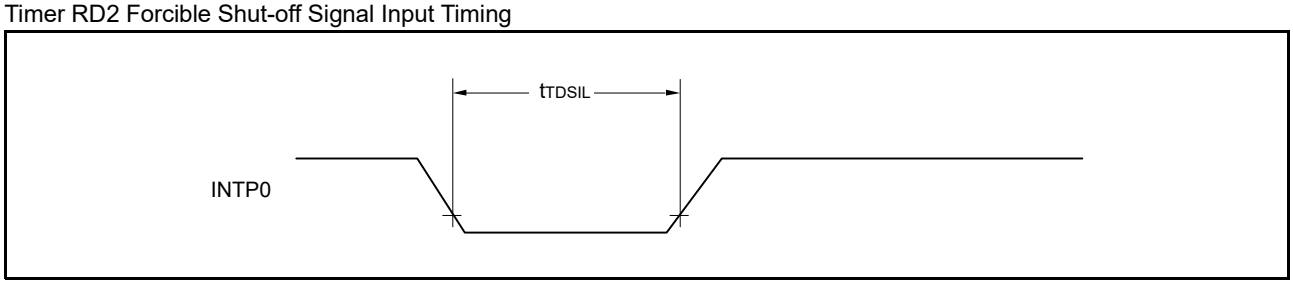
Timer RJ Input Timing



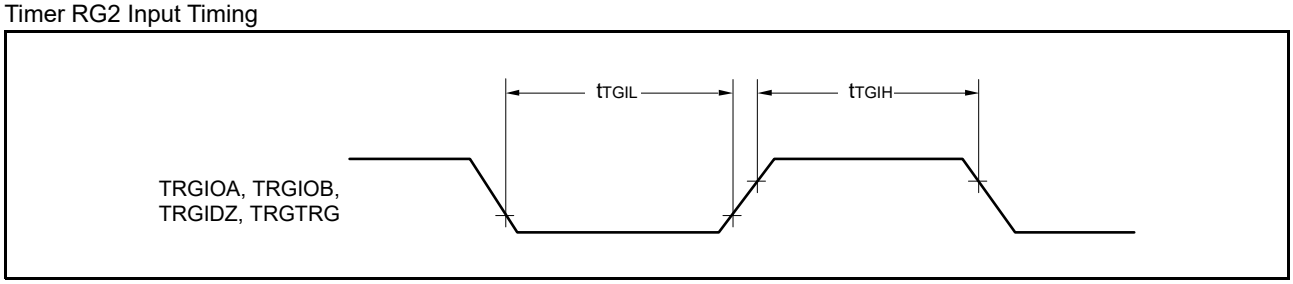
Timer RD2 Input Timing



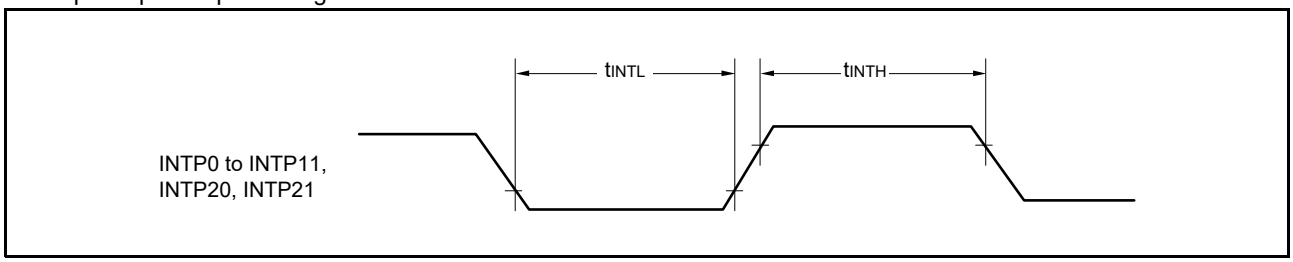
Timer RD2 Forcible Shut-off Signal Input Timing



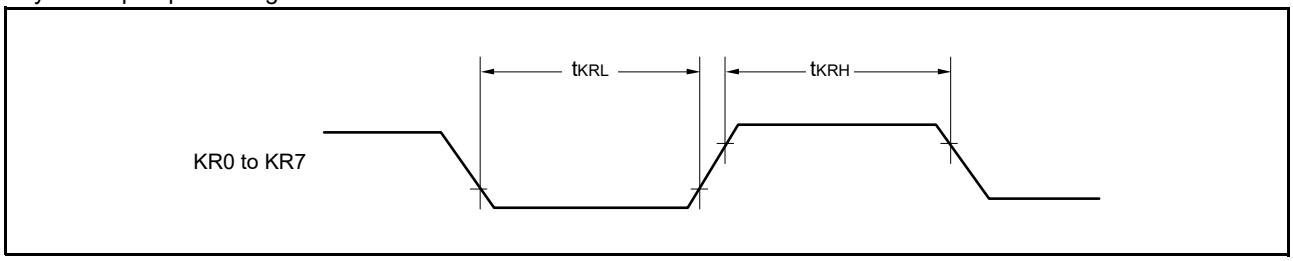
Timer RG2 Input Timing



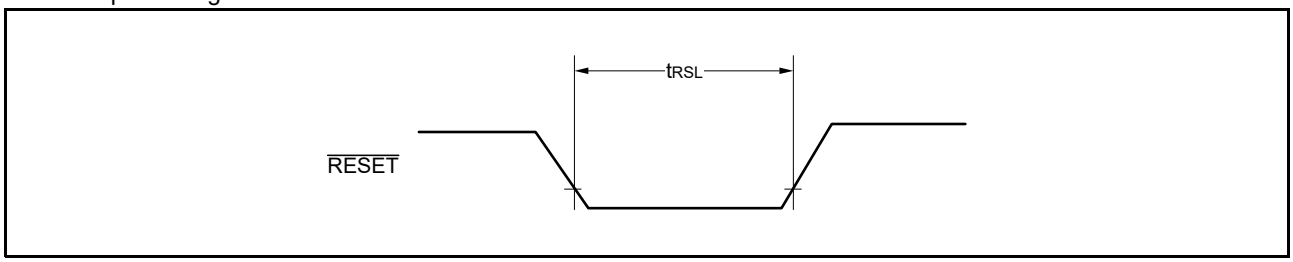
Interrupt Request Input Timing



Key Interrupt Input Timing

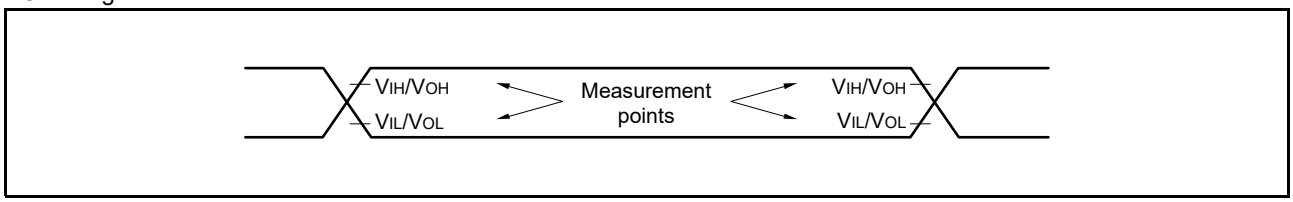


$\overline{\text{RESET}}$ Input Timing



43.5 Characteristics of the Peripheral Functions

AC Timing Measurement Points



43.5.1 Serial array unit

1. In UART communications with devices operating at same voltage levels

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz)

Item	Symbol	Conditions	HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
Transfer rate **Note 1**		1.6 V ≤ EV_{DD0} ≤ 5.5 V		f_{MCK}/6 **Note 2**		f_{MCK}/6 **Note 2**		f_{MCK}/6	bps
		Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} **Note 3**		5.3		4		0.33	Mbps

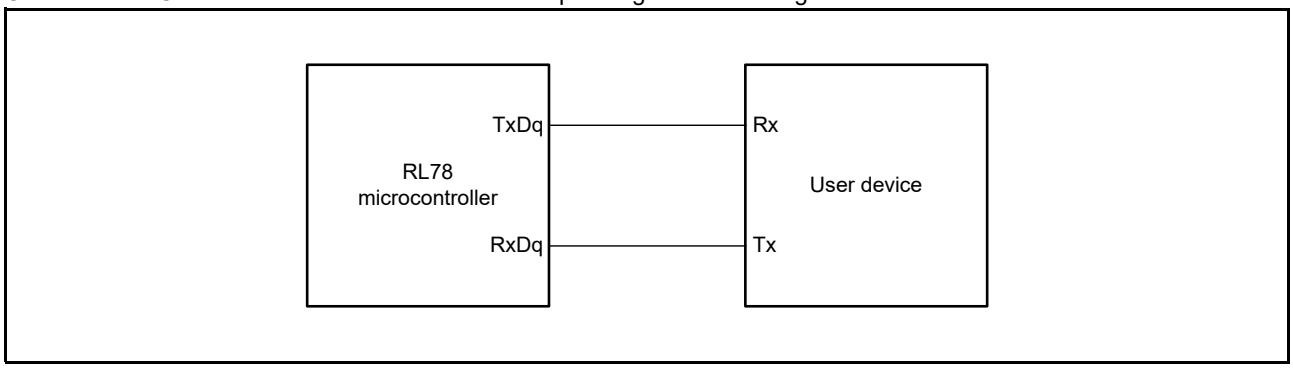
Note 1. The transfer rate in the SNOOZE mode is within the range from 4800 to 9600 bps.

Note 2. The following conditions are required for low-voltage interface when EV_{DD0} < V_{DD}.
2.4 V ≤ EV_{DD0} < 2.7 V: 2.6 Mbps (max.)
1.8 V ≤ EV_{DD0} < 2.4 V: 1.3 Mbps (max.)
1.6 V ≤ EV_{DD0} < 1.8 V: 0.6 Mbps (max.)

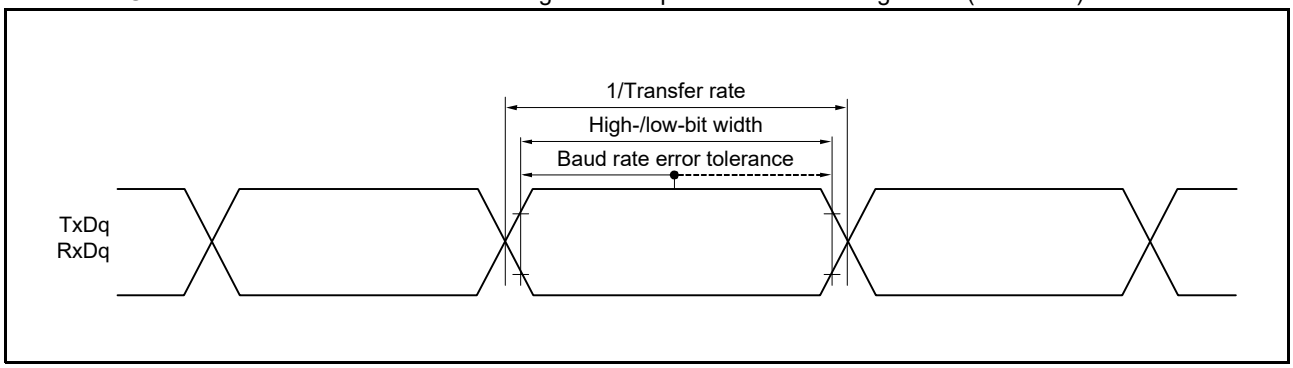
Note 3. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are as follows.
HS (high-speed main) mode: 48 MHz (2.4 V ≤ V_{DD} ≤ 5.5 V)
32 MHz (1.8 V ≤ V_{DD} ≤ 5.5 V)
4 MHz (1.6 V ≤ V_{DD} ≤ 5.5 V)
LS (low-speed main) mode: 24 MHz (1.8 V ≤ V_{DD} ≤ 5.5 V)
4 MHz (1.6 V ≤ V_{DD} ≤ 5.5 V)
LP (low-power main) mode: 2 MHz (1.6 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).

Connection in UART communications with devices operating at same voltage levels



Bit width in UART communications when interfacing devices operate at same voltage level (reference)



Remark 1. q: UART number (q = 0 to 2), g: PIM and POM number (g = 0, 1, 3, 5, 7)

Remark 2. f_{MCK}: Serial array unit operation clock frequency
To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00 to 03, 10, 11)

2. In simplified SPI (CSI) communications in the master mode with devices operating at same voltage levels with the internal SCKp clock (the ratings below are only applicable to CSI00)

(T_A = –40 to +85°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz)

Item	Symbol	Conditions		HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 2/f_{CLK}	4.0 V ≤ EV_{DD0} ≤ 5.5 V	62.5		83.3		1000		ns
			2.7 V ≤ EV_{DD0} ≤ 5.5 V	83.3		125		1000		ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY1}/2 – 7		t_{KCY1}/2 – 10		t_{KCY1}/2 – 50		ns
		2.7 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY1}/2 – 10		t_{KCY1}/2 – 15		t_{KCY1}/2 – 50		ns
SIp setup time (to SCKp↑)**Note 1**	t_{SIK1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V		23		33		110		ns
		2.7 V ≤ EV_{DD0} ≤ 5.5 V		33		50		110		ns
SIp hold time (from SCKp↑) **Note 1**	t_{KSI1}	2.7 V ≤ EV_{DD0} ≤ 5.5 V		10		10		10		ns
Delay time from SCKp↓ to SOp output**Note 2**	t_{KSO1}	C = 20 pF**Note 3**			10		10		10	ns

Note 1. The setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The setting for the SIp setup time becomes “to SCKp↓” and that for the SIp hold time becomes “from SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The setting for the delay time to SOp output becomes “from SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. C is the load capacitance of the SCKp and SOp output lines.

Caution **Select the normal input buffer for the SIp pin and normal output mode for the SOp and SCKp pins by using the port input mode register g (PIMg) and the port output mode register g (POMg).**

Remark 1. The listed values are only valid when the peripheral I/O redirect function of CSI00 is not in use.

Remark 2. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM number (g = 1)

Remark 3. f_{MCK}: Serial array unit operating clock frequency
To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00)

3. In simplified SPI (CSI) communications in the master mode with devices operating at same voltage levels with the internal SCKp clock

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz)

Item	Symbol	Conditions		HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 4/f_{CLK}	2.7 V ≤ EV_{DD0} ≤ 5.5 V	125		166		2000		ns
			2.4 V ≤ EV_{DD0} ≤ 5.5 V	250		250		2000		ns
			1.8 V ≤ EV_{DD0} ≤ 5.5 V	500		500		2000		ns
			1.6 V ≤ EV_{DD0} ≤ 5.5 V	1000		1000		2000		ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY1}/2 – 12		t_{KCY1}/2 – 21		t_{KCY1}/2 – 50		ns
		2.7 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY1}/2 – 18		t_{KCY1}/2 – 25		t_{KCY1}/2 – 50		ns
		2.4 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY1}/2 – 38		t_{KCY1}/2 – 38		t_{KCY1}/2 – 50		ns
		1.8 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		ns
		1.6 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY1}/2 – 100		t_{KCY1}/2 – 100		t_{KCY1}/2 – 100		ns
SIp setup time (to SCKp↑)**Note 1**	t_{SIK1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V		44		54		110		ns
		2.7 V ≤ EV_{DD0} ≤ 5.5 V		44		54		110		ns
		2.4 V ≤ EV_{DD0} ≤ 5.5 V		75		75		110		ns
		1.8 V ≤ EV_{DD0} ≤ 5.5 V		110		110		110		ns
		1.6 V ≤ EV_{DD0} ≤ 5.5 V		220		220		220		ns
SIp hold time (from SCKp↑) **Note 1**	t_{KSI1}	1.6 V ≤ EV_{DD0} ≤ 5.5 V		19		19		19		ns
Delay time from SCKp↓ to SOp output**Note 2**	t_{KSO1}	1.6 V ≤ EV_{DD0} ≤ 5.5 V C = 30 pF**Note 3**			25		25		25	ns

Note 1. The setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The setting for the SIp setup time becomes “to SCKp↓” and that for the SIp hold time becomes “from SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The setting for the delay time to SOp output becomes “from SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. C is the load capacitance of the SCKp and SOp output lines.

Caution Select the normal input buffer for the SIp pin and normal output mode for the SOp and SCKp pins by using the port input mode register g (PIMg) and the port output mode register g (POMg).

Remark 1. p: CSI number (p = 00, 01, 10, 11, 20, 21), m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), g: PIM and POM number (g = 0, 1, 3, 5, 7)

Remark 2. f_{MCK}: Serial array unit operating clock frequency
To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00 to 03, 10, 11)

4. In simplified SPI (CSI) communications in the slave mode with devices operating at same voltage levels with the external SCKp clock

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz) (1/2)

Item	Symbol	Conditions		HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
SCKp cycle time **Note 4**	t_{KCY2}	4.0 V ≤ EV_{DD0} ≤ 5.5 V	20 MHz < f_{MCK}	8/f_{MCK}		8/f_{MCK}		—		ns
			f_{MCK} ≤ 20 MHz	6/f_{MCK}		6/f_{MCK}		6/f_{MCK}		ns
		2.7 V ≤ EV_{DD0} ≤ 5.5 V	16 MHz < f_{MCK}	8/f_{MCK}		8/f_{MCK}		—		ns
			f_{MCK} ≤ 16 MHz	6/f_{MCK}		6/f_{MCK}		6/f_{MCK}		ns
		2.4 V ≤ EV_{DD0} ≤ 5.5 V		6/f_{MCK} and 500		6/f_{MCK} and 500		6/f_{MCK} and 500		ns
		1.8 V ≤ EV_{DD0} ≤ 5.5 V		6/f_{MCK} and 750		6/f_{MCK} and 750		6/f_{MCK} and 750		ns
		1.6 V ≤ EV_{DD0} ≤ 5.5 V		6/f_{MCK} and 1500		6/f_{MCK} and 1500		6/f_{MCK} and 1500		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY2}/2 – 7		t_{KCY2}/2 – 7		t_{KCY2}/2 – 7		ns
		2.7 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY2}/2 – 8		t_{KCY2}/2 – 8		t_{KCY2}/2 – 8		ns
		1.8 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY2}/2 – 18		t_{KCY2}/2 – 18		t_{KCY2}/2 – 18		ns
		1.6 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY2}/2 – 66		t_{KCY2}/2 – 66		t_{KCY2}/2 – 66		ns
SIp setup time (to SCKp↑) **Note 1**	t_{SIK2}	2.7 V ≤ EV_{DD0} ≤ 5.5 V		1/f_{MCK} + 20		1/f_{MCK} + 30		1/f_{MCK} + 30		ns
		1.8 V ≤ EV_{DD0} ≤ 5.5 V		1/f_{MCK} + 30		1/f_{MCK} + 30		1/f_{MCK} + 30		ns
		1.6 V ≤ EV_{DD0} ≤ 5.5 V		1/f_{MCK} + 40		1/f_{MCK} + 40		1/f_{MCK} + 40		ns
SIp hold time (to SCKp↑) **Note 1**	t_{SIK2}	1.8 V ≤ EV_{DD0} ≤ 5.5 V		1/f_{MCK} + 31		1/f_{MCK} + 31		1/f_{MCK} + 31		ns
		1.6 V ≤ EV_{DD0} ≤ 5.5 V		1/f_{MCK} + 250		1/f_{MCK} + 250		1/f_{MCK} + 250		ns

(Notes, Caution, and Remarks are listed on the next page.)

4. In simplified SPI (CSI) communications in the slave mode with devices operating at same voltage levels with the external SCKp clock

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz) (2/2)

Item	Symbol	Conditions		HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
Delay time from SCKp↓ to SOp output **Note 2**	t_{KSO2}	C = 30 pF **Note 3**	2.7 V ≤ EV_{DD0} ≤ 5.5 V		2/f_{MCK} + 44		2/f_{MCK} + 110		2/f_{MCK} + 110	ns
			2.4 V ≤ EV_{DD0} ≤ 5.5 V		2/f_{MCK} + 75		2/f_{MCK} + 110		2/f_{MCK} + 110	ns
			1.8 V ≤ EV_{DD0} ≤ 5.5 V		2/f_{MCK} + 110		2/f_{MCK} + 110		2/f_{MCK} + 110	ns
			1.6 V ≤ EV_{DD0} ≤ 5.5 V		2/f_{MCK} + 220		2/f_{MCK} + 220		2/f_{MCK} + 220	ns

Note 1. The setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The setting for the SIp setup time becomes “to SCKp ↓” and that for the SIp hold time becomes “from SCKp ↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The setting for the delay time to SOp output becomes “from SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. C is the load capacitance of the SOp output line.

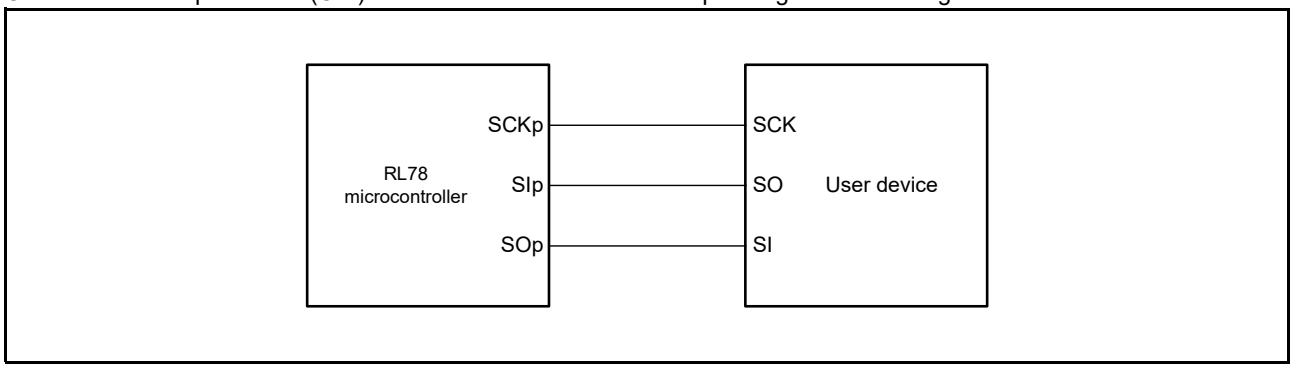
Note 4. The transfer rate in the SNOOZE mode is 1 Mbps maximum.

Caution Select the normal input buffer for the SIp and SCKp pins and normal output mode for the SOp pin by using the port input mode register g (PIMg) and the port output mode register g (POMg).

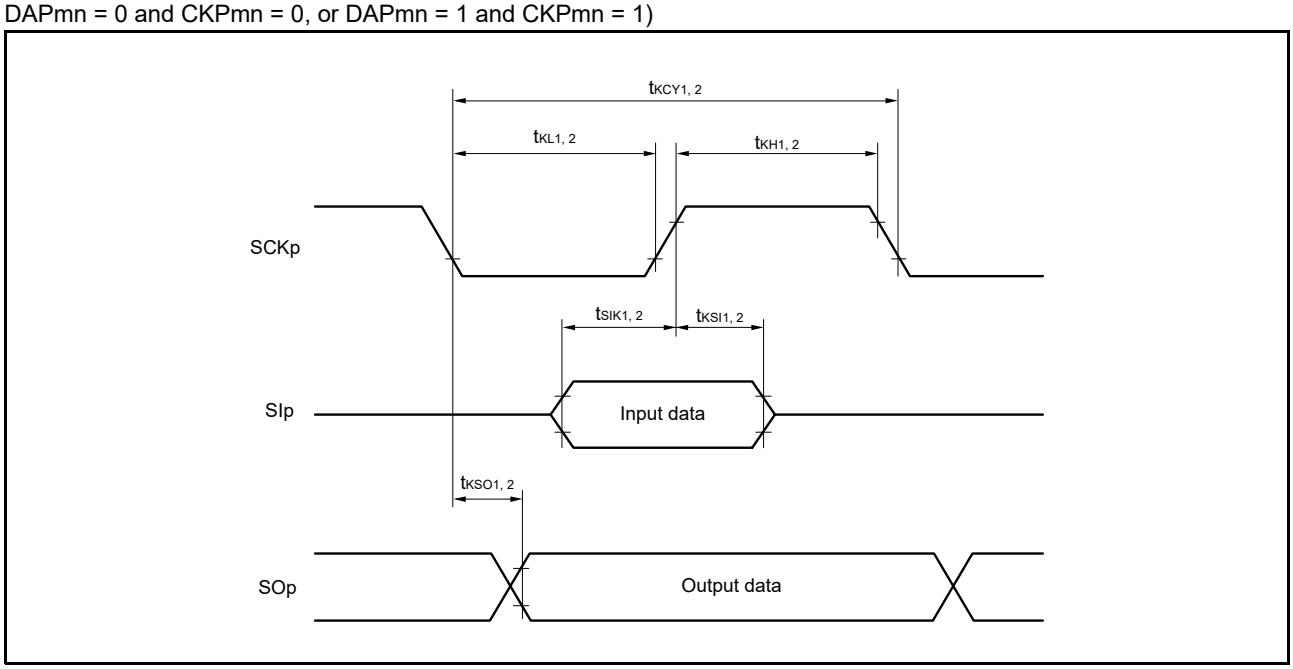
Remark 1. p: CSI number (p = 00, 01, 10, 11, 20, 21), m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), g: PIM and POM number (g = 0, 1, 3, 5, 7)

Remark 2. f_{MCK}: Serial array unit operating clock frequency
To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00 to 03, 10, 11)

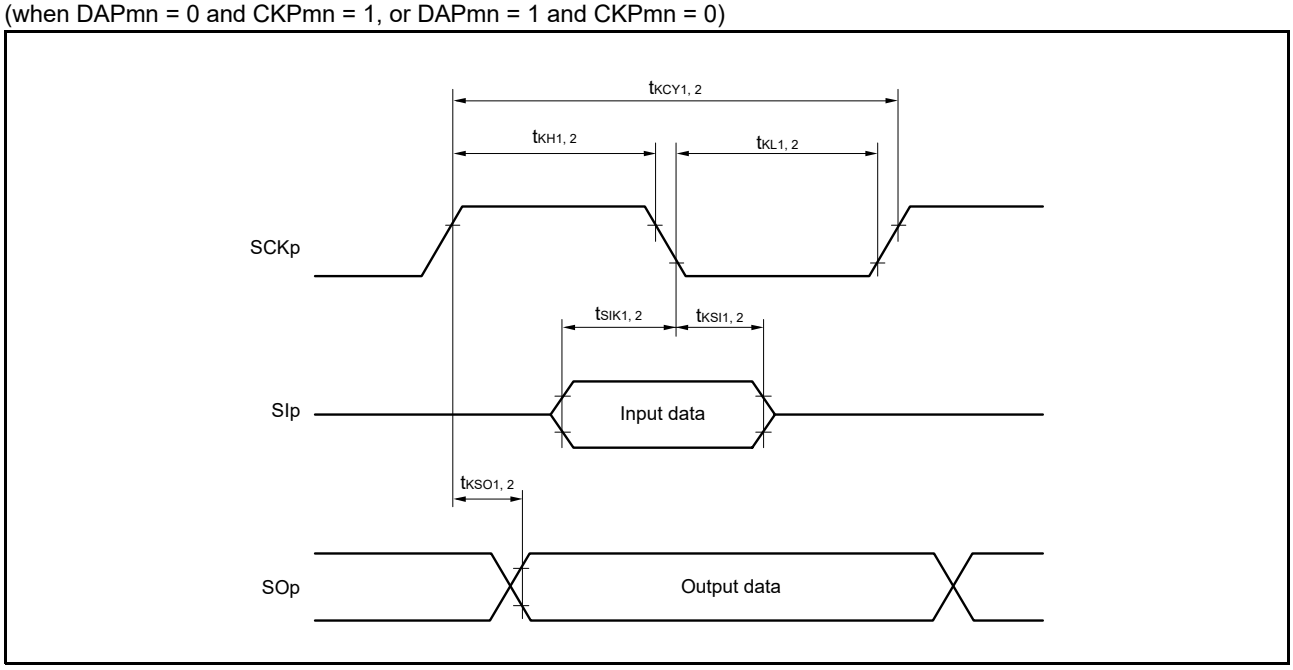
Connection in simplified SPI (CSI) communications with devices operating at same voltage levels



Timing of serial transfer in simplified SPI (CSI) communications with devices operating at same voltage levels (when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)



Timing of serial transfer in simplified SPI (CSI) communications with devices operating at same voltage levels (when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)



Remark 1. p: CSI number (p = 00, 01, 10, 11, 20, 21)

Remark 2. m: Unit number, n: Channel number (mn = 00 to 03, 10, 11)

5. In simplified I^2C communications with devices operating at same voltage levels

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz) (1/2)

Item	Symbol	Conditions	HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
SCLr clock frequency	f_{SCL}	2.7 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ		1000 Note 1		1000 Note 1		400 Note 1	kHz
		1.8 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ		400 Note 1		400 Note 1		400 Note 1	kHz
		1.8 V ≤ EV_{DD0} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ		300 Note 1		300 Note 1		300 Note 1	kHz
		1.6 V ≤ EV_{DD0} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ		250 Note 1		250 Note 1		250 Note 1	kHz
Hold time when SCLr is low	t_{LOW}	2.7 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	475		475		1150		ns
		1.8 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1150		1150		1150		ns
		1.8 V ≤ EV_{DD0} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	1550		1550		1550		ns
		1.6 V ≤ EV_{DD0} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	1850		1850		1850		ns
Hold time when SCLr is high	t_{HIGH}	2.7 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	475		475		1150		ns
		1.8 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1150		1150		1150		ns
		1.8 V ≤ EV_{DD0} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	1550		1550		1550		ns
		1.6 V ≤ EV_{DD0} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	1850		1850		1850		ns
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	1/f_{MCK} + 85 Note 2		1/f_{MCK} + 85 Note 2		1/f_{MCK} + 145 Note 2		ns
		1.8 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1/f_{MCK} + 145 Note 2		1/f_{MCK} + 145 Note 2		1/f_{MCK} + 145 Note 2		ns
		1.8 V ≤ EV_{DD0} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	1/f_{MCK} + 230 Note 2		1/f_{MCK} + 230 Note 2		1/f_{MCK} + 230 Note 2		ns
		1.6 V ≤ EV_{DD0} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	1/f_{MCK} + 290 Note 2		1/f_{MCK} + 290 Note 2		1/f_{MCK} + 290 Note 2		ns

(Notes and Caution are listed on the next page, and Remarks are listed on page 1881.)

5. In simplified I^2C communications with devices operating at same voltage levels

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz) (2/2)

Item	Symbol	Conditions	HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
Data hold time (transmission)	$t_{HD:DAT}$	2.7 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	0	305	0	305	0	305	ns
		1.8 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	0	355	0	355	0	355	ns
		1.8 V ≤ EV_{DD0} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	0	405	0	405	0	405	ns
		1.6 V ≤ EV_{DD0} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	0	405	0	405	0	405	ns

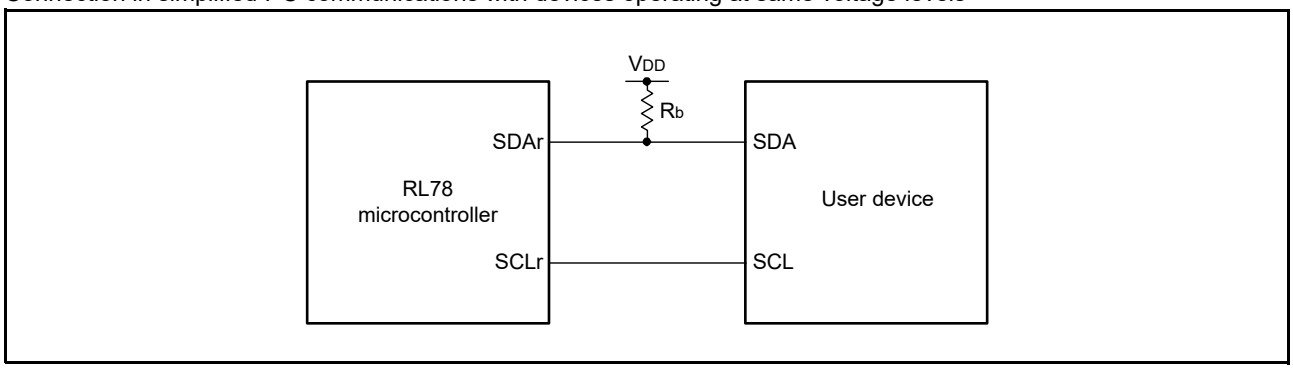
Note 1. The listed frequencies must be no greater than f_{MCK}/4.

Note 2. Set the f_{MCK} value that does not exceed the hold time when SCLr is low or high.

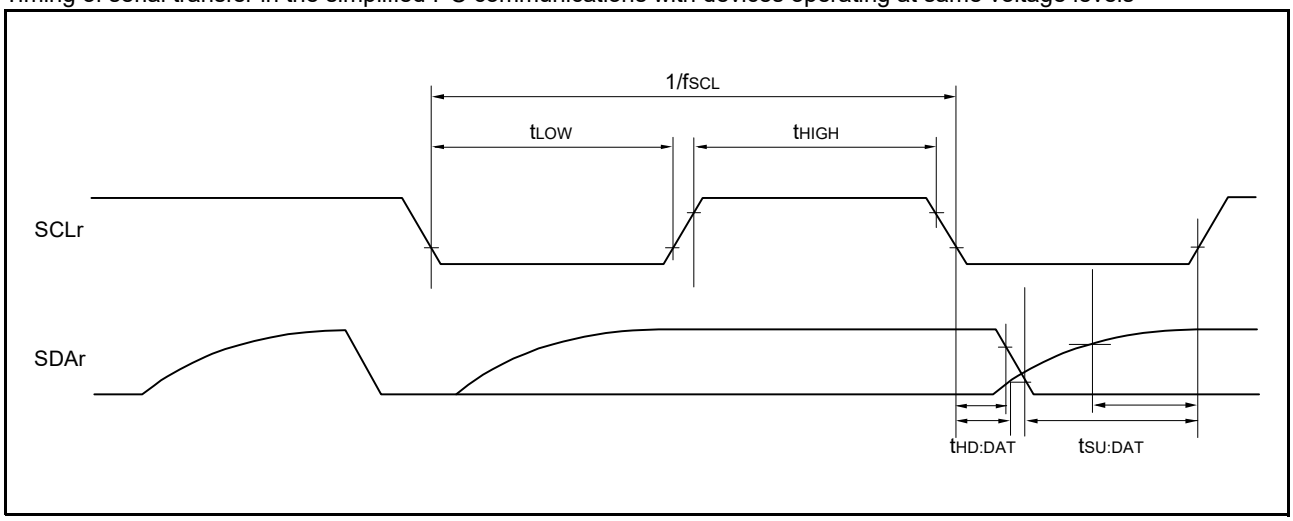
Caution Select the normal input buffer and the N-ch open drain output (V_{DD} withstand voltage for 20- to 52-pin products/ EV_{DD} withstand voltage for 64-pin products) for the SDAr pin and the normal output mode for the SCLr pin by using the port input mode register g (PIMg) and the port output mode register h (POMh).

(Remarks are listed on the next page.)

Connection in simplified I²C communications with devices operating at same voltage levels



Timing of serial transfer in the simplified I²C communications with devices operating at same voltage levels



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, 01, 10, 11, 20, 21), g: PIM number (g = 0, 1, 3, 5, 7), h: POM number (h = 0, 1, 3, 5, 7)

Remark 3. f_{MCK}: Serial array unit operating clock frequency
To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).
m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

6. In UART communications with devices operating at different voltage levels (1.8 V, 2.5 V, 3 V)

(T_A = –40 to +105°C, 1.8 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz) (1/2)

Item	Symbol	Conditions		HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
Transfer rate		Reception	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		f_{MCK}/6 **Note 1**		f_{MCK}/6 **Note 1**		f_{MCK}/6 **Note 1**	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} **Note 4**		5.3		4		0.33	Mbps
			2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		f_{MCK}/6 **Note 1**		f_{MCK}/6 **Note 1**		f_{MCK}/6 **Note 1**	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} **Note 4**		5.3		4		0.33	Mbps
			1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		f_{MCK}/6 **Notes 1, 2, 3**		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 4**		5.3		4		0.33	Mbps

Note 1. Transfer rate in the SNOOZE mode is within the range from 4800 to 9600 bps.

Note 2. Use this rate with EV_{DD0} ≥ V_b.

Note 3. The following conditions are required for low-voltage interface when EV_{DD0} < V_{DD}.
2.4 V ≤ EV_{DD0} < 2.7 V: 2.6 Mbps (max.)
1.8 V ≤ EV_{DD0} < 2.4 V: 1.3 Mbps (max.)

Note 4. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
HS (high-speed main) mode: 48 MHz (2.4 V ≤ V_{DD} ≤ 5.5 V)
32 MHz (1.8 V ≤ V_{DD} ≤ 5.5 V)
4 MHz (1.6 V ≤ V_{DD} ≤ 5.5 V)
LS (low-speed main) mode: 24 MHz (1.8 V ≤ V_{DD} ≤ 5.5 V)
4 MHz (1.6 V ≤ V_{DD} ≤ 5.5 V)
LP (low-power main) mode: 2 MHz (1.6 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} withstand voltage for 20- to 52-pin products/EV_{DD} withstand voltage for 64-pin products) for the TxDq pin by using the port input mode register g (PIMg) and the port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with the TTL input buffer selected.

Remark 1. V_b[V]: Communication line voltage

Remark 2. q: UART number (q = 0 to 3), g: PIM and POM number (g = 0, 1, 3, 5, 7)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00 to 03, 10, 11)

Remark 4. Communications by using UART2 with devices operating at different voltage levels are not possible when bit 1 (PIOR01) of the peripheral I/O redirection register 0 (PIOR0) is set to 1.

6. In UART communications with devices operating at different voltage levels (1.8 V, 2.5 V, 3 V)

(T_A = –40 to +105°C, 1.8 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz) (2/2)

Item	Symbol	Conditions		HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
Transfer rate		Transmission	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		**Note 1**		**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**		2.8**Note 2**	Mbps
			2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		**Note 3**		**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**		1.2**Note 4**	Mbps
			1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		**Notes 5, 6**		**Notes 5, 6**		**Notes 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**		0.43 **Note 7**	Mbps

(Notes and Caution are listed on the next page.)

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 ≤ EV_{DD0} ≤ 5.5 V, 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 rate is calculated as an example when the conditions described in the “Conditions” column are met. See **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 ≤ EV_{DD0} < 4.0 V, 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 rate is calculated as an example when the conditions described in the “Conditions” column are met. See **Note 3** above to calculate the maximum transfer rate under conditions of the customer.

Note 5. Use this rate with EV_{DD0} ≥ 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 1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\} \times 3} \text{ [bps]}$$

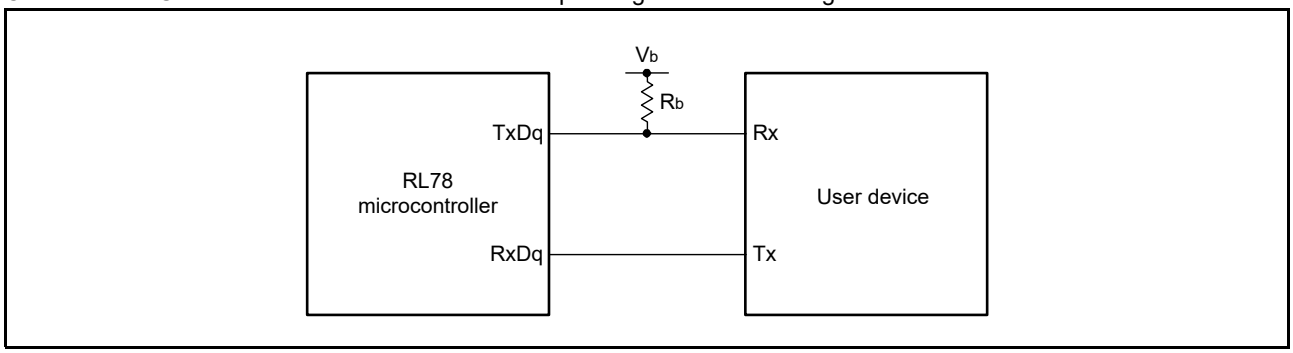
$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

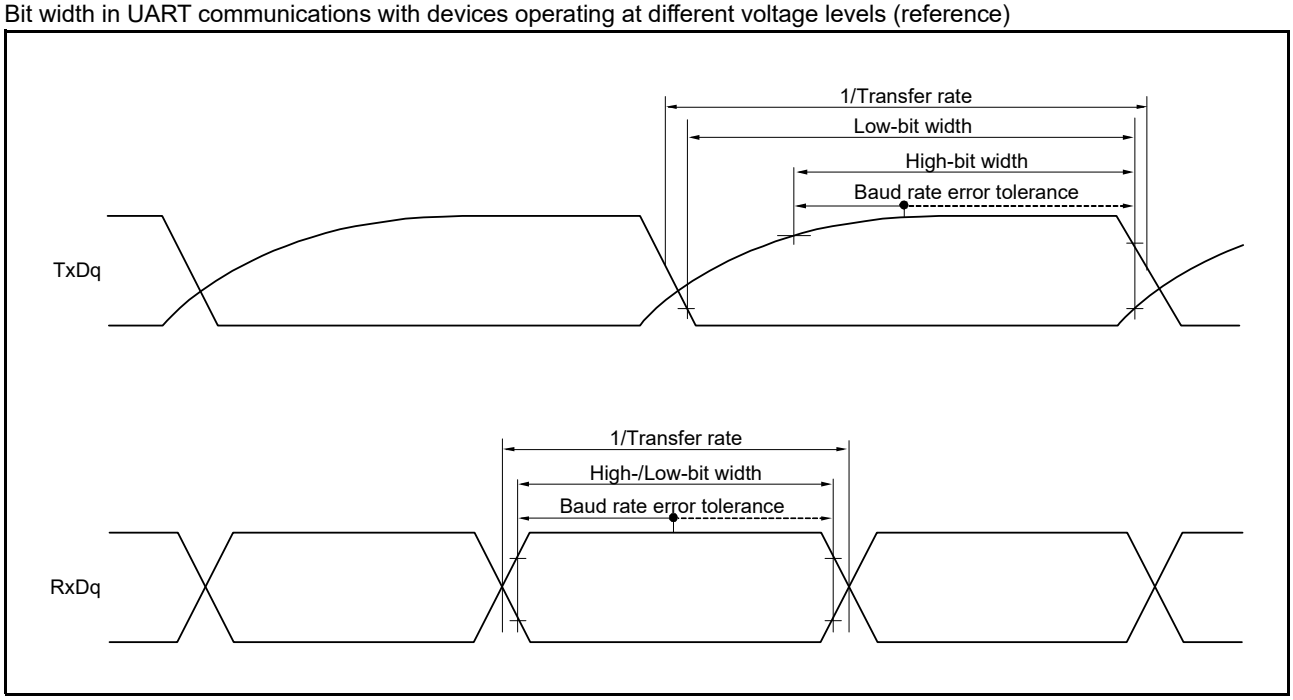
Note 7. This rate is calculated as an example when the conditions described in the “Conditions” column are met. See **Note 6** above to calculate the maximum transfer rate under conditions of the customer.

Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} withstand voltage for 20- to 52-pin products/EV_{DD} withstand voltage for 64-pin products) for the TxDq pin by using the port input mode register g (PIMg) and the port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with the TTL input buffer selected.

Connection in UART communications with devices operating at different voltage levels



Bit width in UART communications with devices operating at different voltage levels (reference)



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 to 3), g: PIM and POM number (g = 0, 1, 3, 5, 7)

Remark 3. f_{MCK}: Serial array unit operating clock frequency
To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00 to 03, 10, 11)

Remark 4. Communications by using UART2 with devices operating at different voltage levels are not possible when bit 1 (PIOR01) of the peripheral I/O redirection register 0 (PIOR0) is set to 1.

7. In simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels (2.5 V or 3 V) with the internal SCKp clock (the ratings below are only applicable to CSI00)

(T_A = –40 to +105°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz) (1/2)

Item	Symbol	Conditions		HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 2/f_{CLK}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ	200		200		2300		ns
			2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ	300		300		2300		ns
SCKp high-level width	t_{KH1}	4.0 V ≤ EV_{DD0} ≤ 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		t_{KCY1}/2 – 50		ns
		2.7 V ≤ EV_{DD0} < 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		t_{KCY1}/2 – 120		ns
SCKp low-level width	t_{KL1}	4.0 V ≤ EV_{DD0} ≤ 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 – 7		t_{KCY1}/2 – 50		ns
		2.7 V ≤ EV_{DD0} < 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 – 10		t_{KCY1}/2 – 50		ns
SIp setup time (to SCKp↑)**Note 1**	t_{SIK1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		58		58		479		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		121		121		479		ns
SIp hold time (from SCKp↑)**Note 1**	t_{KSI1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		10		10		10		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		10		10		10		ns
Delay time from SCKp↓ to SOp output**Note 1**	t_{KSO1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ			60		60		60	ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ			130		130		130	ns

(Notes, Caution, and Remarks are listed on the next page.)

7. In simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels (2.5 V or 3 V) with the internal SCKp clock (the ratings below are only applicable to CSI00)

(T_A = –40 to +105°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz) (2/2)

Item	Symbol	Conditions	HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
SIp setup time (to SCKp↓)**Note 2**	t_{SIK1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ	23		23		110		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ	33		33		110		ns
SIp hold time (from SCKp↓)**Note 2**	t_{KSI1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ	10		10		10		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ	10		10		10		ns
Delay time from SCKp↑ to SOp output**Note 2**	t_{KSO1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		10		10		10	ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		10		10		10	ns

Note 1. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1

Note 2. This setting applies 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} withstand voltage for 20- to 52-pin products/EV_{DD} withstand voltage for 64-pin products) for the SOp and SCKp pins by using the port input mode register g (PIMg) and the port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with the 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 = 1)

Remark 3. f_{MCK}: Serial array unit operating clock frequency
To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00)

Remark 4. The listed values are only valid when the peripheral I/O redirect function of CSI00 is not in use.

8. In simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V) with the internal SCKp clock

(T_A = –40 to +105°C, 1.8 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz) (1/3)

Item	Symbol	Conditions		HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 4/f_{CLK}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	300		300		2300		ns
			2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	500		500		2300		ns
			1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V**Note**, C_b = 30 pF, R_b = 5.5 kΩ	1150		1150		2300		ns
SCKp high-level width	t_{KH1}	4.0 V ≤ EV_{DD0} ≤ 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		t_{KCY1}/2 – 75		ns
		2.7 V ≤ EV_{DD0} < 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		t_{KCY1}/2 – 170		ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V**Note**, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 – 458		t_{KCY1}/2 – 458		t_{KCY1}/2 – 458		ns
SCKp low-level width	t_{KL1}	4.0 V ≤ EV_{DD0} ≤ 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 – 12		t_{KCY1}/2 – 50		ns
		2.7 V ≤ EV_{DD0} < 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 – 18		t_{KCY1}/2 – 50		ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V**Note**, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		ns

Note Use this setting with EV_{DD0} ≥ V_b.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} withstand voltage for 20- to 52-pin products/EV_{DD} withstand voltage for 64-pin products) for the SOp and SCKp pins by using the port input mode register g (PIMg) and the port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with the TTL input buffer selected.

(Remarks are listed on page 1891.)

8. In simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V) with the internal SCKp clock

(T_A = –40 to +105°C, 1.8 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz) (2/3)

Item	Symbol	Conditions	HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
SIp setup time (to SCKp↑)Note 1	t_{SIK1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	81		81		479		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	177		177		479		ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 VNote 2, C_b = 30 pF, R_b = 5.5 kΩ	479		479		479		ns
SIp hold time (from SCKp↑)Note 1	t_{KSI1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	19		19		19		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	19		19		19		ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 VNote 2, C_b = 30 pF, R_b = 5.5 kΩ	19		19		19		ns
Delay time from SCKp↓ to SOp outputNote 1	t_{KSO1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		100		100		100	ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		195		195		195	ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 VNote 2, C_b = 30 pF, R_b = 5.5 kΩ		483		483		483	ns

Note 1. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

Note 2. Use this setting with EV_{DD0} ≥ V_b.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} withstand voltage for 20- to 52-pin products/EV_{DD} withstand voltage for 64-pin products) for the SOp and SCKp pins by using the port input mode register g (PIMg) and the port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with the TTL input buffer selected.

(Remarks are listed on page 1891.)

8. In simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V) with the internal SCKp clock

(T_A = –40 to +105°C, 1.8 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz) (3/3)

Item	Symbol	Conditions	HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
SIp setup time (to SCKp↓)Note 1	t_{SIK1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	44		44		110		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	44		44		110		ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 VNote 2, C_b = 30 pF, R_b = 5.5 kΩ	110		110		110		ns
SIp hold time (from SCKp↓)Note 1	t_{KSI1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	19		19		19		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	19		19		19		ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 VNote 2, C_b = 30 pF, R_b = 5.5 kΩ	19		19		19		ns
Delay time from SCKp↑ to SOp outputNote 1	t_{KSO1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		25		25		25	ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		25		25		25	ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 VNote 2, C_b = 30 pF, R_b = 5.5 kΩ		25		25		25	ns

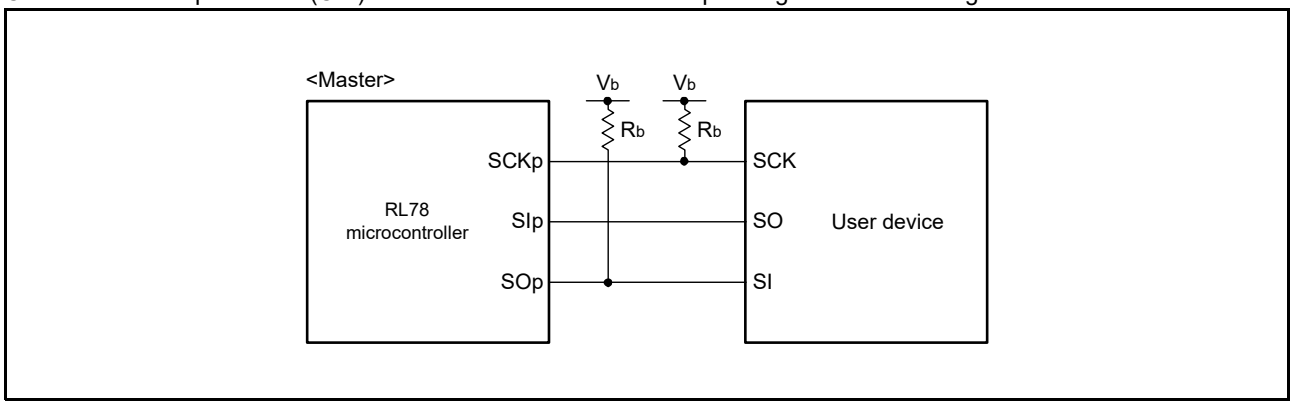
Note 1. This setting applies when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. Use this setting with EV_{DD0} ≥ V_b.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} withstand voltage for 20- to 52-pin products/EV_{DD} withstand voltage for 64-pin products) for the SOp and SCKp pins by using the port input mode register g (PIMg) and the port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with the TTL input buffer selected.

(Remarks are listed on the next page.)

Connection in simplified SPI (CSI) communications with devices operating at different voltage levels



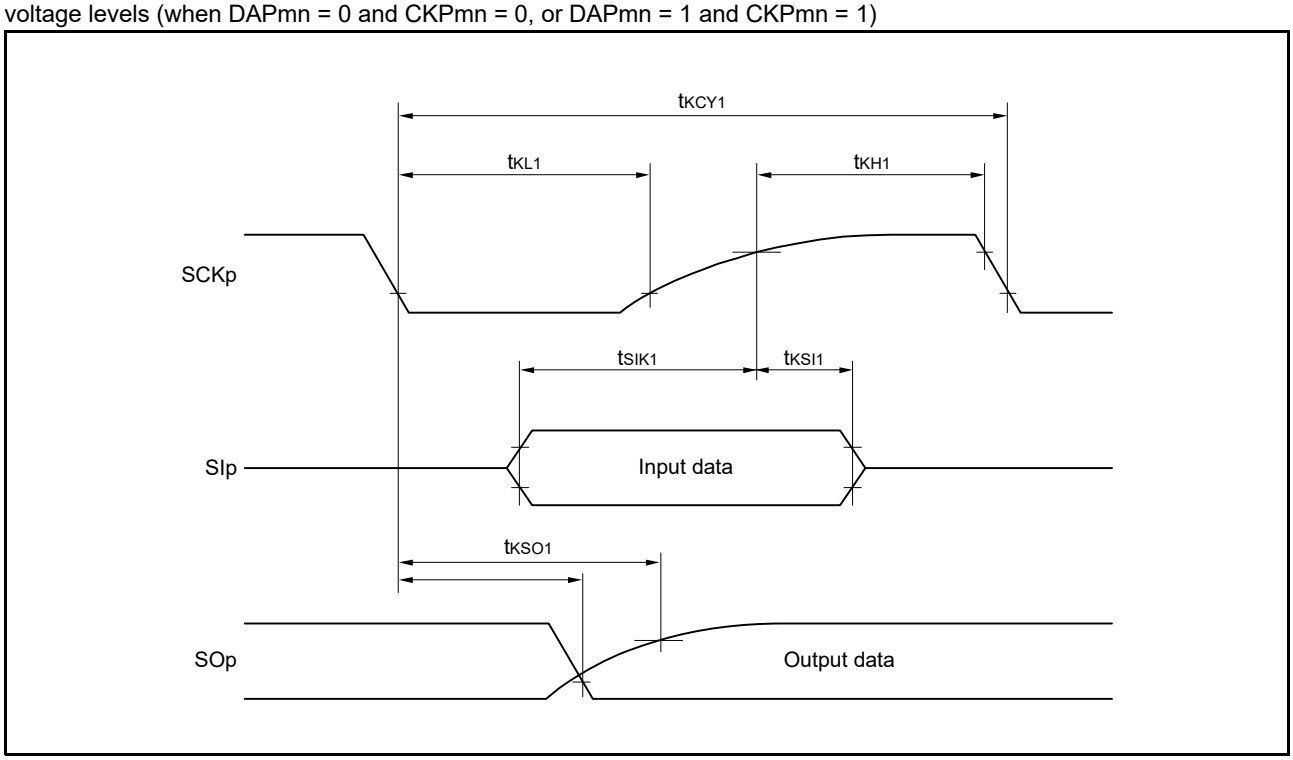
Remark 1. R_b[Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

Remark 2. p: CSI number (p = 00, 01, 10, 20), m: Unit number, n: Channel number (mn = 00, 01, 02, 10), g: PIM and POM number (g = 0, 1, 3, 5, 7)

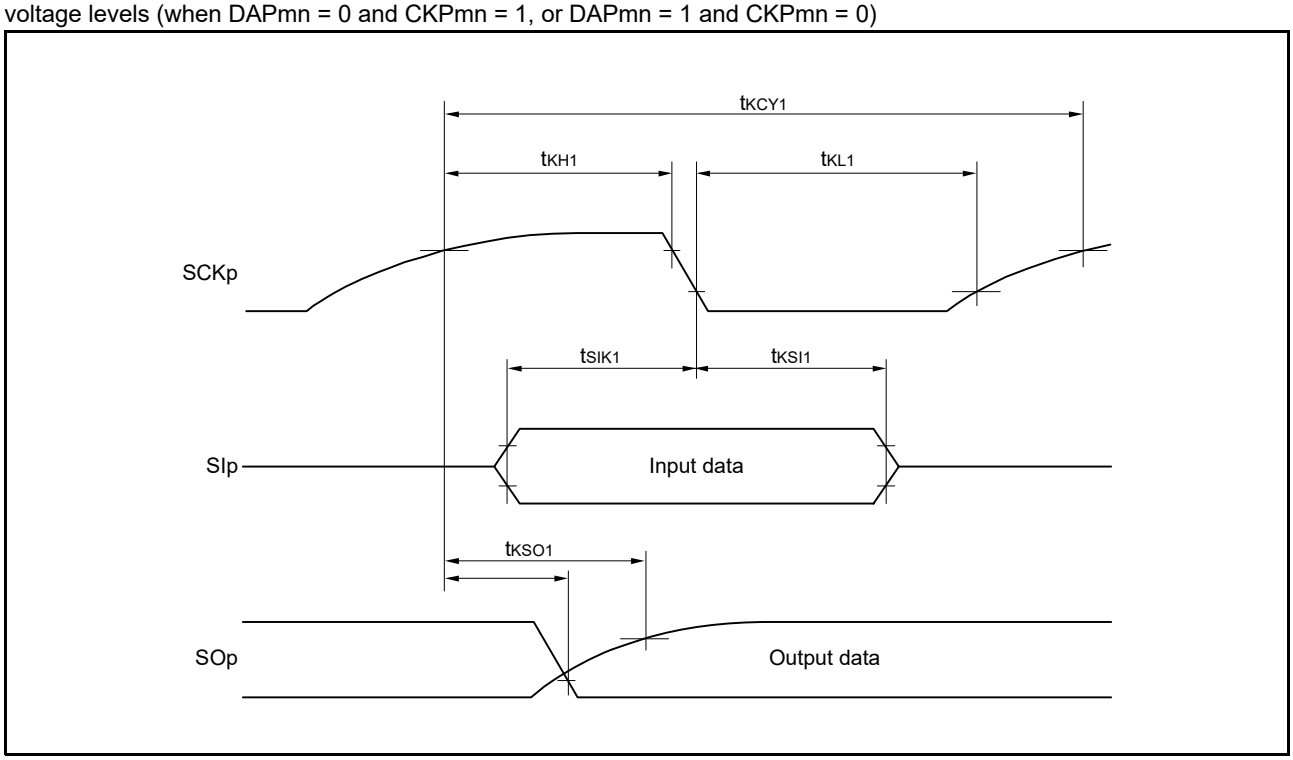
Remark 3. f_{MCK}: Serial array unit operating clock frequency
To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00)

Remark 4. Communications by using CSI01 of 48-, 52-, and 64-pin products, and CSI11 and CSI21 with devices operating at different voltage levels are not possible. Use other CSI channels to handle such communications.

Timing of serial transfer in simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels (when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)



Timing of serial transfer in simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels (when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)



Remark 1. p: CSI number (p = 00, 01, 10, 20), m: Unit number, n: Channel number (mn = 00, 01, 02, 10), g: PIM and POM number (g = 0, 1, 3, 5, 7)

Remark 2. Communications by using CSI01 of 48-, 52-, and 64-pin products, and CSI11 and CSI21 with devices operating at different voltage levels are not possible. Use other CSI channels to handle such communications.

9. In simplified SPI (CSI) communications in the slave mode with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V) with the external SCKp clock

(T_A = –40 to +105°C, 1.8 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz) (1/2)

Item	Symbol	Conditions		HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
SCKp cycle time **Note 1**	t_{KCY2}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	24 MHz < f_{MCK}	14/f_{MCK}		—		—		ns
			20 MHz < f_{MCK} ≤ 24 MHz	12/f_{MCK}		12/f_{MCK}		—		ns
			8 MHz < f_{MCK} ≤ 20 MHz	10/f_{MCK}		10/f_{MCK}		—		ns
			4 MHz < f_{MCK} ≤ 8 MHz	8/f_{MCK}		8/f_{MCK}		—		ns
			f_{MCK} ≤ 4 MHz	6/f_{MCK}		6/f_{MCK}		10/f_{MCK}		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V	24 MHz < f_{MCK}	20/f_{MCK}		—		—		ns
			20 MHz < f_{MCK} ≤ 24 MHz	16/f_{MCK}		16/f_{MCK}		—		ns
			16 MHz < f_{MCK} ≤ 20 MHz	14/f_{MCK}		14/f_{MCK}		—		ns
			8 MHz < f_{MCK} ≤ 16 MHz	12/f_{MCK}		12/f_{MCK}		—		ns
			4 MHz < f_{MCK} ≤ 8 MHz	8/f_{MCK}		8/f_{MCK}		—		ns
			f_{MCK} ≤ 4 MHz	6/f_{MCK}		6/f_{MCK}		10/f_{MCK}		ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V **Note 2**	24 MHz < f_{MCK}	48/f_{MCK}		—		—		ns
			20 MHz < f_{MCK} ≤ 24 MHz	36/f_{MCK}		36/f_{MCK}		—		ns
			16 MHz < f_{MCK} ≤ 20 MHz	32/f_{MCK}		32/f_{MCK}		—		ns
			8 MHz < f_{MCK} ≤ 16 MHz	26/f_{MCK}		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}		10/f_{MCK}		ns

(Notes and Caution are listed on the next page, and Remarks are listed on page 1895.)

9. In simplified SPI (CSI) communications in the slave mode with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V) with the external SCKp clock

(T_A = –40 to +105°C, 1.8 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz) (2/2)

Item	Symbol	Conditions	HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
SCKp high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	t_{KCY2}/2 – 12		t_{KCY2}/2 – 12		t_{KCY2}/2 – 50		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V	t_{KCY2}/2 – 18		t_{KCY2}/2 – 18		t_{KCY2}/2 – 50		ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V **Note 2**	t_{KCY2}/2 – 50		t_{KCY2}/2 – 50		t_{KCY2}/2 – 50		ns
SIp setup time (to SCKp↑) **Note 3**	t_{SIK2}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	1/f_{MCK} + 20		1/f_{MCK} + 20		1/f_{MCK} + 30		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V	1/f_{MCK} + 20		1/f_{MCK} + 20		1/f_{MCK} + 30		ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V **Note 2**	1/f_{MCK} + 30		1/f_{MCK} + 30		1/f_{MCK} + 30		ns
SIp hold time (from SCKp↑) **Note 3**	t_{KSI2}		1/f_{MCK} + 31		1/f_{MCK} + 31		1/f_{MCK} + 31		ns
Delay time from SCKp↓ to SOp output **Note 4**	t_{KSO2}	4.0 V ≤ EV_{DD0} ≤ 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} + 120		2/f_{MCK} + 573	ns
		2.7 V ≤ EV_{DD0} < 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} + 214		2/f_{MCK} + 573	ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V **Note 2**, C_b = 30 pF, R_b = 5.5 kΩ		2/f_{MCK} + 573		2/f_{MCK} + 573		2/f_{MCK} + 573	ns

Note 1. Transfer rate in the SNOOZE mode: 1 Mbps (max.)

Note 2. Use this setting with EV_{DD0} ≥ V_b.

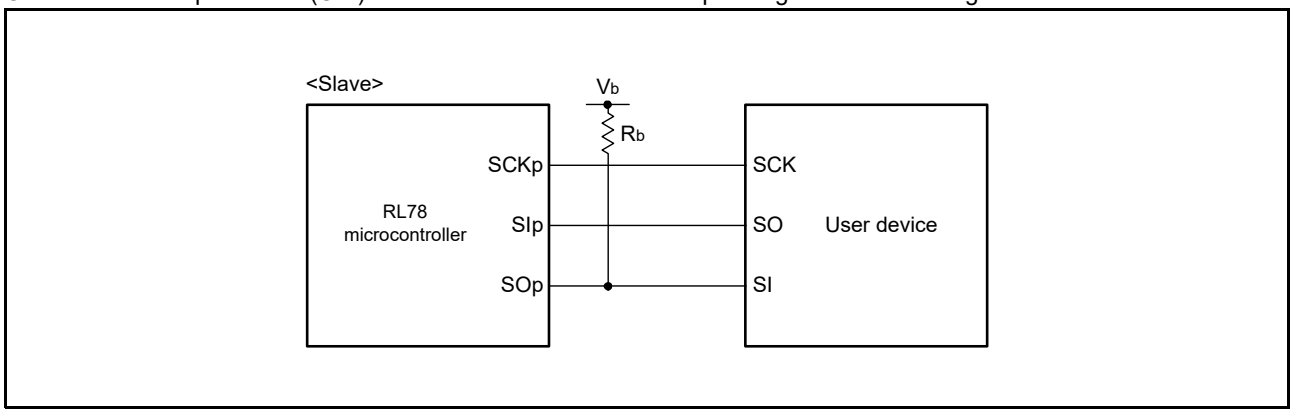
Note 3. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" and SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 4. This setting applies 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 and SCKp pins and the N-ch open drain output (V_{DD} withstand voltage for 20- to 52-pin products/EV_{DD} withstand voltage for 64-pin products) for the SOp pin by using the port input mode register g (PIMg) and the port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with the TTL input buffer selected.

(Remarks are listed on the next page.)

Connection in simplified SPI (CSI) communications with devices operating at different voltage levels



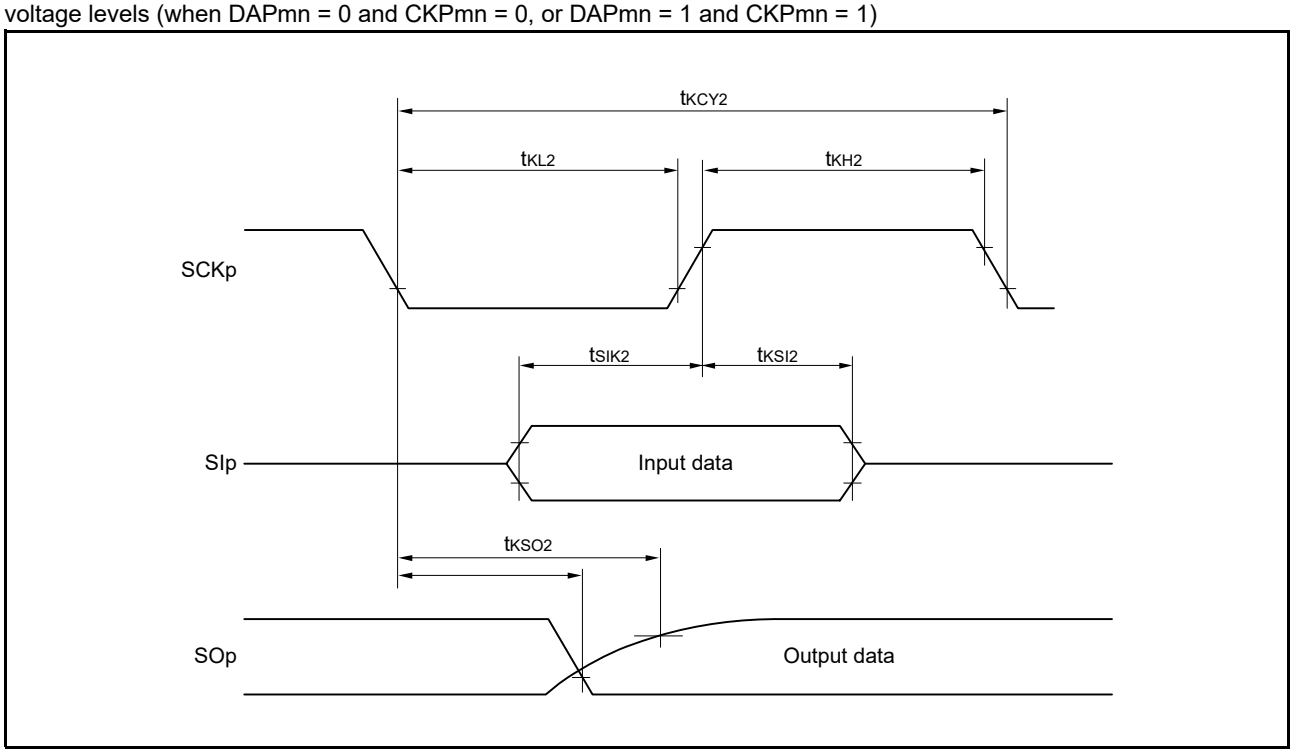
Remark 1. R_b[Ω]: Communication line (SOp) pull-up resistance, C_b[F]: Communication line (SOp) load capacitance, V_b[V]: Communication line voltage

Remark 2. p: CSI number (p = 00, 01, 10, 20), m: Unit number, n: Channel number (mn = 00, 01, 02, 10), g: PIM and POM number (g = 0, 1, 3, 5, 7)

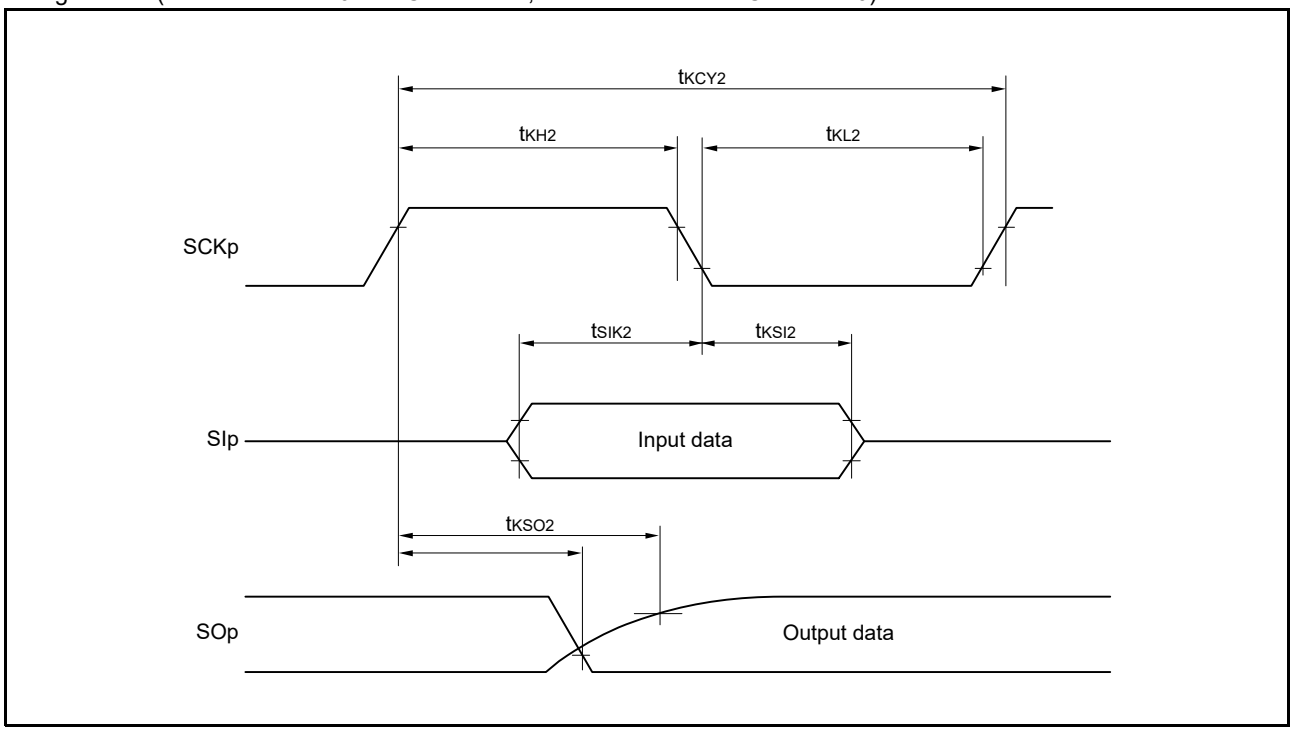
Remark 3. f_{MCK}: Serial array unit operating clock frequency
To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00, 01, 02, 10)

Remark 4. Communications by using CSI01 of 48-, 52-, and 64-pin products, and CSI11 and CSI21 with devices operating at different voltage levels are not possible. Use other CSI channels to handle such communications.

Timing of serial transfer in simplified SPI (CSI) communications in the slave mode with devices operating at different voltage levels (when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)



Timing of serial transfer in simplified SPI (CSI) communications in the slave mode with devices operating at different voltage levels (when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)



Remark 1. p: CSI number (p = 00, 01, 10, 20), m: Unit number, n: Channel number (mn = 00, 01, 02, 10), g: PIM and POM number (g = 0, 1, 3, 5, 7)

Remark 2. Communications by using CSI01 of 48-, 52-, and 64-pin products, and CSI11 and CSI21 with devices operating at different voltage levels are not possible. Use other CSI channels to handle such communications.

10. Simplified I^2C communications with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V)

(T_A = –40 to +105°C, 1.8 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz) (1/2)

Item	Symbol	Conditions	HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
SCLr clock frequency	f_{SCL}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ		1000 **Note 1**		1000 **Note 1**		300 **Note 1**	kHz
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ		1000 **Note 1**		1000 **Note 1**		300 **Note 1**	kHz
		4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ		400 **Note 1**		400 **Note 1**		300 **Note 1**	kHz
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ		400 **Note 1**		400 **Note 1**		300 **Note 1**	kHz
		1.8 V ≤ EV_{DD0} < 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**		300 **Note 1**	kHz
Hold time when SCLr is low	t_{LOW}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	475		475		1550		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	475		475		1550		ns
		4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	1150		1550		1550		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	1150		1550		1550		ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V**Note 2**, C_b = 100 pF, R_b = 5.5 kΩ	1550		1550		1550		ns
Hold time when SCLr is high	t_{HIGH}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	245		245		610		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	200		200		610		ns
		4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	675		675		610		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	600		600		610		ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V**Note 2**, C_b = 100 pF, R_b = 5.5 kΩ	610		610		610		ns

(Notes and Caution are listed on the next page, and Remarks are listed on page 1899.)

10. Simplified I^2C communications with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V)

(T_A = –40 to +105°C, 1.8 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz) (2/2)

Item	Symbol	Conditions	HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
Data setup time (reception)	$t_{SU:DAT}$	4.0 V ≤ EV_{DD0} ≤ 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} + 135 **Note 3**		1/f_{MCK} + 190 **Note 3**		ns
		2.7 V ≤ EV_{DD0} < 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} + 135 **Note 3**		1/f_{MCK} + 190 **Note 3**		ns
		4.0 V ≤ EV_{DD0} ≤ 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**		1/f_{MCK} + 190 **Note 3**		ns
		2.7 V ≤ EV_{DD0} < 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**		1/f_{MCK} + 190 **Note 3**		ns
		1.8 V ≤ EV_{DD0} < 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**		1/f_{MCK} + 190 **Note 3**		ns
Data hold time (transmission)	$t_{HD:DAT}$	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	0	305	0	305	0	305	ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	0	305	0	305	0	305	ns
		4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	0	355	0	355	0	355	ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	0	355	0	355	0	355	ns
		1.8 V ≤ EV_{DD0} < 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	0	405	ns

Note 1. The listed frequencies must be no greater than f_{MCK}/4.

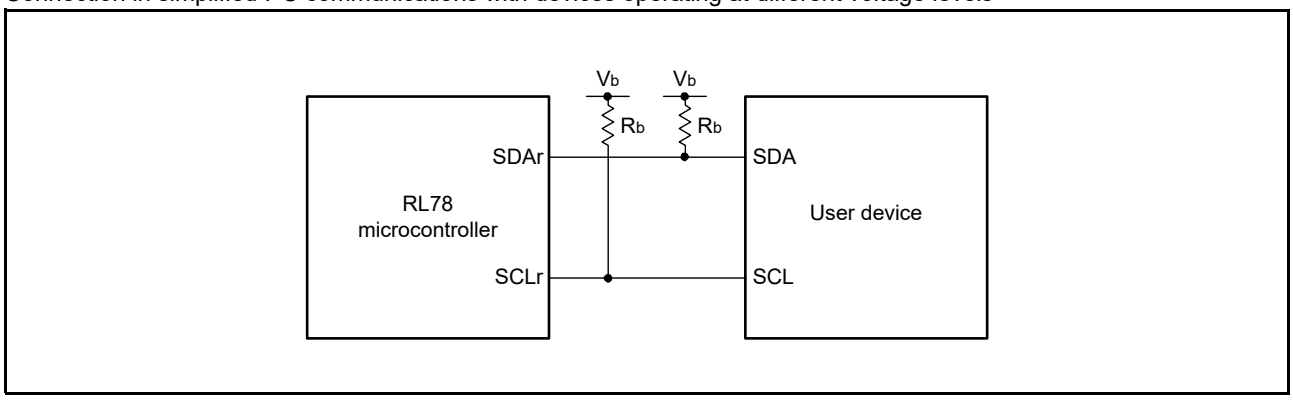
Note 2. Use this setting with EV_{DD0} ≥ V_b.

Note 3. Set the f_{MCK} value that does not exceed the hold time when SCLr is low or high.

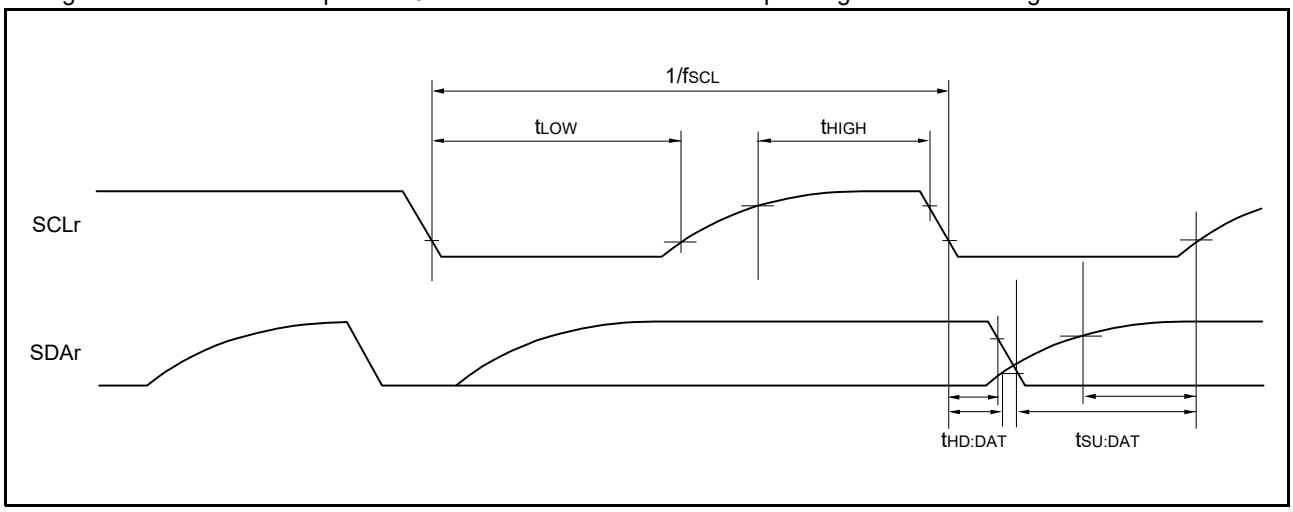
Caution Select the TTL input buffer and the N-ch open drain output (V_{DD} withstand voltage for 20- to 52-pin products/EV_{DD} withstand voltage for 64-pin products) for the SDAr pin and the N-ch open drain output (V_{DD} withstand voltage for 20- to 52-pin products/EV_{DD} withstand voltage for 64-pin products) for the SCLr pin by using the port input mode register g (PIMg) and the port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with the TTL input buffer selected.

(Remarks are listed on the next page.)

Connection in simplified I²C communications with devices operating at different voltage levels



Timing of serial transfer in simplified I²C communications with devices operating at different voltage levels



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, 01, 10, 20), g: PIM and POM number (g = 0, 1, 3, 5, 7)

Remark 3. f_{MCK}: Serial array unit operating clock frequency
To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00, 01, 02, 10)

43.5.2 Serial interface IICA

1. I^2C standard mode

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
SCLA0 clock frequency	f_{SCL}	Standard mode: f_{CLK} ≥ 1 MHz	0		100	kHz
Setup time of restart condition	$t_{SU:STA}$		4.7			μs
Hold time[Note 1]	$t_{HD:STA}$		4.0			μs
Hold time when SCLA0 is low	t_{LOW}		4.7			μs
Hold time when SCLA0 is high	t_{HIGH}		4.0			μs
Data setup time (reception)	$t_{SU:DAT}$		250			ns
Data setup time (transmission)[Note 2]	$t_{HD:DAT}$		0		3.45	μs
Setup time of stop condition	$t_{SU:STO}$		4.0			μs
Path free time	t_{BUF}		4.7			μs

Note 1. The first clock pulse is generated after this period when the start or restart condition is detected.

Note 2. The maximum value of $t_{HD:DAT}$ applies to normal transfer. The clock stretching is inserted on reception of an acknowledgment (ACK) signal.

Caution **The listed values are applicable even when bit 2 (PIOR02) in the peripheral I/O redirection register 0 (PIOR0) is 1. In such cases, the pin characteristics (I_{OH1}, I_{OL1}, V_{OH1}, V_{OL1}) must satisfy the values in the redirect destination.**

Remark The maximum value of communication line capacitance (C_b) and communication line pull-up resistor (R_b) are as follows.
C_b = 400 pF, R_b = 2.7 kΩ

2. I^2C fast mode

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
SCLA0 clock frequency	f_{SCL}	Fast mode: f_{CLK} ≥ 3.5 MHz 1.8 V ≤ EV_{DD0} ≤ 5.5 V	0		400	kHz
Setup time of restart condition	$t_{SU:STA}$	1.8 V ≤ EV_{DD0} ≤ 5.5 V	0.6			µs
Hold time[Note 1]	$t_{HD:STA}$	1.8 V ≤ EV_{DD0} ≤ 5.5 V	0.6			µs
Hold time when SCLA0 is low	t_{LOW}	1.8 V ≤ EV_{DD0} ≤ 5.5 V	1.3			µs
Hold time when SCLA0 is high	t_{HIGH}	1.8 V ≤ EV_{DD0} ≤ 5.5 V	0.6			µs
Data setup time (reception)	$t_{SU:DAT}$	1.8 V ≤ EV_{DD0} ≤ 5.5 V	100			ns
Data hold time (transmission)[Note 2]	$t_{HD:DAT}$	1.8 V ≤ EV_{DD0} ≤ 5.5 V	0		0.9	µs
Setup time of stop condition	$t_{SU:STO}$	1.8 V ≤ EV_{DD0} ≤ 5.5 V	0.6			µs
Bus-free time	t_{BUF}	1.8 V ≤ EV_{DD0} ≤ 5.5 V	1.3			µs

Note 1. The first clock pulse is generated after this period when the start or restart condition is detected.

Note 2. The maximum value of $t_{HD:DAT}$ applies to normal transfer. The clock stretching is inserted on reception of an acknowledgment (ACK) signal.

Caution **The listed values are applicable even when bit 2 (PIOR02) in the peripheral I/O redirection register 0 (PIOR0) is 1. In such cases, the pin characteristics (I_{OH1}, I_{OL1}, V_{OH1}, V_{OL1}) must satisfy the values in the redirect destination.**

Remark The maximum value of communication line capacitance (C_b) and communication line pull-up resistor (R_b) are as follows.
C_b = 320 pF, R_b = 1.1 kΩ

3. I^2C fast mode plus

(T_A = –40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
SCLA0 clock frequency	f_{SCL}	Fast mode plus: f_{CLK} ≥ 10 MHz 2.7 V ≤ EV_{DD0} ≤ 5.5 V	0		1000	kHz
Setup time of restart condition	$t_{SU:STA}$	2.7 V ≤ EV_{DD0} ≤ 5.5 V	0.26			µs
Hold time**Note 1**	$t_{HD:STA}$	2.7 V ≤ EV_{DD0} ≤ 5.5 V	0.26			µs
Hold time when SCLA0 is low	t_{LOW}	2.7 V ≤ EV_{DD0} ≤ 5.5 V	0.5			µs
Hold time when SCLA0 is high	t_{HIGH}	2.7 V ≤ EV_{DD0} ≤ 5.5 V	0.26			µs
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ EV_{DD0} ≤ 5.5 V	50			ns
Data hold time (transmission)**Note 2**	$t_{HD:DAT}$	2.7 V ≤ EV_{DD0} ≤ 5.5 V	0		0.45	µs
Setup time of stop condition	$t_{SU:STO}$	2.7 V ≤ EV_{DD0} ≤ 5.5 V	0.26			µs
Bus-free time	t_{BUF}	2.7 V ≤ EV_{DD0} ≤ 5.5 V	0.5			µs

Note 1. The first clock pulse is generated after this period when the start or restart condition is detected.

Note 2. The maximum value of $t_{HD:DAT}$ applies to normal transfer. The clock stretching is inserted on reception of an acknowledgment (ACK) signal.

Caution The listed values are applicable even when bit 2 (PIOR02) in the peripheral I/O redirection register 0 (PIOR0) is 1. In such cases, the pin characteristics (I_{OH1}, I_{OL1}, V_{OH1}, V_{OL1}) must satisfy the values in the redirect destination.

Remark The maximum value of communication line capacitance (C_b) and communication line pull-up resistor (R_b) are as follows.
C_b = 120 pF, R_b = 1.1 kΩ

4. SMBus/PMBus™ mode (100 kHz Class)

(T_A = –40 to +105°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
SCLA0 clock frequency	f_{SCL}	f_{CLK} ≥ 1 MHz	10		100	kHz
Setup time of restart condition	$t_{SU:STA}$		4.7			µs
Hold time[Note 1]	$t_{HD:STA}$		4			µs
Hold time when SCLA0 is low	t_{LOW}		4.7			µs
Hold time when SCLA0 is high	t_{HIGH}		4			µs
Data setup time (reception)	$t_{SU:DAT}$		250			ns
Data hold time (transmission)[Note 2]	$t_{HD:DAT}$		0		3.45	µs
Setup time of stop condition	$t_{SU:STO}$		4			µs
Clock/data falling time	t_F				0.3	µs
Clock/data rising time	t_R				1	µs
Bus-free time	t_{BUF}		4.7			µs

Note 1. The first clock pulse is generated after this period when the start or restart condition is detected.

Note 2. The maximum value of $t_{HD:DAT}$ applies to normal transfer. The clock stretching is inserted on reception of an acknowledgment (ACK) signal.

Caution **SMBus/PMBus™ communications are disabled when bit 2 (PIOR02) in the peripheral I/O redirection register 0 (PIOR0) is 1.**

Remark The maximum value of communication line pull-up resistor (R_b) is as follows.
R_b = 1.1 kΩ

5. SMBus/PMBus™ mode (400 kHz Class)

(T_A = –40 to +105°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
SCLA0 clock frequency	f_{SCL}	f_{CLK} ≥ 3.5 MHz	10		400	kHz
Setup time of restart condition	$t_{SU:STA}$		0.6			µs
Hold time[Note 1]	$t_{HD:STA}$		0.6			µs
Hold time when SCLA0 is low	t_{LOW}		1.3			µs
Hold time when SCLA0 is high	t_{HIGH}		0.6			µs
Data setup time (reception)	$t_{SU:DAT}$		100			ns
Data hold time (transmission)[Note 2]	$t_{HD:DAT}$		0		0.9	µs
Setup time of stop condition	$t_{SU:STO}$		0.6			µs
Clock/data falling time	t_F				0.3	µs
Clock/data rising time	t_R				0.3	µs
Bus-free time	t_{BUF}		1.3			µs

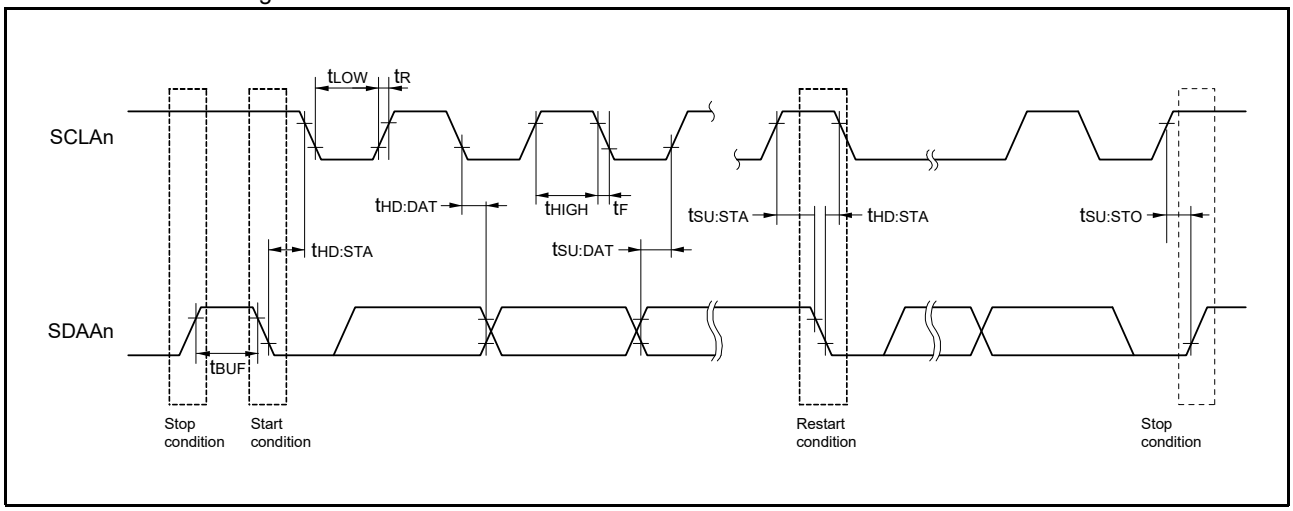
Note 1. The first clock pulse is generated after this period when the start or restart condition is detected.

Note 2. The maximum value of $t_{HD:DAT}$ applies to normal transfer. The clock stretching is inserted on reception of an acknowledgment (ACK) signal.

Caution **SMBus/PMBus™ communications are disabled when bit 2 (PIOR02) in the peripheral I/O redirection register 0 (PIOR0) is 1.**

Remark The maximum value of communication line capacitance (C_b) and communication line pull-up resistor (R_b) are as follows.
C_b = 400 pF, R_b = 1.1 kΩ

IICA serial transfer timing



Remark n = 0

43.6 Analog Characteristics

43.6.1 A/D converter characteristics

1. Normal modes 1 and 2

(T_A = –40 to +105°C, 2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, f_{CLK} ≤ 32 MHz,
reference voltage (+) = AV_{REFP} (ADREFP[1:0] = 01B), reference voltage (–) = AV_{REFM} (ADREFM = 1),
target pins: ANI2 to ANI7, ANI16 to ANI30, internal reference voltage, and temperature sensor output voltage)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Resolution	RES		8		12	Bit
Conversion clock	f_{AD}		1		32	MHz
Overall error**Notes 1, 3, 4, 5**	AINL	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±7.5	LSB
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±9.0	LSB
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±9.0	LSB
<R> Conversion time**Note 6**	t_{CONV}	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	2			µs
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	2			µs
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	2			µs
Zero-scale error**Notes 1, 2, 3, 4, 5**	E_{ZS}	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.17	%FSR
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
Full-scale error**Notes 1, 2, 3, 4, 5**	E_{FS}	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.17	%FSR
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
Integral linearity error**Notes 1, 4, 5**	ILE	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±3.0	LSB
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±3.0	LSB
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±3.0	LSB
Differential linearity error**Note 1**	DLE	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±1.0		LSB
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±1.0		LSB
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±1.0		LSB
Analog input voltage	V_{AIN}		0		AV_{REFP}	V

Note 1. This value does not include the quantization error (±1/2 LSB).

Note 2. This value is indicated as a ratio (%FSR) to the full-scale value.

Note 3. When pins ANI16 to ANI30 are selected as the conversion target, the maximum values are as follows.
Overall error: Add ±3 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.04%FSR to the maximum value.

Note 4. The maximum values are as follows when V_{DD} is selected for reference voltage (+) and V_{SS} is selected for reference voltage (–).
Overall error: Add ±10 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.25%FSR to the maximum value.
Integral linearity error: Add ±4 LSB to the maximum value.

Note 5. When AV_{REFP} < V_{DD}, the maximum values are as follows.
Overall error: Add ±0.75 LSB × (V_{DD} voltage (V) – AV_{REFP} voltage (V)) to the maximum value.
Zero-scale/full-scale error: Add ±0.018%FSR × (V_{DD} voltage (V) – AV_{REFP} voltage (V)) to the maximum value.
Integral linearity error: Add ±0.2 LSB × (V_{DD} voltage (V) – AV_{REFP} voltage (V)) to the maximum value.

Note 6. When the internal reference voltage or the temperature sensor output voltage is selected as the conversion target, the sampling time must be at least 5 µs. Accordingly, use normal mode 2 with the longer sampling time.

2. Normal modes 1 and 2 (advanced mode)

(T_A = –40 to +105°C, 2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, f_{CLK} ≤ 48 MHz,
reference voltage (+) = AV_{REFP} (ADREFP[1:0] = 01B), reference voltage (–) = AV_{REFM} (ADREFM = 1),
target pins: ANI2 to ANI7, ANI16 to ANI30, PGA**Note 1**, S&H**Note 1**, internal reference voltage, and temperature sensor output voltage)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Resolution	RES		8		12	Bit
Conversion clock	f_{AD}		1		48	MHz
Overall error**Notes 2, 4, 5, 6, 7, 9**	AINL	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±7.5	LSB
		4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V, and when the sample & hold circuit is in use (0.25 V ≤ V_{AIN} ≤ V_{DD} – 0.25 V)			±8.5	LSB
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±9.0	LSB
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V, and when the sample & hold circuit is in use (0.25 V ≤ V_{AIN} ≤ V_{DD} – 0.25 V)			±10.0	LSB
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±9.0	LSB
Conversion time**Notes 7, 8**	t_{CONV}	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	1			µs
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	1			µs
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	1.5			µs
Zero-scale error **Notes 2, 3, 4, 5, 6, 7, 9**	E_{ZS}	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.17	%FSR
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
Full-scale error **Notes 2, 3, 4, 5, 6, 7, 9**	E_{FS}	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.17	%FSR
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
Integral linearity error**Notes 2, 5, 6**	ILE	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±3.0	LSB
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±3.0	LSB
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±3.0	LSB
Differential linearity error**Note 2**	DLE	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±1.0		LSB
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±1.0		LSB
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±1.0		LSB
Analog input voltage	V_{AIN}		0		AV_{REFP}	V

(Notes are listed on the next page.)

Note 1. If the sample & hold circuit or PGA is to be A/D converted, V_{DD} must be at least 2.7 V.

Note 2. This value does not include the quantization error (±1/2 LSB).

Note 3. This value is indicated as a ratio (%FSR) to the full-scale value.

Note 4. When pins ANI16 to ANI30 are selected as the conversion target, the maximum values are as follows.
Overall error: Add ±3 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.04%FSR to the maximum value.

Note 5. The maximum values are as follows when V_{DD} is selected for reference voltage (+) and V_{SS} is selected for reference voltage (–).
Overall error: Add ±10 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.25%FSR to the maximum value.
Integral linearity error: Add ±4 LSB to the maximum value.

Note 6. When AV_{REFP} < V_{DD}, the maximum values are as follows.
Overall error: Add ±0.75 LSB × (V_{DD} voltage (V) – AV_{REFP} voltage (V)) to the maximum value.
Zero-scale/full-scale error: Add ±0.018%FSR × (V_{DD} voltage (V) – AV_{REFP} voltage (V)) to the maximum value.
Integral linearity error: Add ±0.2 LSB × (V_{DD} voltage (V) – AV_{REFP} voltage (V)) to the maximum value.

Note 7. Add the following values to the listed values in the cases below.
- 7 f_{AD} when the conversion target includes low-speed conversion ANI (ANI16 to ANI30)
- 12 f_{AD} when the conversion target includes the PGA with the gain of ×4 to ×16.
- 43 f_{AD} when the conversion target includes the PGA with the gain of ×32.

Note 8. When the internal reference voltage or the temperature sensor output voltage is selected as the conversion target, the sampling time must be at least 5 µs. Accordingly, use normal mode 2 with the longer sampling time.

Note 9. When the PGA is selected as the conversion target, the maximum values are as follows. For details, see **43.6.5 PGA characteristics**.
Overall error: Add input offset voltage and amplification rate error of the PGA to the maximum value.
Zero-scale error: Add input offset voltage of the PGA to the maximum value.
Full-scale error: Add amplification rate error of the PGA to the maximum value.

3. Low-voltage modes 1 and 2

(T_A = –40 to +105°C, 1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, f_{CLK} ≤ 32 MHz,
reference voltage (+) = AV_{REFP} (ADREFP[1:0] = 01B), reference voltage (–) = AV_{REFM} (ADREFM = 1),
target pins: ANI2 to ANI7, ANI16 to ANI30, internal reference voltage**Note 1**, and temperature sensor output voltage**Note 1**)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Resolution	RES		8		12	Bit
Conversion clock	f_{AD}		1		24	MHz
Overall error**Notes 2, 4, 5, 6**	AINL	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±9.0	LSB
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±9.0	LSB
		1.8 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±11.5	LSB
		1.6 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±12.0	LSB
Conversion time**Note 7**	t_{CONV}	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	3.33			µs
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	5.00			µs
		1.8 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	10.00			µs
		1.6 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	20.00			µs
Zero-scale error**Notes 2, 3, 4, 5, 6**	E_{ZS}	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
		1.8 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.27	%FSR
		1.6 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.28	%FSR
Full-scale error**Notes 2, 3, 4, 5, 6**	E_{FS}	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
		1.8 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.27	%FSR
		1.6 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.28	%FSR
Integral linearity error**Notes 2, 5, 6**	ILE	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±4.0	LSB
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±4.0	LSB
		1.8 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±4.5	LSB
		1.6 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±4.5	LSB
Differential linearity error**Note 2**	DLE	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±1.5		LSB
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±1.5		LSB
		1.8 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±2.0		LSB
		1.6 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±2.0		LSB
Analog input voltage	V_{AIN}		0		AV_{REFP}	V

(Notes are listed on the next page.)

Note 1. If the internal reference voltage or temperature sensor output voltage is to be A/D converted, V_{DD} must be at least 1.8 V.

Note 2. This value does not include the quantization error (±1/2 LSB).

Note 3. This value is indicated as a ratio (%FSR) to the full-scale value.

Note 4. When pins ANI16 to ANI30 are selected as the conversion target, the maximum values are as follows.
Overall error: Add ±3 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.04%FSR to the maximum value.

Note 5. The maximum values are as follows when V_{DD} is selected for reference voltage (+) and V_{SS} is selected for reference voltage (–).
Overall error: Add ±10 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.25%FSR to the maximum value.
Integral linearity error: Add ±4 LSB to the maximum value.

Note 6. When AV_{REFP} < V_{DD}, the maximum values are as follows.
Overall error: Add ±0.75 LSB × (V_{DD} voltage (V) – AV_{REFP} voltage (V)) to the maximum value.
Zero-scale/full-scale error: Add ±0.018%FSR × (V_{DD} voltage (V) – AV_{REFP} voltage (V)) to the maximum value.
Integral linearity error: Add ±0.2 LSB × (V_{DD} voltage (V) – AV_{REFP} voltage (V)) to the maximum value.

Note 7. When the internal reference voltage or the temperature sensor output voltage is selected as the conversion target, the sampling time must be at least 5 µs. Accordingly, use low-voltage mode 2 with the longer sampling time and the conversion clock (f_{AD}) with a frequency of no more than 16 MHz.

4. Low-voltage modes 1 and 2 (advanced mode)

(T_A = –40 to +105°C, 1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, f_{CLK} ≤ 48 MHz,
reference voltage (+) = AV_{REFP} (ADREFP[1:0] = 01B), reference voltage (–) = AV_{REFM} (ADREFM = 1),
target pins: ANI2 to ANI7, ANI16 to ANI30, PGA**Note 1**, S&H**Note 1**, internal reference voltage**Note 2**, and temperature sensor output voltage**Note 2**)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Resolution	RES		8		12	Bit
Conversion clock	f_{AD}		1		24	MHz
Overall error**Notes 3, 5, 6, 7, 8, 9**	AINL	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V, and when the sample & hold circuit is in use (0.25 V ≤ V_{AIN} ≤ V_{DD} – 0.25 V)			±10.0	LSB
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±9.0	LSB
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V, and when the sample & hold circuit is in use (0.25 V ≤ V_{AIN} ≤ V_{DD} – 0.25 V)			±10.0	LSB
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±9.0	LSB
		1.8 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±11.5	LSB
		1.6 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±12.0	LSB
Conversion time**Note 8**	t_{CONV}	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	3.33			µs
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	5.00			µs
		1.8 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	10.00			µs
		1.6 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	20.00			µs
Zero-scale error **Notes 3, 4, 5, 6, 7, 8, 9**	E_{ZS}	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
		1.8 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.27	%FSR
		1.6 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.28	%FSR
Full-scale error **Notes 3, 4, 5, 6, 7, 8, 9**	E_{FS}	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
		1.8 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.27	%FSR
		1.6 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.28	%FSR
Integral linearity error**Notes 3, 6, 7**	ILE	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±4.0	LSB
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±4.0	LSB
		1.8 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±4.5	LSB
		1.6 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±4.5	LSB
Differential linearity error**Note 3**	DLE	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±1.5		LSB
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±1.5		LSB
		1.8 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±2.0		LSB
		1.6 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±2.0		LSB
Analog input voltage	V_{AIN}		0		AV_{REFP}	V

(Notes are listed on the next page.)

Note 1. If the sample & hold circuit or PGA is to be A/D converted, V_{DD} must be at least 2.7 V.

Note 2. If the internal reference voltage or temperature sensor output voltage is to be A/D converted, V_{DD} must be at least 1.8 V.

Note 3. This value does not include the quantization error (±1/2 LSB).

Note 4. This value is indicated as a ratio (%FSR) to the full-scale value.

Note 5. When pins ANI16 to ANI30 are selected as the conversion target, the maximum values are as follows.
Overall error: Add ±3 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.04%FSR to the maximum value.

Note 6. The maximum values are as follows when V_{DD} is selected for reference voltage (+) and V_{SS} is selected for reference voltage (–).
Overall error: Add±10 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.25%FSR to the maximum value.
Integral linearity error: Add ±4 LSB to the maximum value.

Note 7. When AV_{REFP} < V_{DD}, the maximum values are as follows.
Overall error: Add ±0.75 LSB × (V_{DD} voltage (V) – AV_{REFP} voltage (V)) to the maximum value.
Zero-scale/full-scale error: Add ±0.018%FSR × (V_{DD} voltage (V) – AV_{REFP} voltage (V)) to the maximum value.
Integral linearity error: Add ±0.2 LSB × (V_{DD} voltage (V) – AV_{REFP} voltage (V)) to the maximum value.

Note 8. When the internal reference voltage or the temperature sensor output voltage is selected as the conversion target, the sampling time must be at least 5 µs. Accordingly, use low-voltage mode 2 with the longer sampling time and the conversion clock (f_{AD}) with a frequency of no more than 16 MHz.

Note 9. When the PGA is selected as the conversion target, the maximum values are as follows. For details, see **43.6.5 PGA characteristics**.
Overall error: Add input offset voltage and amplification rate error of the PGA to the maximum value.
Zero-scale error: Add input offset voltage of the PGA to the maximum value.
Full-scale error: Add amplification rate error of the PGA to the maximum value.

5. When the internal reference voltage is selected as reference voltage (+)

(T_A = –40 to +105°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, low-voltage modes 1 and 2, f_{CLK} ≤ 32 MHz**Note 1**, f_{CLK} ≤ 48 MHz**Note 2**, reference voltage (+) = internal reference voltage (ADREFP[1:0] = 10B), reference voltage (–) = AV_{REFM} (ADREFM = 1))

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Resolution	RES		8			Bit
Conversion clock	f_{AD}	1.6 V ≤ V_{DD} ≤ 5.5 V	1		2	MHz
Zero-scale error**Notes 3, 4, 6**	E_{ZS}	1.6 V ≤ V_{DD} ≤ 5.5 V			±0.6	%FSR
Integral linearity error**Notes 3, 6**	ILE	1.6 V ≤ V_{DD} ≤ 5.5 V			±2.0	LSB
Differential linearity error**Note 3**	DLE	1.6 V ≤ V_{DD} ≤ 5.5 V		±1.0		LSB
Analog input voltage	V_{AIN}		0		VBGR **Note 5**	V

Note 1. This applies when the advanced mode is disabled.
Note 2. This applies when the advanced mode is enabled.
Note 3. This value does not include the quantization error (±1/2 LSB).
Note 4. This value is indicated as a ratio (%FSR) to the full-scale value.
Note 5. Refer to **43.6.2 Temperature sensor/internal reference voltage characteristics**.
Note 6. When reference voltage (–) is selected as V_{SS}, the maximum values are as follows.
Zero-scale error: Add ±0.35%FSR to the maximum value.
Integral linearity error: Add ±0.5 LSB to the maximum value.

43.6.2 Temperature sensor/internal reference voltage characteristics

(T_A = –40 to +105°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Temperature sensor output voltage	V_{TMPS25}	ADS register is set to 80H, T_A = +25°C		1.05		V
Internal reference voltage	V_{BGR}	ADS register is set to 81H	1.40	1.48	1.56	V
Temperature coefficient	F_{VTMPS}	Temperature dependency of the temperature sensor voltage		–3.3		mV/°C
Operation stabilization wait time	t_{AMP}		5			μs

43.6.3 D/A converter characteristics

(T_A = –40 to +105°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Resolution	RES	DAC0, DAC1 (DACONF = 0)			10	Bit
		DAC1 (DACONF = 1), DAC2			8	Bit
Overall error	AINL	Rload = 8 MΩ			±2.5	LSB
Differential non-linearity error	ADNL				±1.0	LSB
Settling time	t_{SET}	Cload = 20 pF when DACO is output			6	μs
		During full code conversion using CMP reference			3	μs
		During 1LSB code conversion using CMP reference			1	μs

Caution The voltage on the ANO0 to ANO2 pins must not exceed EV_{DD0}.

43.6.4 Comparator characteristics

(T_A = –40 to +105°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Input voltage range	I_{VREF}	IVREF0 pin, IVREF1 pin input	0		EV_{DD0}	V
	I_{VCMP}	IVCMP0, IVCMP1, IVCMP2, ICMP3 pin input	0		EV_{DD0}	V
Output delay	td	Input amplitude ±100 mV		50	100	ns
Offset voltage	—			±5	±40	mV
Operation stabilization time**Note**	t_{CMP}		1			μs
Input channel switching stabilization wait time	—		0.3			μs

Note The listed values indicate the time until the DC/AC characteristics of the comparator are satisfied following enabling of the comparator operation (CnENB = 1).

43.6.5 PGA characteristics

(T_A = –40 to +105°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V)

Item	Symbol	Conditions			Min.	Typ.	Max.	Unit
Input offset voltage	V_{IOPGA}						±10	mV
Input voltage range[Note 1]	V_{IPGA}				0		0.9 × V_{DD}/ amplification rate	V
Amplification rate error		×4, ×8					±1	%
		×16					±1.5	%
		×32					±2	%
Slew rate[Note 1]	S_{RRPGA}	Rising Vin = V_{DD} × 0.1/ amplification rate to V_{DD} × 0.9/ amplification rate 10 to 90% of output amplitude	4.0 V ≤ V_{DD} ≤ 5.5 V	Other than ×32	3.5			V/µs
			4.0 V ≤ V_{DD} ≤ 5.5 V	×32	3			
			2.7 V ≤ V_{DD} ≤ 4.0 V		0.5			
	S_{FPGA}	Falling Vin = V_{DD} × 0.1/ amplification rate to V_{DD} × 0.9/ amplification rate 90 to 10% of output amplitude	4.0 V ≤ V_{DD} ≤ 5.5 V	Other than ×32	3.5			
			4.0 V ≤ V_{DD} ≤ 5.5 V	×32	3			
			2.7 V ≤ V_{DD} ≤ 4.0 V		0.5			
Operation stabilization wait time[Note 2]	t_{PGA}	×4, ×8					5	µs
		×16, ×32					10	µs

Note 1. A voltage of EV_{DD0} is supplied to the PGAI0 to PGAI3 pins.

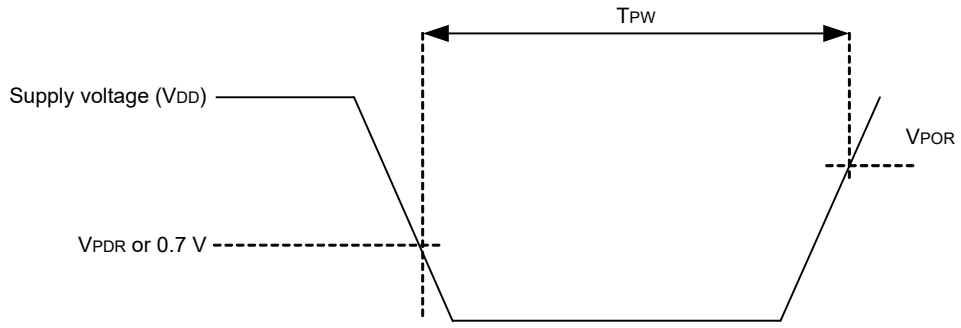
Note 2. The listed values indicate the time until the DC/AC characteristics of PGA operation are satisfied following enabling of the PGA operation (PGAEN = 1).

43.6.6 POR circuit characteristics

(T_A = –40 to +105°C, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Detection voltage	V_{POR}, V_{PDR}		1.43	1.50	1.57	V
Minimum pulse width**Note**	T_{PW}		300			μs

Note This width is the minimum time required for a POR reset when V_{DD} falls below V_{PDR}. This width is also the minimum time required for a POR reset from when V_{DD} falls below 0.7 V to when V_{DD} exceeds V_{POR} in the STOP mode or while the main system clock is stopped through setting bit 0 (HIOSTOP) and bit 7 (MSTOP) in the clock operation status control register (CSC).



43.6.7 LVD circuit characteristics

1. LVD detection voltage in the LVD0 reset mode and interrupt mode

(T_A = –40 to +105°C, $V_{PDR} \leq V_{DD} \leq 5.5$ V, V_{SS} = 0 V)

Item		Symbol	Conditions	Min.	Typ.	Max.	Unit
Detection voltage	Supply voltage level	V_{LVD00}	The power supply voltage is rising.	3.84	3.96	4.08	V
			The power supply voltage is falling.	3.76	3.88	4.00	V
		V_{LVD01}	The power supply voltage is rising.	2.88	2.97	3.06	V
			The power supply voltage is falling.	2.82	2.91	3.00	V
		V_{LVD02}	The power supply voltage is rising.	2.59	2.67	2.75	V
			The power supply voltage is falling.	2.54	2.62	2.70	V
		V_{LVD03}	The power supply voltage is rising.	2.31	2.38	2.45	V
			The power supply voltage is falling.	2.26	2.33	2.40	V
		V_{LVD04}	The power supply voltage is rising.	1.84	1.90	1.95	V
			The power supply voltage is falling.	1.80	1.86	1.91	V
		V_{LVD05}	The power supply voltage is rising.	1.64	1.69	1.74	V
			The power supply voltage is falling.	1.60	1.65	1.70	V
Minimum pulse width		t_{LW}		500			μs
Detection delay time						500	μs

2. LVD detection voltage in the LVD1 reset mode and interrupt mode

(T_A = –40 to +105°C, $V_{PDR} \leq V_{DD} \leq 5.5$ V, V_{SS} = 0 V)

Item		Symbol	Conditions	Min.	Typ.	Max.	Unit
Detection voltage	Supply voltage level	V_{LVD10}	The power supply voltage is rising.	4.08	4.16	4.24	V
			The power supply voltage is falling.	4.00	4.08	4.16	V
		V_{LVD11}	The power supply voltage is rising.	3.88	3.96	4.04	V
			The power supply voltage is falling.	3.80	3.88	3.96	V
		V_{LVD12}	The power supply voltage is rising.	3.68	3.75	3.82	V
			The power supply voltage is falling.	3.60	3.67	3.74	V
		V_{LVD13}	The power supply voltage is rising.	3.48	3.55	3.62	V
			The power supply voltage is falling.	3.40	3.47	3.54	V
		V_{LVD14}	The power supply voltage is rising.	3.28	3.35	3.42	V
			The power supply voltage is falling.	3.20	3.27	3.34	V
		V_{LVD15}	The power supply voltage is rising.	3.07	3.13	3.19	V
			The power supply voltage is falling.	3.00	3.06	3.12	V
		V_{LVD16}	The power supply voltage is rising.	2.91	2.97	3.03	V
			The power supply voltage is falling.	2.85	2.91	2.97	V
		V_{LVD17}	The power supply voltage is rising.	2.76	2.82	2.87	V
			The power supply voltage is falling.	2.70	2.76	2.81	V
		V_{LVD18}	The power supply voltage is rising.	2.61	2.66	2.71	V
			The power supply voltage is falling.	2.55	2.60	2.65	V
		V_{LVD19}	The power supply voltage is rising.	2.45	2.50	2.55	V
			The power supply voltage is falling.	2.40	2.45	2.50	V
		V_{LVD110}	The power supply voltage is rising.	2.35	2.40	2.45	V
			The power supply voltage is falling.	2.30	2.35	2.40	V
		V_{LVD111}	The power supply voltage is rising.	2.25	2.30	2.34	V
			The power supply voltage is falling.	2.20	2.25	2.29	V
		V_{LVD112}	The power supply voltage is rising.	2.15	2.20	2.24	V
			The power supply voltage is falling.	2.10	2.15	2.19	V
		V_{LVD113}	The power supply voltage is rising.	2.05	2.09	2.13	V
			The power supply voltage is falling.	2.00	2.04	2.08	V
		V_{LVD114}	The power supply voltage is rising.	1.94	1.98	2.02	V
			The power supply voltage is falling.	1.90	1.94	1.98	V
		V_{LVD115} **Note**	The power supply voltage is rising.	1.84	1.88	1.91	V
			The power supply voltage is falling.	1.80	1.84	1.87	V
		V_{LVD116} **Note**	The power supply voltage is rising.	1.74	1.78	1.81	V
			The power supply voltage is falling.	1.70	1.74	1.77	V
		V_{LVD117} **Note**	The power supply voltage is rising.	1.64	1.67	1.70	V
			The power supply voltage is falling.	1.60	1.63	1.66	V
Minimum pulse width		t_{LW}		500			μs
Detection delay						500	μs

Note This setting can only be used when LVD0 is disabled.

43.6.8 Power supply voltage rising slope characteristics

(T_A = –40 to +105°C, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Power voltage rising slope	S_{VDD}				54	V/ms

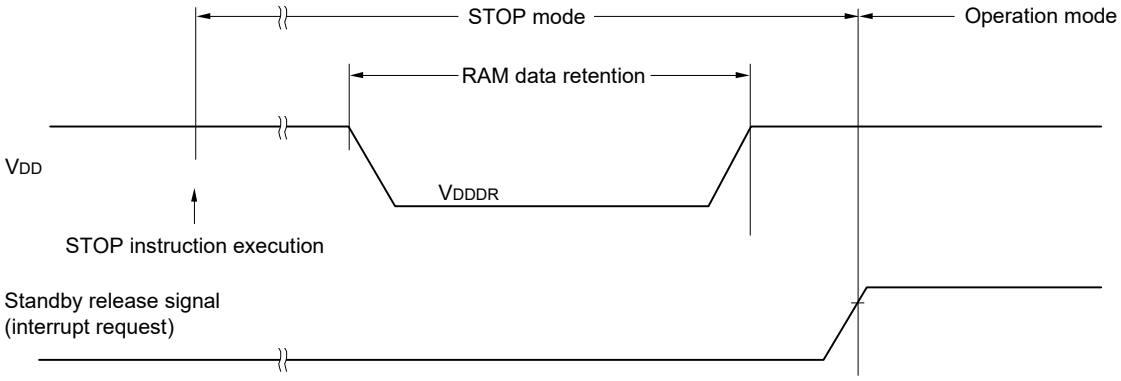
Caution Make sure to keep the internal reset state by the LVD0 circuit or an external reset until V_{DD} reaches the operating voltage range shown in AC characteristics.

43.7 RAM Data Retention Characteristics

(T_A = –40 to +105°C, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Data retention power voltage	V_{DDDR}		1.43Note		5.5	V

Note This voltage depends on the POR detection voltage. When the voltage drops, the data in RAM are retained until a POR is applied, but are not retained following a POR.



43.8 Flash Memory Programming Characteristics

(T_A = –40 to +105°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
CPU/peripheral hardware clock frequency	f_{CLK}		1		48	MHz
Number of code flash rewrites Notes 1, 2, 3	C_{erwr}	Retained for 10 years T_A = 85°C	10,000			Times
		Retained for 20 years T_A = 85°C	1,000			
Number of data flash rewrites Notes 1, 2, 3		Retained for 1 year T_A = 25°C		1,000,000		
		Retained for 5 years T_A = 85°C	100,000			
		Retained for 20 years T_A = 85°C	10,000			

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. The listed numbers of times apply when using the flash memory programmer and the Renesas Electronics self-programming library.

Note 3. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

1. Code flash memory

(T_A = –40 to +105°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item		Symbol	f_{CLK} = 1 MHz			f_{CLK} = 2 MHz, 3 MHz			4 MHz ≤ f_{CLK} < 8 MHz			8 MHz ≤ f_{CLK} < 32 MHz			f_{CLK} = 32 MHz			f_{CLK} = 48 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	4 bytes	t_{P4}	—	75.8	666.6	—	51.5	469.7	—	41.9	387.3	—	37.2	347.4	—	34.2	322.3	—	33.9	319.7	µs
Erasure time	2 Kbytes	t_{E2K}	—	10.4	312.2	—	7.7	258.5	—	6.4	231.8	—	5.8	218.4	—	5.6	214.4	—	5.6	213.9	ms
Blank checking time	4 bytes	t_{BC4}	—	—	38.4	—	—	19.2	—	—	13.1	—	—	10.2	—	—	8.3	—	—	8.1	µs
	2 Kbytes	t_{BC2K}	—	—	2618.9	—	—	1309.5	—	—	658.3	—	—	332.8	—	—	234.1	—	—	223.19	µs
Time taken to forcibly stop erasure		t_{SED}	—	—	19.0	—	—	14.5	—	—	12.3	—	—	11.1	—	—	10.4	—	—	10.3	µs
Security setting time		t_{AWSSAS}	—	18.2	526.4	—	14.4	469.3	—	12.6	441.1	—	11.6	427.1	—	11.3	422.6	—	11.3	422.1	ms
Time until programming starts following cancellation of the STOP instruction		—	20	—	—	20	—	—	20	—	—	20	—	—	20	—	—	20	—	—	µs

Caution The listed values do not include the time until the operations of the flash memory start following execution of an instruction by software.

2. Data flash memory

(T_A = –40 to +105°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item		Symbol	f_{CLK} = 1 MHz			f_{CLK} = 2 MHz, 3 MHz			4 MHz ≤ f_{CLK} < 8 MHz			8 MHz ≤ f_{CLK} < 32 MHz			f_{CLK} = 32 MHz			f_{CLK} = 48 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	1 byte	t_{P4}	—	75.8	666.6	—	51.51	469.7	—	41.9	387.34	—	37.24	347.4	—	34.2	322.3	—	33.92	319.7	µs
Erasure time	256 bytes	t_{E2K}	—	7.8	259.2	—	6.4	232.0	—	5.8	218.5	—	5.5	211.8	—	5.4	209.7	—	5.3	209.5	ms
Blank checking time	1 byte	t_{BC4}	—	—	38.4	—	—	19.2	—	—	13.1	—	—	10.2	—	—	8.3	—	—	8.1	µs
	256 bytes	t_{BC2K}	—	—	1326.1	—	—	663.1	—	—	335.1	—	—	171.2	—	—	121.0	—	—	115.5	µs
Time taken to forcibly stop erasure		t_{SED}	—	—	19.0	—	—	14.5	—	—	12.3	—	—	11.1	—	—	10.4	—	—	10.3	µs
Time until programming starts following cancellation of the STOP instruction		—	20	—	—	20	—	—	20	—	—	20	—	—	20	—	—	20	—	—	µs
Time until reading starts following setting DFLEN to 1		—	0.25	—	—	0.25	—	—	0.25	—	—	0.25	—	—	0.25	—	—	0.25	—	—	ns

Caution The listed values do not include the time until the operations of the flash memory start following execution of an instruction by software.

43.9 Dedicated Flash Memory Programmer Communication (UART)

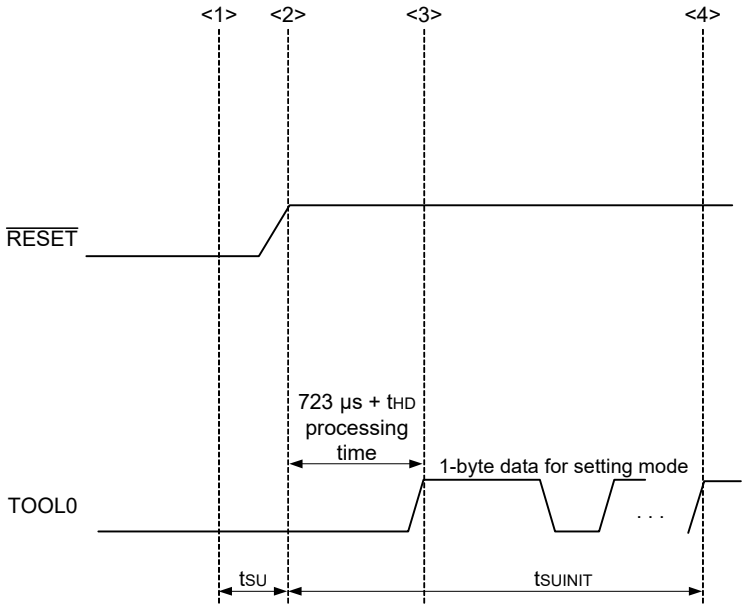
(T_A = –40 to +105°C, 1.8 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Transfer rate		During serial programming	115,200		1,000,000	bps

43.10 Timing of Entry to Flash Memory Programming Modes

(T_A = –40 to +105°C, 1.8 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Time to complete the communication for the initial setting after the external reset is released	t_{SUINIT}	POR and LVD reset must be released before the external reset is released			100	ms
Time to release the external reset after the TOOL0 pin is set to the low level	t_{SU}	POR and LVD reset must be released before the external reset is released	10			μs
Time to hold the TOOL0 pin at the low level after the external reset is released (the processing time of the firmware to control the flash memory is not included)	t_{HD}	POR and LVD reset must be released before the external reset is released	1			ms



<1> The low level is input to the TOOL0 pin.
<2> The external reset is released. Note that the POR and LVD reset must be released before the external reset is released.
<3> The TOOL0 pin is set to the high level.
<4> The baud rate setting is complete upon UART reception.

Remark t_{SUINIT}: The time during which the communications for the initial setting must be completed within 100 ms after the external reset is released.
t_{SU}: Time to release the external reset after the TOOL0 pin is set to the low level
t_{HD}: Time to hold the TOOL0 pin at the low level after the external reset is released. It does not include the processing time of the firmware to control the flash memory.

Section 44 Electrical Characteristics (T_A = –40 to +125°C)

This section describes the electrical characteristics of the following type of products.

- 4C: Industrial applications, T_A = –40 to +125°C
  R7F101Gxx4Cxx

Caution 1. RL78 microcontrollers have on-chip debugging functionality for use in the development and evaluation of user systems. Do not use on-chip debugging with products designated as part of mass production, because using this function may cause the guaranteed number of times the flash memory is rewritten to be exceeded, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when on-chip debugging is used with products designated as part of mass production.

Caution 2. For products that do not have an EV_{DD0} or EV_{SS0} pin, read EV_{DD0} as V_{DD}, and EV_{SS0} as V_{SS}.

Caution 3. The present pins differ depending on the products. For details, see 2.1 Functions of Port Pins through 2.2.1 Functions for each product.

Remark If you use a product under the condition T_A = –40 to +105°C, see **Section 43 Electrical Characteristics (T_A = –40 to +105°C)**.

44.1 Absolute Maximum Ratings

(1/2)

Item	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}		–0.5 to +6.5	V
	EV_{DD0}		–0.5 to +6.5	V
	EV_{SS0}		–0.5 to +0.3	V
REGC pin input voltage	V_{IREGC}	REGC	–0.3 to +2.1 and –0.3 to V_{DD} + 0.3 **Note 1**	V
Input voltage	V_{I1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63, P70 to P77, P120, P140, P141, P146, P147	–0.3 to EV_{DD0} + 0.3 and –0.3 to V_{DD} + 0.3 **Note 2**	V
	V_{I2}	P60, P61 (N-ch open drain)	–0.3 to +6.5	V
	V_{I3}	P20 to P27, P121 to P124, P137, EXCLK, EXCLKS, $\overline{RESET}$	–0.3 to V_{DD} + 0.3 **Note 2**	V
Output voltage	V_{O1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63, P70 to P77, P120, P130, P140, P141, P146, P147	–0.3 to EV_{DD0} + 0.3 and –0.3 to V_{DD} + 0.3 **Note 2**	V
	V_{O2}	P20 to P27, P121, P122	–0.3 to V_{DD} + 0.3 **Note 2**	V
Analog input voltage	V_{AI1}	ANI16 to ANI30	–0.3 to EV_{DD0} + 0.3 and –0.3 to AV_{REFP} + 0.3 **Notes 2, 3**	V
	V_{AI2}	ANI0 to ANI7	–0.3 to V_{DD} + 0.3 and –0.3 to AV_{REFP} + 0.3 **Notes 2, 3**	V

Note 1. Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). The listed value is the absolute maximum rating of the REGC pin. Do not apply a specific voltage to this pin.

Note 2. This voltage must be no higher than 6.5 V.

Note 3. The voltage on a pin in use for A/D conversion must not exceed AV_{REFP} + 0.3.

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. The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

Remark 2. AV_{REFP} refers to the positive reference voltage of the A/D converter.

Remark 3. The reference voltage is V_{SS}.

(2/2)

Item	Symbols	Conditions		Ratings	Unit
High-level output current	I_{OH1}	Per pin	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63, P70 to P77, P120, P130, P140, P141, P146, P147	–40	mA
		Total of all pins –170 mA	P00 to P04, P40 to P43, P120, P130, P140, P141	–70	mA
			P05, P06, P10 to P17, P30, P31, P50 to P55, P62, P63, P70 to P77, P146, P147	–100	mA
	I_{OH2}	Per pin	P20 to P27, P121, P122	–5	mA
		Total of all pins		–20	mA
Low-level output current	I_{OL1}	Per pin	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63, P70 to P77, P120, P130, P140, P141, P146, P147	40	mA
		Total of all pins 170 mA	P00 to P04, P40 to P43, P120, P130, P140, P141	70	mA
			P05, P06, P10 to P17, P30, P31, P50 to P55, P60 to P63, P70 to P77, P146, P147	100	mA
	I_{OL2}	Per pin	P20 to P27, P121, P122	10	mA
		Total of all pins		20	mA
Ambient operating temperature	T_A	In normal operation mode		–40 to +125	°C
		In flash memory programming mode		–40 to +125	°C
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 The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

44.2 Characteristics of the Oscillators

44.2.1 Characteristics of the X1 and XT1 oscillators

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item	Resonator	Conditions	Min.	Typ.	Max.	Unit
X1 clock oscillation allowable input cycle time**Note**	Ceramic resonator/ crystal resonator		0.05		1	μs
XT1 clock oscillation frequency (f_{XT})**Note**	Crystal resonator			32.768		kHz

Note The listed time and frequency indicate permissible ranges of the oscillators. For actual applications, request the resonator manufacturer for evaluation of the resonators on the oscillator circuit mounted on a board so you can use appropriate values. Refer to **44.4 AC Characteristics** for instruction execution time.

Caution **Since the CPU is started by the high-speed on-chip oscillator clock after release from the reset state, the user should use the oscillation stabilization time counter status register (OSTC) to check the X1 clock oscillation stabilization time. Sufficiently evaluate the oscillation stabilization time with the resonator to be used, and then specify the values for the oscillation stabilization time in the OSTC register and the oscillation stabilization time select register (OSTS).**

44.2.2 Characteristics of the on-chip oscillators

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
High-speed on-chip oscillator clock frequency	f_{IH}			1		48	MHz
High-speed on-chip oscillator clock frequency accuracy **Note 1**		HIPREC = 1	+105 to +125°C	–1.5		+1.5	%
			+85 to +105°C	–1.5		+1.5	%
			–20 to +85°C	–1.0		+1.0	%
			–40 to –20°C	–1.5		+1.5	%
		HIPREC = 0 **Note 4**		–15		0	%
High-speed on-chip oscillator clock correction resolution					0.05		%
Middle-speed on-chip oscillator clock frequency **Note 2**	f_{IM}			1		4	MHz
Middle-speed on-chip oscillator clock frequency accuracy **Note 1**				–12		+12	%
Middle-speed on-chip oscillator clock correction resolution					0.15		%
Middle-speed on-chip oscillator frequency temperature coefficient						±0.17 **Note 3**	%/°C
Low-speed on-chip oscillator clock frequency **Note 2**	f_{IL}				32.768		kHz
Low-speed on-chip oscillator clock frequency accuracy **Note 1**				–15		+15	%
Low-speed on-chip oscillator clock correction resolution					0.3		%
Low-speed on-chip oscillator frequency temperature coefficient						±0.21 **Note 3**	%/°C

Note 1. The accuracy values were obtained in testing of this product.

Note 2. The listed values only indicate the characteristics of the oscillators. Refer to **44.4 AC Characteristics** for instruction execution time.

Note 3. These values were obtained in the evaluation.

Note 4. This condition applies when the setting of the FRQSEL3 bit is 1.

44.2.3 Characteristics of the PLL oscillator

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
PLL input frequency	f_{PLLIN}	High-speed system clock (f_{MX}) or high-speed on-chip oscillator clock (f_{IH})		8		MHz
PLL output frequency	f_{PLL}	f_{PLLIN} × 12		96		MHz
		f_{PLLIN} × 8		64		MHz
Lock-up wait time		Wait time after PLL output is enabled until the output frequency is stabilized	50			μs
Interval wait time		Wait time after PLL stop until PLL operation is set again	4			μs
Setting wait time		Required wait time after the PLL input clock is stabilized and the PLL setting is determined until startup settings are made	1			μs

44.3 DC Characteristics

44.3.1 Pin characteristics

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (1/7)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Allowable high-level output current**Note 1**	I_{OH1}	Per pin for P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63, P70-P77, P120, P130, P140, P141, P146, P147	2.7 V ≤ EV_{DD0} ≤ 5.5 V			–10.0 **Note 2**	mA
		Total of P00 to P04, P40 to P43, P120, P130, P140, P141 (when duty ≤ 70%**Note 3**)	4.0 V ≤ EV_{DD0} ≤ 5.5 V			–24.0	mA
			2.7 V ≤ EV_{DD0} < 4.0 V			–10.0	mA
		Total of P05, P06, P10 to P17, P30, P31, P50 to P55, P62, P63, P70 to P77, P146, P147 (when duty ≤ 70%**Note 3**)	4.0 V ≤ EV_{DD0} ≤ 5.5 V			–42.0	mA
			2.7 V ≤ EV_{DD0} < 4.0 V			–17.0	mA
		Total of all pins (when duty ≤ 70%**Note 3**)	2.7 V ≤ EV_{DD0} ≤ 5.5 V			–54.0	mA
	I_{OH2}	Per pin for P20 to P27, P121, P122	4.0 V ≤ V_{DD} ≤ 5.5 V			–3.0 **Note 2**	mA
			2.7 V ≤ V_{DD} < 4.0 V			–1.0 **Note 2**	mA
		Total of all pins (when duty ≤ 70%**Note 3**)	4.0 V ≤ V_{DD} ≤ 5.5 V			–14	mA
			2.7 V ≤ V_{DD} < 4.0 V			–8	mA

Note 1. Device operation is guaranteed at the listed currents even if current is flowing from the EV_{DD0} or V_{DD} pin to an output pin.

Note 2. The combination of these and other pins must not exceed the total current value.

Note 3. The listed output current values apply when the duty cycle is no greater than 70%.
Use the following formula to calculate the output current when the duty cycle is greater than 70%, where n is the duty cycle (%).

- Total output current from all pins = (I_{OH} × 0.7)/(n × 0.01)
  Example when I_{OH} = –10.0 mA, n = 80%
  Total output current from all pins = (–10.0 × 0.7)/(80 × 0.01) ≈ –8.7 mA
  Note that the duty cycle has no effect on the current that is allowed to flow into a single pin. No current higher than the absolute maximum rating must not flow into a single pin.

Caution **The following pins do not output high-level signals in the N-ch open-drain mode.**
P00, P02 to P04, P10 to P15, P17, P30, P50, P51, P55, and P71 to P74

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (2/7)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Allowable low-level output current Note 1	I_{OL1}	Per pin for P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63,P70 to P77, P120, P130, P140, P141, P146, P147				17.0 Note 2	mA
		Per pin for P60, P61	2.7 V ≤ EV_{DD0} < 5.5 V			15.0 Note 2	mA
		Total of P00 to P04, P40 to P43, P120, P130, P140, P141 (when duty ≤ 70% Note 3)	4.0 V ≤ EV_{DD0} ≤ 5.5 V			34.0	mA
			2.7 V ≤ EV_{DD0} < 4.0 V			15.0	mA
		Total of P05, P06, P10 to P17, P30, P31, P50 to P55, P60 to P63, P70 to P77, P146, P147 (when duty ≤ 70% Note 3)	4.0 V ≤ EV_{DD0} ≤ 5.5 V			34.0	mA
			2.7 V ≤ EV_{DD0} < 4.0 V			15.0	mA
		Total of all pins (when duty ≤ 70% Note 3)				68.0	mA
	I_{OL2}	Per pin for P20 to P27, P121, P122	4.0 V ≤ V_{DD} ≤ 5.5 V			8.5 Note 2	mA
			2.7 V ≤ V_{DD} < 4.0 V			1.5 Note 2	mA
		Total of all pins (when duty ≤ 70% Note 3)	4.0 V ≤ V_{DD} ≤ 5.5 V			14.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			14.0	mA

Note 1. Device operation is guaranteed at the listed currents even if current is flowing from an output pin to the EV_{SS0} or V_{SS} pin.

Note 2. The combination of these and other pins must not exceed the total current value.

Note 3. The listed output current values apply when the duty cycle is no greater than 70%.
Use the following formula to calculate the output current when the duty cycle is greater than 70%, where n is the duty cycle (%).

- Total output current from all pins = (I_{OH} × 0.7)/(n × 0.01)
  Example when I_{OH} = –10.0 mA, n = 80%
  Total output current from all pins = (–10.0 × 0.7)/(80 × 0.01) ≈ –8.7 mA
  Note that the duty cycle has no effect on the current that is allowed to flow into a single pin. No current higher than the absolute maximum rating must not flow into a single pin.

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (3/7)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Input voltage, high	V_{IH1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63, P70 to P77, P120, P140, P141, P146, P147	Normal input buffer	0.8 EV_{DD0}		EV_{DD0}	V
	V_{IH2}	P01, P03, P04, P10, P11, P14 to P17, P30, P50, P55, P73	TTL input buffer 4.0 V ≤ EV_{DD0} ≤ 5.5 V	2.2		EV_{DD0}	V
			TTL input buffer 3.3 V ≤ EV_{DD0} < 4.0 V	2.0		EV_{DD0}	V
			TTL input buffer 2.7 V ≤ EV_{DD0} < 3.3 V	1.5		EV_{DD0}	V
	V_{IH3}	P20 to P27		0.7 V_{DD}		V_{DD}	V
	V_{IH4}	P60, P61	I/O port mode	0.7 EV_{DD0}		6.0	V
	V_{IH5}	P121 to P124, P137, EXCLK, EXCLKS, $\overline{RESET}$		0.8 V_{DD}		V_{DD}	V
	V_{IH6}	P60, P61	SMBus input mode	1.35		EV_{DD0}	V
Input voltage, low	V_{IL1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63, P70 to P77, P120, P140, P141, P146, P147	Normal input buffer	0		0.2 EV_{DD0}	V
	V_{IL2}	P01, P03, P04, P10, P11, P14 to P17, P30, P50, P55, P73	TTL input buffer 4.0 V ≤ EV_{DD0} ≤ 5.5 V	0		0.8	V
			TTL input buffer 3.3 V ≤ EV_{DD0} < 4.0 V	0		0.5	V
			TTL input buffer 2.7 V ≤ EV_{DD0} < 3.3 V	0		0.32	V
	V_{IL3}	P20 to P27		0		0.3 V_{DD}	V
	V_{IL4}	P60, P61	I/O port mode	0		0.3 EV_{DD0}	V
	V_{IL5}	P121 to P124, P137, EXCLK, EXCLKS, $\overline{RESET}$		0		0.2 V_{DD}	V
	V_{IL6}	P60, P61	SMBus input mode			0.8	V

Caution **The maximum value of V_{IH} of pins P00, P02 to P04, P10 to P15, P17, P30, P50, P51, P55, and P71 to P74 is EV_{DD0}, even in the N-ch open-drain mode.**

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (4/7)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Output voltage, high	V_{OH1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63, P70 to P77, P120, P130, P140, P141, P146, P147	4.0 V ≤ EV_{DD0} ≤ 5.5 V, I_{OH1} = –10.0 mA	EV_{DD0} – 1.5			V
			4.0 V ≤ EV_{DD0} ≤ 5.5 V, I_{OH1} = –3.0 mA	EV_{DD0} – 0.7			V
			2.7 V ≤ EV_{DD0} ≤ 5.5 V, I_{OH1} = –2.0 mA	EV_{DD0} – 0.6			V
	V_{OH2}	P20 to P27, P121, P122	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH2} = –3.0 mA	V_{DD} – 0.7			V
			2.7 V ≤ V_{DD} < 4.0 V, I_{OH2} = –1.0 mA	V_{DD} – 0.5			V

Caution **Pins P00, P02 to P04, P10 to P15, P17, P30, P50, P51, P55, P71 to P74 do not output high-level signals in the N-ch open-drain mode.**

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (5/7)

Item	Symbol	Conditions			Min.	Typ.	Max.	Unit
Output voltage, low	V_{OL1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63, P70 to P77, P120, P130, P140, P141, P146, P147	4.0 V ≤ EV_{DD0} ≤ 5.5 V	I_{OL1} = 17.0 mA			1.3	V
			4.0 V ≤ EV_{DD0} ≤ 5.5 V	I_{OL1} = 8.5 mA			0.7	V
			2.7 V ≤ EV_{DD0} ≤ 5.5 V	I_{OL1} = 3.0 mA			0.6	V
			2.7 V ≤ EV_{DD0} ≤ 5.5 V	I_{OL1} = 1.5 mA			0.4	V
	V_{OL2}	P20 to P27, P121, P122	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL2} = 6.0 mA				0.7	V
			2.7 V ≤ V_{DD} < 4.0 V, I_{OL2} = 1.5 mA				0.5	V
	V_{OL3}	P60, P61	4.0 V ≤ EV_{DD0} ≤ 5.5 V, I_{OL3} = 7.0 mA				2.0	V
			4.0 V ≤ EV_{DD0} ≤ 5.5 V, I_{OL3} = 5.0 mA				0.4	V
			2.7 V ≤ EV_{DD0} ≤ 5.5 V, I_{OL3} = 3.0 mA				0.4	V

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (6/7)

Item	Symbol	Conditions			Min.	Typ.	Max.	Unit
Output current[Note]	CCDIOL	P10, P11, P16, P17, P60 to P63	CCSm = 01H	4.0 V ≤ EV_{DD0} ≤ 5.5 V	1.0	1.8	2.6	mA
				2.7 V ≤ EV_{DD0} < 4.0 V	0.8	1.5	2.3	mA
			CCSm = 02H	4.0 V ≤ EV_{DD0} ≤ 5.5 V	3.0	4.9	6.5	mA
				3.0 V ≤ EV_{DD0} < 4.0 V	2.7	4.3	5.9	mA
			CCSm = 03H	4.0 V ≤ EV_{DD0} ≤ 5.5 V	6.6	10.0	13.2	mA
				3.3 V ≤ EV_{DD0} < 4.0 V	6.0	9.1	12.1	mA
		P60, P61	CCSm = 04H	4.0 V ≤ EV_{DD0} ≤ 5.5 V	10.2	15.0	19.8	mA
				3.3 V ≤ EV_{DD0} < 4.0 V	9.4	13.8	18.2	mA

Note The listed currents apply when the output current control function is enabled.

(T_A = –40 to +125°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (7/7)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Input leakage current, high	ILIH1	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P60 to P63, P70 to P77, P120, P140, P141, P146, P147	V_I = EV_{DD0}			1	μA
	ILIH2	P20 to P27, P137, $\overline{\text{RESET}}$	V_I = V_{DD}			1	μA
	ILIH3	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	V_I = V_{DD}			1	μA
Input leakage current, low	ILIL1	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63, P70 to P77, P120, P130, P140, P141, P146, P147	V_I = EV_{SS0}			1	μA
	ILIL2	P20 to P27, P137, $\overline{\text{RESET}}$	V_I = V_{SS}			1	μA
	ILIL3	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	V_I = V_{SS}			1	μA
On-chip pull-up resistance	RU	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P62, P63, P70 to P77, P120 to P122, P140, P141, P146, P147	V_I = EV_{SS0}, input port	10	20	100	kΩ

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

44.3.2 Supply current characteristics

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (1/5)

Item	Symbol	Conditions					Min.	Typ.	Max.	Unit
Supply current **Note 1**	I_{DD1}	Operating mode	HS (high-speed main) mode	f_{PLL} = 96 MHz f_{CLK} = 48 MHz (MCM0 = 0)**Note 2**	Normal operation	V_{DD} = 5.0 V		5.5	19.7	mA
						V_{DD} = 2.7 V		5.5	19.7	
				f_{PLL} = 96 MHz f_{CLK} = 48 MHz (MCM = 1)**Note 4**	Normal operation	V_{DD} = 5.0 V		5.3	19.4	mA
						V_{DD} = 2.7 V		5.3	19.4	
				f_{IH} = 48 MHz**Note 2**	Normal operation	V_{DD} = 5.0 V		4.6	13.3	mA
						V_{DD} = 2.7 V		4.6	13.3	
				f_{PLL} = 64 MHz f_{CLK} = 32 MHz (MCM0 = 0)**Note 2**	Normal operation	V_{DD} = 5.0 V		3.9	13.6	mA
						V_{DD} = 2.7 V		3.9	13.5	
				f_{PLL} = 64 MHz f_{CLK} = 32 MHz (MCM = 1)**Note 4**	Normal operation	V_{DD} = 5.0 V		3.7	13.3	mA
						V_{DD} = 2.7 V		3.7	13.3	
				f_{IH} = 32 MHz**Note 2**	Basic operation	V_{DD} = 5.0 V		1.6	—	mA
						V_{DD} = 2.7 V		1.6	—	
					Normal operation	V_{DD} = 5.0 V		3.3	9.3	mA
						V_{DD} = 2.7 V		3.3	9.3	
			LS (low-speed main) mode	f_{IH} = 24 MHz**Note 2**	Normal operation	V_{DD} = 5.0 V		2.5	7.1	mA
						V_{DD} = 2.7 V		2.5	7.1	
				f_{IH} = 16 MHz**Note 2**	Normal operation	V_{DD} = 5.0 V		1.8	5.1	mA
						V_{DD} = 2.7 V		1.8	5.1	
				f_{IM} = 4 MHz**Note 3**	Normal operation	V_{DD} = 5.0 V		0.5	1.6	mA
						V_{DD} = 2.7 V		0.5	1.6	
			LP (low-power main) mode	f_{IM} = 2 MHz**Note 3**	Normal operation	V_{DD} = 5.0 V		0.2	968	µA
						V_{DD} = 2.7 V		0.2	968	
				f_{IM} = 1 MHz**Note 3**	Normal operation	V_{DD} = 5.0 V		0.1	701	µA
						V_{DD} = 2.7 V		0.1	701	
			HS (high-speed main) mode	f_{MX} = 20 MHz**Note 4**, Square wave input	Normal operation	V_{DD} = 5.0 V		2.0	5.9	mA
						V_{DD} = 2.7 V		2.0	5.9	
			LS (low-speed main) mode	f_{MX} = 20 MHz**Note 4**, Square wave input	Normal operation	V_{DD} = 5.0 V		1.9	5.8	mA
						V_{DD} = 2.7 V		1.9	5.8	
				f_{MX} = 20 MHz**Note 4**, Resonator connection	Normal operation	V_{DD} = 5.0 V		2.1	6.0	mA
						V_{DD} = 2.7 V		2.1	6.0	
				f_{MX} = 10 MHz**Note 4**, Square wave input	Normal operation	V_{DD} = 5.0 V		1.0	3.2	mA
						V_{DD} = 2.7 V		1.0	3.2	
				f_{MX} = 10 MHz**Note 4**, Resonator connection	Normal operation	V_{DD} = 5.0 V		1.1	3.4	mA
						V_{DD} = 2.7 V		1.1	3.4	
				f_{MX} = 8 MHz**Note 4**, Square wave input	Normal operation	V_{DD} = 5.0 V		0.8	2.7	mA
						V_{DD} = 2.7 V		0.8	2.6	

(Notes and Remarks are listed on the next page.)

(T_A = –40 to +125°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (2/5)

Item	Symbol	Conditions					Min.	Typ.	Max.	Unit
Supply current **Note 1**	I_{DD1}	Operating mode	LS (low-speed main) mode	f_{MX} = 8 MHz**Note 4**, Resonator connection	Normal operation	V_{DD} = 5.0 V		0.9	2.8	mA
						V_{DD} = 2.7 V		0.9	2.8	

Note 1. The listed currents are the total currents flowing into V_{DD} and EV_{DD0}, including the input leakage currents flowing when the level of the input pin is fixed to V_{DD}, EV_{DD0} or V_{SS}, EV_{SS0}. The currents in the Typ. column do not include the peripheral operating current when the CPU is placed in the HS (high-speed main), LS (low-speed main), or LV (low-voltage main) mode. The currents in the Max. column include the peripheral operating current, but do not include those of the FAA, A/D converter, sample & hold circuit, D/A converter, PGA, comparator, TRNG, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors, and those flowing when the data flash memory is being rewritten.

Note 2. The listed currents apply when the high-speed system clock, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock are stopped.

Note 3. The listed currents apply when the high-speed on-chip oscillator, high-speed system clock, low-speed on-chip oscillator, and subsystem clock are stopped.

Note 4. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock are stopped.

Remark 1. f_{IH}: High-speed on-chip oscillator clock frequency

Remark 2. f_{IM}: Middle-speed on-chip oscillator clock frequency

Remark 3. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

Remark 4. f_{PLL}: PLL clock frequency (up to 96 MHz)

Remark 5. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 6. The typical value for the ambient operating temperature (T_A) is 25°C unless otherwise specified.

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (3/5)

Item	Symbol	Conditions					Min.	Typ.	Max.	Unit
Supply current **Note 1**	I_{DD1}	Operating mode	Subsystem clock operation mode	f_{SUB} = 32.768 kHz **Note 2**, Low-speed on-chip oscillator operation	Normal operation	T_A = –40°C		3.9	16.8	μA
						T_A = +25°C		4.7	17.4	
						T_A = +50°C		6.3	30.9	
						T_A = +70°C		9.7	52.3	
						T_A = +85°C		15.3	83.2	
						T_A = +105°C		30.6	177.3	
						T_A = +125°C		61.3	324.1	
				f_{SUB} = 32.768 kHz **Note 3**, Square wave input	Normal operation	T_A = –40°C		3.5	16.3	μA
						T_A = +25°C		4.9	22.0	
						T_A = +50°C		5.9	31.7	
						T_A = +70°C		9.2	53.9	
						T_A = +85°C		14.7	81.8	
						T_A = +105°C		30.3	180.4	
						T_A = +125°C		71.6	398.7	
				f_{SUB} = 32.768 kHz **Note 3**, Resonator connection	Normal operation	T_A = –40°C		3.6	13.4	μA
						T_A = +25°C		4.3	14.1	
						T_A = +50°C		5.8	27.2	
						T_A = +70°C		9.2	50.0	
						T_A = +85°C		14.9	79.7	
						T_A = +105°C		30.0	174.3	
						T_A = +125°C		59.5	319.4	

Note 1. The listed currents are the total currents flowing into V_{DD} and EV_{DD0}, including the input leakage currents flowing when the level of the input pin is fixed to V_{DD}, EV_{DD0} or V_{SS}, EV_{SS0}. The currents in the Typ. and Max. columns do not include the peripheral operating current when the CPU is operating with the sub-system clock.

Note 2. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and subsystem clock are stopped. They do not include the currents flowing into the RTC, 32-bit interval timer, and watchdog timer.

Note 3. The listed currents apply when the high-speed on-chip oscillator, high-speed system clock, middle-speed on-chip oscillator, and low-speed on-chip oscillator are stopped, and the low power consumption oscillation 3 is specified (AMPHS1, AMPHS0 = 11B). They do not include the currents flowing into the RTC, 32-bit interval timer, and watchdog timer.

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

Remark 2. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (4/5)

Item	Symbol	Conditions				Min.	Typ.	Max.	Unit
Supply current**Note 1**	I_{DD2} **Note 2**	HALT mode	HS (high-speed main) mode	f_{PLL} = 96 MHz f_{CLK} = 48 MHz (MCM0 = 0)**Note 2**	V_{DD} = 5.0 V		1.57	14.37	mA
					V_{DD} = 2.7 V		1.57	14.37	
				f_{PLL} = 96 MHz f_{CLK} = 48 MHz (MCM = 1)**Note 4**	V_{DD} = 5.0 V		1.39	14.13	mA
					V_{DD} = 2.7 V		1.39	14.13	
				f_{IH} = 48 MHz**Note 2**	V_{DD} = 5.0 V		0.73	8.06	mA
					V_{DD} = 2.7 V		0.73	8.06	
				f_{PLL} = 64 MHz f_{CLK} = 32 MHz (MCM0 = 0)**Note 2**	V_{DD} = 5.0 V		1.19	9.90	mA
					V_{DD} = 2.7 V		1.18	9.90	
				f_{PLL} = 64 MHz f_{CLK} = 32 MHz (MCM = 1)**Note 4**	V_{DD} = 5.0 V		1.01	9.66	mA
					V_{DD} = 2.7 V		1.00	9.66	
				f_{IH} = 32 MHz**Note 3**	V_{DD} = 5.0 V		0.62	5.68	mA
					V_{DD} = 2.7 V		0.61	5.68	
			LS (low-speed main) mode	f_{IH} = 24 MHz**Note 3**	V_{DD} = 5.0 V		0.51	4.42	mA
					V_{DD} = 2.7 V		0.50	4.42	
				f_{IH} = 16 MHz**Note 3**	V_{DD} = 5.0 V		0.48	3.27	mA
					V_{DD} = 2.7 V		0.48	3.27	
				f_{IM} = 4 MHz**Note 4**	V_{DD} = 5.0 V		0.10	1.09	mA
					V_{DD} = 2.7 V		0.10	1.09	
			LP (low-power main) mode	f_{IM} = 2 MHz**Note 4**	V_{DD} = 5.0 V		0.04	731	µA
					V_{DD} = 2.7 V		0.04	731	
				f_{IM} = 1 MHz**Note 4**	V_{DD} = 5.0 V		0.03	583	µA
					V_{DD} = 2.7 V		0.03	583	
			HS (high-speed main) mode	f_{MX} = 20 MHz**Note 5**, Square wave input	V_{DD} = 5.0 V		0.25	3.52	mA
					V_{DD} = 2.7 V		0.23	3.50	
			LS (low-speed main) mode	f_{MX} = 20 MHz**Note 5**, Square wave input	V_{DD} = 5.0 V		0.26	3.53	mA
					V_{DD} = 2.7 V		0.23	3.50	
				f_{MX} = 20 MHz**Note 5**, Resonator connection	V_{DD} = 5.0 V		0.44	3.78	mA
					V_{DD} = 2.7 V		0.44	3.77	
				f_{MX} = 10 MHz**Note 5**, Square wave input	V_{DD} = 5.0 V		0.16	2.00	mA
					V_{DD} = 2.7 V		0.14	1.99	
				f_{MX} = 10 MHz**Note 5**, Resonator connection	V_{DD} = 5.0 V		0.30	2.20	mA
					V_{DD} = 2.7 V		0.30	2.19	
				f_{MX} = 8 MHz**Note 5**, Square wave input	V_{DD} = 5.0 V		0.14	1.70	mA
					V_{DD} = 2.7 V		0.12	1.68	
				f_{MX} = 8 MHz**Note 5**, Resonator connection	V_{DD} = 5.0 V		0.23	1.83	mA
					V_{DD} = 2.7 V		0.23	1.82	

(Notes and Remarks are listed on the next page.)

Note 1. The listed currents are the total currents flowing into V_{DD} and EV_{DD0}, including the input leakage currents flowing when the level of the input pin is fixed to V_{DD}, EV_{DD0} or V_{SS}, EV_{SS0}. The currents in the Typ. column do not include the peripheral operating current when the CPU is placed in the HS (high-speed main), LS (low-speed main), or LV (low-voltage main) mode. The currents in the Max. column include the peripheral operating current, but do not include those flowing into the FAA, A/D converter, sample & hold circuit, D/A converter, PGA, comparator, TRNG, LVD circuit, I/O port, and on-chip pull-up-/pull-down resistors, and those flowing when the data flash memory is being rewritten. The currents in the Max. column include that of the RTC when the CPU is placed in the HALT mode.

Note 2. The listed currents apply when the HALT instruction has been fetched from the flash memory for execution.

Note 3. The listed currents apply when the high-speed system clock, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock are stopped.

Note 4. The listed currents apply when the high-speed on-chip oscillator, high-speed system clock, low-speed on-chip oscillator, and subsystem clock are stopped.

Note 5. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock are stopped.

Remark 1. f_{IH}: High-speed on-chip oscillator clock frequency

Remark 2. f_{IM}: Middle-speed on-chip oscillator clock frequency

Remark 3. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

Remark 4. f_{PLL}: PLL clock frequency (up to 96 MHz)

Remark 5. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 6. The typical value for the ambient operating temperature (T_A) is 25°C unless otherwise specified.

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (5/5)

Item	Symbol	Conditions				Min.	Typ.	Max.	Unit
Supply current **Note 1**	I_{DD2} **Note 2**	HALT mode	Subsystem clock operation mode	f_{SUB} = 32.768 kHz **Note 3**, Low-speed on-chip oscillator operation	T_A = –40°C		0.97	12.34	µA
					T_A = +25°C		1.55	12.63	
					T_A = +50°C		2.80	25.52	
					T_A = +70°C		5.54	45.91	
					T_A = +85°C		10.41	75.72	
					T_A = +105°C		23.12	165.90	
					T_A = +125°C		49.38	305.79	
				f_{SUB} = 32.768 kHz, Square wave input **Note 4**	T_A = –40°C		0.27	11.36	µA
					T_A = +25°C		1.48	16.75	
					T_A = +50°C		2.19	26.07	
					T_A = +70°C		4.93	47.35	
					T_A = +85°C		9.37	73.72	
					T_A = +105°C		22.71	168.74	
					T_A = +125°C		59.16	379.68	
				f_{SUB} = 32.768 kHz, Resonator connection **Note 5**	T_A = –40°C		0.40	8.83	µA
					T_A = +25°C		0.94	9.53	
					T_A = +50°C		2.16	22.41	
					T_A = +70°C		4.91	43.76	
					T_A = +85°C		9.71	72.66	
					T_A = +105°C		22.43	163.33	
					T_A = +125°C		48.89	304.34	
	I_{DD3}	STOP mode	Realtime clock stopped **Note 6**		T_A = –40°C		0.16	10.00	µA
					T_A = +25°C		0.63	10.00	
					T_A = +50°C		1.80	20.00	
					T_A = +70°C		4.30	40.00	
					T_A = +85°C		9.30	70.00	
					T_A = +105°C		22.00	160.00	
					T_A = +125°C		50.00	300.00	
			128Hz realtime clock operation **Note 7**		T_A = –40°C		0.24	11.00	µA
					T_A = +25°C		0.71	11.00	
					T_A = +50°C		1.95	22.00	
					T_A = +70°C		4.60	45.00	
					T_A = +85°C		9.50	80.00	
					T_A = +105°C		23.00	170.00	
					T_A = +125°C		52.00	320.00	

(Notes and Remarks are listed on the next page.)

Note 1. The listed currents are the total currents flowing into V_{DD} and EV_{DD0}, including the input leakage currents flowing when the level of the input pin is fixed to V_{DD}, EV_{DD0} or V_{SS}, EV_{SS0}. The currents in the Typ. and Max. columns do not include the peripheral operating current when the CPU is operating with the sub-system clock, or when the CPU is placed in the STOP mode, but include that of the RTC when in the HALT mode.

Note 2. The listed currents apply when the HALT instruction has been fetched from the flash memory for execution.

Note 3. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and subsystem clock are stopped, including the current flowing into the RTC. They do not include the currents flowing into the 32-bit interval timer and watchdog timer.

Note 4. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and low-speed on-chip oscillator are stopped, including the current flowing into the RTC. They do not include the currents flowing into the 32-bit interval timer and watchdog timer.

Note 5. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and low-speed on-chip oscillator are stopped, and when RTCLPC is set to 1 and the low power consumption oscillation 3 is specified (AMPHS1, AMPHS0 = 11B), including the current flowing into the RTC. They do not include the currents flowing into the 32-bit interval timer and watchdog timer.

Note 6. The listed currents apply when the low-speed on-chip oscillator and subsystem clock oscillation are stopped. They do not include the currents flowing into the RTC, 32-bit interval timer and watchdog timer. For the current for operation of the subsystem clock in the STOP mode, refer to that in the HALT mode.

Note 7. The listed currents apply when the low-speed on-chip oscillator is stopped, and when RTCLPC is set to 1 and the low power consumption oscillation 3 is specified (AMPHS1, AMPHS0 = 11B), including the current flowing into the RTC. They do not include the currents flowing into the 32-bit interval timer and watchdog timer.

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

Remark 2. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

Peripheral Functions (Common to all products)

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (1/2)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
High-speed on-chip oscillator operating current	I_{FIH} **Note 1**	HIPREC = 0			380	—	μA
		HIPREC = 1			240	—	μA
Middle-speed on-chip oscillator operating current	I_{FIM} **Note 1**				20	—	μA
Low-speed on-chip oscillator operating current	I_{FIL} **Note 1**				0.3	—	μA
RTC operating current	I_{RTC} **Notes 1, 2, 3**	f_{RTCCLK} = 32.768 kHz			0.005	—	μA
		f_{RTCCLK} = 128 Hz			0.002	—	μA
32-bit interval timer operating current	I_{IT} **Notes 1, 2, 4**				0.04	—	μA
Watchdog timer operating current	I_{WDT} **Notes 1, 2, 5**	f_{IL} = 32.768 kHz (typ.)			0.32	—	μA
A/D converter operating current	I_{ADC} **Notes 1, 6**	Conversion at maximum speed	Normal mode, AV_{REFP} = V_{DD} = 5.0 V		0.95	1.6	mA
			Low-voltage mode, AV_{REFP} = V_{DD} = 3.0 V		0.54	0.81	mA
AV_{REFP} current	I_{ADREF} **Note 7**	AV_{REFP} = 5.0 V			60	—	μA
A/D converter internal reference voltage current	I_{ADREF} **Note 1**				114	—	μA
Temperature sensor operating current	I_{TMPS} **Note 1**				110	—	μA
D/A converter operating current	I_{DAC} **Notes 1, 8**	Per channel	10-bit DAC, V_{DD} = 5.0 V		223	—	μA
			8-bit DAC, V_{DD} = 5.0 V		120	—	μA
Comparator operating current	I_{CMP} **Notes 1, 9**	Per channel			100	—	μA
<R> PGA operating current	I_{PGA} **Notes 1, 10**				460	—	μA
Sample & hold operating current	I_{SH} **Notes 1, 11**	Per channel			800	—	μA
LVD operating current	I_{LVD0} **Notes 1, 12**				0.03	—	μA
	I_{LVD1} **Notes 1, 12**				0.03	—	μA
FAA operating current	I_{FAA} **Notes 1, 13**	f_{CLK} = 48 MHz			11.0	—	mA
		f_{CLK} = 32 MHz			7.3	—	mA
True random number generator operating current	I_{TRNG}				1.6	—	mA
SMBus operating current	I_{SMBUS}				250	—	μA
Self-programming operating current	I_{FSP} **Notes 1, 14**	–40 to +105°C			2.5	12.2	mA
Data flash rewrite operating current	I_{BGO} **Notes 1, 15**				2.5	12.2	mA

(Notes and Remarks are listed on the next page.)

(T_A = –40 to +125°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (2/2)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
SNOOZE operating current	I_{SNOZ} **Note 1**	ADC to be in use	The ADC is shifting to the SNOOZE mode. **Note 16**		0.7	1.2	mA
			The ADC is operating in the low-voltage mode, AV_{REFP} = V_{DD} = 3.0 V		1.2	2.0	
		Simplified SPI (CSI)/UART to be in use			0.7	1.07	
Low-speed peripheral clock supply current	I_{SXP} **Notes 1, 17**	RTCLPC = 0			0.27	—	μA
Output current control operating current	I_{CCDA} **Notes 1, 18**	The setting of the CCDE register is not 00H.			100	—	μA
	I_{CCDP} **Notes 19, 20**	Per single controlled current drive port	Low-level output current setting: Hi-Z		30	—	μA
			Low-level output current setting: 2 to 15 mA		210	—	μA

Note 1. This current flows into V_{DD}.

Note 2. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, and high-speed system clock are stopped.

Note 3. This current only flows to the realtime clock (RTC). It does not include the operating current of the low-speed on-chip oscillator or the XT1 oscillator. The supply current of the RL78 microcontrollers is the sum of I_{DD1} or I_{DD2}, and IRTC when the realtime clock is operating in the operation mode or HALT mode. When the low-speed on-chip oscillator is selected, IFIL should be included in the supply current. I_{DD2} in the subsystem clock operation mode includes the operating current of the realtime clock.

Note 4. This current only flows to the 32-bit interval timer. It does not include the operating current of the low-speed on-chip oscillator or the XT1 oscillator. The supply current of the RL78 microcontrollers is the sum of I_{DD1} or I_{DD2}, and I_{IT} when the 32-bit interval timer is operating or in the HALT mode.

Note 5. This current only flows to the watchdog timer. It includes the operating current of the low-speed on-chip oscillator. The supply current of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3}, and I_{WDT} when the watchdog timer is operating.

Note 6. This current only flows to the A/D converter. The supply current of the RL78 microcontrollers is the sum of I_{DD1} or I_{DD2}, and I_{ADC} when the A/D converter is operating or in the HALT mode.

Note 7. This current flows into AV_{REFP}.

Note 8. This current only flows to the D/A converter. The supply current of the RL78 microcontrollers is the sum of I_{DD1} or I_{DD2}, and I_{DAC} when the D/A converter is operating.

Note 9. This current only flows to the comparator circuit. The supply current of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3}, and I_{CMP} when the comparator circuit is operating.

Note 10. This current only flows to the PGA circuit. The supply current of the RL78 microcontrollers is the sum of I_{DD1} or I_{DD2}, and I_{PGA} when the PGA circuit is operating.

Note 11. This current only flows to the sample & hold circuit. The supply current of the RL78 microcontrollers is the sum of I_{DD1} or I_{DD2}, and I_{SH} when the sample & hold circuit is operating.

Note 12. This current only flows to the LVD circuit. The supply current of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3}, and I_{LVD} when the LVD circuit is operating.

Note 13. This current only flows to the FAA circuit. The supply current of the RL78 microcontrollers is the sum of I_{DD1} or I_{DD2}, and I_{FAA} when the FAA circuit is operating.

Note 14. This current only flows during self-programming.

Note 15. This current only flows while the data flash memory is being rewritten.

Note 16. For shift time to the SNOOZE mode, see **20.9 SNOOZE Mode Function**.

Note 17. This current is added to the supply current in the STOP mode when the setting of RTCLPC is 0 with the subsystem clock X (f_{SX}) oscillating, or in the HALT mode when the setting of RTCLPC is 0 with the subsystem clock X (f_{SX}) selected as the CPU clock.

Note 18. This current is added to the supply current when the controlled current drive port is set.

Note 19. This current does not include the current flowing into the I/O ports.

Note 20. This current flows into EV_{DD0} and EV_{DD1}.

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

Remark 2. f_{SX}: Subsystem clock X frequency

Remark 3. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 4. The typical value for the ambient operating temperature is 25°C.

44.4 AC Characteristics

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (1/2)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Instruction cycle	T_{CY}	Main system clock (f_{MAIN}) operation	HS (high-speed main) mode (Prefetch ON)	0.02083		1	μs
			HS (high-speed main) mode (Prefetch OFF)	0.03125		1	μs
			LS (low-speed main) mode	0.04167		1	μs
			LP (low-power main) mode	0.5		1	μs
		Subsystem clock (f_{SUB}) operation		26.041	30.5	31.3	μs
		Self-programming mode	HS (high-speed main) mode	0.03125		1	μs
			LS (low-speed main) mode	0.04167		1	μs
External system clock frequency	f_{EX}			1.0		20.0	MHz
	f_{EXS}			32		38.4	kHz
External system clock input high-level width, low-level width	t_{EXH}, t_{EXL}			24			ns
	t_{EXHS}, t_{EXLS}			13.7			μs
TI00 to TI03 input high-level width, low-level width	t_{TIH}, t_{TIL}			1/f_{MCK} + 10			ns
Timer RJ input cycle	t_C	TRJIO		100			ns
Timer RJ input high-level width, low-level width	t_{TJIH}, t_{TJIL}	TRJIO		40			ns
Timer RD2 input high-level width, low-level width	t_{TDIH}, t_{TDIL}	TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1, TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1		3/f_{CLK}			ns
Timer RD2 forcible shut-off signal input low-level width	t_{TDSIL}	P137/INTP0	2 MHz ≤ f_{CLK} ≤ 48 MHz	1			μs
			f_{CLK} ≤ 2 MHz	1/f_{CLK} + 1			μs
Timer RG2 input high-level width, low-level width	t_{TGIH}, t_{TGIL}	TRGIOA, TRGIOB, TRGIDZ, TRGTRG		2.5/f_{CLK}			ns
TO00 to TO03 TKBO00, TKBO01, TKBO10, TKBO11, TKBO20, TKBO21, TRJIO0, TRJO0, TRGIOA, TRGIOB, TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1, TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1 output frequency	f_{TO}	HS (high-speed main) mode LS (low-speed main) mode	4.0 V ≤ EV_{DD0} ≤ 5.5 V			16	MHz
			2.7 V ≤ EV_{DD0} < 4.0 V			8	MHz
		LP (low-power main) mode				2	MHz
PCLBUZ0, PCLBUZ1 output frequency	f_{PCL}	HS (high-speed main) mode LS (low-speed main) mode	4.0 V ≤ EV_{DD0} ≤ 5.5 V			16	MHz
			2.7 V ≤ EV_{DD0} < 4.0 V			8	MHz
		LP (low-power main) mode				2	MHz
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}	INTP0, INTP20, INTP21	2.7 V ≤ V_{DD} ≤ 5.5 V	1			μs
		INTP1 to INTP11	2.7 V ≤ EV_{DD0} ≤ 5.5 V	1			μs
Key interrupt input high-level width, low-level width	t_{KRH}, t_{KRL}	KR0 to KR7	2.7 V ≤ EV_{DD0} ≤ 5.5 V	250			ns

(Remark is listed on the next page.)

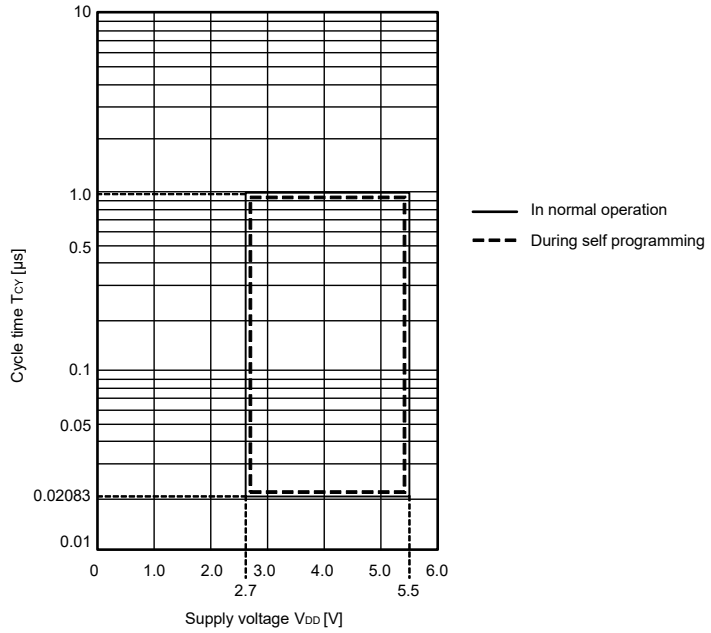
(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (2/2)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
$\overline{\text{RESET}}$ low-level width	t_{RSL}		10			μs

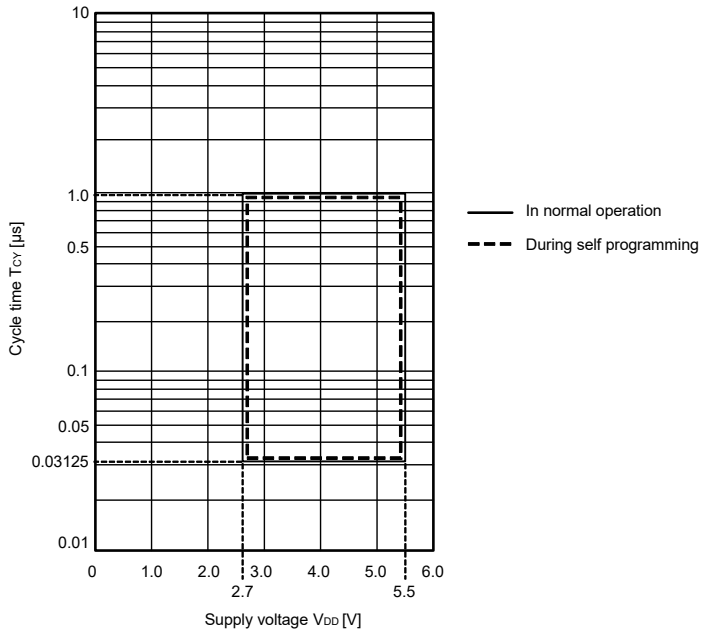
Remark f_{MCK}: Timer array unit operating clock frequency
To set this operating clock, use the CKSmn0 and CKSmn1 bits of the timer mode register mn (TMRmn).
m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Minimum Instruction Execution Time during Main System Clock Operation

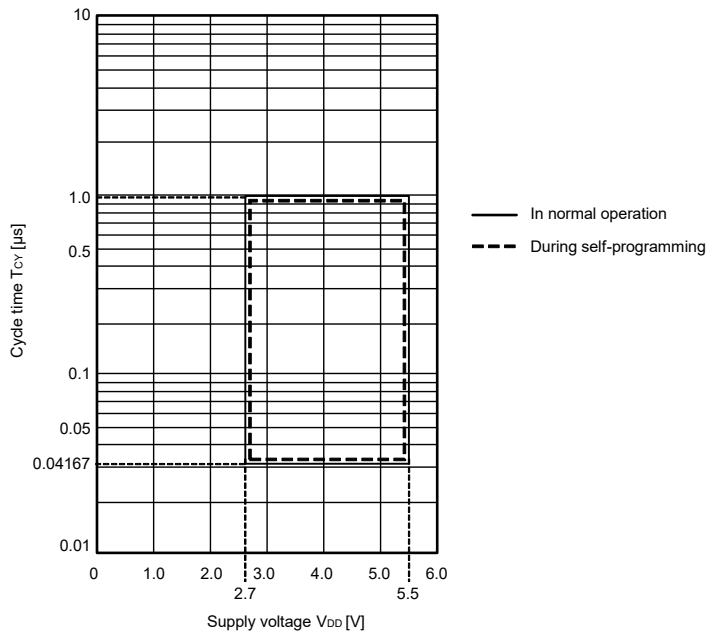
T_{CY} vs V_{DD} (HS (high-speed main) mode) prefetch ON



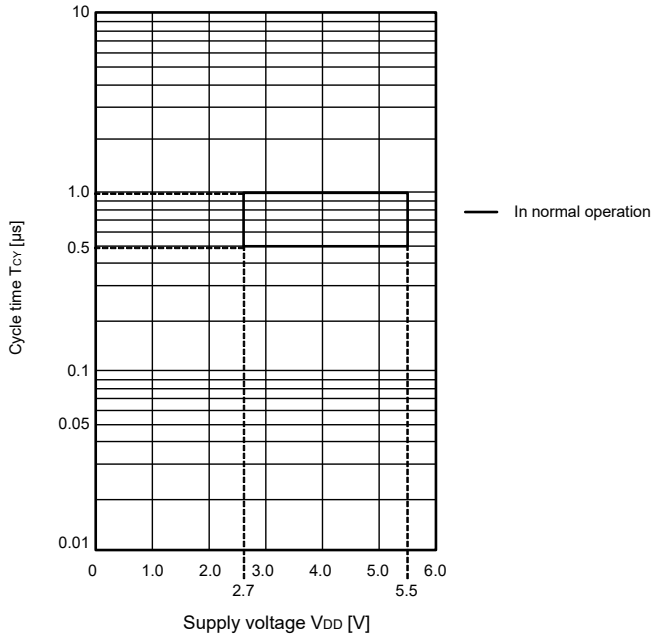
T_{CY} vs V_{DD} (HS (high-speed main) mode) prefetch OFF



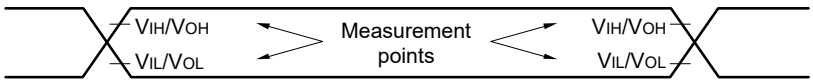
T_{CY} vs V_{DD} (LS (low-speed main) mode)



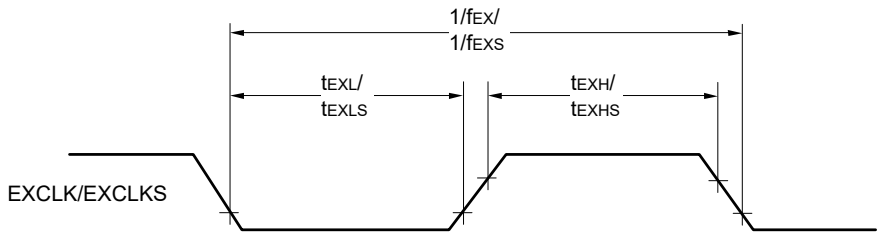
T_{CY} vs V_{DD} (LP (low-power main) mode)



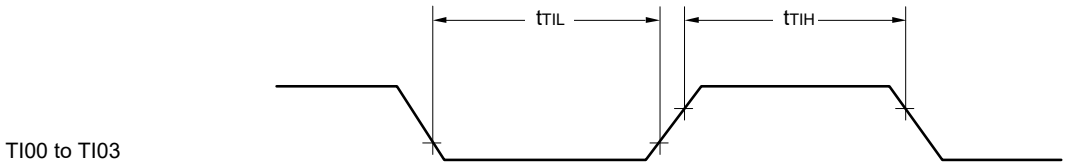
AC Timing Measurement Points

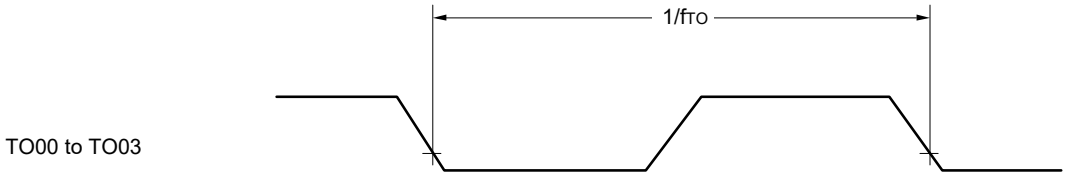


External System Clock Timing



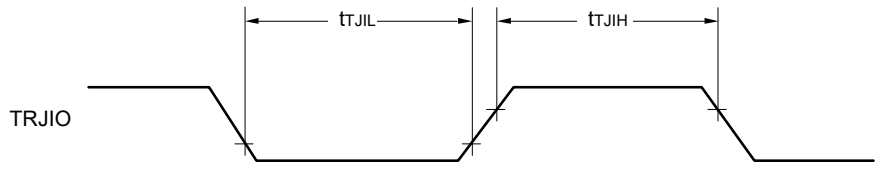
TI/TO Timing



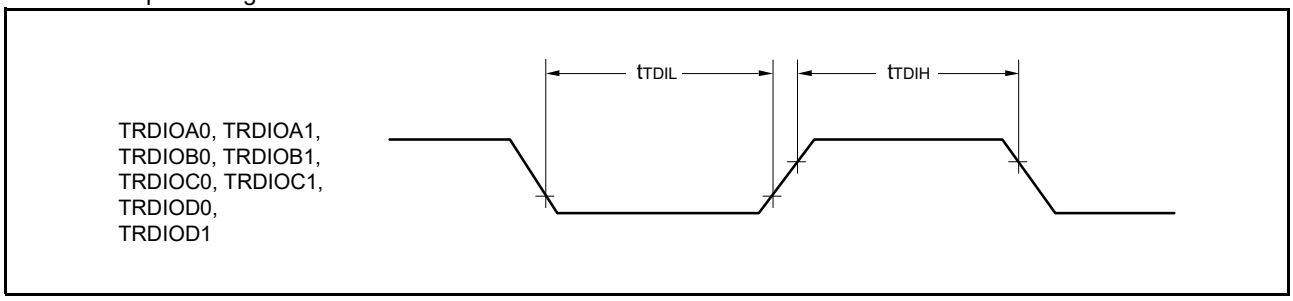


TKBO00, TKBO01, TKBO10, TKBO11,
TKBO20, TKBO21,
TRJIO0, TRJO0, TRGIOA, TRGIOB,
TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1,
TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1

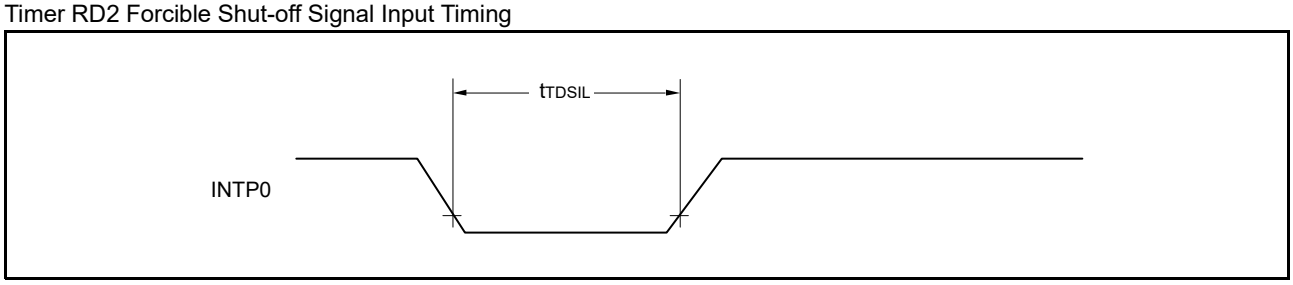
Timer RJ Input Timing



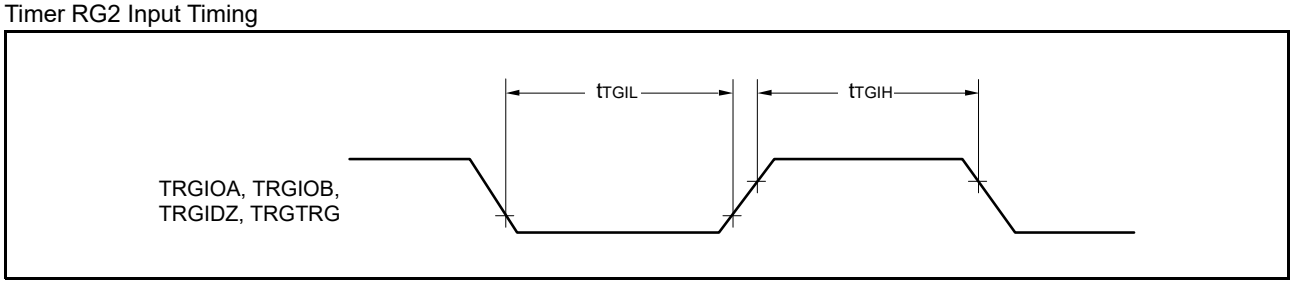
Timer RD2 Input Timing



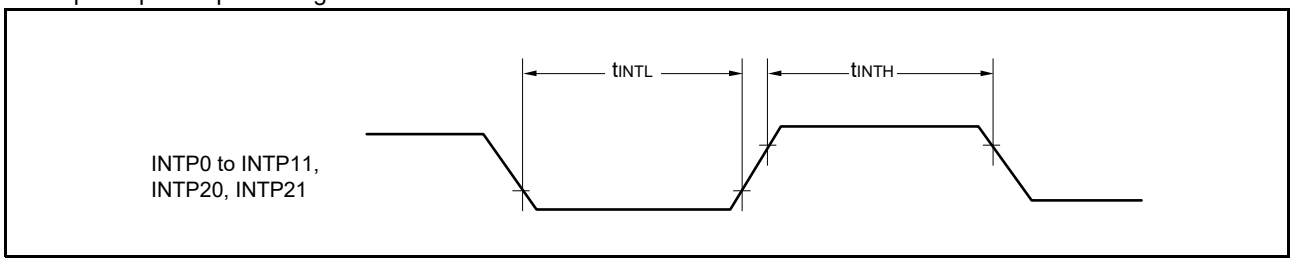
Timer RD2 Forcible Shut-off Signal Input Timing



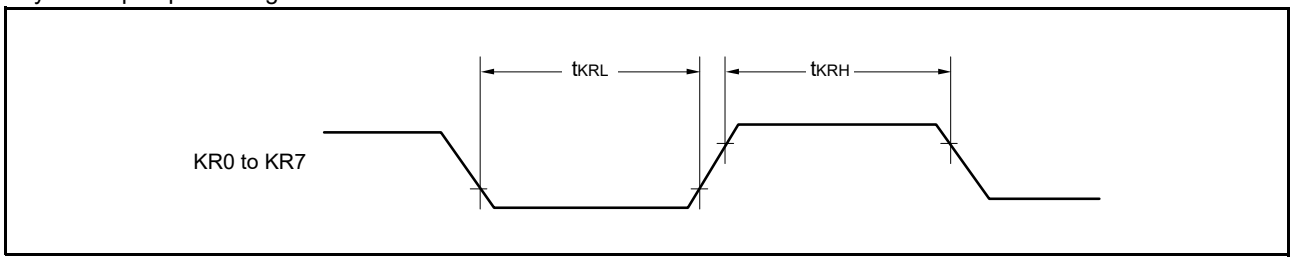
Timer RG2 Input Timing



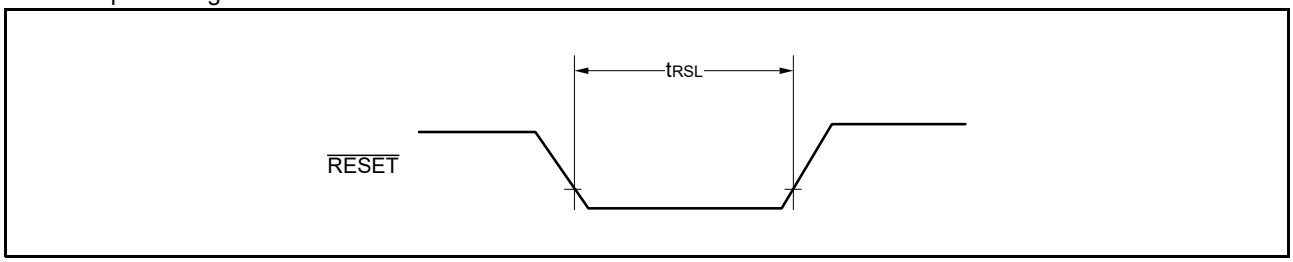
Interrupt Request Input Timing



Key Interrupt Input Timing

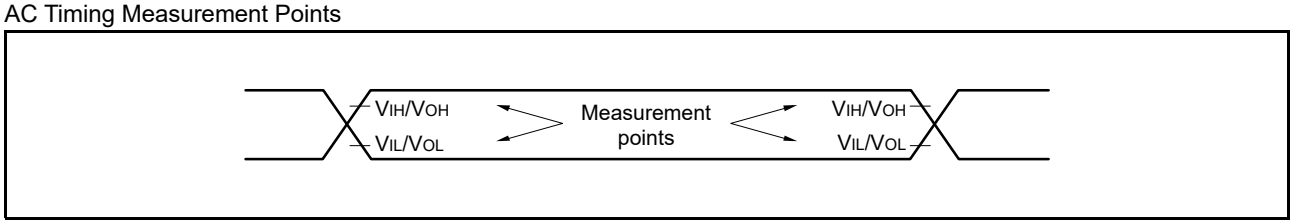


$\overline{\text{RESET}}$ Input Timing



44.5 Characteristics of the Peripheral Functions

AC Timing Measurement Points



44.5.1 Serial array unit

1. In UART communications with devices operating at same voltage levels

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz)

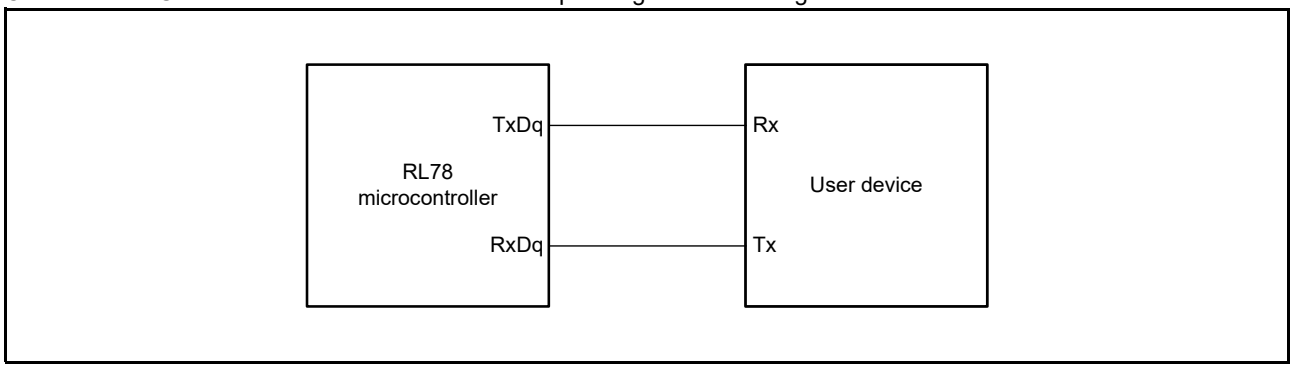
Item	Symbol	Conditions	HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
Transfer rate **Note 1**		2.7 V ≤ EV_{DD0} ≤ 5.5 V		f_{MCK}/6		f_{MCK}/6		f_{MCK}/6	bps
		Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} **Note 2**		5.3		4		0.33	Mbps

Note 1. The transfer rate in the SNOOZE mode is within the range from 4800 to 9600 bps.

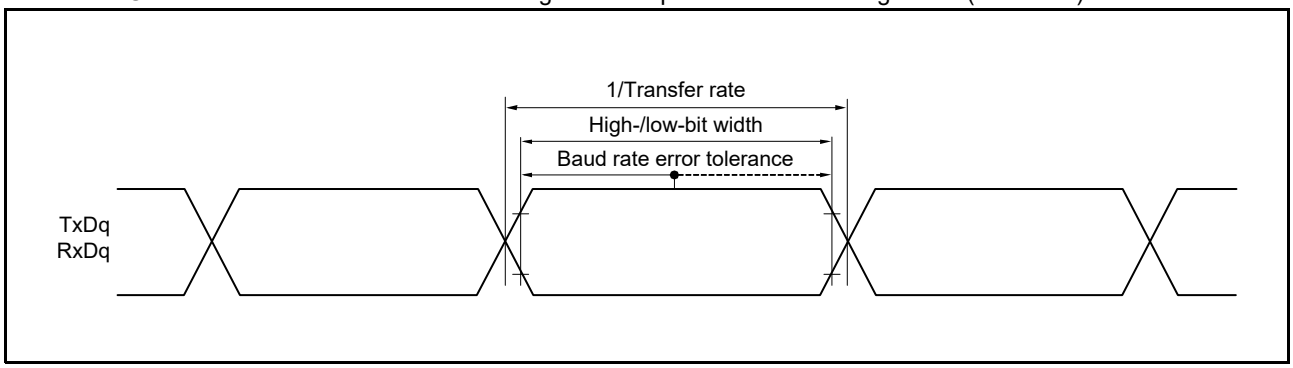
Note 2. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are as follows.
HS (high-speed main) mode: 48 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)
LS (low-speed main) mode: 24 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)
LP (low-power main) mode: 2 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)

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).

Connection in UART communications with devices operating at same voltage levels



Bit width in UART communications when interfacing devices operate at same voltage level (reference)



Remark 1. q: UART number (q = 0 to 2), g: PIM and POM number (g = 0, 1, 3, 5, 7)

Remark 2. f_{MCK}: Serial array unit operation clock frequency
To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00 to 03, 10, 11)

2. In simplified SPI (CSI) communications in the master mode with devices operating at same voltage levels with the internal SCKp clock

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz)

Item	Symbol	Conditions		HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 4/f_{CLK}	2.7 V ≤ EV_{DD0} ≤ 5.5 V	125		166		2000		ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY1}/2 – 12		t_{KCY1}/2 – 21		t_{KCY1}/2 – 50		ns
		2.7 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY1}/2 – 18		t_{KCY1}/2 – 25		t_{KCY1}/2 – 50		ns
SIp setup time (to SCKp↑)**Note 1**	t_{SIK1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V		44		54		110		ns
		2.7 V ≤ EV_{DD0} ≤ 5.5 V		44		54		110		ns
SIp hold time (from SCKp↑) **Note 1**	t_{KSI1}	2.7 V ≤ EV_{DD0} ≤ 5.5 V		19		19		19		ns
Delay time from SCKp↓ to SOp output**Note 2**	t_{KSO1}	2.7 V ≤ EV_{DD0} ≤ 5.5 V C = 30 pF**Note 3**			25		25		25	ns

Note 1. The setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The setting for the SIp setup time becomes “to SCKp↓” and that for the SIp hold time becomes “from SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The setting for the delay time to SOp output becomes “from SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. C is the load capacitance of the SCKp and SOp output lines.

Caution Select the normal input buffer for the SIp pin and normal output mode for the SOp and SCKp pins by using the port input mode register g (PIMg) and the port output mode register g (POMg).

Remark 1. p: CSI number (p = 00, 01, 10, 11, 20, 21), m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), g: PIM and POM number (g = 0, 1, 3, 5, 7)

Remark 2. f_{MCK}: Serial array unit operating clock frequency
To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00 to 03, 10, 11)

3. In simplified SPI (CSI) communications in the slave mode with devices operating at same voltage levels with the external SCKp clock

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz)

Item	Symbol	Conditions		HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
SCKp cycle time **Note 4**	t_{KCY2}	4.0 V ≤ EV_{DD0} ≤ 5.5 V	20 MHz < f_{MCK}	8/f_{MCK}		8/f_{MCK}		—		ns
			f_{MCK} ≤ 20 MHz	6/f_{MCK}		6/f_{MCK}		6/f_{MCK}		ns
		2.7 V ≤ EV_{DD0} ≤ 5.5 V	16 MHz < f_{MCK}	8/f_{MCK}		8/f_{MCK}		—		ns
			f_{MCK} ≤ 16 MHz	6/f_{MCK}		6/f_{MCK}		6/f_{MCK}		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY2}/2 – 7		t_{KCY2}/2 – 7		t_{KCY2}/2 – 7		ns
		2.7 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY2}/2 – 8		t_{KCY2}/2 – 8		t_{KCY2}/2 – 8		ns
SIp setup time (to SCKp↑) **Note 1**	t_{SIK2}	2.7 V ≤ EV_{DD0} ≤ 5.5 V		1/f_{MCK} + 20		1/f_{MCK} + 30		1/f_{MCK} + 30		ns
SIp hold time (to SCKp↑) **Note 1**	t_{SIK2}	2.7 V ≤ EV_{DD0} ≤ 5.5 V		1/f_{MCK} + 31		1/f_{MCK} + 31		1/f_{MCK} + 31		ns
Delay time from SCKp↓ to SOp output **Note 2**	t_{KSO2}	C = 30 pF **Note 3**	2.7 V ≤ EV_{DD0} ≤ 5.5 V		2/f_{MCK} + 44		2/f_{MCK} + 110		2/f_{MCK} + 110	ns

Note 1. The setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The setting for the SIp setup time becomes “to SCKp ↓” and that for the SIp hold time becomes “from SCKp ↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The setting for the delay time to SOp output becomes “from SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. C is the load capacitance of the SOp output line.

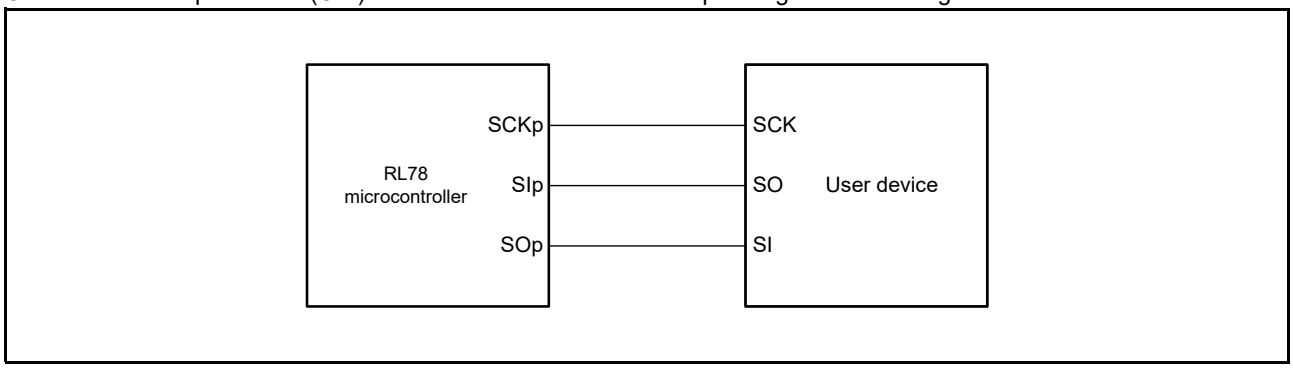
Note 4. The transfer rate in the SNOOZE mode is 1 Mbps maximum.

Caution Select the normal input buffer for the SIp and SCKp pins and normal output mode for the SOp pin by using the port input mode register g (PIMg) and the port output mode register g (POMg).

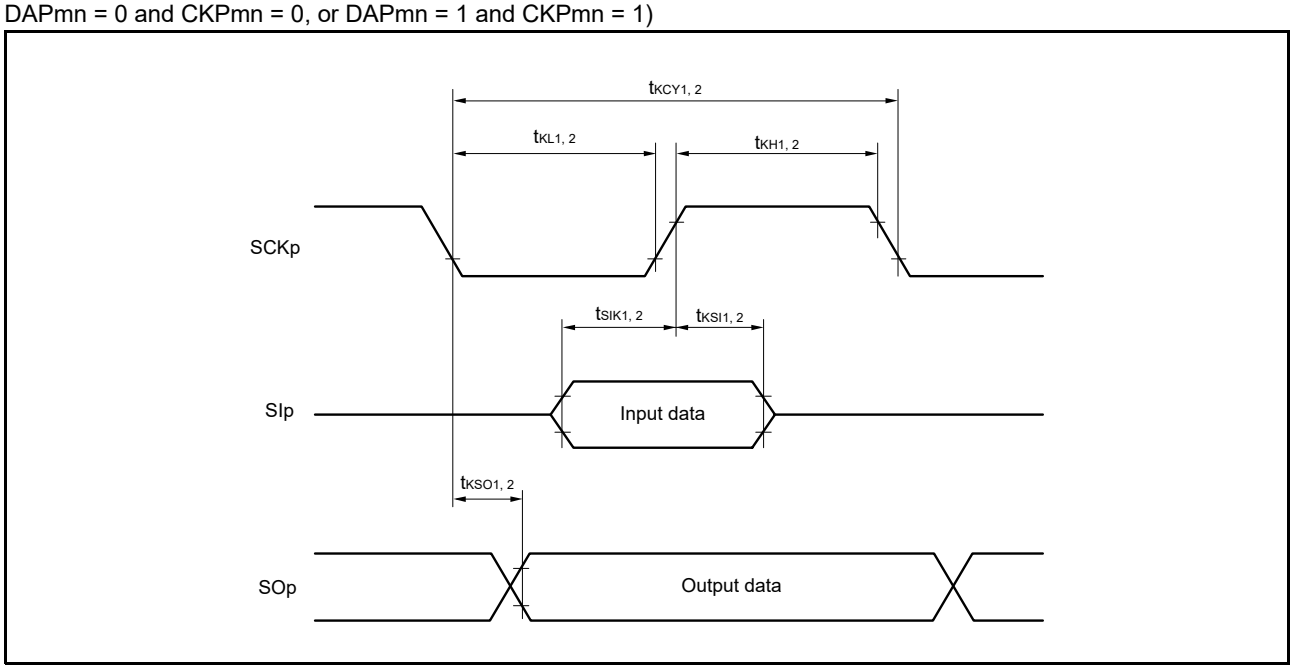
Remark 1. p: CSI number (p = 00, 01, 10, 11, 20, 21), m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), g: PIM and POM number (g = 0, 1, 3, 5, 7)

Remark 2. f_{MCK}: Serial array unit operating clock frequency
To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00 to 03, 10, 11)

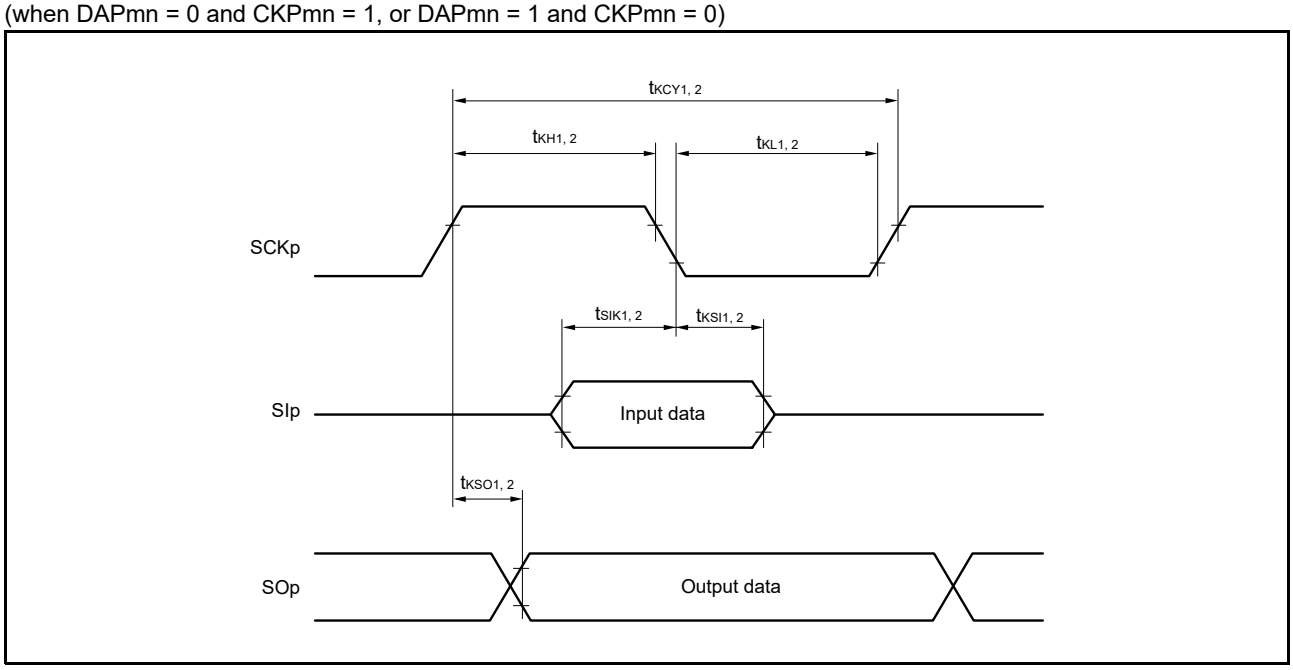
Connection in simplified SPI (CSI) communications with devices operating at same voltage levels



Timing of serial transfer in simplified SPI (CSI) communications with devices operating at same voltage levels (when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)



Timing of serial transfer in simplified SPI (CSI) communications with devices operating at same voltage levels (when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)



Remark 1. p: CSI number (p = 00, 01, 10, 11, 20, 21)

Remark 2. m: Unit number, n: Channel number (mn = 00 to 03, 10, 11)

4. In simplified I^2C communications with devices operating at same voltage levels

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz)

Item	Symbol	Conditions	HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
SCLr clock frequency	f_{SCL}	2.7 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ		1000 **Note 1**		1000 **Note 1**		400 **Note 1**	kHz
Hold time when SCLr is low	t_{LOW}	2.7 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	475		475		1150		ns
Hold time when SCLr is high	t_{HIGH}	2.7 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	475		475		1150		ns
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	1/f_{MCK} + 85 **Note 2**		1/f_{MCK} + 85 **Note 2**		1/f_{MCK} + 145 **Note 2**		ns
Data hold time (transmission)	$t_{HD:DAT}$	2.7 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	0	305	0	305	0	305	ns

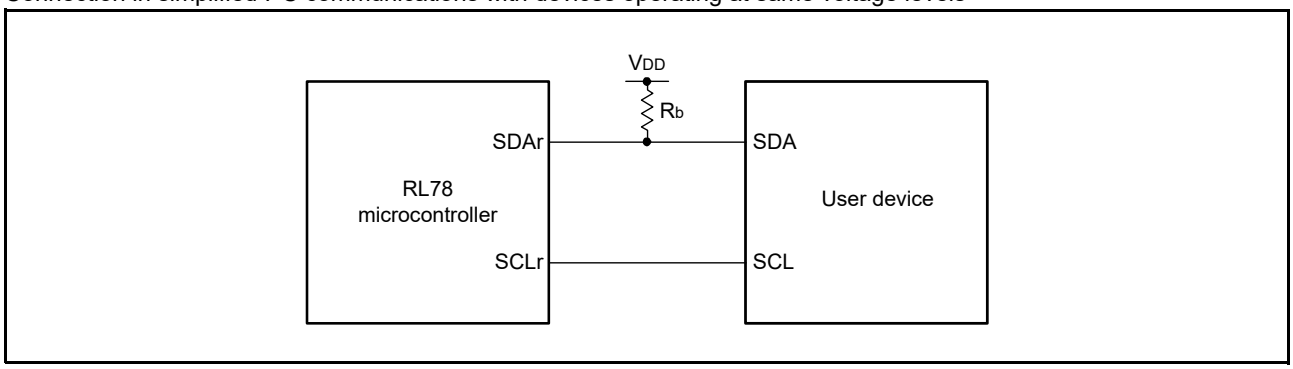
Note 1. The listed frequencies must be no greater than f_{MCK}/4.

Note 2. Set the f_{MCK} value that does not exceed the hold time when SCLr is low or high.

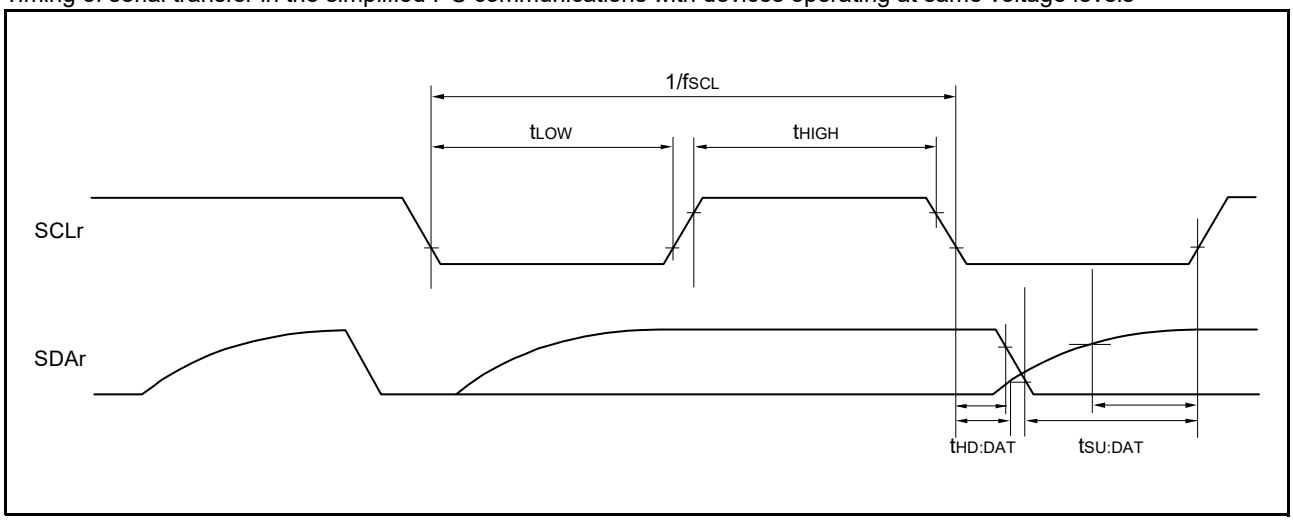
Caution Select the normal input buffer and the N-ch open drain output (V_{DD} withstand voltage for 20- to 52-pin products/ EV_{DD} withstand voltage for 64-pin products) mode for the SDAr pin and the normal output mode for the SCLr pin by using the port input mode register g (PIMg) and the port output mode register h (POMh).

(Remarks are listed on the next page.)

Connection in simplified I²C communications with devices operating at same voltage levels



Timing of serial transfer in the simplified I²C communications with devices operating at same voltage levels



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, 01, 10, 11, 20, 21), g: PIM number (g = 0, 1, 3, 5, 7), h: POM number (h = 0, 1, 3, 5, 7)

Remark 3. f_{MCK}: Serial array unit operating clock frequency
To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).
m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

5. In UART communications with devices operating at different voltage levels (2.5 V, 3 V)

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz)

Item	Symbol	Conditions		HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
Transfer rate		Reception	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		f_{MCK}/6 **Note 1**		f_{MCK}/6 **Note 1**		f_{MCK}/6 **Note 1**	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} **Note 2**		5.3		4		0.33	Mbps
			2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		f_{MCK}/6 **Note 1**		f_{MCK}/6 **Note 1**		f_{MCK}/6 **Note 1**	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} **Note 2**		5.3		4		0.33	Mbps
		Trans-mission	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		**Note 3**		**Note 3**		**Note 3**	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 4**		2.8 **Note 4**		2.8 **Note 4**	Mbps
			2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		**Note 5**		**Note 5**		**Note 5**	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 6**		1.2 **Note 6**		1.2 **Note 6**	Mbps

(Notes, Caution, and Remarks are listed on the next page.)

Note 1. Transfer rate in the SNOOZE mode is within the range from 4800 to 9600 bps.

Note 2. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
HS (high-speed main) mode: 48 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)
LS (low-speed main) mode: 24 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)
LP (low-power main) mode: 2 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)

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 4.0 V ≤ EV_{DD0} ≤ 5.5 V, 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 4. This rate is calculated as an example when the conditions described in the “Conditions” column are met. See **Note 3** above to calculate the maximum transfer rate under conditions of the customer.

Note 5. 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 ≤ EV_{DD0} < 4.0 V, 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 6. This rate is calculated as an example when the conditions described in the “Conditions” column are met. See **Note 5** 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} withstand voltage for 20- to 52-pin products/EV_{DD} withstand voltage for 64-pin products) for the TxDq pin by using the port input mode register g (PIMg) and the port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with the TTL input buffer selected.**

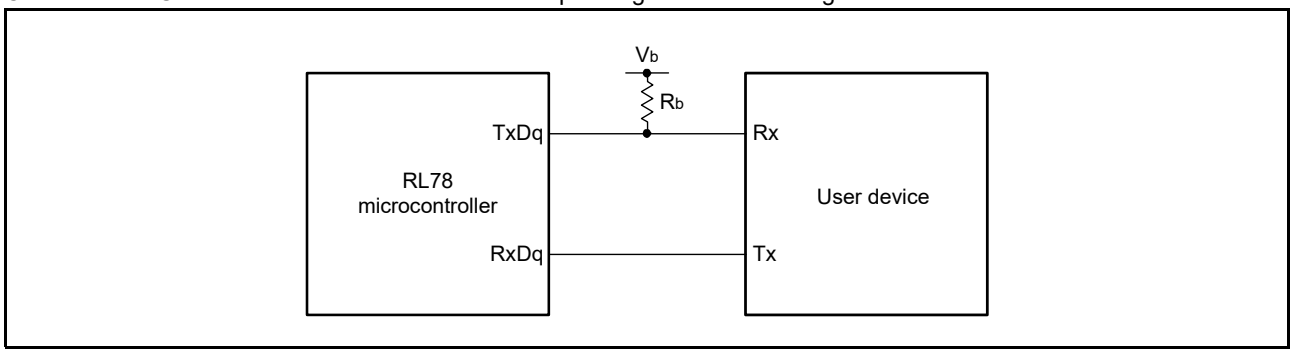
Remark 1. V_b[V]: Communication line voltage

Remark 2. q: UART number (q = 0 to 3), g: PIM and POM number (g = 0, 1, 3, 5, 7)

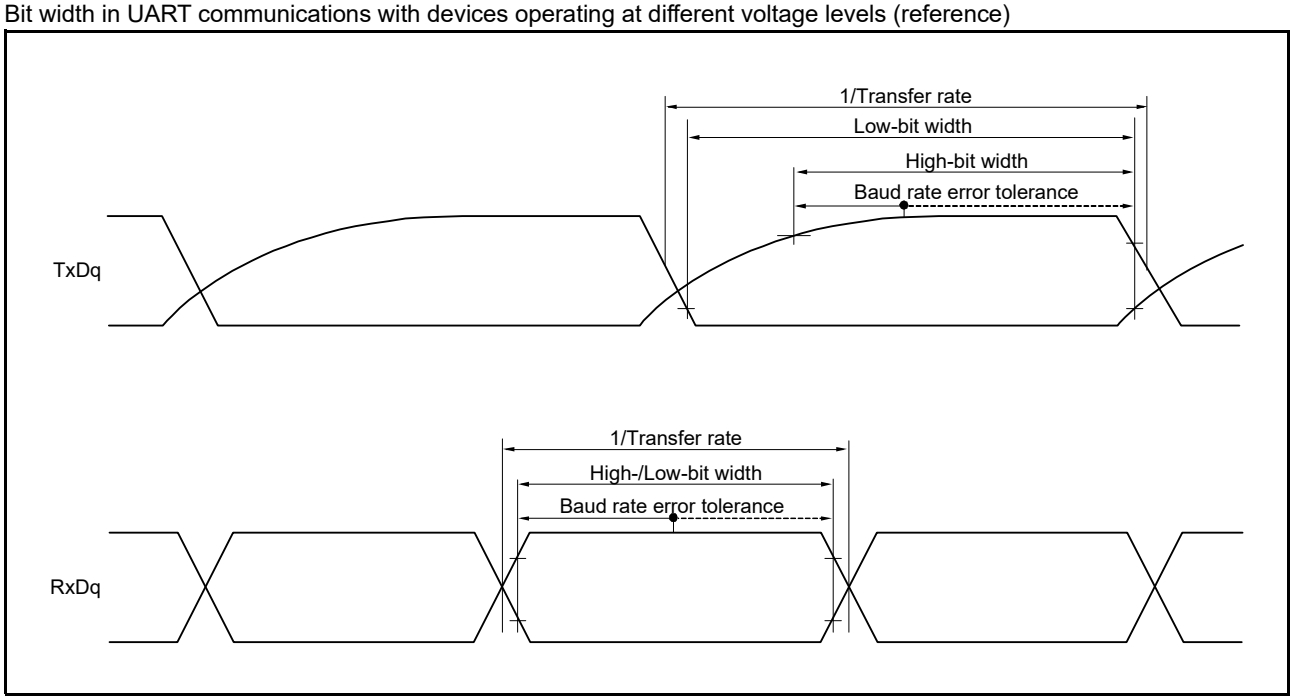
Remark 3. f_{MCK}: Serial array unit operation clock frequency
To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn) (m: Unit number, n: Channel number = 00 to 03, 10, 11).

Remark 4. Communications by using UART2 with devices operating at different voltage levels are not possible when bit 1 (PIOR01) of the peripheral I/O redirection register 0 (PIOR0) is set to 1.

Connection in UART communications with devices operating at different voltage levels



Bit width in UART communications with devices operating at different voltage levels (reference)



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 to 3), g: PIM and POM number (g = 0, 1, 3, 5, 7)

Remark 3. f_{MCK}: Serial array unit operating clock frequency
To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00 to 03, 10, 11)

Remark 4. Communications by using UART2 with devices operating at different voltage levels are not possible when bit 1 (PIOR01) of the peripheral I/O redirection register 0 (PIOR0) is set to 1.

6. In simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels (2.5 V or 3 V) with the internal SCKp clock

(T_A = –40 to +125°C, 1.8 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz) (1/2)

Item	Symbol	Conditions		HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 4/f_{CLK}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	300		300		2300		ns
			2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	500		500		2300		ns
SCKp high-level width	t_{KH1}	4.0 V ≤ EV_{DD0} ≤ 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		t_{KCY1}/2 – 75		ns
		2.7 V ≤ EV_{DD0} < 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		t_{KCY1}/2 – 170		ns
SCKp low-level width	t_{KL1}	4.0 V ≤ EV_{DD0} ≤ 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 – 12		t_{KCY1}/2 – 50		ns
		2.7 V ≤ EV_{DD0} < 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 – 18		t_{KCY1}/2 – 50		ns
SIp setup time (to SCKp↑)**Note**	t_{SIK1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		81		81		479		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		177		177		479		ns
SIp hold time (from SCKp↑)**Note**	t_{KSI1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		19		19		19		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		19		19		19		ns
Delay time from SCKp↓ to SOp output**Note**	t_{KSO1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ			100		100		100	ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			195		195		195	ns

(Note and Caution are listed on the next page, and Remarks are listed on page 1962.)

6. In simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels (2.5 V or 3 V) with the internal SCKp clock

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz) (2/2)

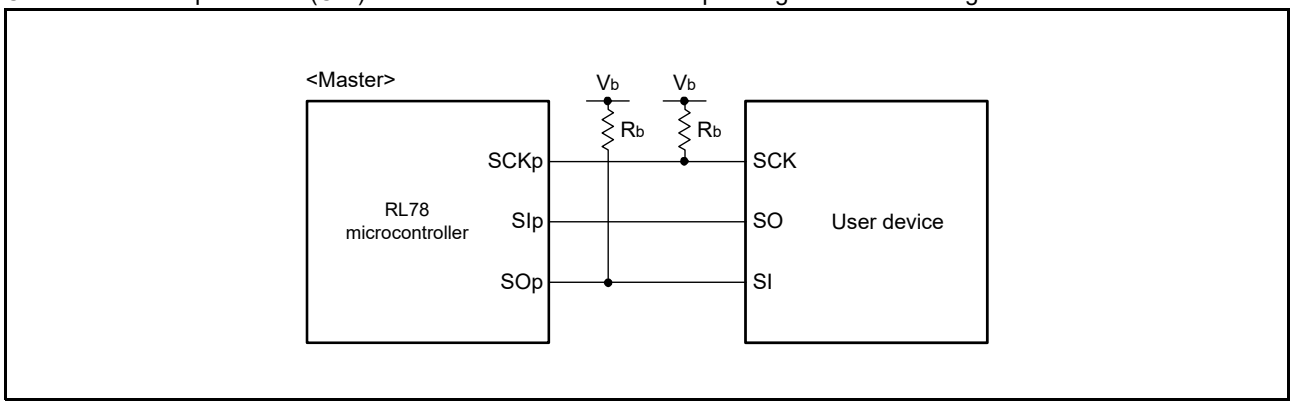
Item	Symbol	Conditions	HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
SIp setup time (to SCKp↓)**Note**	t_{SIK1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	44		44		110		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	44		44		110		ns
SIp hold time (from SCKp↓)**Note**	t_{KSI1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	19		19		19		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	19		19		19		ns
Delay time from SCKp↑ to SOp output**Note**	t_{KSO1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		25		25		25	ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		25		25		25	ns

Note This setting applies 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} withstand voltage for 20- to 52-pin products/EV_{DD} withstand voltage for 64-pin products) for the SOp and SCKp pins by using the port input mode register g (PIMg) and the port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with the TTL input buffer selected.

(Remarks are listed on the next page.)

Connection in simplified SPI (CSI) communications with devices operating at different voltage levels



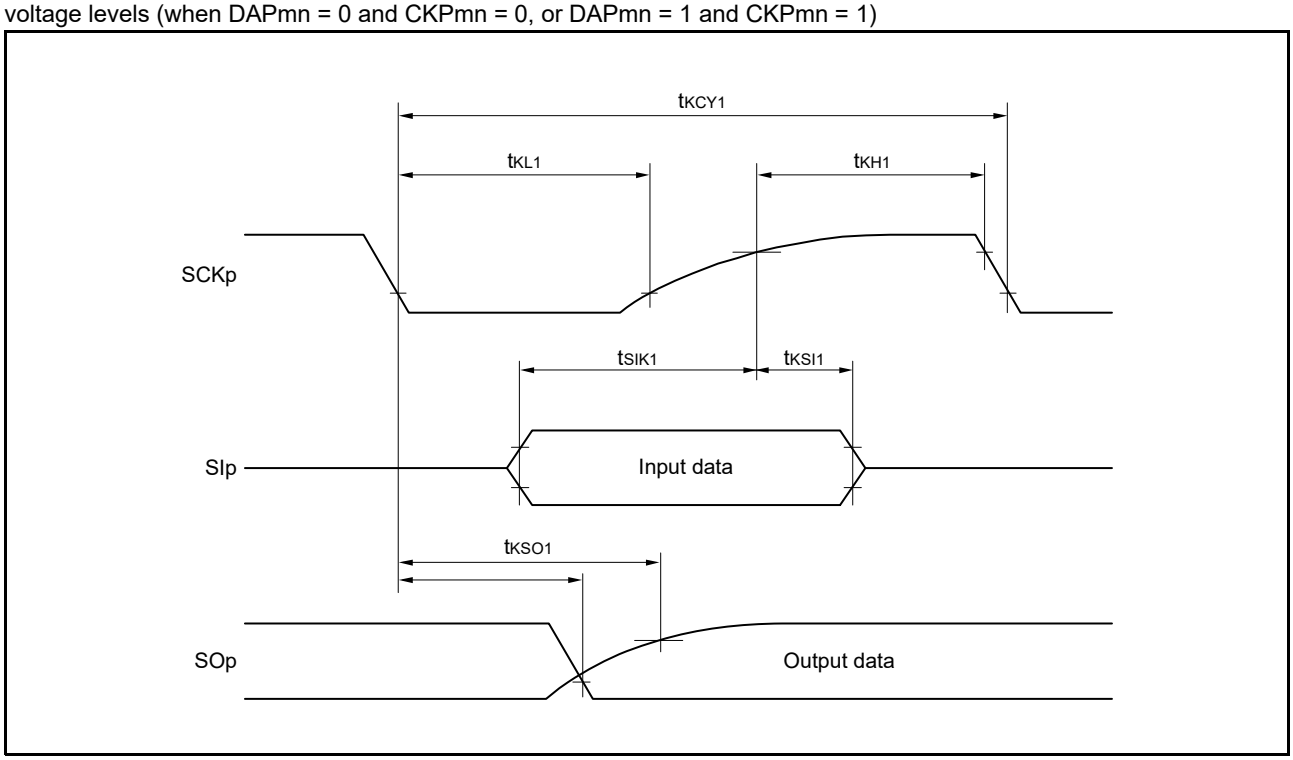
Remark 1. R_b[Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

Remark 2. p: CSI number (p = 00, 01, 10, 20), m: Unit number, n: Channel number (mn = 00, 01, 02, 10), g: PIM and POM number (g = 0, 1, 3, 5, 7)

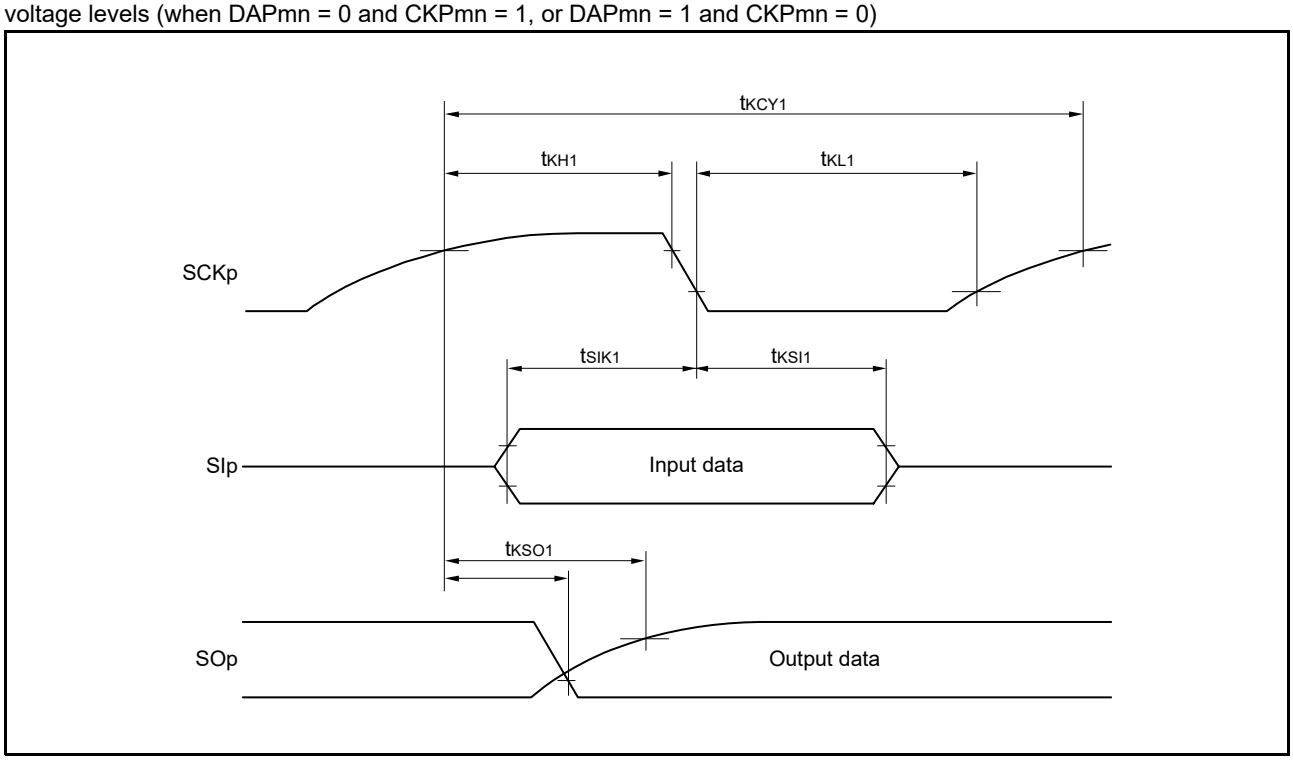
Remark 3. f_{MCK}: Serial array unit operating clock frequency
To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00)

Remark 4. Communications by using CSI01 of 48-, 52-, and 64-pin products, and CSI11 and CSI21 with devices operating at different voltage levels are not possible. Use other CSI channels to handle such communications.

Timing of serial transfer in simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels (when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)



Timing of serial transfer in simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels (when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)



Remark 1. p: CSI number (p = 00, 01, 10, 20), m: Unit number, n: Channel number (mn = 00, 01, 02, 10), g: PIM and POM number (g = 0, 1, 3, 5, 7)

Remark 2. Communications by using CSI01 of 48-, 52-, and 64-pin products, and CSI11 and CSI21 with devices operating at different voltage levels are not possible. Use other CSI channels to handle such communications.

7. In simplified SPI (CSI) communications in the slave mode with devices operating at different voltage levels (2.5 V or 3 V) with the external SCKp clock

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz)

Item	Symbol	Conditions		HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
SCKp cycle time **Note 1**	t_{KCY2}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	24 MHz < f_{MCK}	14/f_{MCK}		—		—		ns
			20 MHz < f_{MCK} ≤ 24 MHz	12/f_{MCK}		12/f_{MCK}		—		ns
			8 MHz < f_{MCK} ≤ 20 MHz	10/f_{MCK}		10/f_{MCK}		—		ns
			4 MHz < f_{MCK} ≤ 8 MHz	8/f_{MCK}		8/f_{MCK}		—		ns
			f_{MCK} ≤ 4 MHz	6/f_{MCK}		6/f_{MCK}		10/f_{MCK}		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V	24 MHz < f_{MCK}	20/f_{MCK}		—		—		ns
			20 MHz < f_{MCK} ≤ 24 MHz	16/f_{MCK}		16/f_{MCK}		—		ns
			16 MHz < f_{MCK} ≤ 20 MHz	14/f_{MCK}		14/f_{MCK}		—		ns
			8 MHz < f_{MCK} ≤ 16 MHz	12/f_{MCK}		12/f_{MCK}		—		ns
			4 MHz < f_{MCK} ≤ 8 MHz	8/f_{MCK}		8/f_{MCK}		—		ns
			f_{MCK} ≤ 4 MHz	6/f_{MCK}		6/f_{MCK}		10/f_{MCK}		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		t_{KCY2}/2 – 12		t_{KCY2}/2 – 12		t_{KCY2}/2 – 50		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		t_{KCY2}/2 – 18		t_{KCY2}/2 – 18		t_{KCY2}/2 – 50		ns
SIp setup time (to SCKp↑) **Note 2**	t_{SIK2}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		1/f_{MCK} + 20		1/f_{MCK} + 20		1/f_{MCK} + 30		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		1/f_{MCK} + 20		1/f_{MCK} + 20		1/f_{MCK} + 30		ns
SIp hold time (from SCKp↑) **Note 2**	t_{KSI2}			1/f_{MCK} + 31		1/f_{MCK} + 31		1/f_{MCK} + 31		ns
Delay time from SCKp↓ to SOp output **Note 3**	t_{KSO2}	4.0 V ≤ EV_{DD0} ≤ 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} + 120		2/f_{MCK} + 573	ns
		2.7 V ≤ EV_{DD0} < 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} + 214		2/f_{MCK} + 573	ns

Note 1. Transfer rate in the SNOOZE mode: 1 Mbps (max.)

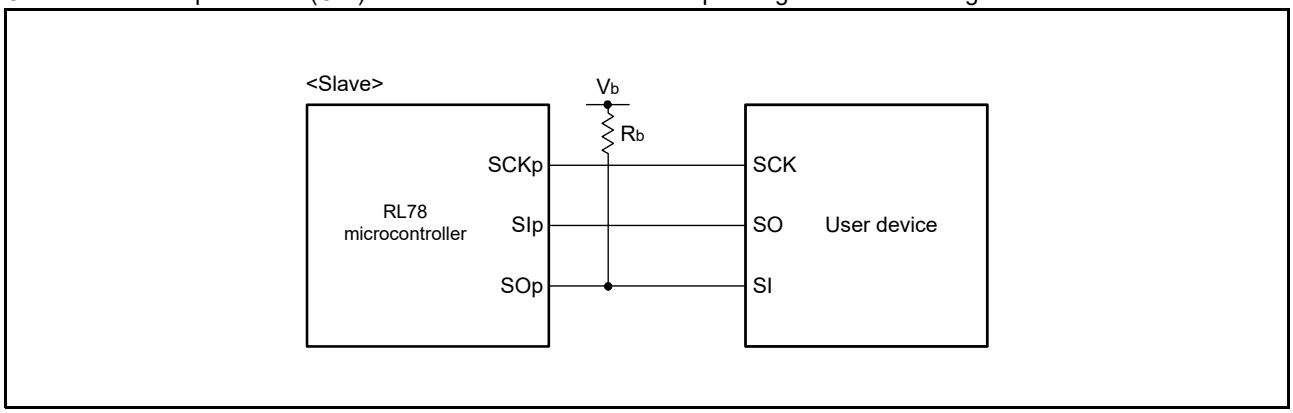
Note 2. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" and SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. This setting applies 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 and SCKp pins and the N-ch open drain output (V_{DD} withstand voltage for 20- to 52-pin products/EV_{DD} withstand voltage for 64-pin products) for the SOp pin by using the port input mode register g (PIMg) and the port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with the TTL input buffer selected.

(Remarks are listed on the next page.)

Connection in simplified SPI (CSI) communications with devices operating at different voltage levels



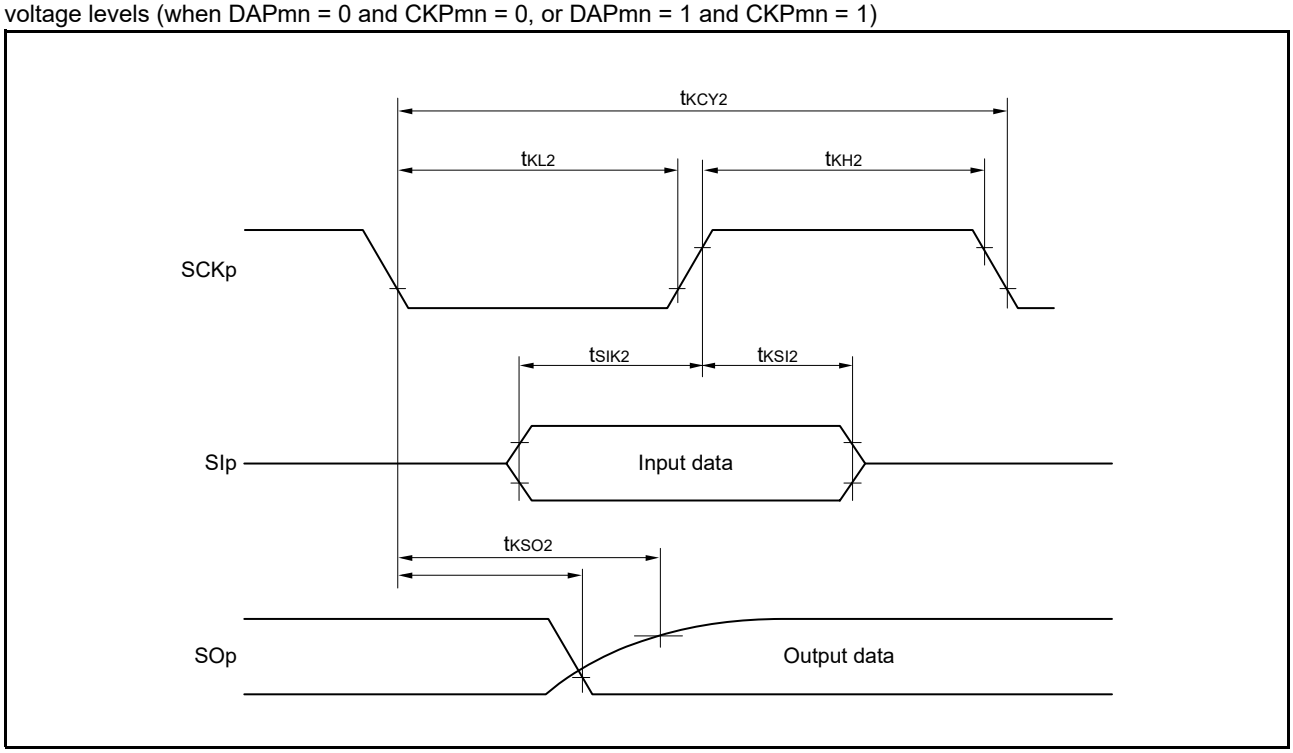
Remark 1. R_b[Ω]: Communication line (SOp) pull-up resistance, C_b[F]: Communication line (SOp) load capacitance, V_b[V]: Communication line voltage

Remark 2. p: CSI number (p = 00, 01, 10, 20), m: Unit number, n: Channel number (mn = 00, 01, 02, 10), g: PIM and POM number (g = 0, 1, 3, 5, 7)

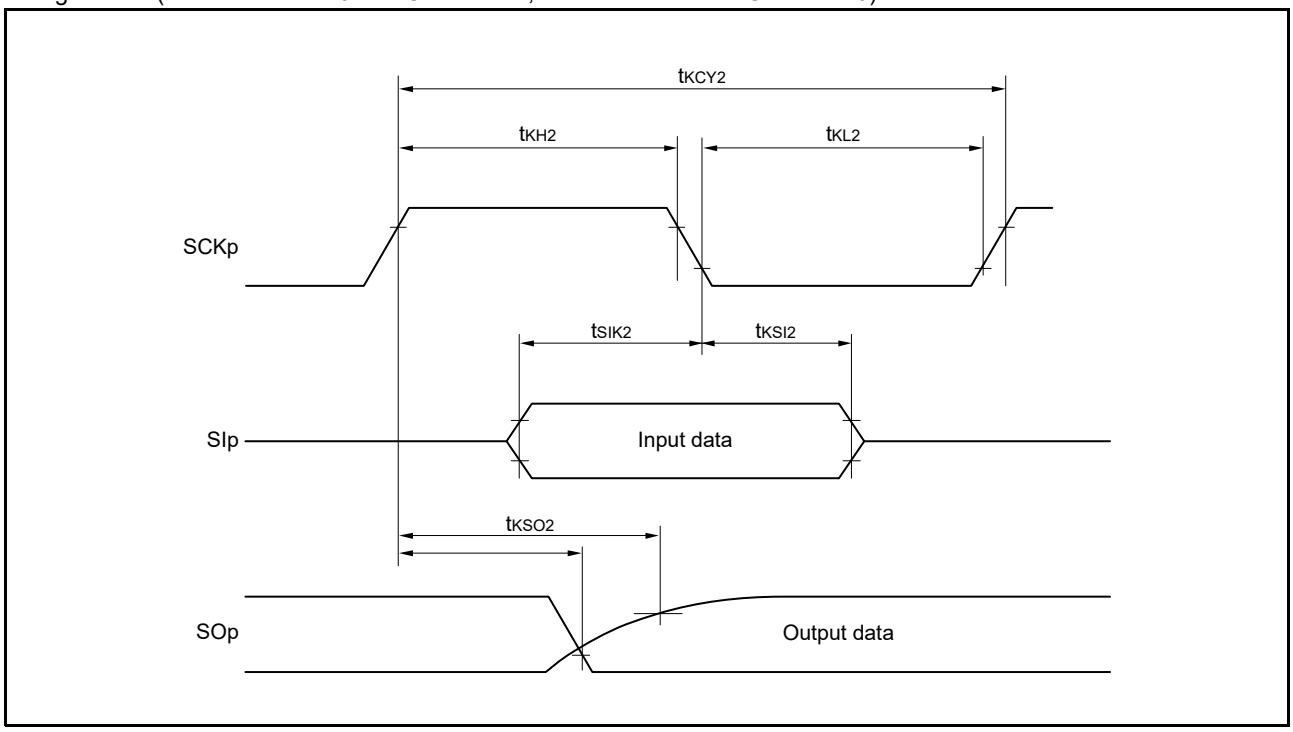
Remark 3. f_{MCK}: Serial array unit operating clock frequency
To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00, 01, 02, 10)

Remark 4. Communications by using CSI01 of 48-, 52-, and 64-pin products, and CSI11 and CSI21 with devices operating at different voltage levels are not possible. Use other CSI channels to handle such communications.

Timing of serial transfer in simplified SPI (CSI) communications in the slave mode with devices operating at different voltage levels (when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)



Timing of serial transfer in simplified SPI (CSI) communications in the slave mode with devices operating at different voltage levels (when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)



Remark 1. p: CSI number (p = 00, 01, 10, 20), m: Unit number, n: Channel number (mn = 00, 01, 02, 10), g: PIM and POM number (g = 0, 1, 3, 5, 7)

Remark 2. Communications by using CSI01 of 48-, 52-, and 64-pin products, and CSI11 and CSI21 with devices operating at different voltage levels are not possible. Use other CSI channels to handle such communications.

8. Simplified I^2C communications with devices operating at different voltage levels (2.5 V or 3 V)

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz) (1/2)

Item	Symbol	Conditions	HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
SCLr clock frequency	f_{SCL}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ		1000 **Note 1**		1000 **Note 1**		300 **Note 1**	kHz
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ		1000 **Note 1**		1000 **Note 1**		300 **Note 1**	kHz
		4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ		400 **Note 1**		400 **Note 1**		300 **Note 1**	kHz
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ		400 **Note 1**		400 **Note 1**		300 **Note 1**	kHz
Hold time when SCLr is low	t_{LOW}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	475		475		1550		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	475		475		1550		ns
		4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	1150		1550		1550		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	1150		1550		1550		ns
Hold time when SCLr is high	t_{HIGH}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	245		245		610		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	200		200		610		ns
		4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	675		675		610		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	600		600		610		ns

(Notes and Caution are listed on the next page, and Remarks are listed on page 1969.)

8. Simplified I^2C communications with devices operating at different voltage levels (2.5 V or 3 V)

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V, f_{MCK} ≤ 32 MHz) (2/2)

Item	Symbol	Conditions	HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
Data setup time (reception)	$t_{SU:DAT}$	4.0 V ≤ EV_{DD0} ≤ 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 2**		1/f_{MCK} + 135 **Note 2**		1/f_{MCK} + 190 **Note 2**		ns
		2.7 V ≤ EV_{DD0} < 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 2**		1/f_{MCK} + 135 **Note 2**		1/f_{MCK} + 190 **Note 2**		ns
		4.0 V ≤ EV_{DD0} ≤ 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 2**		1/f_{MCK} + 190 **Note 2**		1/f_{MCK} + 190 **Note 2**		ns
		2.7 V ≤ EV_{DD0} < 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 2**		1/f_{MCK} + 190 **Note 2**		1/f_{MCK} + 190 **Note 2**		ns
Data hold time (transmission)	$t_{HD:DAT}$	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	0	305	0	305	0	305	ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	0	305	0	305	0	305	ns
		4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	0	355	0	355	0	355	ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	0	355	0	355	0	355	ns

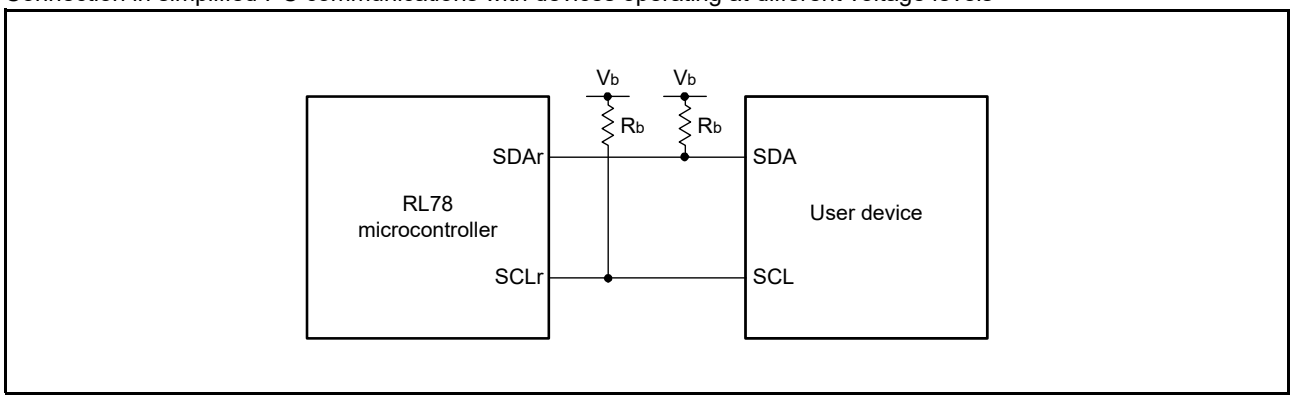
Note 1. The listed frequencies must be no greater than f_{MCK}/4.

Note 2. Set the f_{MCK} value that does not exceed the hold time when SCLr is low or high.

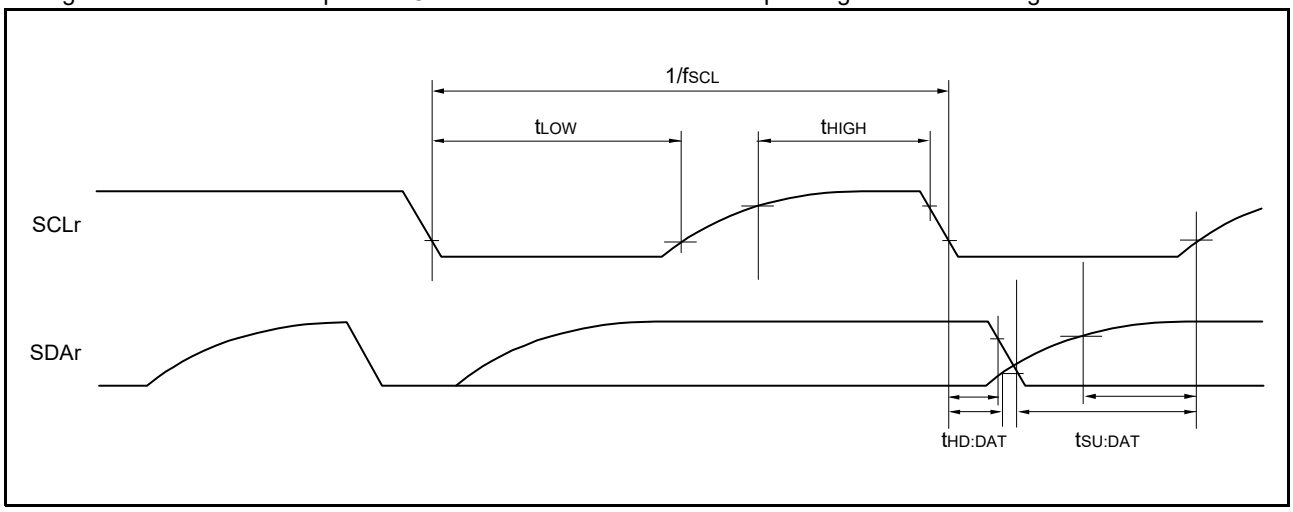
Caution Select the TTL input buffer and the N-ch open drain output (V_{DD} withstand voltage for 20- to 52-pin products/EV_{DD} withstand voltage for 64-pin products) for the SDAr pin and the N-ch open drain output (V_{DD} withstand voltage for 20- to 52-pin products/EV_{DD} withstand voltage for 64-pin products) for the SCLr pin by using the port input mode register g (PIMg) and the port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with the TTL input buffer selected.

(Remarks are listed on the next page.)

Connection in simplified I²C communications with devices operating at different voltage levels



Timing of serial transfer in simplified I²C communications with devices operating at different voltage levels



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, 01, 10, 20), g: PIM and POM number (g = 0, 1, 3, 5, 7)

Remark 3. f_{MCK}: Serial array unit operating clock frequency
To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00, 01, 02, 10)

44.5.2 Serial interface IICA

1. I^2C standard mode

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
SCLA0 clock frequency	f_{SCL}	Standard mode: f_{CLK} ≥ 1 MHz	0		100	kHz
Setup time of restart condition	$t_{SU:STA}$		4.7			μs
Hold time[Note 1]	$t_{HD:STA}$		4.0			μs
Hold time when SCLA0 is low	t_{LOW}		4.7			μs
Hold time when SCLA0 is high	t_{HIGH}		4.0			μs
Data setup time (reception)	$t_{SU:DAT}$		250			ns
Data setup time (transmission)[Note 2]	$t_{HD:DAT}$		0		3.45	μs
Setup time of stop condition	$t_{SU:STO}$		4.0			μs
Path free time	t_{BUF}		4.7			μs

Note 1. The first clock pulse is generated after this period when the start or restart condition is detected.

Note 2. The maximum value of $t_{HD:DAT}$ applies to normal transfer. The clock stretching is inserted on reception of an acknowledgment (ACK) signal.

Caution **The listed values are applicable even when bit 2 (PIOR02) in the peripheral I/O redirection register 0 (PIOR0) is 1. In such cases, the pin characteristics (I_{OH1}, I_{OL1}, V_{OH1}, V_{OL1}) must satisfy the values in the redirect destination.**

Remark The maximum value of communication line capacitance (C_b) and communication line pull-up resistor (R_b) are as follows.
C_b = 400 pF, R_b = 2.7 kΩ

2. I^2C fast mode

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
SCLA0 clock frequency	f_{SCL}	Fast mode: f_{CLK} ≥ 3.5 MHz	0		400	kHz
Setup time of restart condition	$t_{SU:STA}$		0.6			μs
Hold time[Note 1]	$t_{HD:STA}$		0.6			μs
Hold time when SCLA0 is low	t_{LOW}		1.3			μs
Hold time when SCLA0 is high	t_{HIGH}		0.6			μs
Data setup time (reception)	$t_{SU:DAT}$		100			ns
Data hold time (transmission)[Note 2]	$t_{HD:DAT}$		0		0.9	μs
Setup time of stop condition	$t_{SU:STO}$		0.6			μs
Bus-free time	t_{BUF}		1.3			μs

Note 1. The first clock pulse is generated after this period when the start or restart condition is detected.

Note 2. The maximum value of $t_{HD:DAT}$ applies to normal transfer. The clock stretching is inserted on reception of an acknowledgment (ACK) signal.

Caution **The listed values are applicable even when bit 2 (PIOR02) in the peripheral I/O redirection register 0 (PIOR0) is 1. In such cases, the pin characteristics (I_{OH1}, I_{OL1}, V_{OH1}, V_{OL1}) must satisfy the values in the redirect destination.**

Remark The maximum value of communication line capacitance (C_b) and communication line pull-up resistor (R_b) are as follows.
C_b = 320 pF, R_b = 1.1 kΩ

3. I^2C fast mode plus

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
SCLA0 clock frequency	f_{SCL}	Fast mode plus: f_{CLK} ≥ 10 MHz	0		1000	kHz
Setup time of restart condition	$t_{SU:STA}$		0.26			µs
Hold time**Note 1**	$t_{HD:STA}$		0.26			µs
Hold time when SCLA0 is low	t_{LOW}		0.5			µs
Hold time when SCLA0 is high	t_{HIGH}		0.26			µs
Data setup time (reception)	$t_{SU:DAT}$		50			ns
Data hold time (transmission)**Note 2**	$t_{HD:DAT}$		0		0.45	µs
Setup time of stop condition	$t_{SU:STO}$		0.26			µs
Bus-free time	t_{BUF}		0.5			µs

Note 1. The first clock pulse is generated after this period when the start or restart condition is detected.

Note 2. The maximum value of $t_{HD:DAT}$ applies to normal transfer. The clock stretching is inserted on reception of an acknowledgment (ACK) signal.

Caution **The listed values are applicable even when bit 2 (PIOR02) in the peripheral I/O redirection register 0 (PIOR0) is 1. In such cases, the pin characteristics (I_{OH1}, I_{OL1}, V_{OH1}, V_{OL1}) must satisfy the values in the redirect destination.**

Remark The maximum value of communication line capacitance (C_b) and communication line pull-up resistor (R_b) are as follows.
C_b = 120 pF, R_b = 1.1 kΩ

4. SMBus/PMBus™ mode (100 kHz Class)

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
SCLA0 clock frequency	f_{SCL}	f_{CLK} ≥ 1 MHz	10		100	kHz
Setup time of restart condition	$t_{SU:STA}$		4.7			µs
Hold time[Note 1]	$t_{HD:STA}$		4			µs
Hold time when SCLA0 is low	t_{LOW}		4.7			µs
Hold time when SCLA0 is high	t_{HIGH}		4			µs
Data setup time (reception)	$t_{SU:DAT}$		250			ns
Data hold time (transmission)[Note 2]	$t_{HD:DAT}$		0		3.45	µs
Setup time of stop condition	$t_{SU:STO}$		4			µs
Clock/data falling time	t_F				0.3	µs
Clock/data rising time	t_R				1	µs
Bus-free time	t_{BUF}		4.7			µs

Note 1. The first clock pulse is generated after this period when the start or restart condition is detected.

Note 2. The maximum value of $t_{HD:DAT}$ applies to normal transfer. The clock stretching is inserted on reception of an acknowledgment (ACK) signal.

Caution **SMBus/PMBus™ communications are disabled when bit 2 (PIOR02) in the peripheral I/O redirection register 0 (PIOR0) is 1.**

Remark The maximum value of communication line pull-up resistor (R_b) is as follows.
R_b = 1.1 kΩ

5. SMBus/PMBus™ mode (400 kHz Class)

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
SCLA0 clock frequency	f_{SCL}	f_{CLK} ≥ 3.5 MHz	10		400	kHz
Setup time of restart condition	$t_{SU:STA}$		0.6			µs
Hold time[Note 1]	$t_{HD:STA}$		0.6			µs
Hold time when SCLA0 is low	t_{LOW}		1.3			µs
Hold time when SCLA0 is high	t_{HIGH}		0.6			µs
Data setup time (reception)	$t_{SU:DAT}$		100			ns
Data hold time (transmission)[Note 2]	$t_{HD:DAT}$		0		0.9	µs
Setup time of stop condition	$t_{SU:STO}$		0.6			µs
Clock/data falling time	t_F				0.3	µs
Clock/data rising time	t_R				0.3	µs
Bus-free time	t_{BUF}		1.3			µs

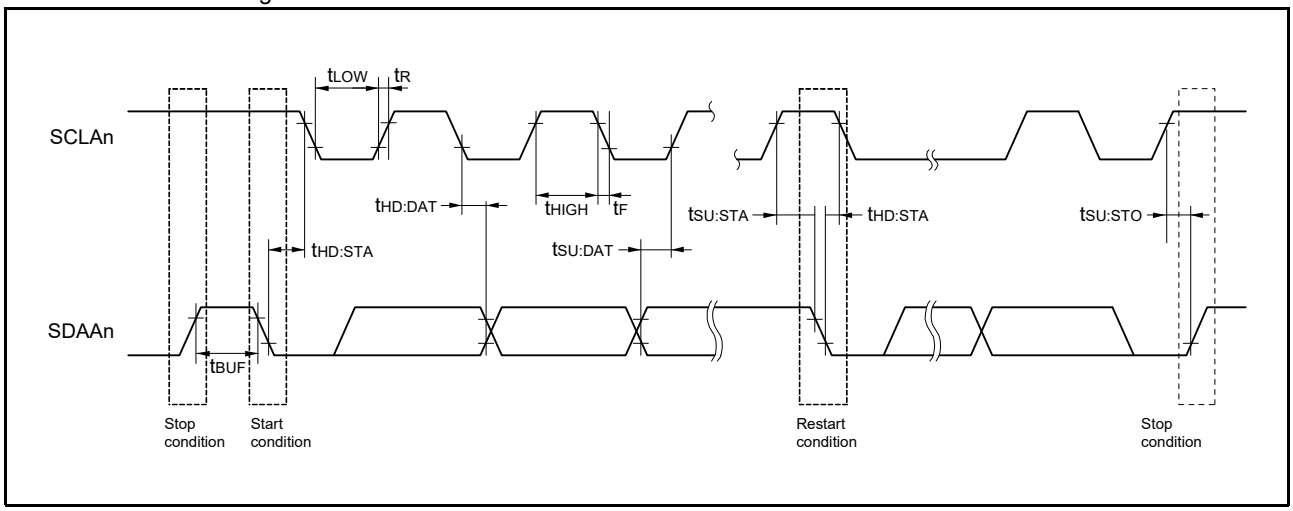
Note 1. The first clock pulse is generated after this period when the start or restart condition is detected.

Note 2. The maximum value of $t_{HD:DAT}$ applies to normal transfer. The clock stretching is inserted on reception of an acknowledgment (ACK) signal.

Caution **SMBus/PMBus™ communications are disabled when bit 2 (PIOR02) in the peripheral I/O redirection register 0 (PIOR0) is 1.**

Remark The maximum value of communication line capacitance (C_b) and communication line pull-up resistor (R_b) are as follows.
C_b = 400 pF, R_b = 1.1 kΩ

IICA serial transfer timing



Remark n = 0

44.6 Analog Characteristics

44.6.1 A/D converter characteristics

1. Normal modes 1 and 2

(T_A = –40 to +125°C, 2.7 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, f_{CLK} ≤ 32 MHz,
reference voltage (+) = AV_{REFP} (ADREFP[1:0] = 01B), reference voltage (–) = AV_{REFM} (ADREFM = 1),
target pins: ANI2 to ANI7, ANI16 to ANI30, internal reference voltage, and temperature sensor output voltage)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Resolution	RES		8		12	Bit
Conversion clock	f_{AD}		1		32	MHz
Overall error **Notes 1, 3, 4, 5**	AINL	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±7.5	LSB
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±9.0	LSB
Conversion time **Note 6**	t_{CONV}	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	2			µs
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	2			µs
Zero-scale error **Notes 1, 2, 3, 4, 5**	E_{ZS}	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.17	%FSR
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
Full-scale error **Notes 1, 2, 3, 4, 5**	E_{FS}	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.17	%FSR
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
Integral linearity error **Notes 1, 4, 5**	ILE	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±3.0	LSB
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±3.0	LSB
Differential linearity error **Note 1**	DLE	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±1.0		LSB
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±1.0		LSB
Analog input voltage	V_{AIN}		0		AV_{REFP}	V

Note 1. This value does not include the quantization error (±1/2 LSB).

Note 2. This value is indicated as a ratio (%FSR) to the full-scale value.

Note 3. When pins ANI16 to ANI30 are selected as the conversion target, the maximum values are as follows.
Overall error: Add ±3 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.04%FSR to the maximum value.

Note 4. The maximum values are as follows when V_{DD} is selected for reference voltage (+) and V_{SS} is selected for reference voltage (–).
Overall error: Add ±10 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.25%FSR to the maximum value.
Integral linearity error: Add ±4 LSB to the maximum value.

Note 5. When AV_{REFP} < V_{DD}, the maximum values are as follows.
Overall error/zero-scale/full-scale error: Add ±0.75 LSB × (V_{DD} voltage (V) – AV_{REFP} voltage (V)) to the maximum value.
Integral linearity error: Add ±0.2 LSB × (V_{DD} voltage (V) – AV_{REFP} voltage (V)) to the maximum value.

Note 6. When the internal reference voltage or the temperature sensor output voltage is selected as the conversion target, the sampling time must be at least 5 µs. Accordingly, use normal mode 2 with the longer sampling time.

2. Normal modes 1 and 2 (advanced mode)

(T_A = –40 to +125°C, 2.7 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, f_{CLK} ≤ 48 MHz,
reference voltage (+) = AV_{REFP} (ADREFP[1:0] = 01B), reference voltage (–) = AV_{REFM} (ADREFM = 1),
target pins: ANI2 to ANI7, ANI16 to ANI30, internal reference voltage, and temperature sensor output voltage)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Resolution	RES		8		12	Bit
Conversion clock	f_{AD}		1		48	MHz
Overall error **Notes 1, 3, 4, 5, 6, 7**	AINL	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±7.5	LSB
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±9.0	LSB
Conversion time **Notes 7, 8**	t_{CONV}	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	1			μs
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	1			μs
Zero-scale error **Notes 1, 2, 3, 4, 5, 6, 7**	E_{ZS}	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.17	%FSR
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
Full-scale error **Notes 1, 2, 3, 4, 5, 6, 7**	E_{FS}	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.17	%FSR
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
Integral linearity error **Notes 1, 4, 5**	ILE	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±3.0	LSB
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±3.0	LSB
Differential linearity error **Note 1**	DLE	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±1.0		LSB
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±1.0		LSB
Analog input voltage	V_{AIN}		0		AV_{REFP}	V

Note 1. This value does not include the quantization error (±1/2 LSB).

Note 2. This value is indicated as a ratio (%FSR) to the full-scale value.

Note 3. When pins ANI16 to ANI30 are selected as the conversion target, the maximum values are as follows.
Overall error: Add ±3 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.04%FSR to the maximum value.

Note 4. The maximum values are as follows when V_{DD} is selected for reference voltage (+) and V_{SS} is selected for reference voltage (–).
Overall error: Add ±10 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.25%FSR to the maximum value.
Integral linearity error: Add ±4 LSB to the maximum value.

Note 5. When AV_{REFP} < V_{DD}, the maximum values are as follows.
Overall error/zero-scale/full-scale error: Add ±0.75 LSB × (V_{DD} voltage (V) – AV_{REFP} voltage (V)) to the maximum value.
Integral linearity error: Add ±0.2 LSB × (V_{DD} voltage (V) – AV_{REFP} voltage (V)) to the maximum value.

Note 6. When the sample & hold circuit is selected as the conversion target, the maximum values are as follows.
Overall error: Add ±1 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.03%FSR to the maximum value.

Note 7. Add the following values to the listed values in the cases below.
- 7 f_{AD} when the conversion target includes low-speed conversion ANI (ANI16 to ANI30)
- 12 f_{AD} when the conversion target includes the PGA with the gain of ×4 to ×16.
- 43 f_{AD} when the conversion target includes the PGA with the gain of ×32.

Note 8. When the internal reference voltage or the temperature sensor output voltage is selected as the conversion target, the sampling time must be at least 5 μs. Accordingly, use normal mode 2 with the longer sampling time.

3. Low-voltage modes 1 and 2

(T_A = –40 to +125°C, 2.7 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, f_{CLK} ≤ 32 MHz,
reference voltage (+) = AV_{REFP} (ADREFP[1:0] = 01B), reference voltage (–) = AV_{REFM} (ADREFM = 1),
target pins: ANI2 to ANI7, ANI16 to ANI30, PGA, S&H, internal reference voltage, and temperature sensor output voltage)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Resolution	RES		8		12	Bit
Conversion clock	f_{AD}		1		24	MHz
Overall error **Notes 1, 3, 4, 5**	AINL	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±9.0	LSB
Conversion time **Note 6**	t_{CONV}	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	3.33			μs
Zero-scale error **Notes 1, 2, 3, 4, 5**	E_{ZS}	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
Full-scale error **Notes 1, 2, 3, 4, 5**	E_{FS}	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
Integral linearity error **Notes 1, 4, 5**	ILE	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±4.0	LSB
Differential linearity error **Note 1**	DLE	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±1.5		LSB
Analog input voltage	V_{AIN}		0		AV_{REFP}	V

Note 1. This value does not include the quantization error (±1/2 LSB).

Note 2. This value is indicated as a ratio (%FSR) to the full-scale value.

Note 3. When pins ANI16 to ANI30 are selected as the conversion target, the maximum values are as follows.
Overall error: Add ±3 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.04%FSR to the maximum value.

Note 4. The maximum values are as follows when V_{DD} is selected for reference voltage (+) and V_{SS} is selected for reference voltage (–).
Overall error: Add ±10 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.25%FSR to the maximum value.
Integral linearity error: Add ±4 LSB to the maximum value.

Note 5. When AV_{REFP} < V_{DD}, the maximum values are as follows.
Overall error/zero-scale/full-scale error: Add ±0.75 LSB × (V_{DD} voltage (V) – AV_{REFP} voltage (V)) to the maximum value.
Integral linearity error: Add ±0.2 LSB × (V_{DD} voltage (V) – AV_{REFP} voltage (V)) to the maximum value.

Note 6. When the internal reference voltage or the temperature sensor output voltage is selected as the conversion target, the sampling time must be at least 5 μs. Accordingly, use low-voltage mode 2 with the longer sampling time and the conversion clock (f_{AD}) with a frequency of no more than 16 MHz.

4. Low-voltage modes 1 and 2 (advanced mode)

(T_A = –40 to +125°C, 2.7 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, f_{CLK} ≤ 48 MHz,
reference voltage (+) = AV_{REFP} (ADREFP[1:0] = 01B), reference voltage (–) = AV_{REFM} (ADREFM = 1),
target pins: ANI2 to ANI7, ANI16 to ANI30, PGA, S&H, internal reference voltage, and temperature sensor output voltage)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Resolution	RES		8		12	Bit
Conversion clock	f_{AD}		1		24	MHz
Overall error **Notes 1, 3, 4, 5, 6**	AINL	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±9.0	LSB
Conversion time **Note 7**	t_{CONV}	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	3.63			μs
Zero-scale error **Notes 1, 2, 3, 4, 5, 6**	E_{ZS}	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
Full-scale error **Notes 1, 2, 3, 4, 5, 6**	E_{FS}	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
Integral linearity error **Notes 1, 4, 5**	ILE	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±4.0	LSB
Differential linearity error **Note 1**	DLE	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±1.5		LSB
Analog input voltage	V_{AIN}		0		AV_{REFP}	V

Note 1. This value does not include the quantization error (±1/2 LSB).

Note 2. This value is indicated as a ratio (%FSR) to the full-scale value.

Note 3. When pins ANI16 to ANI30 are selected as the conversion target, the maximum values are as follows.
Overall error: Add ±3 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.04%FSR to the maximum value.

Note 4. The maximum values are as follows when V_{DD} is selected for reference voltage (+) and V_{SS} is selected for reference voltage (–).
Overall error: Add±10 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.25%FSR to the maximum value.
Integral linearity error: Add ±4 LSB to the maximum value.

Note 5. When AV_{REFP} < V_{DD}, the maximum values are as follows.
Overall error/zero-scale/full-scale error: Add ±0.75 LSB × (V_{DD} voltage (V) – AV_{REFP} voltage (V)) to the maximum value.
Integral linearity error: Add ±0.2 LSB × (V_{DD} voltage (V) – AV_{REFP} voltage (V)) to the maximum value.

Note 6. When the sample & hold circuit is selected as the conversion target, the maximum values are as follows.
Overall error: Add ±1 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.03%FSR to the maximum value.

Note 7. When the internal reference voltage or the temperature sensor output voltage is selected as the conversion target, the sampling time must be at least 5 μs. Accordingly, use low-voltage mode 2 with the longer sampling time and the conversion clock (f_{AD}) with a frequency of no more than 16 MHz.

5. When the internal reference voltage is selected as reference voltage (+)

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, low-voltage modes 1 and 2, f_{CLK} ≤ 32 MHz**Note 1**, f_{CLK} ≤ 48 MHz**Note 2**, reference voltage (+) = internal reference voltage (ADREFP[1:0] = 10B), reference voltage (–) = AV_{REFM} (ADREFM = 1))

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Resolution	RES		8			BIt
Conversion clock	f_{AD}		1		2	MHz
Zero-scale error**Notes 3, 4, 6**	E_{ZS}				±0.6	%FSR
Integral linearity error**Notes 3, 6**	ILE				±2.0	LSB
Differential linearity error**Note 3**	DLE			±1.0		LSB
Analog input voltage	V_{AIN}		0		V_{BGR} **Note 5**	V

Note 1. This applies when the advanced mode is disabled.

Note 2. This applies when the advanced mode is enabled.

Note 3. This value does not include the quantization error (±1/2 LSB).

Note 4. This value is indicated as a ratio (%FSR) to the full-scale value.

Note 5. Refer to **44.6.2 Temperature sensor/internal reference voltage characteristics**.

Note 6. When reference voltage (–) is selected as V_{SS}, the maximum values are as follows.
Zero-scale error: Add ±0.35%FSR to the maximum value.
Integral linearity error: Add ±0.5 LSB to the maximum value.

44.6.2 Temperature sensor/internal reference voltage characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Temperature sensor output voltage	V_{TMPS25}	ADS register is set to 80H, T_A = +25°C		1.05		V
Internal reference voltage	V_{BGR}	ADS register is set to 81H	1.40	1.48	1.56	V
Temperature coefficient	F_{VTMPS}	Temperature dependency of the temperature sensor voltage		–3.3		mV/°C
Operation stabilization wait time	t_{AMP}		5			μs

44.6.3 D/A converter characteristics

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Resolution	RES	DAC0, DAC1 (DACONF = 0)			10	Bit
		DAC1 (DACONF = 1), DAC2			8	Bit
Overall error	AINL	Rload = 8 MΩ			±2.5	LSB
Differential non-linearity error	ADNL				±1.0	LSB
Settling time	t_{SET}	Cload = 20 pF when DACO is output			6	μs
		During full code conversion using CMP reference			3	μs
		During 1LSB code conversion using CMP reference			1	μs

Caution The voltage on the ANO0 to ANO2 pins must not exceed EV_{DD0}.

44.6.4 Comparator characteristics

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Input voltage range	I_{VREF}	IVREF0 pin, IVREF1 pin input	0		EV_{DD0}	V
	I_{VCMP}	IVCMP0, IVCMP1, IVCMP2, ICMP3 pin input	0		EV_{DD0}	V
Output delay	td	Input amplitude ±100 mV		50	100	ns
Offset voltage	—			±5	±40	mV
Operation stabilization time**Note**	t_{CMP}		1			μs
Input channel switching stabilization wait time	—		0.3			μs

Note The listed values indicate the time until the DC/AC characteristics of the comparator are satisfied following enabling of the comparator operation (CnENB = 1).

44.6.5 PGA characteristics

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V)

Item	Symbol	Conditions			Min.	Typ.	Max.	Unit
Input offset voltage	V_{IOPGA}						±10	mV
Input voltage range[Note 1]	V_{IPGA}				0		0.9 × V_{DD}/ amplification rate	V
Amplification rate error		×4, ×8					±1	%
		×16					±1.5	%
		×32					±2	%
Slew rate[Note 1]	S_{RRPGA}	Rising Vin = V_{DD} × 0.1/ amplification rate to V_{DD} × 0.9/ amplification rate 10 to 90% of output amplitude	4.0 V ≤ V_{DD} ≤ 5.5 V	Other than ×32	3.5			V/µs
			4.0 V ≤ V_{DD} ≤ 5.5 V	×32	3			
			2.7 V ≤ V_{DD} ≤ 4.0 V		0.5			
	S_{FPGA}	Falling Vin = V_{DD} × 0.1/ amplification rate to V_{DD} × 0.9/ amplification rate 90 to 10% of output amplitude	4.0 V ≤ V_{DD} ≤ 5.5 V	Other than ×32	3.5			
			4.0 V ≤ V_{DD} ≤ 5.5 V	×32	3			
			2.7 V ≤ V_{DD} ≤ 4.0 V		0.5			
Operation stabilization wait time[Note 2]	t_{PGA}	×4, ×8					5	µs
		×16, ×32					10	µs

Note 1. A voltage of EV_{DD0} is supplied to the PGAI0 to PGAI3 pins.

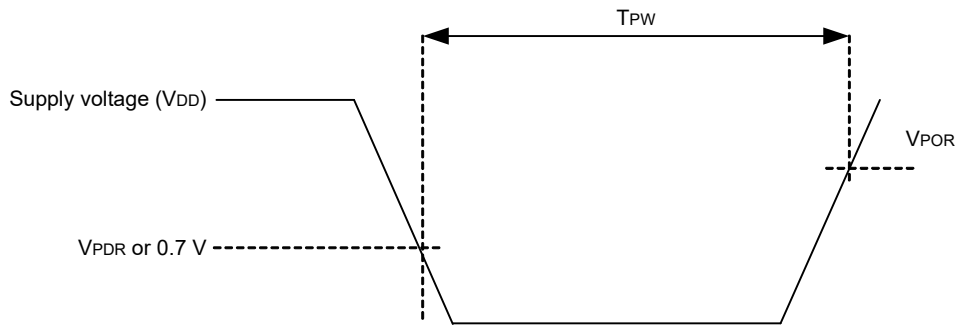
Note 2. The listed values indicate the time until the DC/AC characteristics of PGA operation are satisfied following enabling of the PGA operation (PGAEN = 1).

44.6.6 POR circuit characteristics

(T_A = –40 to +125°C, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Detection voltage	V_{POR}, V_{PDR}		1.43	1.50	1.57	V
Minimum pulse width[Note]	T_{PW}		300			μs

Note This width is the minimum time required for a POR reset when V_{DD} falls below V_{PDR}. This width is also the minimum time required for a POR reset from when V_{DD} falls below 0.7 V to when V_{DD} exceeds V_{POR} in the STOP mode or while the main system clock is stopped through setting bit 0 (HIOSTOP) and bit 7 (MSTOP) in the clock operation status control register (CSC).



44.6.7 LVD circuit characteristics

1. LVD detection voltage in the LVD0 reset mode and interrupt mode

(T_A = –40 to +125°C, $V_{PDR} \leq V_{DD} \leq 5.5$ V, V_{SS} = 0 V)

Item		Symbol	Conditions	Min.	Typ.	Max.	Unit
Detection voltage	Supply voltage level	V_{LVD00}	The power supply voltage is rising.	3.84	3.96	4.08	V
			The power supply voltage is falling.	3.76	3.88	4.00	V
		V_{LVD01}	The power supply voltage is rising.	2.88	2.97	3.06	V
			The power supply voltage is falling.	2.82	2.91	3.00	V
Minimum pulse width		t_{LW}		500			μs
Detection delay time						500	μs

2. LVD detection voltage in the LVD1 reset mode and interrupt mode

(T_A = –40 to +125°C, $V_{PDR} \leq V_{DD} \leq 5.5$ V, V_{SS} = 0 V)

Item		Symbol	Conditions	Min.	Typ.	Max.	Unit
Detection voltage	Supply voltage level	V_{LVD10}	The power supply voltage is rising.	4.08	4.16	4.24	V
			The power supply voltage is falling.	4.00	4.08	4.16	V
		V_{LVD11}	The power supply voltage is rising.	3.88	3.96	4.04	V
			The power supply voltage is falling.	3.80	3.88	3.96	V
		V_{LVD12}	The power supply voltage is rising.	3.68	3.75	3.82	V
			The power supply voltage is falling.	3.60	3.67	3.74	V
		V_{LVD13}	The power supply voltage is rising.	3.48	3.55	3.62	V
			The power supply voltage is falling.	3.40	3.47	3.54	V
		V_{LVD14}	The power supply voltage is rising.	3.28	3.35	3.42	V
			The power supply voltage is falling.	3.20	3.27	3.34	V
		V_{LVD15}	The power supply voltage is rising.	3.07	3.13	3.19	V
			The power supply voltage is falling.	3.00	3.06	3.12	V
		V_{LVD16}	The power supply voltage is rising.	2.91	2.97	3.03	V
			The power supply voltage is falling.	2.85	2.91	2.97	V
		V_{LVD17}	The power supply voltage is rising.	2.76	2.82	2.87	V
			The power supply voltage is falling.	2.70	2.76	2.81	V
Minimum pulse width		t_{LW}		500			μs
Detection delay						500	μs

44.6.8 Power supply voltage rising slope characteristics

(T_A = –40 to +125°C, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Power voltage rising slope	S_{VDD}				54	V/ms

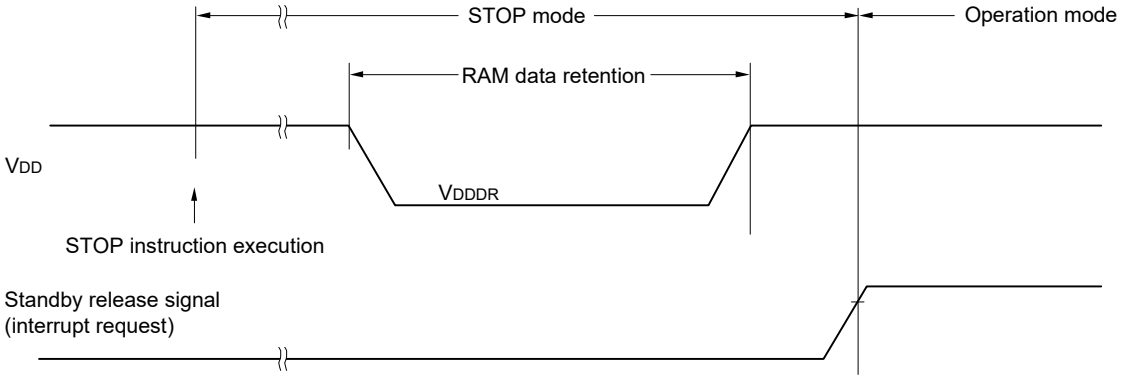
Caution Make sure to keep the internal reset state by the LVD0 circuit or an external reset until V_{DD} reaches the operating voltage range shown in AC characteristics.

44.7 RAM Data Retention Characteristics

(T_A = –40 to +125°C, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Data retention power voltage	V_{DDDR}		1.43[Note]		5.5	V

Note This voltage depends on the POR detection voltage. When the voltage drops, the data in RAM are retained until a POR is applied, but are not retained following a POR.



44.8 Flash Memory Programming Characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
CPU/peripheral hardware clock frequency	f_{CLK}		1		48	MHz
Number of code flash rewrites[Notes 1, 2, 3]	C_{erwr}	Retained for 10 years T_A = 85°C	10,000			Times
		Retained for 20 years T_A = 85°C	1,000			
Number of data flash rewrites[Notes 1, 2, 3]		Retained for 1 year T_A = 25°C		1,000,000		
		Retained for 5 years T_A = 85°C	100,000			
		Retained for 20 years T_A = 85°C	10,000			

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. The listed numbers of times apply when using the flash memory programmer and the Renesas Electronics self-programming library.

Note 3. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

1. Code flash memory

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item		Symbol	f_{CLK} = 1 MHz			f_{CLK} = 2 MHz, 3 MHz			4 MHz ≤ f_{CLK} < 8 MHz			8 MHz ≤ f_{CLK} < 32 MHz			f_{CLK} = 32 MHz			f_{CLK} = 48 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	4 bytes	t_{P4}	—	75.8	666.6	—	51.5	469.7	—	41.9	387.3	—	37.2	347.4	—	34.2	322.3	—	33.9	319.7	µs
Erasure time	2 Kbytes	t_{E2K}	—	10.4	312.2	—	7.7	258.5	—	6.4	231.8	—	5.8	218.4	—	5.6	214.4	—	5.6	213.9	ms
Blank checking time	4 bytes	t_{BC4}	—	—	38.4	—	—	19.2	—	—	13.1	—	—	10.2	—	—	8.3	—	—	8.1	µs
	2 Kbytes	t_{BC2K}	—	—	2618.9	—	—	1309.5	—	—	658.3	—	—	332.8	—	—	234.1	—	—	223.19	µs
Time taken to forcibly stop erasure		t_{SED}	—	—	19.0	—	—	14.5	—	—	12.3	—	—	11.1	—	—	10.4	—	—	10.3	µs
Security setting time		t_{AWSSAS}	—	18.2	526.4	—	14.4	469.3	—	12.6	441.1	—	11.6	427.1	—	11.3	422.6	—	11.3	422.1	ms
Time until programming starts following cancellation of the STOP instruction		—	20	—	—	20	—	—	20	—	—	20	—	—	20	—	—	20	—	—	µs

Caution The listed values do not include the time until the operations of the flash memory start following execution of an instruction by software.

2. Data flash memory

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item		Symbol	f_{CLK} = 1 MHz			f_{CLK} = 2 MHz, 3 MHz			4 MHz ≤ f_{CLK} < 8 MHz			8 MHz ≤ f_{CLK} < 32 MHz			f_{CLK} = 32 MHz			f_{CLK} = 48 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	1 byte	t_{P4}	—	75.8	666.6	—	51.51	469.7	—	41.9	387.34	—	37.24	347.4	—	34.2	322.3	—	33.92	319.7	µs
Erasure time	256 bytes	t_{E2K}	—	7.8	259.2	—	6.4	232.0	—	5.8	218.5	—	5.5	211.8	—	5.4	209.7	—	5.3	209.5	ms
Blank checking time	1 byte	t_{BC4}	—	—	38.4	—	—	19.2	—	—	13.1	—	—	10.2	—	—	8.3	—	—	8.1	µs
	256 bytes	t_{BC2K}	—	—	1326.1	—	—	663.1	—	—	335.1	—	—	171.2	—	—	121.0	—	—	115.5	µs
Time taken to forcibly stop erasure		t_{SED}	—	—	19.0	—	—	14.5	—	—	12.3	—	—	11.1	—	—	10.4	—	—	10.3	µs
Time until programming starts following cancellation of the STOP instruction		—	20	—	—	20	—	—	20	—	—	20	—	—	20	—	—	20	—	—	µs
Time until reading starts following setting DFLEN to 1		—	0.25	—	—	0.25	—	—	0.25	—	—	0.25	—	—	0.25	—	—	0.25	—	—	ns

Caution The listed values do not include the time until the operations of the flash memory start following execution of an instruction by software.

44.9 Dedicated Flash Memory Programmer Communication (UART)

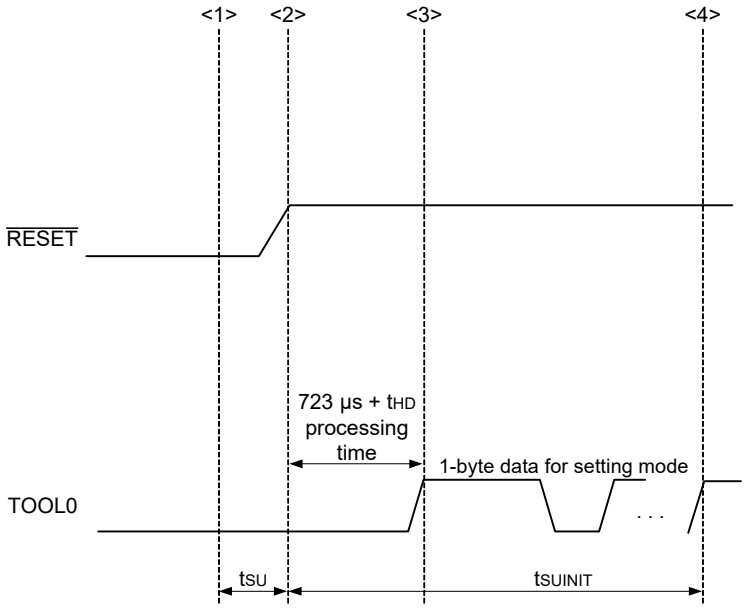
(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Transfer rate		During serial programming	115,200		1,000,000	bps

44.10 Timing of Entry to Flash Memory Programming Modes

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Time to complete the communication for the initial setting after the external reset is released	t_{SUINIT}	POR and LVD reset must be released before the external reset is released			100	ms
Time to release the external reset after the TOOL0 pin is set to the low level	t_{SU}	POR and LVD reset must be released before the external reset is released	10			μs
Time to hold the TOOL0 pin at the low level after the external reset is released (the processing time of the firmware to control the flash memory is not included)	t_{HD}	POR and LVD reset must be released before the external reset is released	1			ms



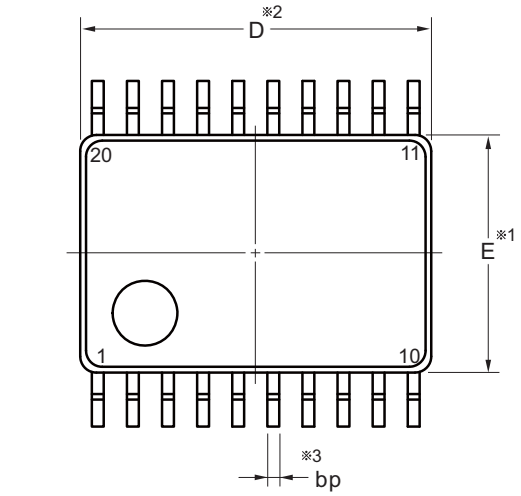
<1> The low level is input to the TOOL0 pin.
<2> The external reset is released. Note that the POR and LVD reset must be released before the external reset is released.
<3> The TOOL0 pin is set to the high level.
<4> The baud rate setting is complete upon UART reception.

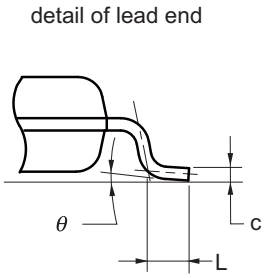
Remark t_{SUINIT}: The time during which the communications for the initial setting must be completed within 100 ms after the external reset is released.
t_{SU}: Time to release the external reset after the TOOL0 pin is set to the low level
t_{HD}: Time to hold the TOOL0 pin at the low level after the external reset is released. It does not include the processing time of the firmware to control the flash memory.

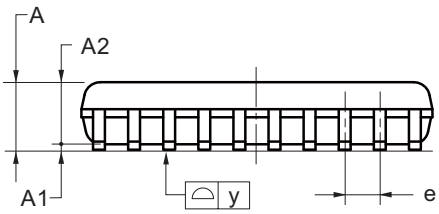
Section 45 Package Drawings

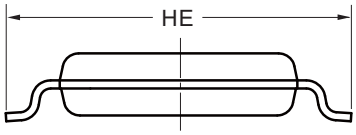
45.1 20-pin Products

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LSSOP20-4.4x6.5-0.65	PLSP0020JB-A	P20MA-65-NAA-1	0.1









NOTE

1.Dimensions "※1" and "※2" do not include mold flash.

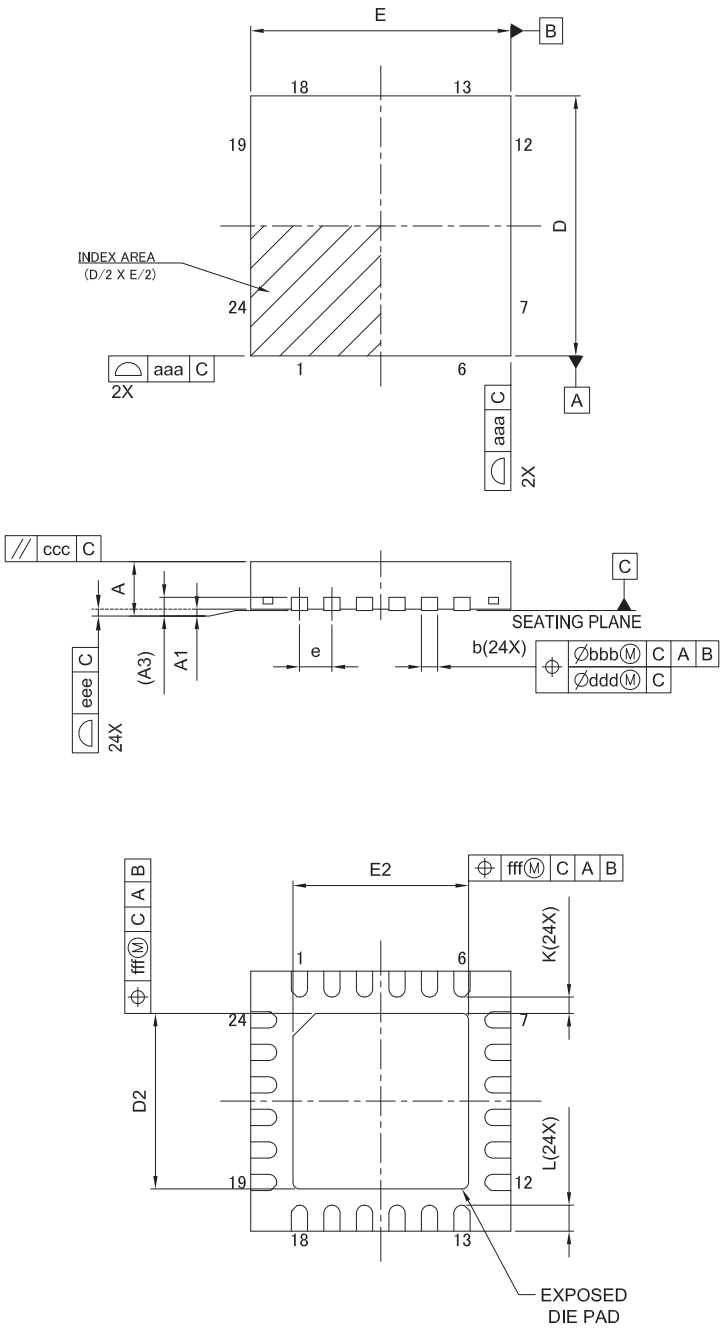
2.Dimension "※3" does not include trim offset.

(UNIT:mm)

ITEM	DIMENSIONS
D	6.50±0.10
E	4.40±0.10
HE	6.40±0.20
A	1.45 MAX.
A1	0.10±0.10
A2	1.15
e	0.65±0.12
bp	$0.22^{+0.10}_{-0.05}$
c	$0.15^{+0.05}_{-0.02}$
L	0.50±0.20
y	0.10
θ	0° to 10°

45.2 24-pin Products

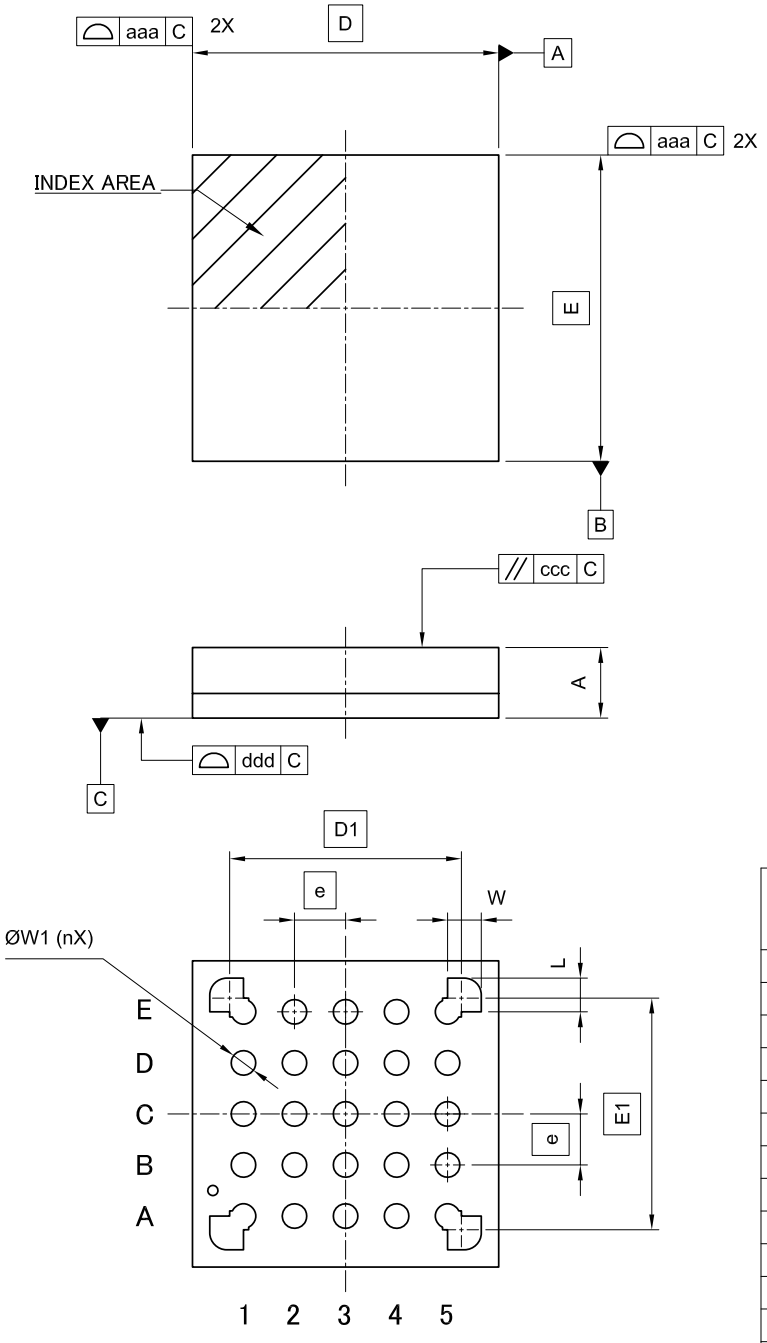
JEITA Package Code	RENESAS Code	MASS(Typ.)[g]
P-HWFQFN24-4×4-0.50	PWQN0024KG-A	0.04



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	0.02	0.05
A_3		0.203 REF.	
b	0.18	0.25	0.30
D		4.00 BSC	
E		4.00 BSC	
e		0.50 BSC	
L	0.35	0.40	0.45
K	0.20	—	—
D_2	2.65	2.70	2.75
E_2	2.65	2.70	2.75
aaa		0.15	
bbb		0.10	
ccc		0.10	
ddd		0.05	
eee		0.08	
fff		0.10	

45.3 25-pin Products

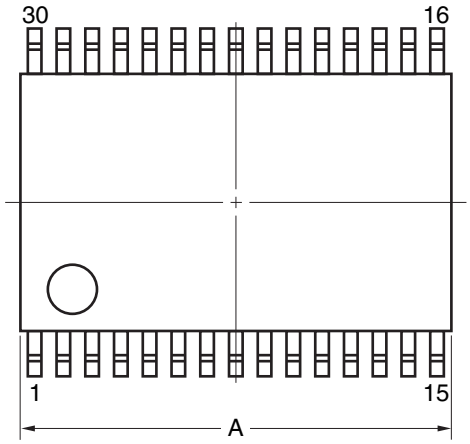
JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-WLGA25-3x3-0.50	PWLG0025KB-A	0.01



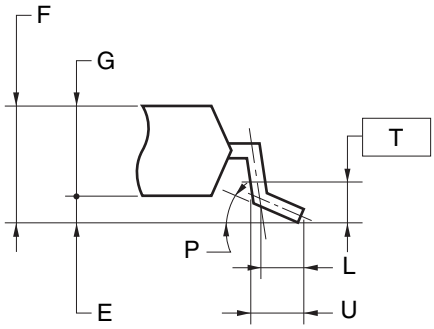
Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	—	3.00	—
E	—	3.00	—
D1		2.27	
E1		2.27	
A	—	—	0.76
W1	0.19	0.24	0.29
W	—	0.330	—
L	—	0.330	—
e		0.50	
aaa	—	—	0.10
ccc	—	—	0.20
ddd	—	—	0.08
n	—	25	—

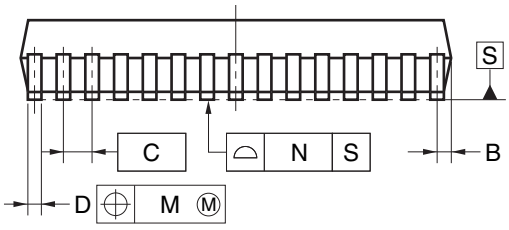
45.4 30-pin Products

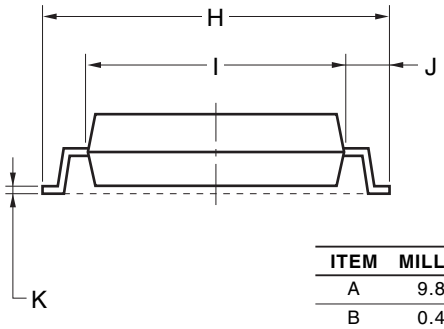
JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LSSOP30-0300-0.65	PLSP0030JB-B	S30MC-65-5A4-3	0.18



detail of lead end





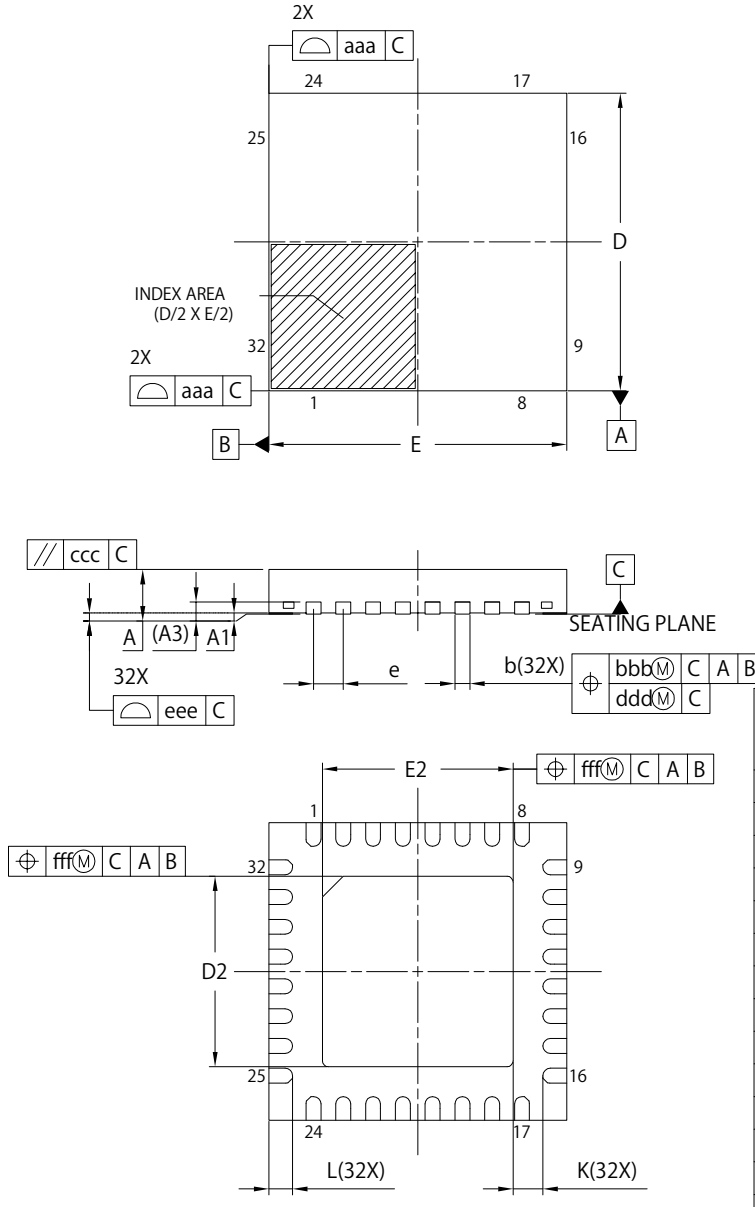


NOTE
Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

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	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
T	0.25
U	0.6±0.15

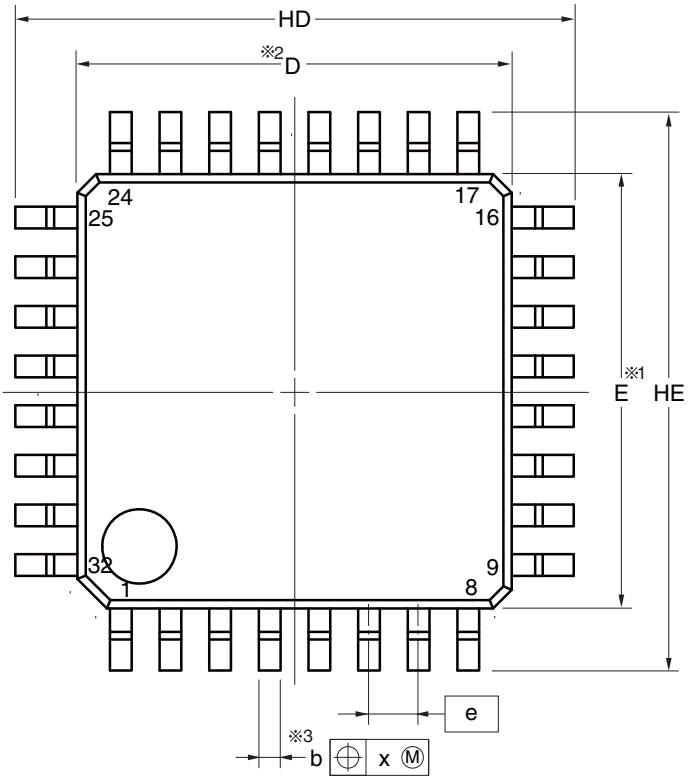
45.5 32-pin Products

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN032-5x5-0.50	PWQN0032KE-A	0.06

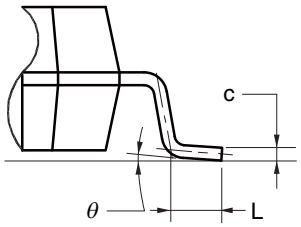


Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	0.02	0.05
A_3		0.203 REF.	
b	0.18	0.25	0.30
D		5.00 BSC	
E		5.00 BSC	
e		0.50 BSC	
L	0.35	0.40	0.45
K	0.20	—	—
D_2	3.15	3.20	3.25
E_2	3.15	3.20	3.25
aaa		0.15	
bbb		0.10	
ccc		0.10	
ddd		0.05	
eee		0.08	
fff		0.10	

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP32-7x7-0.80	PLQP0032GB-A	P32GA-80-GBT-1	0.2



detail of lead end



(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

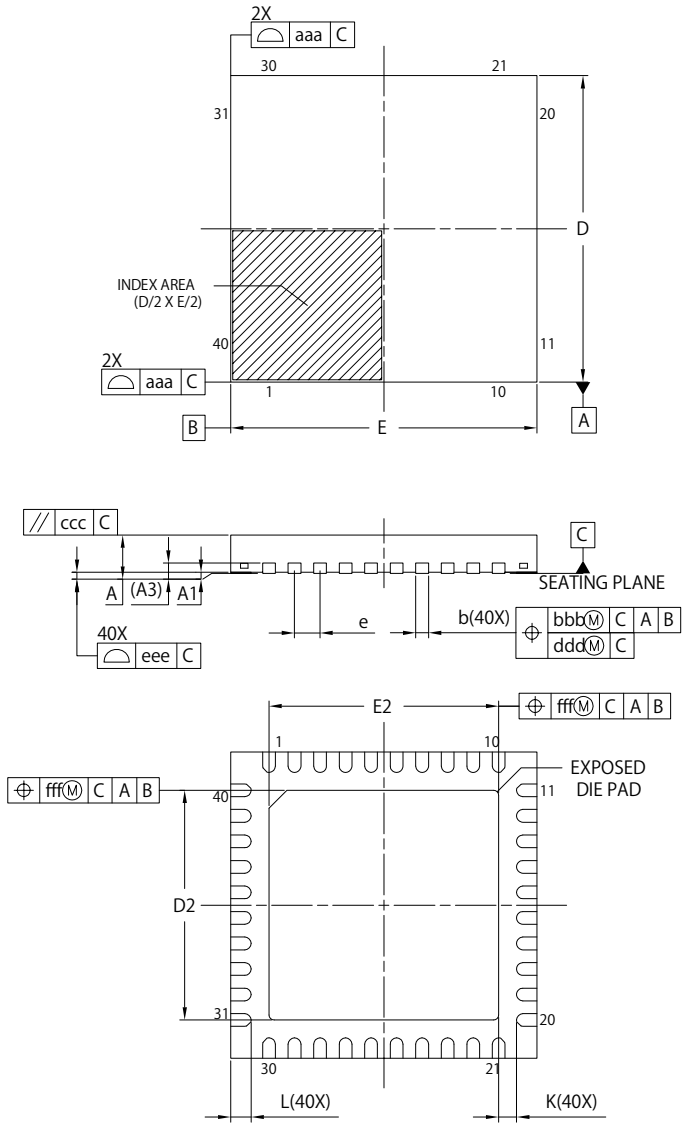
NOTE

1.Dimensions “※1” and “※2” do not include mold flash.

2.Dimension “※3” does not include trim offset.

45.6 40-pin Products

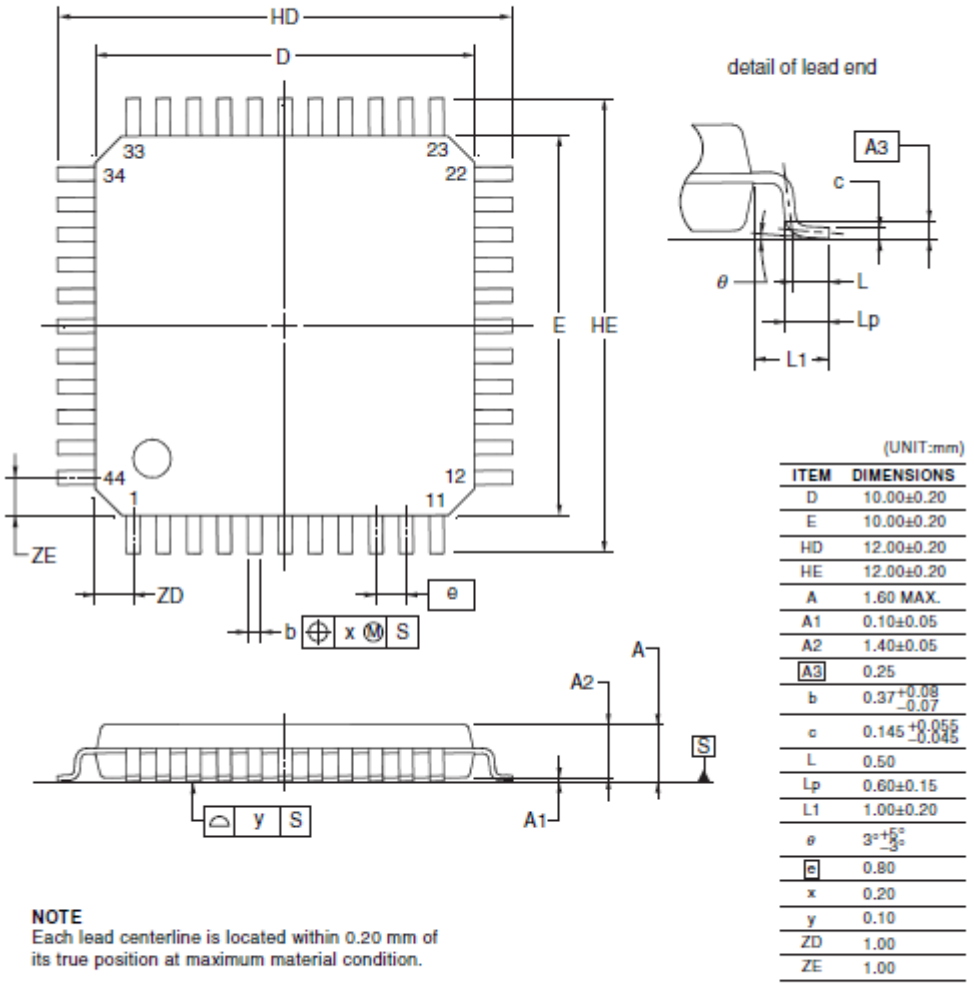
JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN040-6x6-0.50	PWQN0040KD-A	0.08



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	0.02	0.05
A_3		0.203 REF.	
b	0.18	0.25	0.30
D		6.00 BSC	
E		6.00 BSC	
e		0.50 BSC	
L	0.30	0.40	0.50
K	0.20	—	—
D_2	4.45	4.50	4.55
E_2	4.45	4.50	4.55
aaa		0.15	
bbb		0.10	
ccc		0.10	
ddd		0.05	
eee		0.08	
fff		0.10	

45.7 44-pin Products

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP44-10x10-0.80	PLQP0044GC-A	P44GB-80-UES-2	0.36



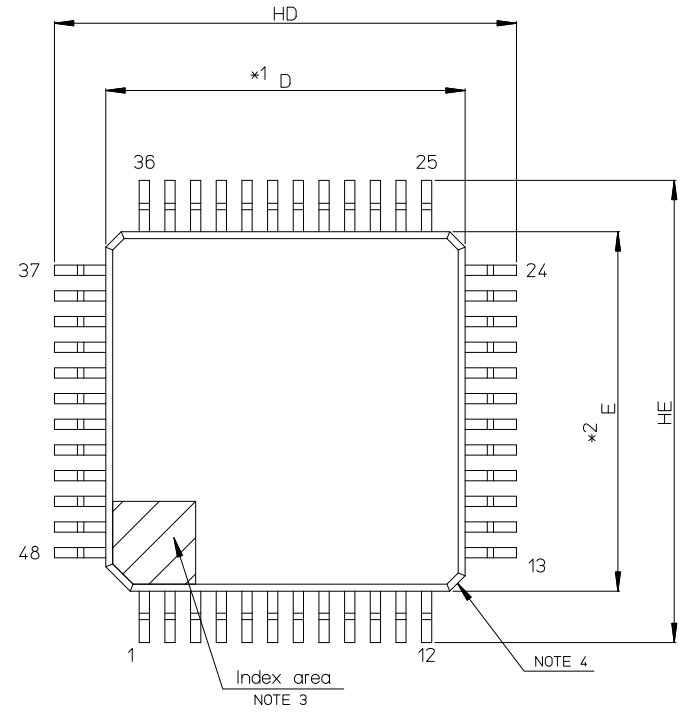
(UNIT:mm)

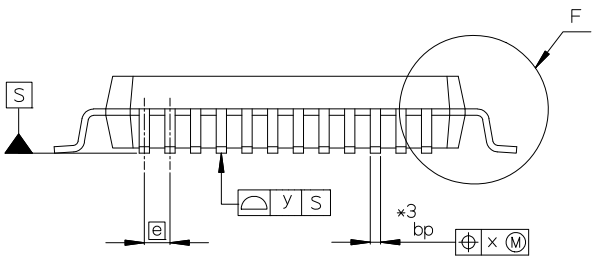
ITEM	DIMENSIONS
D	10.00±0.20
E	10.00±0.20
HD	12.00±0.20
HE	12.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	$0.37^{+0.08}_{-0.07}$
c	$0.145^{+0.055}_{-0.045}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.80
x	0.20
y	0.10
ZD	1.00
ZE	1.00

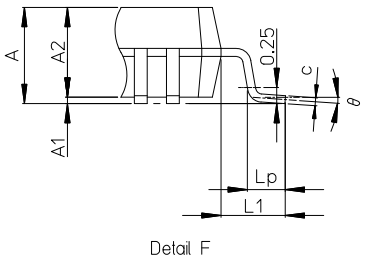
NOTE
Each lead centerline is located within 0.20 mm of its true position at maximum material condition.

45.8 48-pin Products

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP48-7x7-0.50	PLQP0048KB-B	——	0.2g





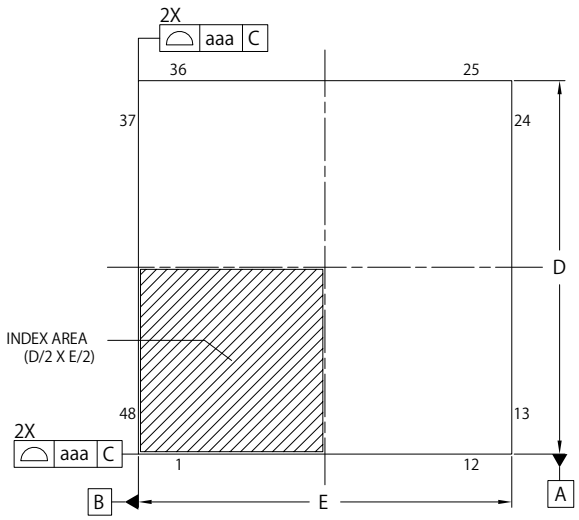


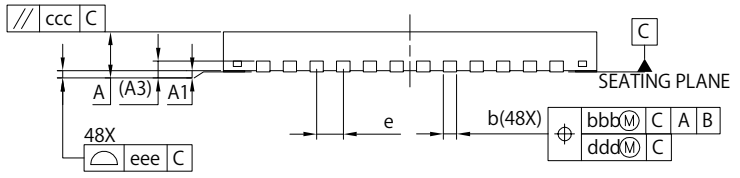
Detail F

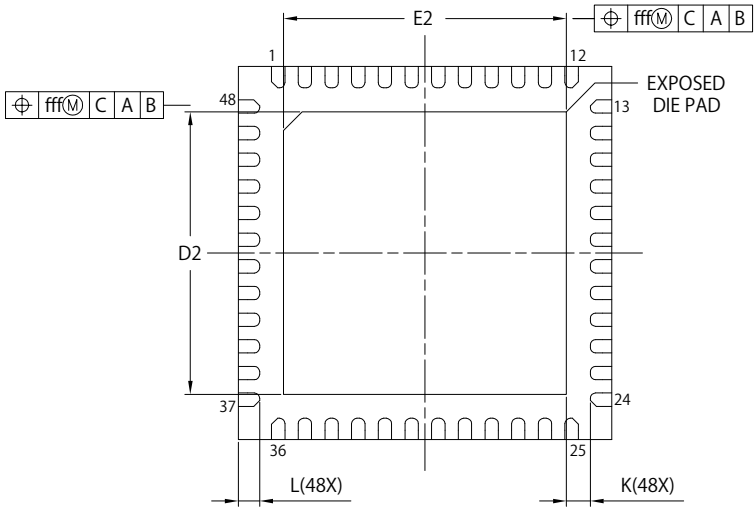
NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL; SIZE MAY VARY.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	6.9	7.0	7.1
E	6.9	7.0	7.1
A2	——	1.4	——
HD	8.8	9.0	9.2
HE	8.8	9.0	9.2
A	——	——	1.7
A1	0.05	——	0.15
bp	0.17	0.20	0.27
c	0.09	——	0.20
θ	0°	3.5°	8°
e	——	0.5	——
x	——	——	0.08
y	——	——	0.08
Lp	0.45	0.6	0.75
L1	——	1.0	——

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN048-7x7-0.50	PWQN0048KC-A	0.13 g



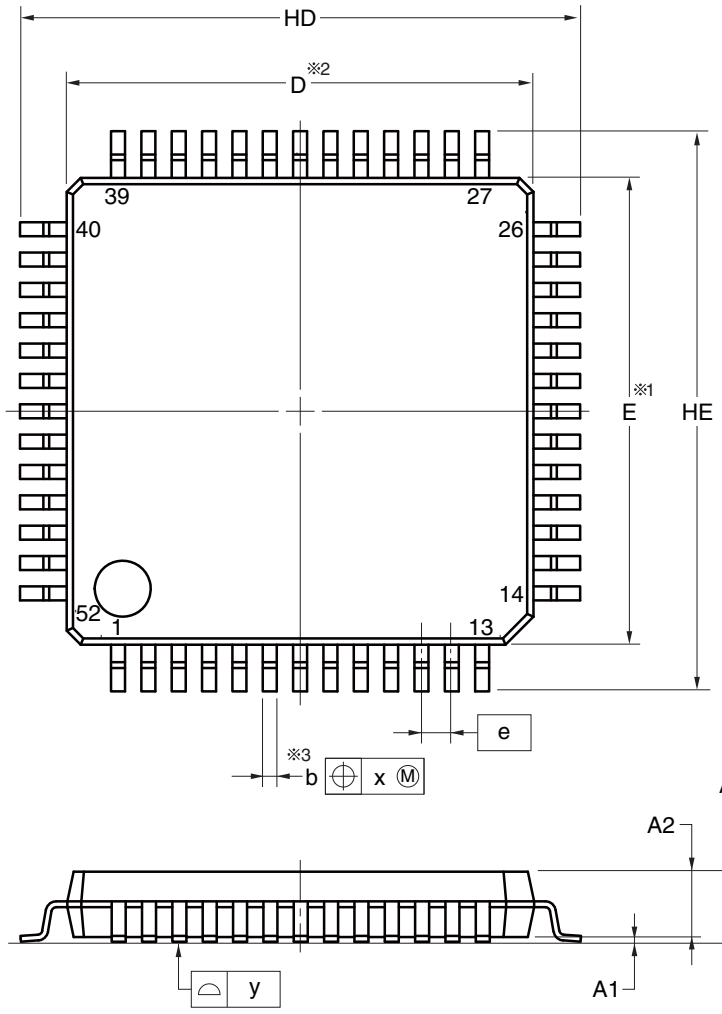




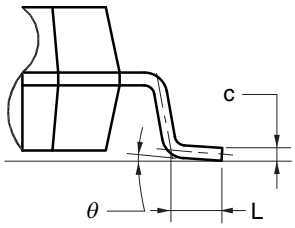
Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	0.02	0.05
A_3		0.203 REF.	
b	0.20	0.25	0.30
D		7.00 BSC	
E		7.00 BSC	
e		0.50 BSC	
L	0.30	0.40	0.50
K	0.20	—	—
D_2	5.25	5.30	5.35
E_2	5.25	5.30	5.35
aaa		0.15	
bbb		0.10	
ccc		0.10	
ddd		0.05	
eee		0.08	
fff		0.10	

45.9 52-pin Products

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP52-10x10-0.65	PLQP0052JA-A	P52GB-65-GBS-1	0.3



detail of lead end



(UNIT:mm)

ITEM	DIMENSIONS
D	10.00±0.10
E	10.00±0.10
HD	12.00±0.20
HE	12.00±0.20
A	1.70 MAX.
A1	0.10±0.05
A2	1.40
b	0.32±0.05
c	0.145±0.055
L	0.50±0.15
θ	0° to 8°
e	0.65
x	0.13
y	0.10

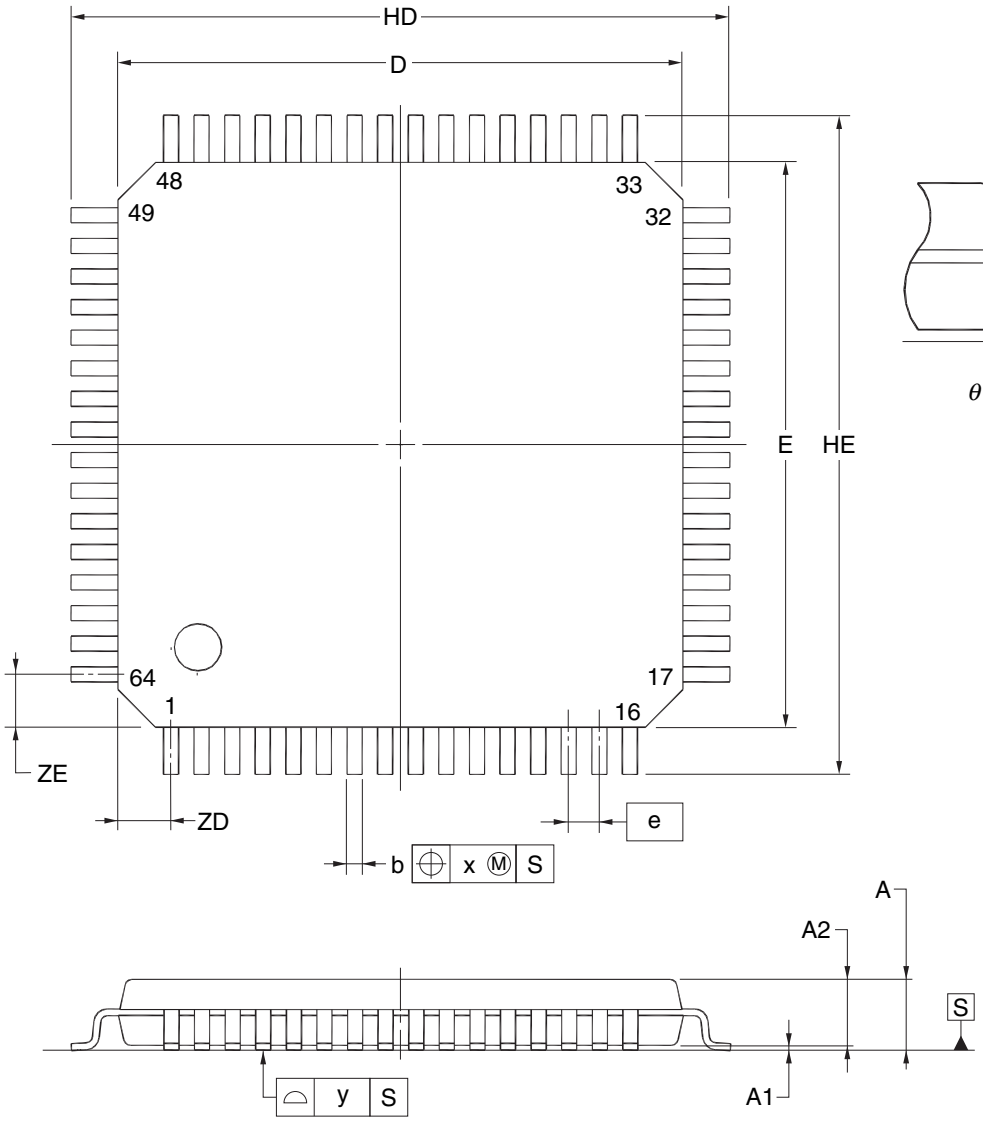
NOTE

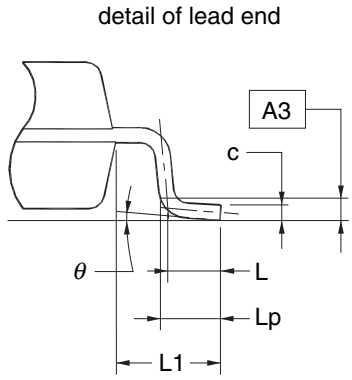
1.Dimensions "※1" and "※2" do not include mold flash.

2.Dimension "※3" does not include trim offset.

45.10 64-pin Products

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP64-12x12-0.65	PLQP0064JA-A	P64GK-65-UET-2	0.51





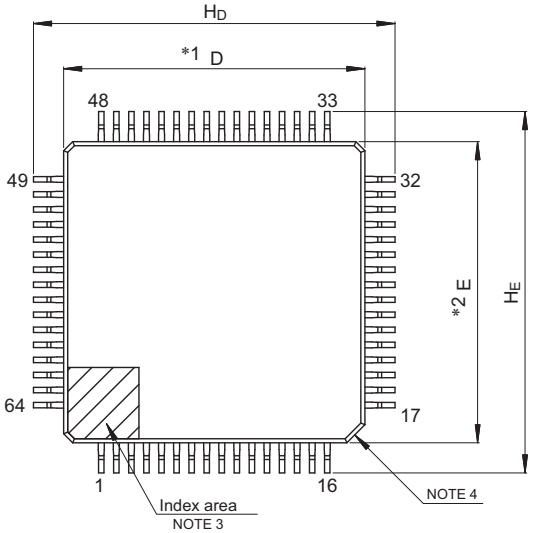
(UNIT:mm)

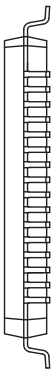
ITEM	DIMENSIONS
D	12.00±0.20
E	12.00±0.20
HD	14.00±0.20
HE	14.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	$0.32^{+0.08}_{-0.07}$
c	$0.145^{+0.055}_{-0.045}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.65
x	0.13
y	0.10
ZD	1.125
ZE	1.125

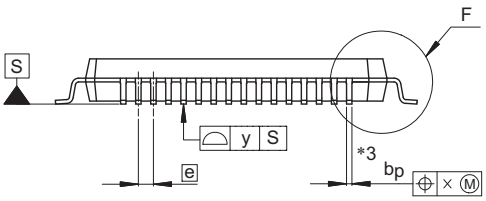
NOTE
Each lead centerline is located within 0.13 mm of its true position at maximum material condition.

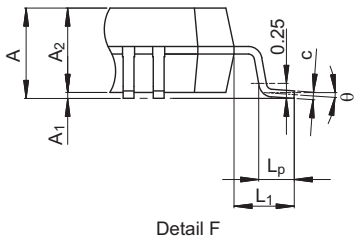
JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP64-10x10-0.50	PLQP0064KB-C	—	0.3

Unit: mm









Detail F

NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	9.9	10.0	10.1
E	9.9	10.0	10.1
A_2	—	1.4	—
H_D	11.8	12.0	12.2
H_E	11.8	12.0	12.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.15	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

Appendix A Revision History

A.1 Major Revisions in This Edition

(1/2)

Page	Description	Classification
Section 1 Outline		
p.5	Figure 1 - 1 Part Number, Memory Size, and Package of RL78/G24 was modified.	(d)
p.6	Table 1 - 1 List of Ordering Part Numbers was modified.	(d)
Section 4 Flexible Application Accelerator (FAA)		
p.252	4.17.2 Transfer of the program and data for the FAA, which are stored in the code flash memory, respectively to the instruction code memory and data memory: The descriptions were modified.	(c)
Section 7 Port Functions		
p.318	Table 7 - 5 Examples of Register and Output Latch Settings for Alternate Functions (19/22) was modified.	(a)
Section 10 Timer Array Unit (TAU)		
p.456	Figure 10 - 36 States of Operation following Set and Reset Signals was modified.	(a)
p.501	Figure 10 - 75 Procedure for Operations When the PWM Function Is to Be Used (2/2): Remark was modified.	(a)
Section 12 Timer RD2		
p.537	Figure 12 - 1 Block Diagram of Timer RD2 was modified.	(a)
p.542	Figure 12 - 5 Format of Timer RD Timer-KB PWM Output Gating Mode Control Register (TRDBCR) was modified.	(a)
p.672	12.5.9 Timer-KB PWM output gating mode was modified.	(c)
p.680	Table 12 - 28 Registers Associated with Timer RD2 Interrupts was modified.	(a)
Section 20 A/D Converter (ADC)		
p.1096	Figure 20 - 7 Format of A/D Converter Mode Register 1 (ADM1): Note 1 was modified.	(a)
p.1108	Figure 20 - 14 Format of Analog Input Channel Specification Registers for Advanced Mode (ADSn): Caution 3 was modified.	(c)
p.1113	Figure 20 - 19 Format of A/D Conversion Sampling Mode Specification Register (ADSPMOD): Caution was modified.	(c)
Section 24 Serial Array Unit (SAU)		
p.1290	Figure 24 - 48 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0) was modified.	(a)
p.1322	Figure 24 - 73 Flowchart of SNOOZE Mode Operation (Once Startup) was modified.	(a)
Section 25 Serial Interface IICA (IICA)		
p.1443	Figure 25 - 31 Slave Operation Flowchart (1) was modified.	(a)
Section 43 Electrical Characteristics (T_A = –40 to +105°C)		
p.1862	43.3.2 Supply current characteristics was modified.	(a)
p.1905	43.6.1 A/D converter characteristics: 1. Normal modes 1 and 2 was modified.	(a)

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

(2/2)

Page	Description	Classification
Section 44 Electrical Characteristics (T_A = –40 to +125°C)		
p.1940	44.3.2 Supply current characteristics was modified.	(a)

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

Here is the revision history of the preceding editions. Section indicates the section of each edition.

(1/4)

Edition	Description	Section
Rev.0.80	First edition issued.	Throughout
Rev.1.00	1.1 Features was modified.	Section 1 Outline
	Figure 1 - 1 Part Number, Memory Size, and Package of RL78/G24 was modified.	
	Table 1 - 1 List of Ordering Part Numbers was modified.	
	1.3.1 20-pin products: The figure was modified.	
	Table 1 - 2 Multiplexed Pin Functions of the 20-pin Products was modified.	
	1.3.2 24-pin products: The figure was modified.	
	Table 1 - 3 Multiplexed Pin Functions of the 24-pin Products was modified.	
	1.3.4 30-pin products: The figure was modified.	
	Table 1 - 5 Multiplexed Pin Functions of the 30-pin Products was modified.	
	1.3.5 32-pin products: The figure was modified.	
	Table 1 - 6 Multiplexed Pin Functions of the 32-pin Products was modified.	
	1.3.9 52-pin products: The figure was modified.	
	Table 1 - 10 Multiplexed Pin Functions of the 52-pin Products was modified.	
	2.2.3 VBAT pin was added.	Section 2 Pin Functions
	Table 2 - 2 Connections of Unused Pins was modified.	
	Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) was modified.	Section 3 CPU Architecture
	Table 4 - 15 List of FAA Addresses and Access Size for Registers (2nd SFRs) of Peripheral Functions	Section 4 Flexible Application Accelerator (FAA)
	Figure 9 - 22 Format of Peripheral Clock Control Register (PCKC): Note was modified.	Section 9 Clock Generator
	9.6.4 Example of setting the PLL circuit was modified.	
	Table 9 - 2 Examples of Transitions of the CPU Clock and SFR Settings was modified.	
	Table 12 - 13 TRDCMPm Register Functions in Timer-KB PWM Output Gating Mode was modified.	Section 12 Timer RD2
	15.1 Functions of 16-bit Timers KB30, KB31, and KB32: The descriptions were modified.	Section 15 16-bit Timers KB30, KB31, and KB32
	15.2.1 16-bit timer KB compare register nm (TKBCRnm): The descriptions were modified.	
	15.2.2 16-bit timer KB trigger compare register n (TKBTGCRn): The descriptions were modified.	
	Figure 15 - 9 Format of 16-bit Timer KB Operation Control Register n2 (TKBCTLn2): Caution 2 was modified, and Caution 3 was deleted.	
	Figure 15 - 21 Format of 16-bit Timer KB Skipping Control Register n (TKBTCTLn) was modified.	
	Figure 15 - 23 Format of External Interrupt Control Register n (INTPCTLn) was modified.	
	Figure 15 - 24 16-bit Timer KB3n Operation Setting Example (Operation Start Flow) was modified.	
	15.4.1 Counter basic operation, 2. Clear operation: The descriptions were modified.	
	15.4.4 Batch overwrite operation, 1. Procedure of batch overwrite was modified.	
	15.4.5 Skipping of interrupt requests and A/D conversion triggers was added.	
	Figure 15 - 46 Configuration in Standalone Mode (Period Controlled by the TKBCRn0 Register) was modified.	

(2/4)

Edition	Description	Section
Rev.1.00	15.4.6 Standalone mode (period controlled by the TKBCRn0 register), 5. Sample of register settings in standalone mode (period controlled by the TKBCRn0 register) was modified.	Section 15 16-bit Timers KB30, KB31, and KB32
	Figure 15 - 50 Configuration in Standalone Mode (Period Controlled by External Trigger Input) was modified.	
	15.4.7 Standalone mode (period controlled by external trigger input), 4. Sample of register settings in standalone mode (period controlled by external trigger input) was modified.	
	15.4.8 Simultaneous start/stop mode, 2. Simultaneous start/stop mode, Master: Sample of register settings in standalone mode was modified.	
	15.4.8 Simultaneous start/stop mode, 3. Simultaneous start/stop mode, Slave: Sample of register settings in simultaneous start/stop mode was modified.	
	Figure 15 - 53 Configuration in Simultaneous Start/Stop Mode was modified.	
	15.4.9 Simultaneous start/clear mode, 1. Outline of functions: The descriptions were modified.	
	15.4.9 Simultaneous start/clear mode, 2. Simultaneous start/clear mode, Master: Sample register settings in standalone mode was modified.	
	15.4.9 Simultaneous start/clear mode, 3. Simultaneous start/clear mode, Slave: Sample register settings in simultaneous start/clear mode was modified.	
	Figure 15 - 55 Configuration in Simultaneous Start/Clear Mode (Period Controlled by Master) was modified.	
	15.4.10 Interleaved power factor correction (PFC) output mode: 3. List of register settings in interleaved PFC output mode was modified.	
	15.5 Optional Functions of 16-bit Timers KB30, KB31, and KB32: Remark was modified.	
	15.5.2 PWM output dithering function, 1. Available operating mode: The table was modified.	
	15.5.3 PWM output smooth start function, 1. Operating mode available for PWM output smooth start function: The table was modified.	
	15.5.4 PWM output gating function (without combining with PWM output smooth start function), 1. Operating mode available for PWM output gating function was modified: The table was modified.	
	15.5.5 PWM output gating function (combining with PWM output smooth start function): 1. Operating mode available for PWM output gating function was modified.	
	15.5.6 Maximum frequency limit function was modified.	
	15.5.7 Multi-phase function was added.	
	15.6.1 Forced output stop functions 1 and 2: 2. Start of or release from forced output stop functions 1 and 2 was modified.	
	Table 15 - 11 Configuration of Hardware for the Forced Output Stop Function was modified.	
	15.6.3 Registers controlling forced output stop function was modified.	
	Figure 15 - 90 Format of Forced Output Stop Function Control Register n3 (TKBPACTLn3): Caution 2 was added.	
	Figure 15 - 91 Format of Forced Output Stop Function Control Register n4 (TKBPACTLn4): Caution was deleted.	
	15.6.3.5 Pulse characteristics measurement capture register np (TKBPAPLSnp): The title was modified.	
	Figure 15 - 92 Format of Pulse Characteristics Measurement Capture Register np (TKBPAPLSnp): The title was modified.	
	15.6.3.6 Pulse characteristics measurement capture register npL (TKBPAPLSnpL): The title was modified.	

(3/4)

Edition	Description	Section
Rev.1.00	Figure 15 - 93 Format of Pulse Characteristics Measurement Capture Register npL (TKBPAPLSnpL): The title was modified.	Section 15 16-bit Timers KB30, KB31, and KB32
	Figure 15 - 103 Forced Output Stop Function 2 When TKBPAFIXnp = 1 (that is, the Fixed Off Function): The descriptions were modified.	
	15.9 Operation of the Pulse Characteristics Measurement Function was modified.	
	Table 15 - 18 Relationship between INTPm Functions, Register Settings, and Active Signal Width, and Notes 2, 4, and 5 were modified.	
	Figure 15 – 121 Timing of the Forced Output Stop Request Signal and Timer Restart Request Signal Generated from INTP0, INTP20, and INTP21, and Notes 1, 3, and 4 were modified.	
	Table 22 - 3 Relationship between Functions of Comparator n, Register Settings, and Active Signal Width, and Notes 3 and 5 were modified.	Section 22 Comparator Module (CMP)
	Figure 22 - 19 Timing for Generating the Forced Output Stop Request Signal and Timer Restart Request Signal by Comparator n, and Notes 3 and 4 were modified.	
	Figure 23 - 4 Format of PGA Control Register (PGACTL): Caution 3 was modified.	Section 23 Programmable Gain Amplifier (PGA)
	Figure 23 - 5 Format of PGA Input Channel Select Register (PGAINS): Caution 2 was modified.	
	Figure 23 - 7 Operation Stopping Flow of Programmable Gain Amplifier: Caution 1 was modified.	
	Section 26 was entirely modified.	Section 26 Digital Addressable Lighting Interface (DALI)
	36.2.2 Setting of flash read protection: The description was modified.	Section 36 Security Functions
	43.1 Absolute Maximum Ratings was modified.	Section 43 Electrical Characteristics (T_A = –40 to +105°C)
	43.2.1 Characteristics of the X1 and XT1 oscillators: The condition was modified.	
	43.2.3 Characteristics of the PLL oscillator: Note was deleted.	
	43.3.1 Pin characteristics: Notes 4 to 9 were modified.	
	43.3.1 Pin characteristics: Notes 4 to 8 were modified.	
	43.3.1 Pin characteristics was modified.	
	43.3.2 Supply current characteristics was modified.	
	43.3.2 Supply current characteristics: Peripheral Functions (Common to all products) was modified, and Note 20 was added.	
	43.4 AC Characteristics was modified.	
	43.5.1 Serial array unit: Note 3 was modified.	
	43.6.1 A/D converter characteristics: 1. Normal modes 1 and 2, and Notes 3 and 5 were modified.	
	43.6.1 A/D converter characteristics: 2. Normal modes 1 and 2 (advanced mode), and Notes 4, 6, and 9 were modified.	
	43.6.1 A/D converter characteristics, 3. Low-voltage modes 1 and 2: Note 6 was modified.	
	43.6.1 A/D converter characteristics: 4. Low-voltage modes 1 and 2 (advanced mode), and Notes 7 and 9 were modified.	
	43.6.2 Temperature sensor/internal reference voltage characteristics was modified.	
	43.6.3 D/A converter characteristics was modified.	
	43.8 Flash Memory Programming Characteristics was modified.	
	44.1 Absolute Maximum Ratings was modified.	Section 44 Electrical Characteristics (T_A = –40 to +125°C)

(4/4)

Edition	Description	Section
Rev.1.00	44.2.2 Characteristics of the on-chip oscillators was modified.	Section 44 Electrical Characteristics (T_A = –40 to +125°C)
	44.2.3 Characteristics of the PLL oscillator: The condition was modified, and Note was deleted.	
	44.3.1 Pin characteristics was modified.	
	44.3.2 Supply current characteristics was modified.	
	44.3.2 Supply current characteristics: Peripheral Functions (Common to all products) was modified, and Note 20 was added.	
	44.4 AC Characteristics was modified.	
	44.5.1 Serial array unit: 2. In simplified SPI (CSI) communications in the master mode with devices operating at same voltage levels with the internal SCKp clock (the ratings below are only applicable to CSI00) was deleted.	
	44.5.1 Serial array unit: 7. In simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels (2.5 V or 3 V) with the internal SCKp clock (the ratings below are only applicable to CSI00) was deleted.	
	44.5.1 Serial array unit: 7. In simplified SPI (CSI) communications in the slave mode with devices operating at different voltage levels (2.5 V or 3 V) with the external SCKp clock was modified.	
	44.6.1 A/D converter characteristics: 1. Normal modes 1 and 2, and Note 3 were modified.	
	44.6.1 A/D converter characteristics: 2. Normal modes 1 and 2 (advanced mode), and Note 3 were modified.	
	44.6.2 Temperature sensor/internal reference voltage characteristics was modified.	
	44.6.3 D/A converter characteristics was modified.	
	44.8 Flash Memory Programming Characteristics was modified.	

RL78/G24 User's Manual: Hardware

Publication Date: Rev.0.80 Feb 20, 2023
Rev.1.10 Nov 1, 2023

Published by: Renesas Electronics Corporation

RL78/G24

Renesas Electronics Corporation

R01UH0961EJ0110