User's Manual

16

RL78/I1A

User's Manual: Hardware

16-Bit Single-Chip Microcontrollers

Renesas Electronics
www.renesas.com

Rev.3.40 Mar 2024

Notice

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

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

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

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

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Contact Information

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

Trademarks

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

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

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

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

How to Use This Manual

Readers

This manual is intended for user engineers who wish to understand the functions of the RL78/I1A and design and develop application systems and programs for these devices.
The target products are as follows.

- 20-pin: R5F1076C
- 30-pin: R5F107AE, R5F107AC
- 38-pin: R5F107DE

Purpose

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

Organization

The RL78/I1A manual is separated into two parts: this manual and the software edition (common to the RL78 Microcontroller).

RL78/I1A **User's Manual** **Hardware** **(This Manual)**	**RL78 Family** **User's Manual** **Software**
• Pin functions • Internal block functions • Interrupts • Other on-chip peripheral functions • Electrical specifications	• 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 Microcontroller instructions:
  → Refer to the separate document **RL78 Family User's Manual: Software (R01US0015E)**.

Conventions

Data significance:	Higher digits on the left and lower digits on the right	
Active low representations:	$\overline{\times\times\times}$ (overscore over pin and signal name)	
Note:	Footnote for item marked with **Note** in the text	
Caution:	Information requiring particular attention	
Remark:	Supplementary information	
Numerical representations:	Binary	···×××× or ××××B
	Decimal	···××××
	Hexadecimal	···××××H

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

Documents Related to Devices

Document Name	Document No.
RL78/I1A User's Manual: Hardware	This manual
RL78 Family User's Manual: Software	R01US0015E

Documents Related to Flash Memory Programming

Document Name	Document No.
PG-FP5 Flash Memory Programmer User's Manual	R20UT0008E

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 MPUs & MCUs 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

CHAPTER 1 OUTLINE

1.1 Features

- O Operation clocks
  - 1 to 32 MHz (when using high-speed on-chip oscillator clock)
  - 32.768 kHz (when using subsystem clock)
- O General-purpose registers: (8-bit register × 8) × 4 banks
- O ROM: 32/64 KB, RAM: 2/4 KB, Data flash memory: 4 KB
- O 16-bit timers KB0 to KB2, and KC0 for PWM output
  - 16-bit timers KB0 to KB2: maximum 6 outputs (3 ch × 2)
    - Smooth start function, dithering function, forced output stop function (by comparator or external interrupt), and interleave function for PFC control supported
    - Average resolution 1 nsec output, 64 MHz (when using PLL) + dithering option
  - 16-bit timer KC0 (3 ch)
    PWM output gating function by interlocking with 16-bit timers KB0, KB1, and KB2
- O On-chip comparator (4 to 6 channels)/programmable gain amplifier (1 channel)
- O Flash memory
  Prohibition of block erase and rewriting (security function)
  Back ground operation (BGO): Instructions can be executed from the code flash memory while rewriting the data flash memory.
- O Self-programming (with boot swap function/flash shield window function)
- O On-chip debug function
- O Power-on-reset (POR) circuit and voltage detector (LVD)
- O On-chip multiplier and divider/multiply-accumulator
  - 16 bits × 16 bits = 32 bits (Unsigned or signed)
  - 32 bits ÷ 32 bits = 32 bits (Unsigned)
  - 16 bits × 16 bits + 32 bits = 32 bits (Unsigned or signed)
- O On-chip BCD adjustment
- O I/O ports: 13 to 29
- O Timer
  - Timer array unit: 16-bit timer × 8 channels
  - Watchdog timer: 1 channel
  - Real-time clock: 1 channel
  - 12-bit Interval timer: 1 channel
- O Serial interface
  - Simplified SPI (CSI [Note]), UART, I^2C (SM/PM bus)
    UART supports DALI for light communication and DMX512
- O 8/10-bit resolution A/D converter (V_{DD} = 2.7 to 5.5 V): 6 to 11 channels
- O Power supply voltage: V_{DD} = 2.7 to 5.5 V
- O Operating ambient temperature: T_A = –40 to +105°C, T_A = –40 to +125°C

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.

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

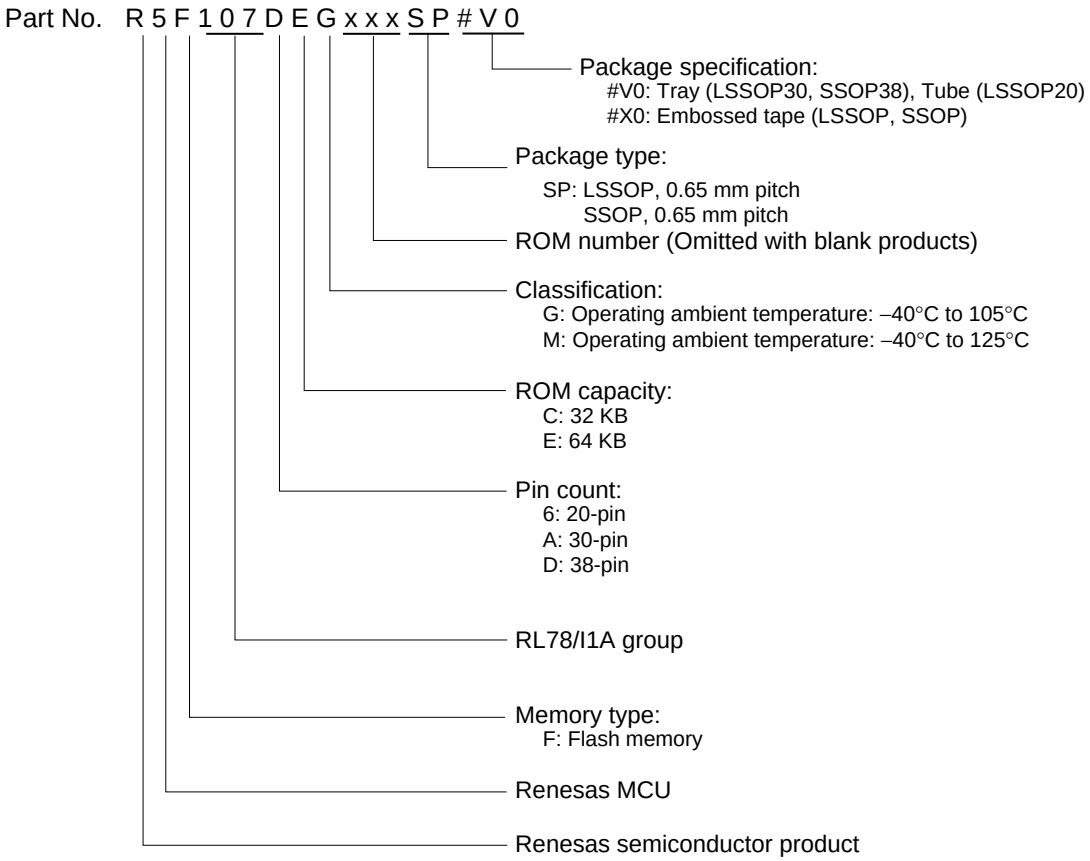
O ROM, RAM capacities

Flash ROM	Data flash	RAM	RL78/I1A		
			20 pins	30 pins	38 pins
64 KB	4 KB	4 KB[Note]	–	R5F107AE	R5F107DE
32 KB	4 KB	2 KB	R5F1076C	R5F107AC	–

Note This is about 3 KB when the self-programming function and data flash function are used. (For details, see **CHAPTER 3**.)

1.2 List of Part Numbers

Figure 1-1. Part Number, Memory Size, and Package of RL78/I1A



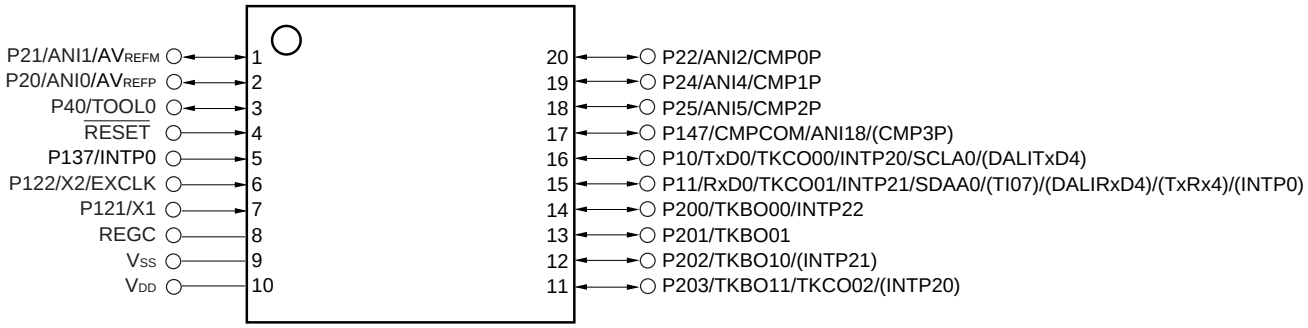
Pin count	Package	Operating Ambient Temperature	Part Number
20 pins	20-pin plastic LSSOP (4.4 × 6.5)	T_A = –40 to +105°C	R5F1076CGSP#V0, R5F1076CGSP#X0
		T_A = –40 to +125°C	R5F1076CMSP#V0, R5F1076CMSP#X0
30 pins	30-pin plastic LSSOP (7.62 mm (300))	T_A = –40 to +105°C	R5F107ACGSP#V0, R5F107AEGSP#V0, R5F107ACGSP#X0, R5F107AEGSP#X0
		T_A = –40 to +125°C	R5F107ACMSP#V0, R5F107AEMSP#V0, R5F107ACMSP#X0, R5F107AEMSP#X0
38 pins	38-pin plastic SSOP (7.62 mm (300))	T_A = –40 to +105°C	R5F107DEGSP#V0, R5F107DEGSP#X0
		T_A = –40 to +125°C	R5F107DEMSP#V0, R5F107DEMSP#X0

Caution The ordering part numbers represent the numbers at the time of publication. For the latest ordering part numbers, refer to the target product page of the Renesas Electronics website.

1.3 Pin Configuration (Top View)

1.3.1 20-pin products

- 20-pin plastic LSSOP (4.4 x 6.5)



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

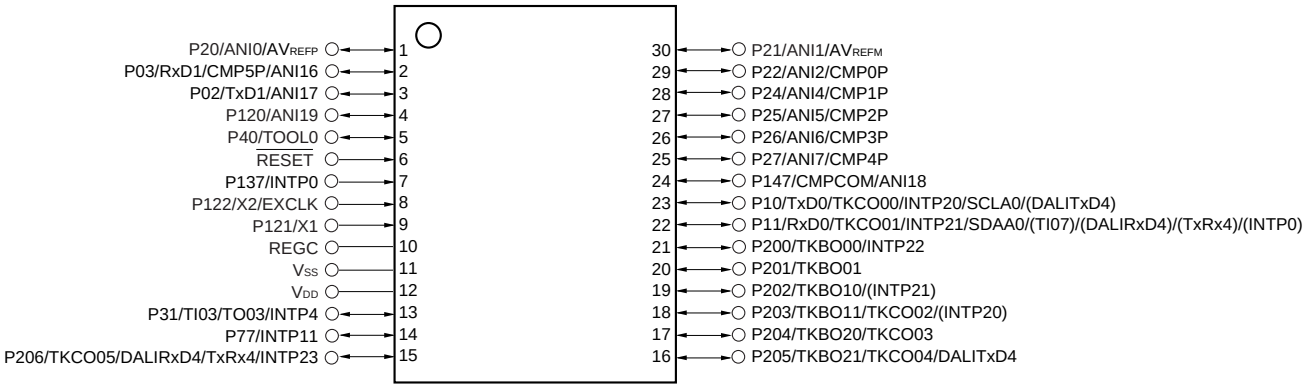
Remarks 1. For pin identification, see **1.4 Pin Identification**.

2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR1) or the input switch control register (ISC). See **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR1)** and **Figure 15-20 Format of Input Switch Control Register (ISC)**.

3. The shared function CMP3P can be assigned to P147 by setting the CMPSEL0 bit in the comparator input switch control register (CMPSEL).

1.3.2 30-pin products

- 30-pin plastic LSSOP (7.62 mm (300))



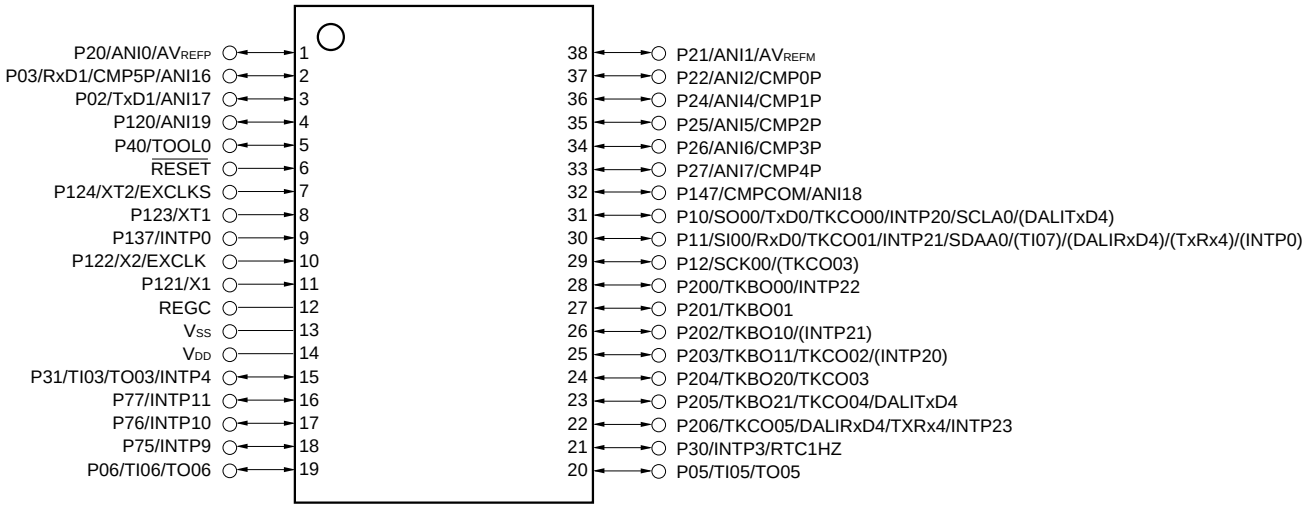
Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).

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

2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR1) or the input switch control register (ISC). See **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR1)** and **Figure 15-20 Format of Input Switch Control Register (ISC)**.

1.3.3 38-pin products

- 38-pin plastic SSOP (7.62 mm (300))



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

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

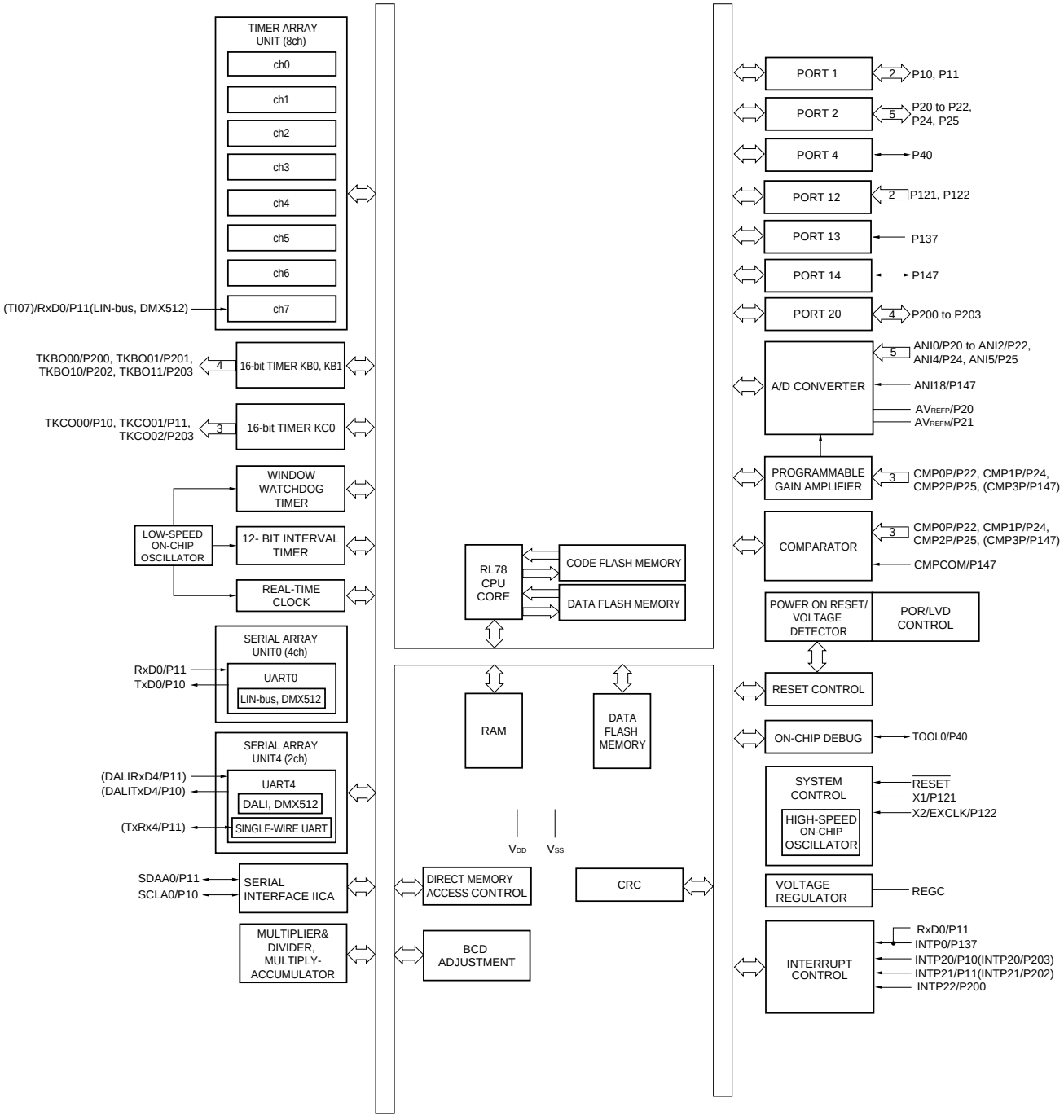
2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR1) or the input switch control register (ISC). See **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR1)** and **Figure 15-20 Format of Input Switch Control Register (ISC)**.

1.4 Pin Identification

ANI0 to ANI2, ANI4 to ANI7, ANI16 to ANI19:	Analog Input
AV_{REFM}:	Analog Reference Voltage Minus
AV_{REFP}:	Analog Reference Voltage Plus
CMP0P to CMP5P:	Comparator Analog Input
CMPCOM:	Comparator External Reference Voltage
EXCLK:	External Clock Input (Main System Clock)
EXCLKS:	External Clock Input (Subsystem Clock)
INTP0, INTP3, INTP4, INTP9, INTP10, INTP11, INTP20 to INTP23:	Interrupt Request from Peripheral
P02, P03, P05, P06:	Port 0
P10 to P12:	Port 1
P20 to P22, P24 to P27:	Port 2
P30, P31:	Port 3
P40:	Port 4
P75 to P77:	Port 7
P120 to P124:	Port 12
P137:	Port 13
P147:	Port 14
P200 to P206:	Port 20
REGC:	Regulator Capacitance
$\overline{\text{RESET}}$:	Reset
RTC1HZ:	Real-time Clock Correction Clock (1 Hz) Output
RxD0, RxD1, DALIRxD4:	Receive Data
SCK00:	Serial Clock Input/Output
SCLA0:	Serial Clock Input/Output
SDAA0:	Serial Data Input/Output
SI00:	Serial Data Input
SO00:	Serial Data Output
TI03, TI05, TI06, TI07:	Timer Input
TO03, TO05, TO06, TKBO00, TKBO01 to TKBO20, TKBO21, TKCO00 to TKCO05:	Timer Output
TOOL0:	Data Input/Output for Tool
TxRx4:	Serial Data Input/Output for Single Wired UART
TxD0, TxD1 DALITxD4:	Transmit Data
V_{DD}:	Power Supply
V_{SS}:	Ground
X1, X2:	Crystal Oscillator (Main System Clock)
XT1, XT2:	Crystal Oscillator (Subsystem Clock)

1.5 Block Diagram

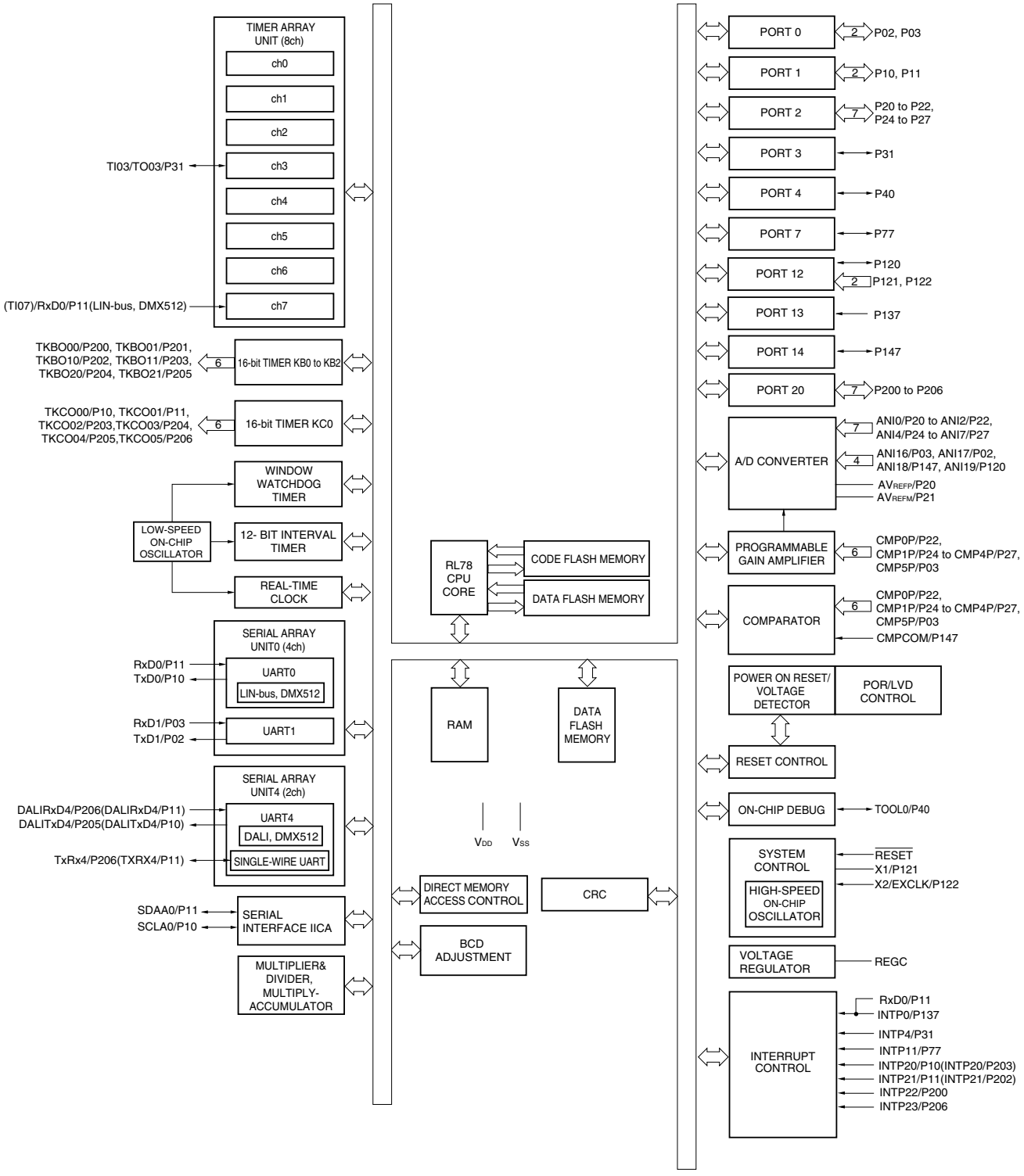
1.5.1 20-pin products



Remarks 1. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR1) or the input switch control register (ISC). See **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR1)** and **Figure 15-20 Format of Input Switch Control Register (ISC)**.

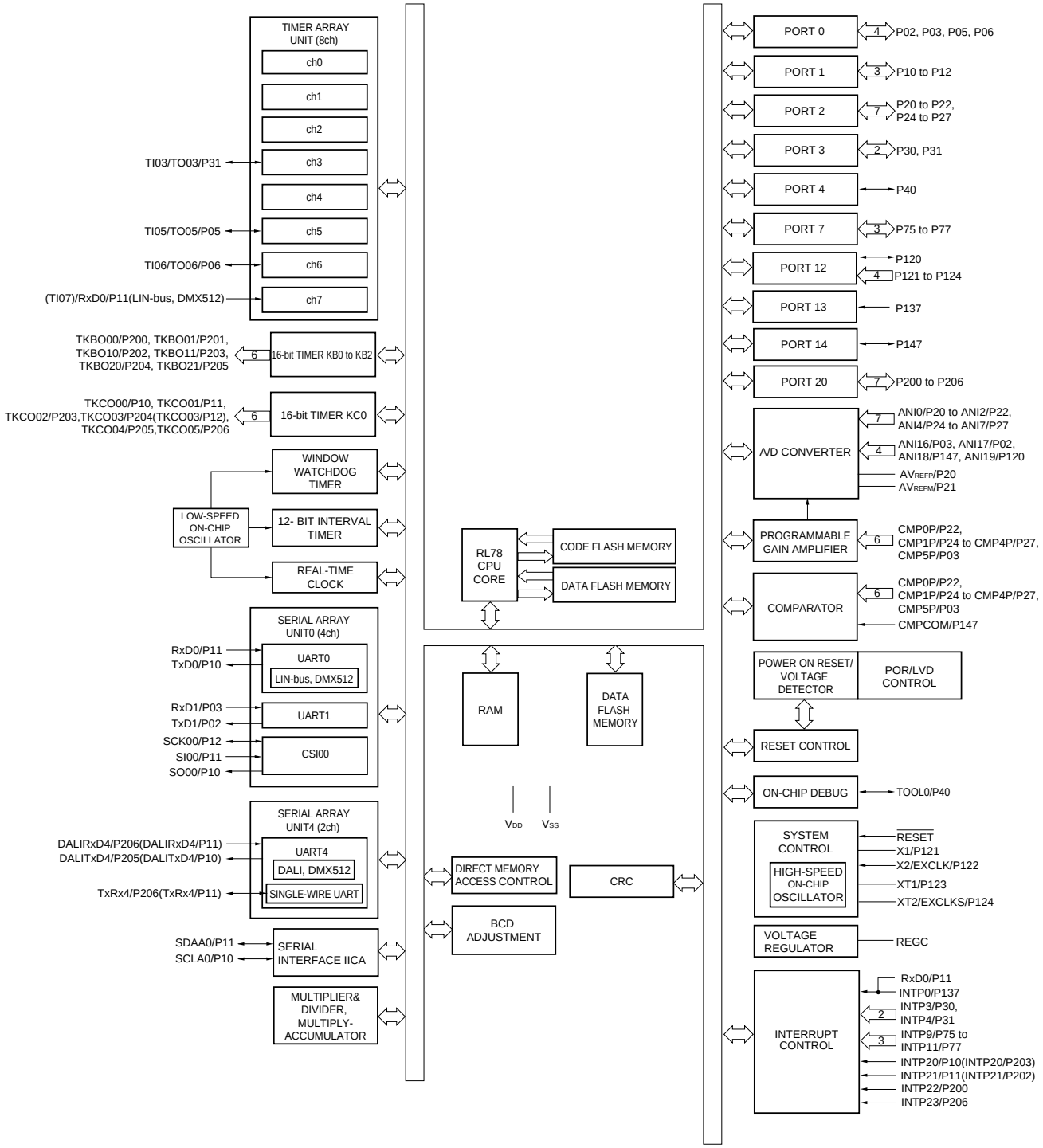
2. The shared function CMP3P can be assigned to P147 by setting the CMPSEL0 bit in the comparator input switch control register (CMPSEL).

1.5.2 30-pin products



Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR1) or the input switch control register (ISC). See **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR1)** and **Figure 15-20 Format of Input Switch Control Register (ISC)**.

1.5.3 38-pin products



Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR1) or the input switch control register (ISC). See **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR1)** and **Figure 15-20 Format of Input Switch Control Register (ISC)**.

1.6 Outline of Functions

Caution This outline describes the functions at the time when Peripheral I/O redirection register (PIOR1) is set to 00H.

(1/3)

Item		20-pin	30-pin		38-pin
		R5F1076C	R5F107AC	R5F107AE	R5F107DE
Code flash memory (KB)		32	32	64	64
Data flash memory (KB)		4	4	4	4
RAM (KB)		2	2	4 Note 1	4 Note 1
Address space		1 MB			
Main system clock	High-speed system clock	X1 (crystal/ceramic) oscillation, external main system clock input (EXCLK) HS (High-speed main) mode: 1 to 20 MHz (V_{DD} = 2.7 to 5.5 V), LS (Low-speed main) mode: 1 to 8 MHz (V_{DD} = 2.7 to 5.5 V)			
	High-speed on-chip oscillator	HS (High-speed main) mode: 1 to 32 MHz (V_{DD} = 2.7 to 5.5 V), LS (Low-speed main) mode: 1 to 8 MHz (V_{DD} = 2.7 to 5.5 V)			
Clock for 16-bit timers KB0 to KB2, and KC0		64 MHz (TYP.)			
Subsystem clock (38-pin products only)		XT1 (crystal) oscillation, external subsystem clock input (EXCLKS) 32.768 kHz			
Low-speed on-chip oscillator		15 kHz (TYP.)			
General-purpose register		(8-bit register × 8) × 4 banks			
Minimum instruction execution time		0.03125 μs (High-speed on-chip oscillator: f_{IH} = 32 MHz operation)			
		0.05 μs (High-speed system clock: f_{MX} = 20 MHz operation)			
		30.5 μs (Subsystem clock: f_{SUB} = 32.768 kHz operation) (38-pin products only)			
Instruction set		• 8-bit operation, 16-bit operation • Multiplication (8 bits × 8 bits) • Bit manipulation (Set, reset, test, and Boolean operation), etc.			
I/O port	Total	16	26		34
	CMOS I/O	13	23		29
	CMOS input	3	3		5
	CMOS output	–	–		–
Timer	16-bit timer TAU	8 channels (no timer output)	8 channels (timer output: 1, PWM output: 1 Note 2)		8 channels (timer outputs: 3, PWM outputs: 3 Note 2)
	16-bit timer KB	2 channels (PWM outputs: 4)	3 channels (PWM outputs: 6)		
	16-bit timer KC	1 channel (PWM outputs: 3)	1 channel (PWM outputs: 6)		

Notes 1. This is about 3 KB when the self-programming function and data flash function are used. (For details, see **CHAPTER 3**.)

2. The number of PWM outputs varies depending on the setting of channels in use (the number of masters and slaves) (see **6.9.3 Operation as multiple PWM output function**).

(2/3)

Item		20-pin	30-pin	38-pin
		R5F1076C	R5F107AC, R5F107AE	R5F107DE
Timer	Watchdog timer	1 channel		
	Real-time clock (RTC)	1 channel[Notes 1, 2]		
	12-bit interval timer (IT)	1 channel		
	RTC output	–		1 1 Hz (subsystem clock: f_{SUB} = 32.768 kHz)
8/10-bit resolution A/D converter		6 channels	11 channels	11 channels
Comparator		4 channels	6 channels	6 channels
Programmable gain amplifier		1 channel		
	Input[Note 3]	4 channels	6 channels	6 channels
Serial interface		[20-pin] • UART (Supporting LIN-bus and DMX512): 1 channel • UART (Supporting DALI communication): 1 channel [30-pin products] • UART (Supporting LIN-bus and DMX512): 1 channel • UART: 1 channel • UART (Supporting DALI communication): 1 channel [38-pin products] • Simplified SPI (CSI): 1 channel/UART (Supporting LIN-bus and DMX512): 1 channel • UART: 1 channel • UART (Supporting DALI communication): 1 channel		
	I^2C bus	1 channel	1 channel	1 channel
Multiplier and divider/multiply-accumulator		• 16 bits × 16 bits = 32 bits (Unsigned or signed) • 32 bits ÷ 32 bits = 32 bits (Unsigned) • 16 bits × 16 bits + 32 bits = 32 bits (Unsigned or signed)		
DMA controller		2 channels		
Vectored interrupt sources	Internal	27	30	30
	External	7	10	11
Reset		• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-reset • Internal reset by voltage detector • Internal reset by illegal instruction execution[Note 4] • Internal reset by RAM parity error • Internal reset by illegal-memory access		

Notes 1. The subsystem clock (f_{SUB}) can be selected as the operating clock only for 38-pin products.

2. The 20- and 30-pin products can only be used as the constant-period interrupt function.

3. The comparator input is alternatively used with analog input pin (ANI pin).

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

(3/3)

Item	20-pin	30-pin	38-pin
	R5F1076C	R5F107AC, R5F107AE	R5F107DE
Power-on-reset circuit	• Power-on-reset: 1.51 V (TYP.) • Power-down-reset: 1.50 V (TYP.)		
Voltage detector	• Rising edge: 2.81 V to 4.06 V (6 stages) • Falling edge: 2.75 V to 3.98 V (6 stages)		
On-chip debug function	Provided		
Power supply voltage	V_{DD} = 2.7 to 5.5 V		
Operating ambient temperature	T_A = –40 to +105°C (G: Industrial applications), T_A = –40 to +125°C (M: Industrial applications)		

CHAPTER 2 PIN FUNCTIONS

2.1 Port Function

The I/O buffer power supply for pins is provided by V_{DD}.

Set in each port I/O, buffer, pull-up resistor is also valid for alternate functions.

2.1.1 20-pin products

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P10	8-1-2	I/O	Input port	TxD0/TKCO00/INTP20/ SCLA0/(DALITxD4)	Port 1. 2-bit I/O port. Input of P10 and P11 can be set to TTL input buffer. Output of P10 and P11 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port
P11				RxD0/TKCO01/INTP21/ SDAA0/(TI07)/ (DALIRxD4)/(TxRx4)/ (INTP0)	
P20	4-3-1	I/O	Analog input port	ANI0/AV_{REFP}	Port 2. 5-bit I/O port. Can be set to analog input[Note 1]. Input/output can be specified in 1-bit units.
P21				ANI1/AV_{REFM}	
P22	4-18-1			ANI2/CMP0P	
P24				ANI4/CMP1P	
P25				ANI5/CMP2P	
P40	7-1-1	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P121	2-2-1	Input	Input port	X1	Port 12. 2-bit input only port.
P122				X2/EXCLK	
P137	2-1-2	Input	Input port	INTP0	Port 13. 1-bit input only port.
P147	7-9-1	I/O	Analog input port	ANI18/CMPCOM/ (CMP3P)	Port 14. 1-bit I/O port. P147 can be set to analog input[Note 2]. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P200	7-1-6	I/O	Input port	TKBO00/INTP22	Port 20. 4-bit I/O port. Output of P200 to P203 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P201				TKBO01	
P202				TKBO10/(INTP21)	
P203				TKBO11/TKCO02/ (INTP20)	
$\overline{\text{RESET}}$	2-1-1	Input	–	–	Input only pin for external reset. When external reset is not used, connect this pin to V_{DD} directly or via a resistor.

Notes 1. Setting digital or analog to each pin can be done in A/D port configuration register (ADPC).

2. When the each pin is used as input, specify them as either digital or analog in Port mode control register x (PMCx) (This register can be specified in 1-bit unit).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR1) or the input switch control register (ISC). See **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR1)** and **Figure 15-20 Format of Input Switch Control Register (ISC)**.

2.1.2 30-pin products

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P02	7-3-2	I/O	Analog input port	TxD1/ANI17	Port 0. 2-bit I/O port. Input of P03 can be set to TTL input buffer. Output of P02 can be set to N-ch open-drain output (V_{DD} tolerance). P02 and P03 can be set to analog input[Note 1]. Input/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.
P03	8-18-1			RxD1/CMP5P/ANI16	
P10	8-1-2	I/O	Input port	TxD0/TKCO00/INTP20/ SCLA0/(DALITxD4)	Port 1. 2-bit I/O port. Input of P10 and P11 can be set to TTL input buffer. Output of P10 and P11 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P11				RxD0/TKCO01/INTP21/ SDAA0/(TI07)/ (DALIRxD4)/(TxRx4)/ (INTP0)	
P20	4-3-1	I/O	Analog input port	ANI0/AV_{REFP}	Port 2. 7-bit I/O port. Can be set to analog input[Note 2]. Input/output can be specified in 1-bit units.
P21				ANI1/AV_{REFM}	
P22	4-18-1			ANI2/CMP0P	
P24				ANI4/CMP1P	
P25				ANI5/CMP2P	
P26				ANI6/CMP3P	
P27				ANI7/CMP4P	
P31	7-1-1	I/O	Input port	TI03/TO03/INTP4	Port 3. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P40	7-1-1	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting at input port.

Notes 1. When the each pin is used as input, specify them as either digital or analog in Port mode control register x (PMCx) (This register can be specified in 1-bit unit).

2. Setting digital or analog to each pin can be done in A/D port configuration register (ADPC).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR1) or the input switch control register (ISC). See **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR1)** and **Figure 15-20 Format of Input Switch Control Register (ISC)**.

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P77	7-1-1	I/O	Input port	INTP11	Port 7. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P120	7-3-1	I/O	Input port	ANI19	Port 12. 1-bit I/O port and 2-bit input port. P120 can be set to analog input[Note]. For only P120, input/output can be specified in 1-bit units. For only P120, use of an on-chip pull-up resistor can be specified by a software setting at input port.
P121	2-2-1	Input		X1	
P122				X2/EXCLK	
P137	2-1-2	Input	Input port	INTP0	Port 13. 1-bit input only port.
P147	7-9-1	I/O	Analog input port	ANI18/CMPCOM	Port 14. 1-bit I/O port. P147 can be set to analog input[Note]. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P200	7-1-6	I/O	Input port	TKBO00/INTP22	Port 20. 7-bit I/O port. Output of P200 to P206 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P201				TKBO01	
P202				TKBO10/(INTP21)	
P203				TKBO11/TKCO02/ (INTP20)	
P204				TKBO20/TKCO03	
P205				TKBO21/TKCO04/ DALITxD4	
P206				TKCO05/DALIRxD4/ TxRx4/INTP23	
$\overline{\text{RESET}}$	2-1-1	Input	–	–	Input only pin for external reset. When external reset is not used, connect this pin to V_{DD} directly or via a resistor.

Note When the each pin is used as input, specify them as either digital or analog in Port mode control register x (PMCx) (This register can be specified in 1-bit unit).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR1) or the input switch control register (ISC). See **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR1)** and **Figure 15-20 Format of Input Switch Control Register (ISC)**.

2.1.3 38-pin products

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P02	7-3-2	I/O	Analog input port	TxD1/ANI17	Port 0. 4-bit I/O port. Input of P03 can be set to TTL input buffer. Output of P02 can be set to N-ch open-drain output (V_{DD} tolerance). P02 and P03 can be set to analog input[Note 1]. Input/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.
P03	8-18-1			RxD1/CMP5P/ANI16	
P05	7-1-1		Input port	TI05/TO05	
P06				TI06/TO06	
P10	8-1-2	I/O	Input port	SO00/TxD0/TKCO00/ INTP20/SCLA0/ (DALITxD4)	Port 1. 3-bit I/O port. Input of P10 and P11 can be set to TTL input buffer. Output of P10 to P12 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P11				SI00/RxD0/TKCO01/ INTP21/SDAA0/(TI07)/ (DALIRxD4)/(TxRx4)/ (INTP0)	
P12	7-1-2			SCK00/(TKCO03)	
P20	4-3-1	I/O	Analog input port	ANI0/AV_{REFP}	Port 2. 7-bit I/O port. Can be set to analog input[Note 2]. Input/output can be specified in 1-bit units.
P21				ANI1/AV_{REFM}	
P22	4-18-1			ANI2/CMP0P	
P24				ANI4/CMP1P	
P25				ANI5/CMP2P	
P26				ANI6/CMP3P	
P27				ANI7/CMP4P	
P30	7-1-1	I/O	Input port	INTP3/RTC1HZ	Port 3. 2-bit I/O port. Input/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.
P31				TI03/TO03/INTP4	
P40	7-1-1	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting at input port.

Notes 1. When the each pin is used as input, specify them as either digital or analog in Port mode control register x (PMCx) (This register can be specified in 1-bit unit).

2. Setting digital or analog to each pin can be done in A/D port configuration register (ADPC).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR1) or the input switch control register (ISC). See **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR1)** and **Figure 15-20 Format of Input Switch Control Register (ISC)**.

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P75	7-1-1	I/O	Input port	INTP9	Port 7. 3-bit I/O port. Input/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.
P76				INTP10	
P77				INTP11	
P120	7-3-1	I/O	Input port	ANI19	Port 12. 1-bit I/O port and 4-bit input only port. P120 can be set to analog input[Note]. For only P120, input/output can be specified in 1-bit units. For only P120, use of an on-chip pull-up resistor can be specified by a software setting at input port.
P121	2-2-1	Input		X1	
P122				X2/EXCLK	
P123				XT1	
P124				XT2/EXCLKS	
P137	2-1-2	Input	Input port	INTP0	Port 13. 1-bit input only port.
P147	7-9-1	I/O	Analog input port	ANI18/CMPCOM	Port 14. 1-bit I/O port. P147 can be set to analog input[Note]. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P200	7-1-6	I/O	Input port	TKBO00/INTP22	Port 20. 7-bit I/O port. Output of P200 to P206 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P201				TKBO01	
P202				TKBO10/(INTP21)	
P203				TKBO11/TKCO02/ (INTP20)	
P204				TKBO20/TKCO03	
P205				TKBO21/TKCO04/ DALITxD4	
P206				TKCO05/DALIRxD4/ TxRx4/INTP23	
$\overline{\text{RESET}}$	2-1-1	Input	–	–	Input only pin for external reset. When external reset is not used, connect this pin to V_{DD} directly or via a resistor.

Note When the each pin is used as input, specify them as either digital or analog in Port mode control register x (PMCx) (This register can be specified in 1-bit unit).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR1) or the input switch control register (ISC). See **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR1)** and **Figure 15-20 Format of Input Switch Control Register (ISC)**.

2.2 Functions Other than Port Pins

2.2.1 Functions for each product

(1/2)

Function Name	38-pin	30-pin	20-pin
ANI0	√	√	√
ANI1	√	√	√
ANI2	√	√	√
ANI4	√	√	√
ANI5	√	√	√
ANI6	√	√	–
ANI7	√	√	–
ANI16	√	√	–
ANI17	√	√	–
ANI18	√	√	√
ANI19	√	√	–
CMP0P	√	√	√
CMP1P	√	√	√
CMP2P	√	√	√
CMP3P	√	√	√
CMP4P	√	√	–
CMP5P	√	√	–
CMPCOM	√	√	√
DALIRxD4	√	√	(√)
DALITxD4	√	√	(√)
INTP0	√	√	√
INTP3	√	–	–
INTP4	√	√	–
INTP9	√	–	–
INTP10	√	–	–
INTP11	√	√	–
INTP20	√	√	√
INTP21	√	√	√
INTP22	√	√	√
INTP23	√	√	–
REGC	√	√	√
RTC1HZ	√	–	–

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

(2/2)

Function Name	38-pin	30-pin	20-pin
$\overline{\text{RESET}}$	√	√	√
RxD0	√	√	√
RxD1	√	√	–
TxD0	√	√	√
TxD1	√	√	–
SCK00	√	–	–
SI00	√	–	–
SO00	√	–	–
SCLA0	√	√	√
SDAA0	√	√	√
TI03	√	√	–
TI05	√	–	–
TI06	√	–	–
TI07	√	√	√
TO03	√	√	–
TO05	√	–	–
TO06	√	–	–
TKBO00	√	√	√
TKBO01	√	√	√
TKBO10	√	√	√
TKBO11	√	√	√
TKBO20	√	√	–
TKBO21	√	√	–
TKCO00	√	√	√
TKCO01	√	√	√
TKCO02	√	√	√
TKCO03	√	√	–
TKCO04	√	√	–
TKCO05	√	√	–
TxRx4	√	√	√
X1, X2	√	√	√
EXCLK	√	√	√
XT1, XT2	√	–	–
EXCLKS	√	–	–
V_{DD}	√	√	√
AV_{REFP}	√	√	√
AV_{REFM}	√	√	√
V_{SS}	√	√	√
TOOL0	√	√	√

(The **Caution** and **Remark** are listed on the page after the next page.)

2.2.2 Description of functions

Function Name	I/O	Function
ANI0 to ANI7, ANI16 to ANI19	Input	A/D converter analog input (see **Figure 12-44 Analog Input Pin Connection**)
CMP0P to CMP5P	Input	Comparator 0 to 5 analog inputs
CMPCOM	Input	Comparator external reference potential input
DALIRxD4	Input	Serial data input to DALI/UART4
DALITxD4	output	Serial data output from DALI/UART4
INTP0, INTP3, INTP4, INTP9 to INTP11, INTP20 to INTP23	Input	External interrupt request input pin for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.
REGC	–	Pin for connecting regulator output stabilization capacitance for internal operation. Connect this pin to V_{SS} via a capacitor (0.47 to 1 μF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.
RTC1HZ	Output	Real-time clock correction clock (1 Hz) output
$\overline{\text{RESET}}$	Input	This is the active-low system reset input pin. When the external reset pin is not used, connect this pin directly or via a resistor to V_{DD}.
RxD0, RxD1	Input	Serial data input pin of serial interface UART0 and UART1
TxD0, TxD1	Output	Serial data output pin of serial interface UART0 and UART1
SCK00	I/O	Serial clock I/O pin of serial interface CSI00
SI00	Input	Serial data input pin of serial interface CSI00
SO00	Output	Serial data output pin of serial interface CSI00
SCLA0	I/O	Serial clock I/O pin of serial interface IICA0
SDAA0	I/O	Serial data I/O pin of serial interface IICA0
TI03, TI05 to TI07	Input	The pins for inputting an external count clock/capture trigger to 16-bit timers 03, 05 to 07
TO03, TO05, TO06	Output	Timer output pins of 16-bit timers 03, 05 and 06
TKBO00, TKBO01, TKBO10, TKBO11, TKBO20, TKBO21	Output	16-bit timers KB0 to KB2 output
TKCO00 to TKCO05	Output	16-bit timer KC0 output
TxRx4	I/O	Serial data I/O for single-wired UART
X1, X2	–	Resonator connection for main system clock
EXCLK	Input	External clock input for main system clock
XT1, XT2	–	Resonator connection for subsystem clock
EXCLKS	Input	External clock input for subsystem clock
V_{DD}	–	Positive power supply for all pins
AV_{REFP}	Input	A/D converter reference potential (+ side) input
AV_{REFM}	Input	A/D converter reference potential (– side) input
V_{SS}	–	Ground potential for all pins
TOOL0	I/O	Data I/O for flash memory programmer/debugger

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

(The **Caution** and **Remark** are listed on the next page.)

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

Table 2-1. Relationships Between P40/TOOL0 and Operation Mode After Reset Release

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

For details, see 28.3 Serial Programming Method.

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

2.3 Connection of Unused Pins

Table 2-2 shows the connections of unused pins.

Remark The pins mounted depend on the product. See **1.3 Pin Configuration (Top View)** and **2.1 Port Function.**

Table 2-2. Connection of Unused Pins (38-pin Products) (1/2)

Pin Name	I/O	Recommended Connection of Unused Pins
P02	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open. <When N-ch open-drain> Set the port output latch to 0 and leave open with low level out put.
P03		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P05		
P06		
P10		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open. <When N-ch open-drain> Set the port output latch to 0 and leave open with low level out put.
P11		
P12		
P20		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P21		
P22		
P24		
P25		
P26		
P27		
P30		
P31		
P40		Input: Independently connect to V_{DD} or leave open. Output: Leave open.
P75		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P76		
P77		

Table 2-2. Connection of Unused Pins (38-pin Products) (2/2)

Pin Name	I/O	Recommended Connection of Unused Pins
P120	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P121	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P122		
P123		
P124		
P137		
P147	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P200		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open. <When N-ch open-drain> Set the port output latch to 0 and leave open with low level out put.
P201		
P202		
P203		
P204		
P205		
P206		
$\overline{\text{RESET}}$	Input	Connect directly or via a resistor to V_{DD}.
REGC	–	Connect to V_{SS} via capacitor (0.47 to 1 μF).

2.4 Block Diagrams of Pins

Figures 2-1 to 2-12 show the block diagrams of the pins described in **2.1.1 20-pin products** to **2.1.3 38-pin products**.

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

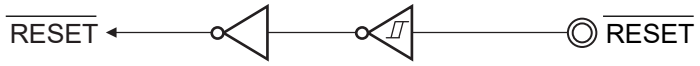
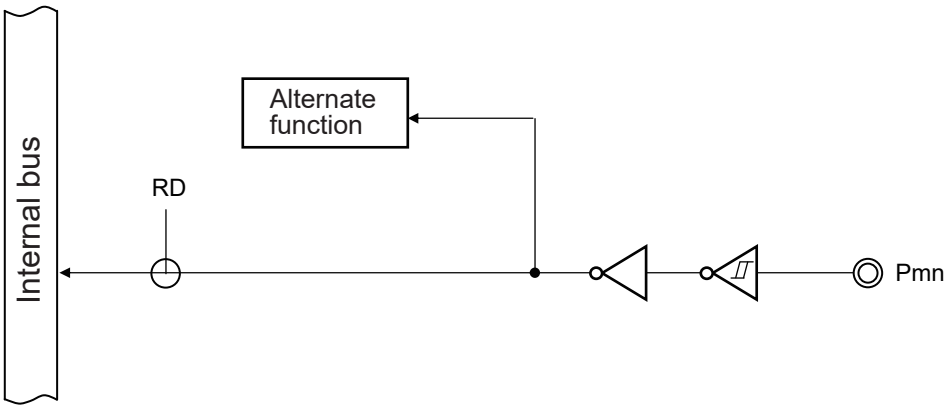
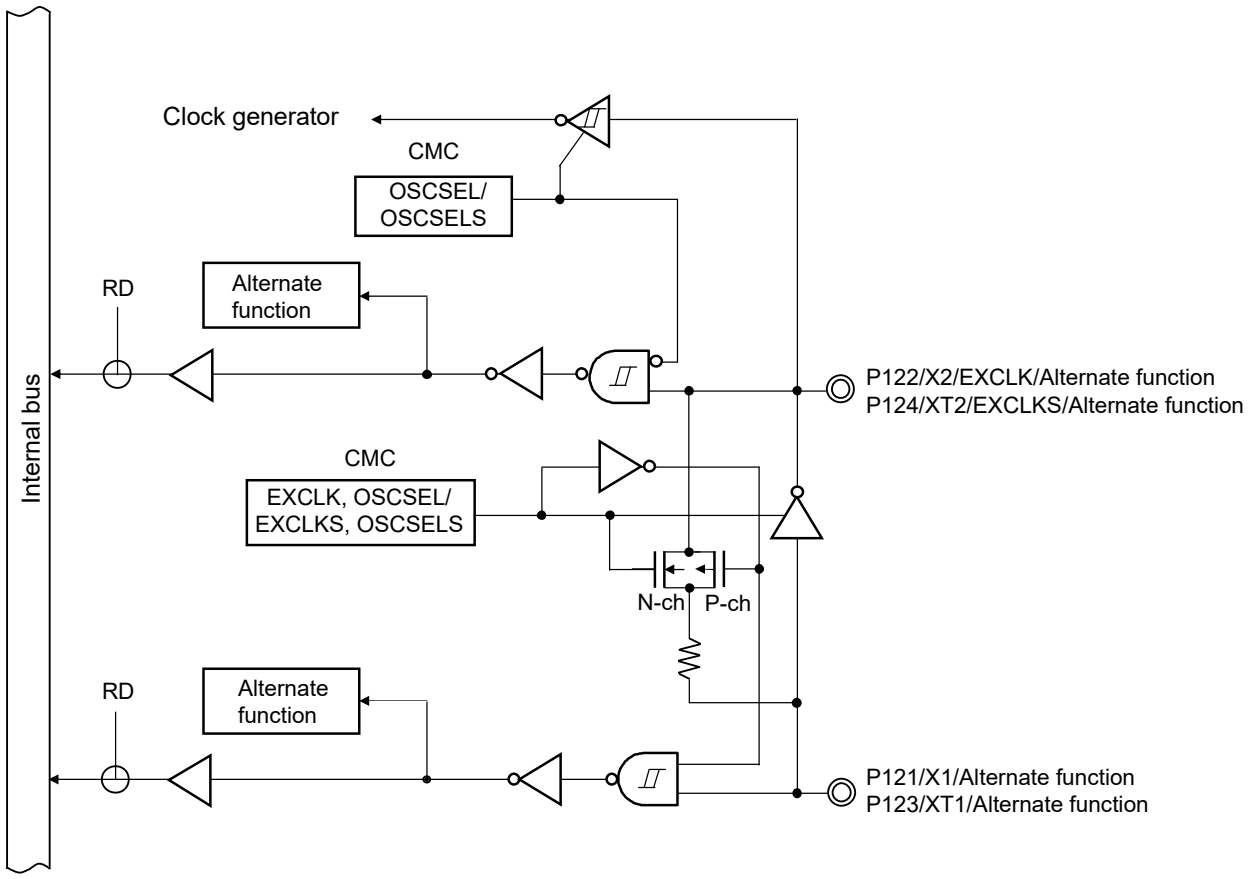


Figure 2-2. Pin Block Diagram for Pin Type 2-1-2



Remark For alternate functions, see **2.1 Port Function**.

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



Remark For alternate functions, see **2.1 Port Function**.

Figure 2-4. Pin Block Diagram for Pin Type 4-3-1

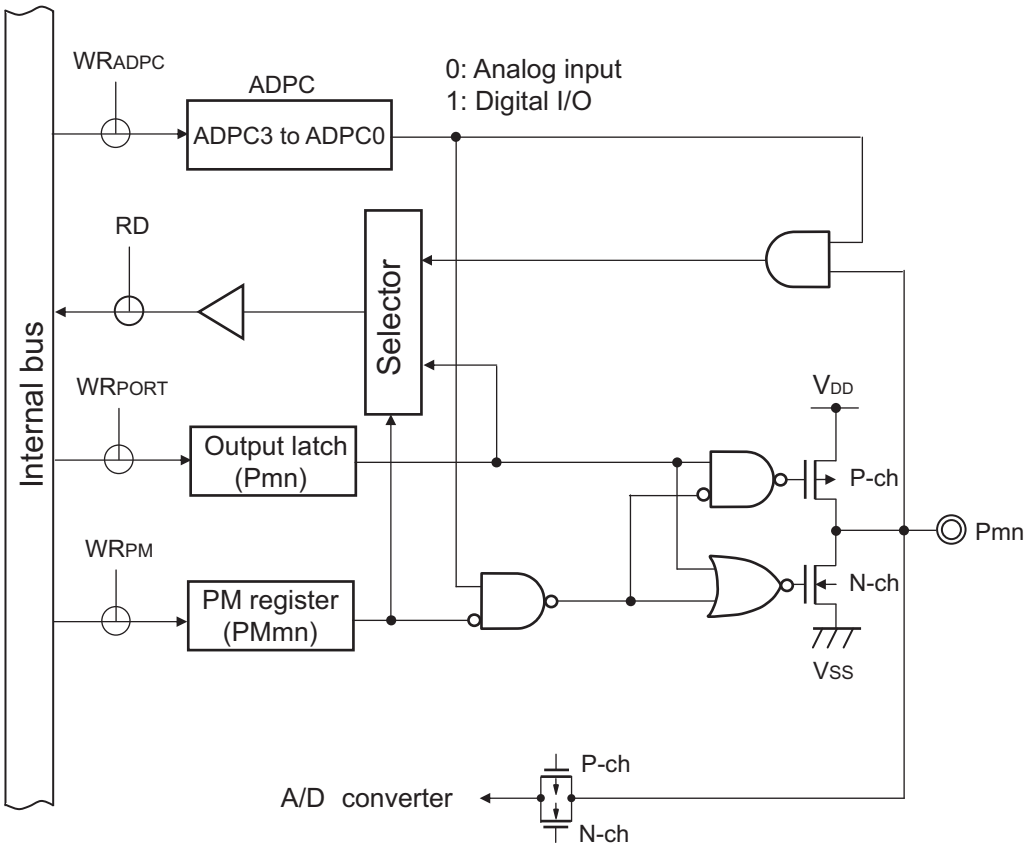


Figure 2-5. Pin Block Diagram for Pin Type 4-18-1

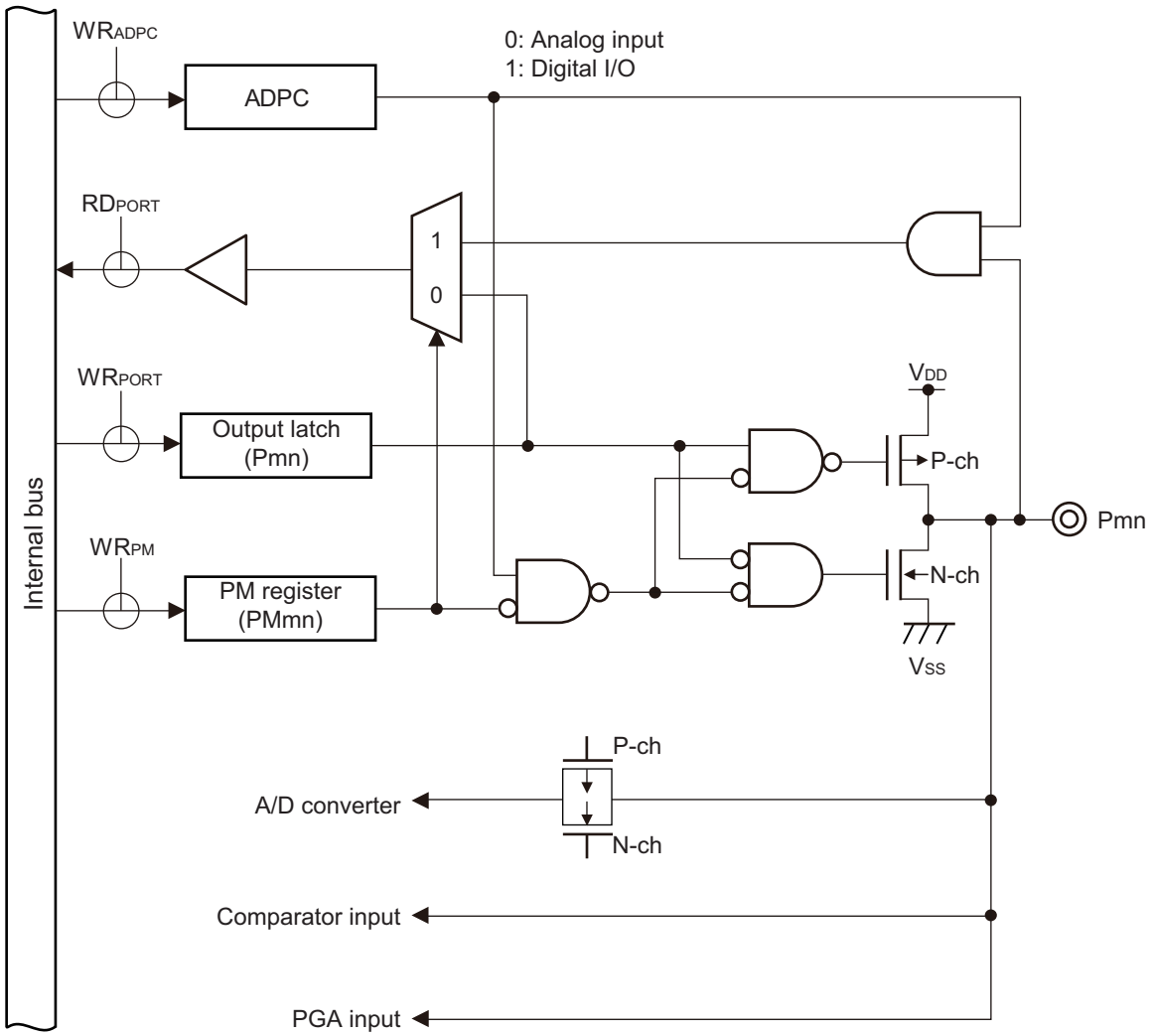
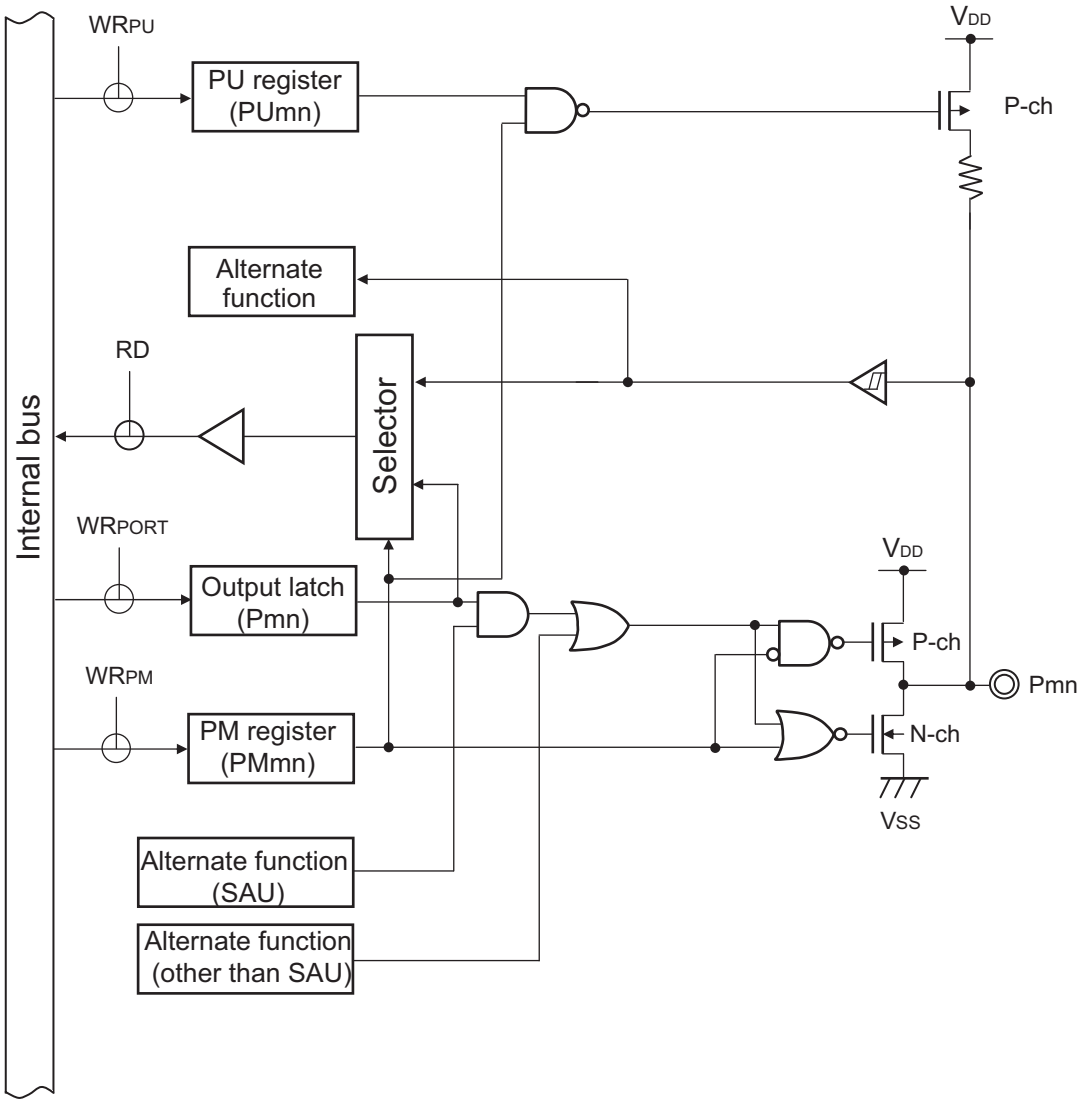


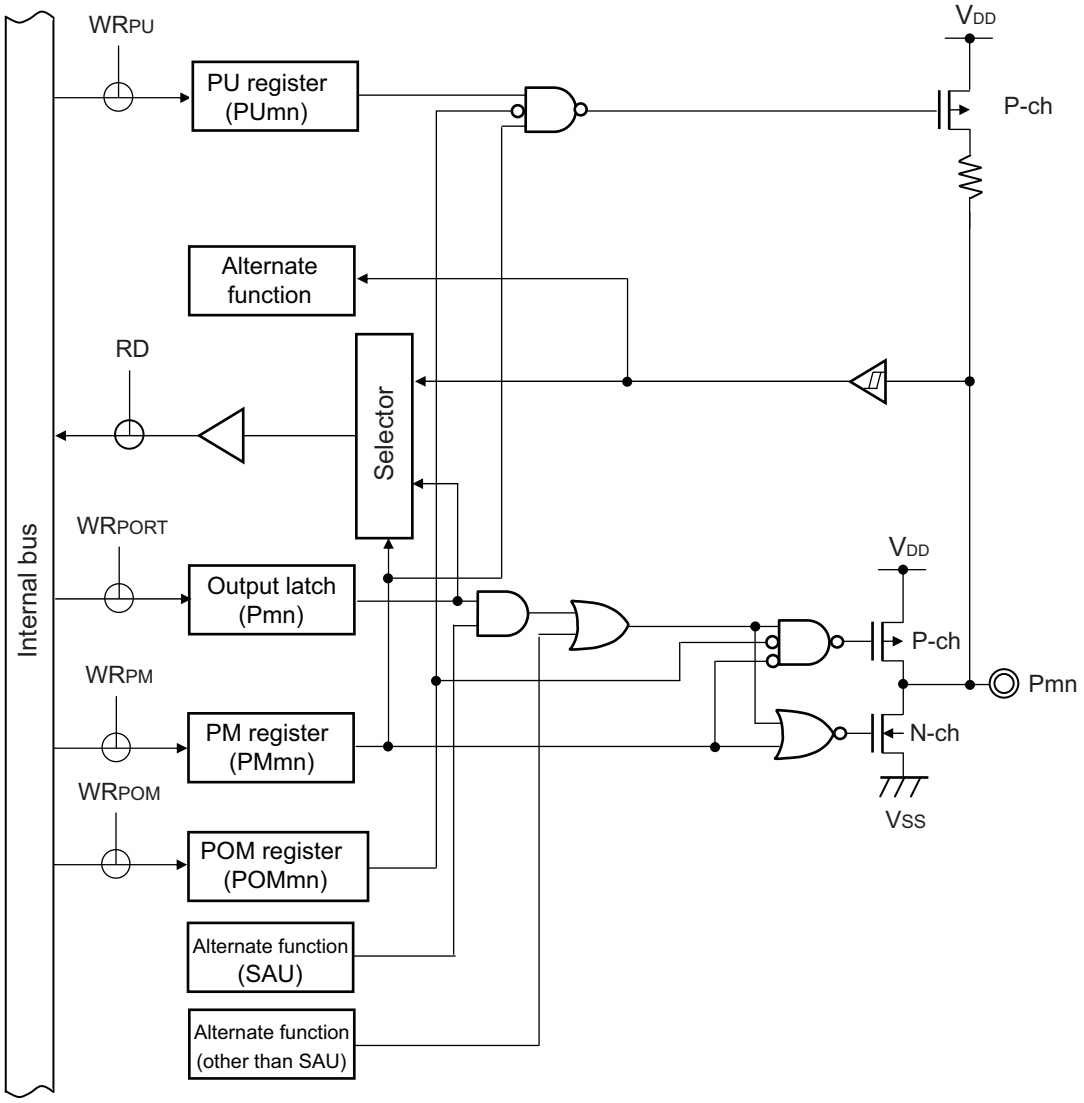
Figure 2-6. Pin Block Diagram for Pin Type 7-1-1



Remarks 1. For alternate functions, see **2.1 Port Function**.

2. SAU: Serial array unit

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

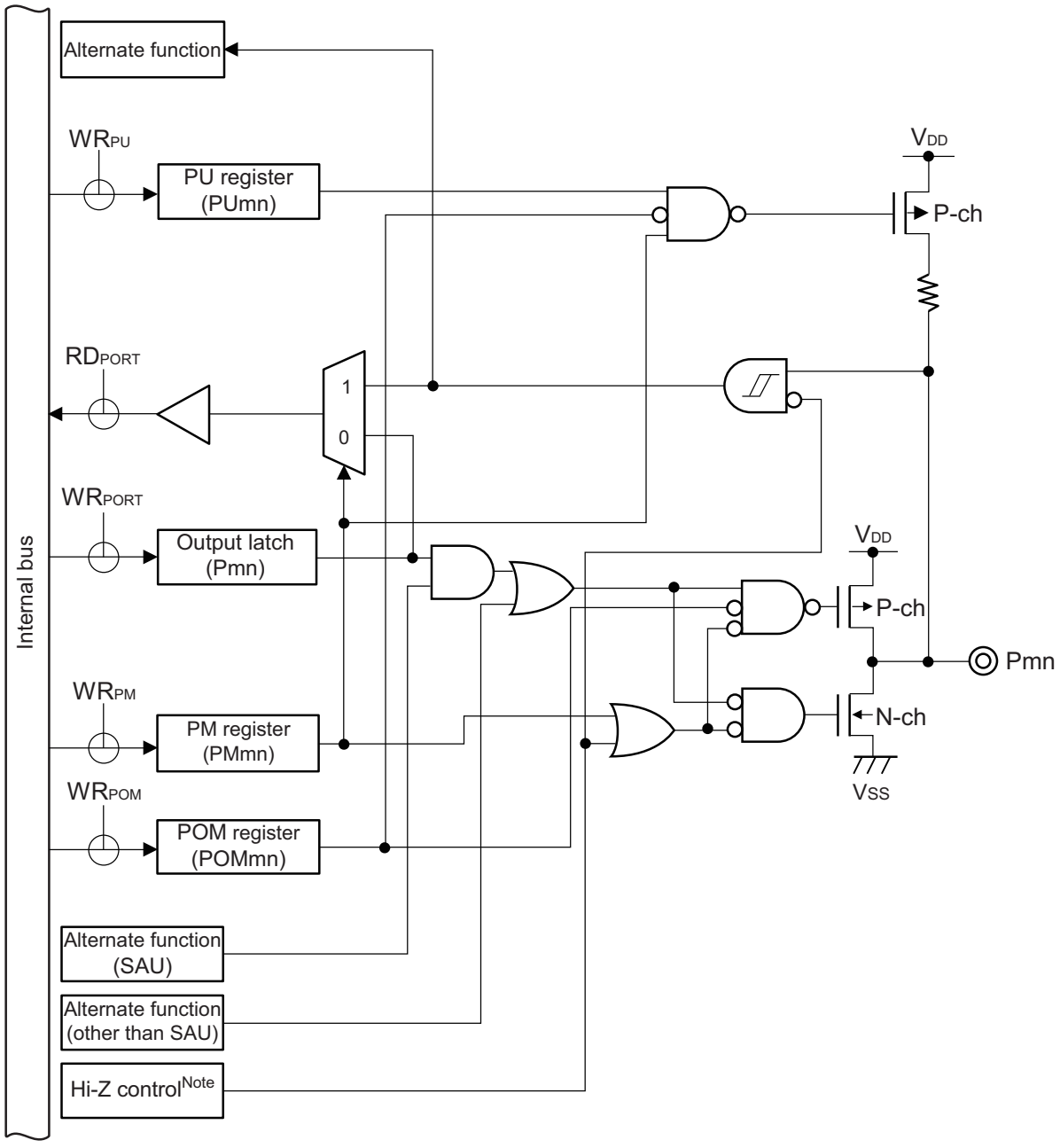


Caution The input buffer is enabled even if the type 7-1-2 pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMxx). This may lead to a through current flowing through the type 7-1-2 pin when the voltage level on this pin is intermediate. Changing the output level when the N-ch open drain output mode is selected may cause a glitch (V_{DD} level).

Remarks 1. For alternate functions, see **2.1 Port Function**.

2. SAU: Serial array unit

Figure 2-8. Pin Block Diagram for Pin Type 7-1-6



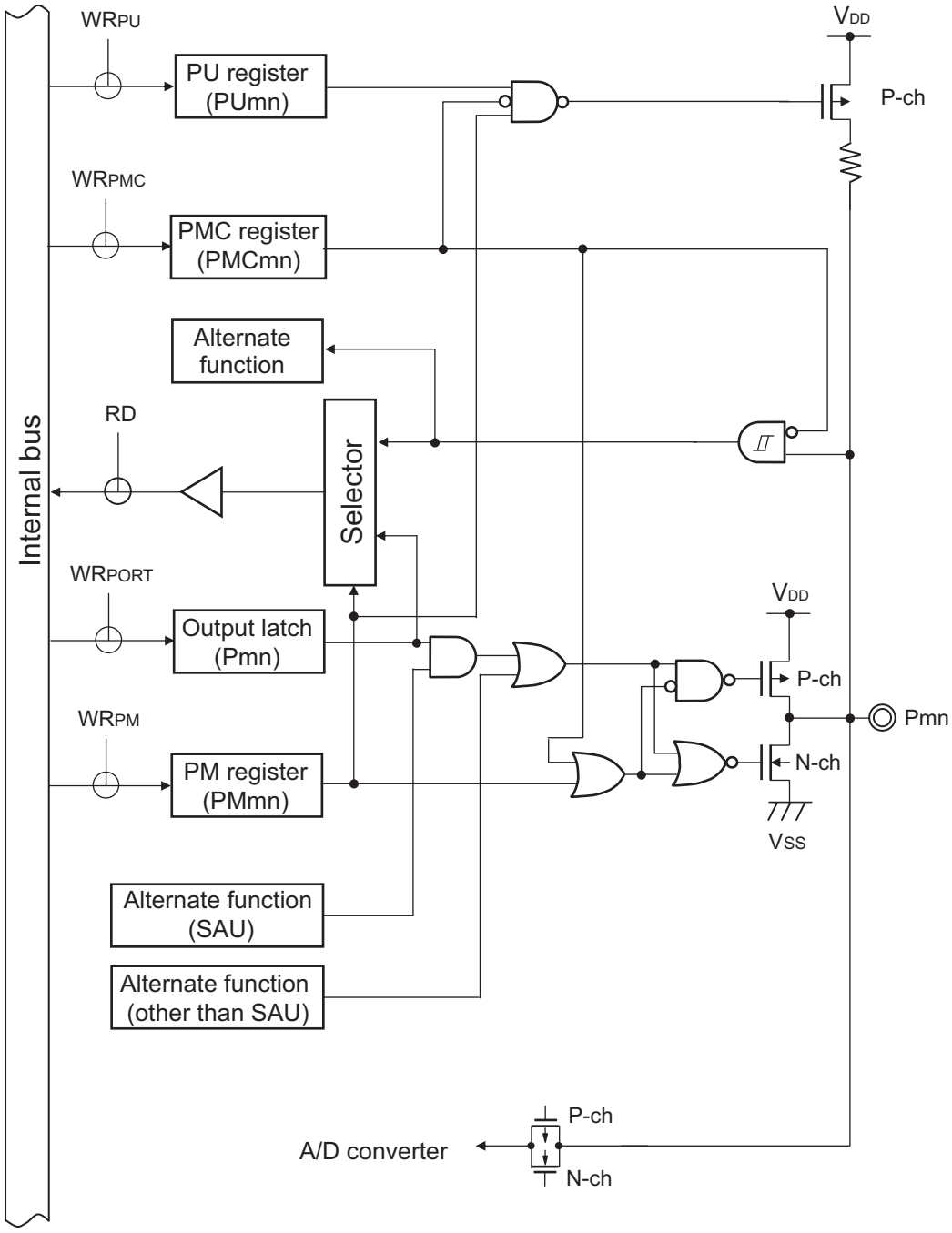
Note P206 does not provide the Hi-Z control function.

Caution The input buffer is enabled even if the type 7-1-6 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 (POMxx). This may lead to a through current flowing through the type 7-1-6 pin when the voltage level on this pin is intermediate. Changing the output level when the N-ch open drain output mode is selected may cause a glitch (V_{DD} level).

Remarks 1. For alternate functions, see **2.1 Port Function**.

2. SAU: Serial array unit

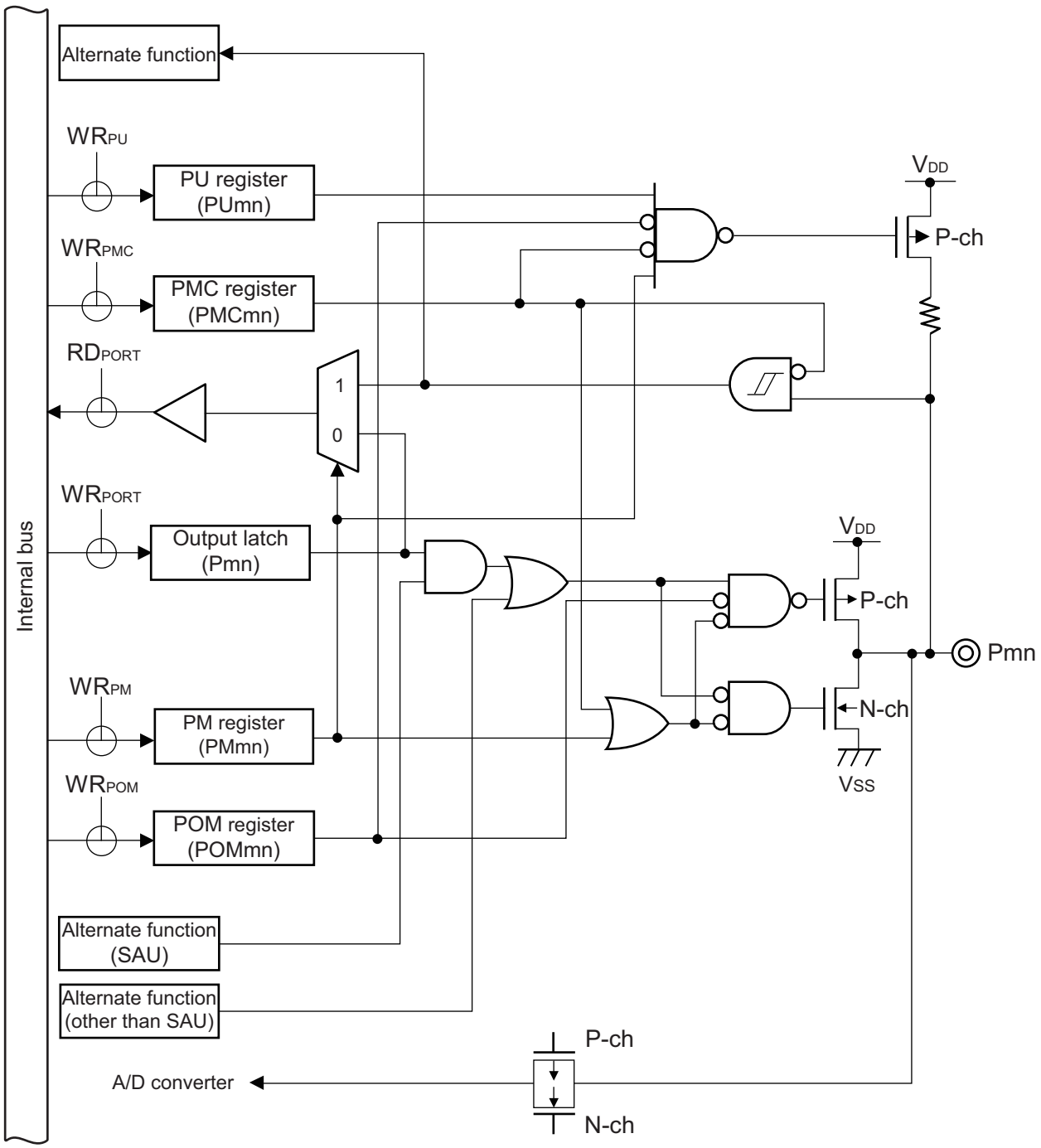
Figure 2-9. Pin Block Diagram for Pin Type 7-3-1



Remarks 1. For alternate functions, see **2.1 Port Function**.

2. SAU: Serial array unit

Figure 2-10. Pin Block Diagram for Pin Type 7-3-2

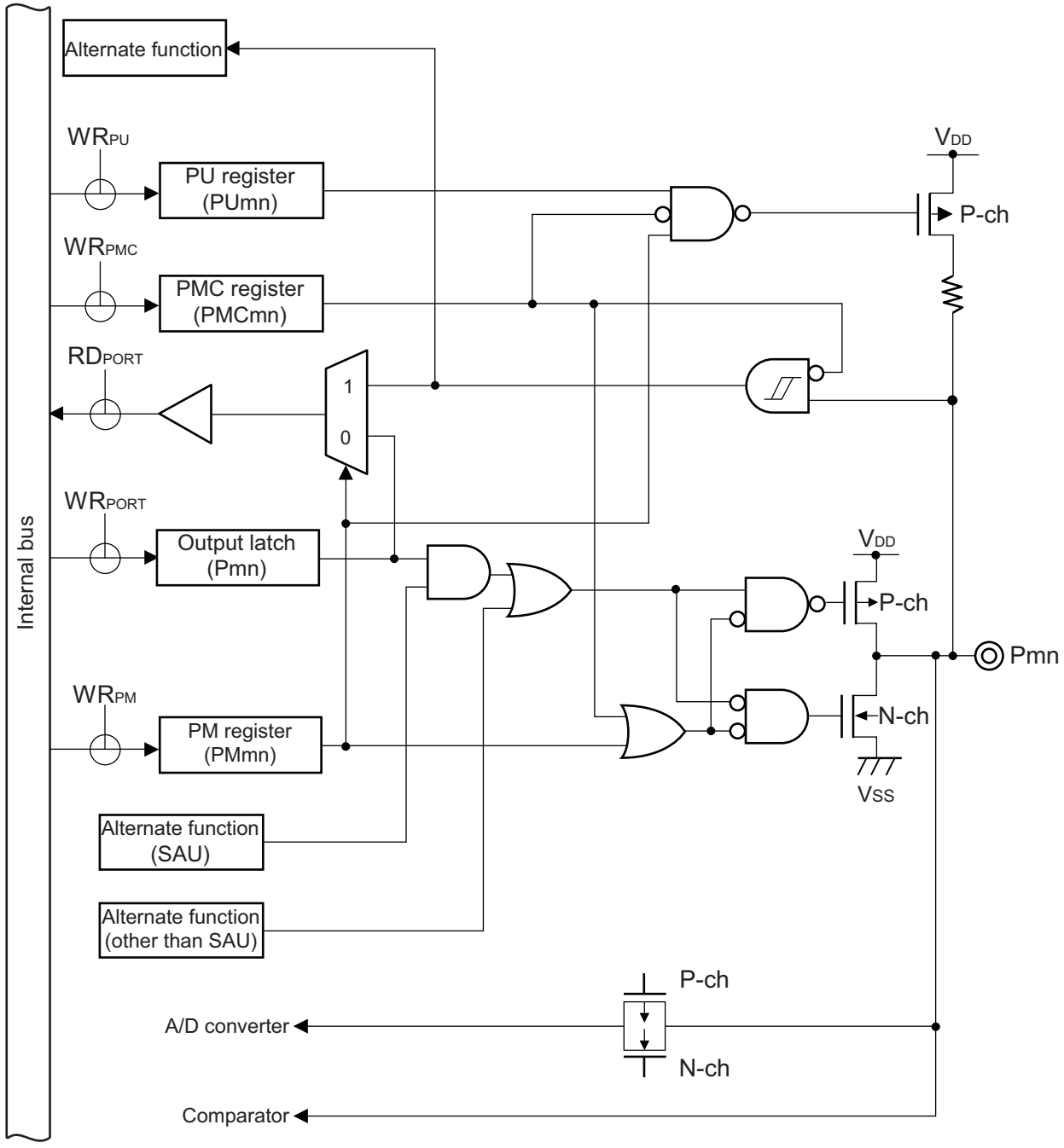


Caution The input buffer is enabled even if the type 7-3-2 pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMxx). This may lead to a through current flowing through the type 7-3-2 pin when the voltage level on this pin is intermediate. Changing the output level when the N-ch open drain output mode is selected may cause a glitch (V_{DD} level).

Remarks 1. For alternate functions, see **2.1 Port Function**.

2. SAU: Serial array unit

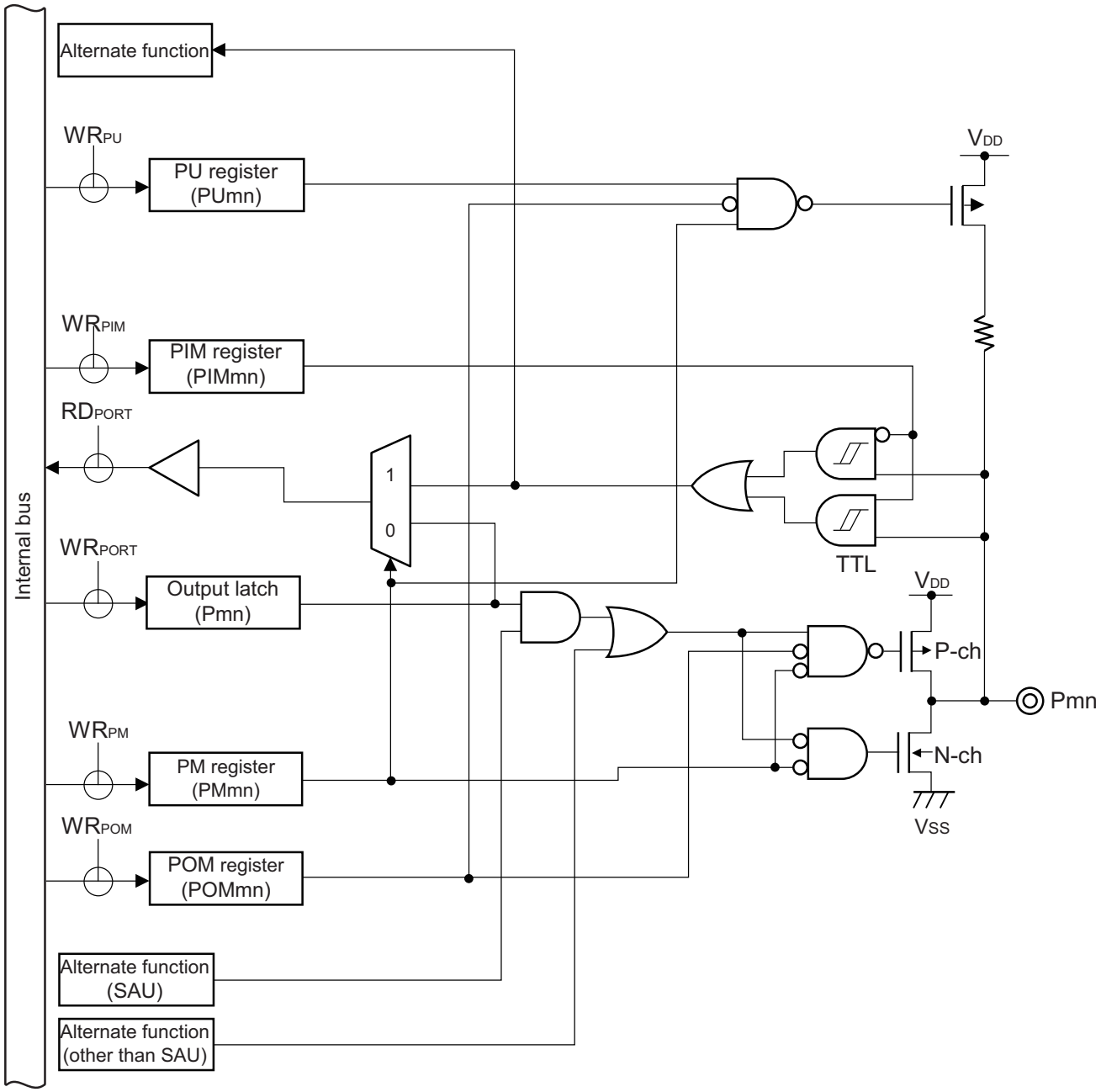
Figure 2-11. Pin Block Diagram for Pin Type 7-9-1



Remarks 1. For alternate functions, see **2.1 Port Function**.

2. SAU: Serial array unit

Figure 2-12. Pin Block Diagram for Pin Type 8-1-2



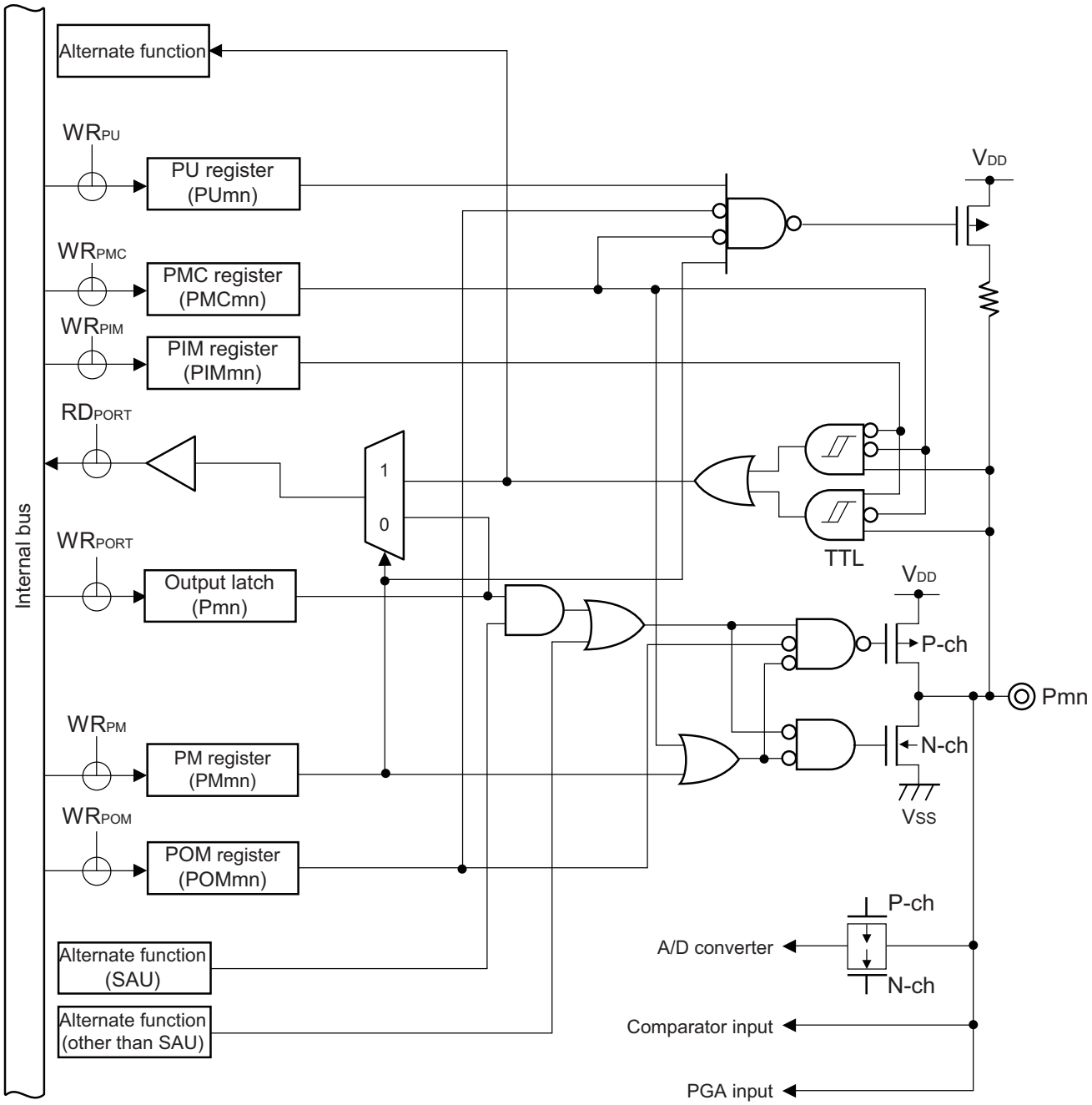
Cautions 1. The input buffer is enabled even if the type 8-1-2 pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMxx). This may lead to a through current flowing through the type 8-1-2 pin when the voltage level on this pin is intermediate. Changing the output level when the N-ch open drain output mode is selected may cause a glitch (V_{DD} level).

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

Remarks 1. For alternate functions, see **2.1 Port Function**.

2. SAU: Serial array unit

Figure 2-13. Pin Block Diagram for Pin Type 8-18-1



Remarks 1. For alternate functions, see **2.1 Port Function**.

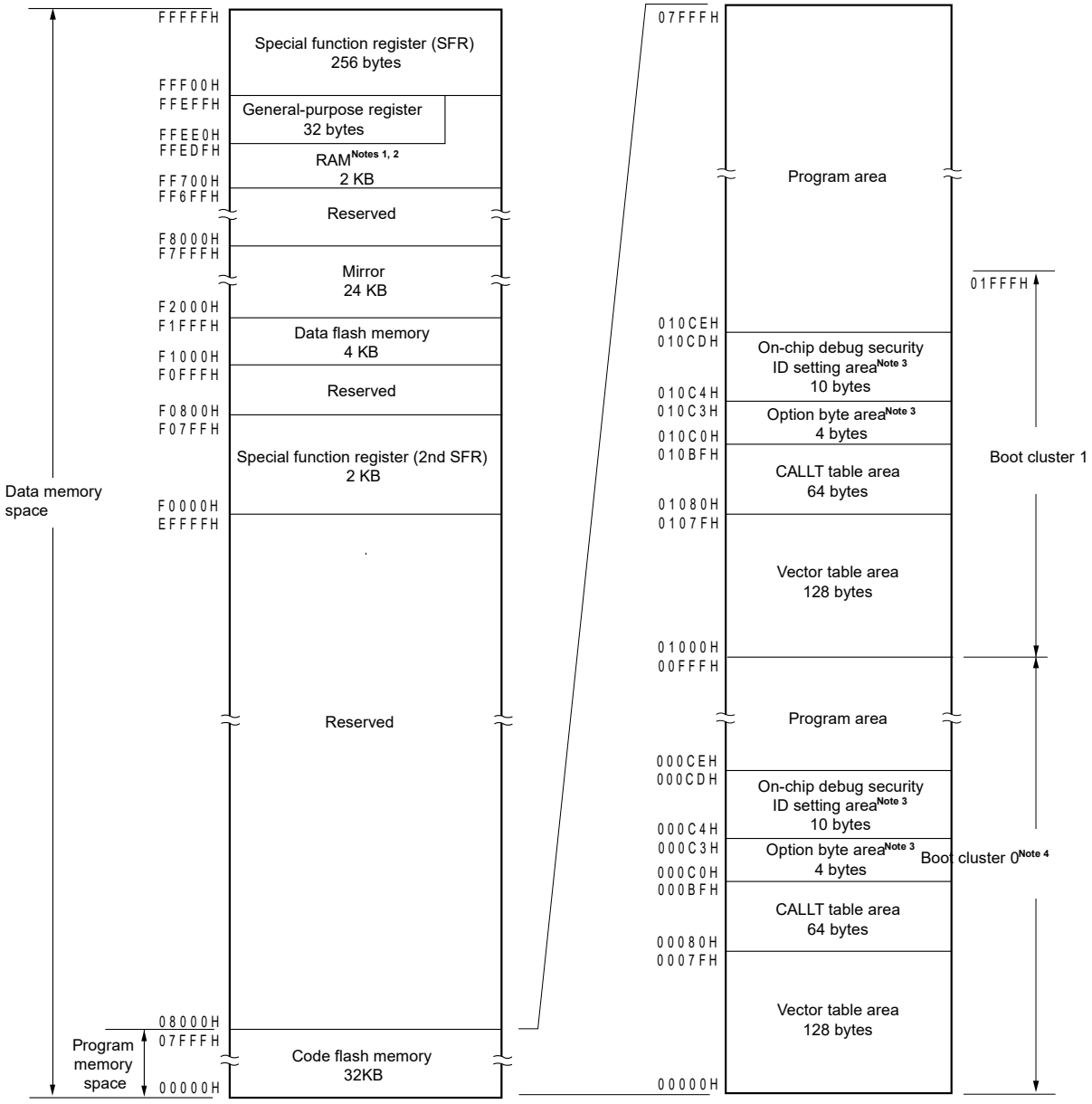
2. SAU: Serial array unit

CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Products in the RL78/I1A can access a 1 MB address space. Figures 3-1 and 3-2 show the memory maps.

Figure 3-1. Memory Map (R5F1076C, R5F107AC)



Notes 1. Do not allocate RAM addresses which are used as a stack area, a data buffer, a branch destination of vector interrupt processing, and a DMA transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

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

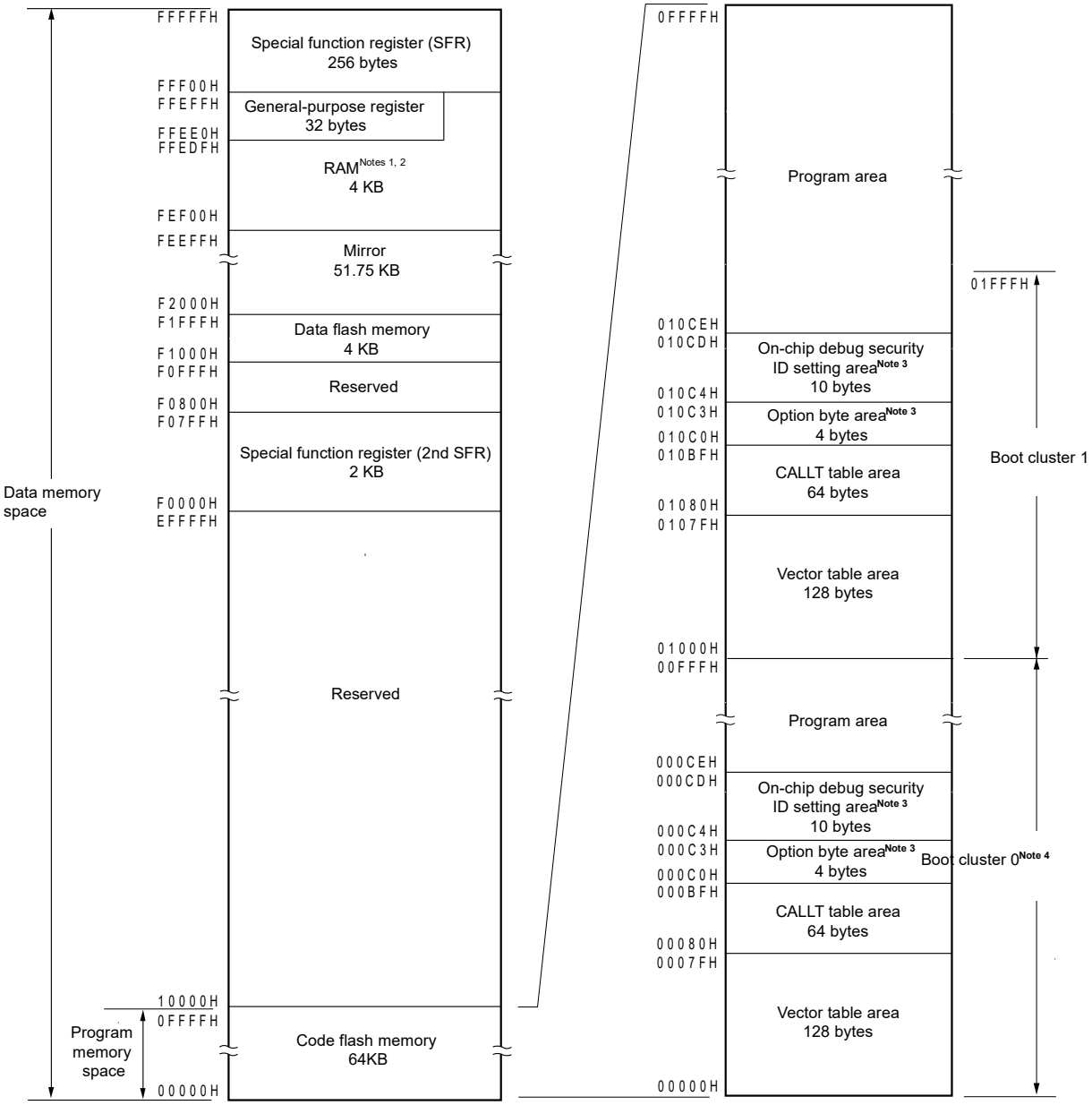
3. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

When boot swap is used: Set the option bytes to 000C0H to 000C3H and 010C0H to 010C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 010C4H to 010CDH.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **28.6 Security Settings**).

Caution While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed and the RAM area + 10 bytes when instructions are fetched from RAM areas, respectively. Reset signal generation sets RAM parity error resets to enabled (RPERDIS = 0). For details, see 25.3.3 RAM parity error detection function.

Figure 3-2. Memory Map (R5F107AE, R5F107DE)



Notes 1. Do not allocate RAM addresses which are used as a stack area, a data buffer, a branch destination of vector interrupt processing, and a DMA transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory. Also, use of the area FEF00H to FF309H is prohibited, because this area is used for each library.

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

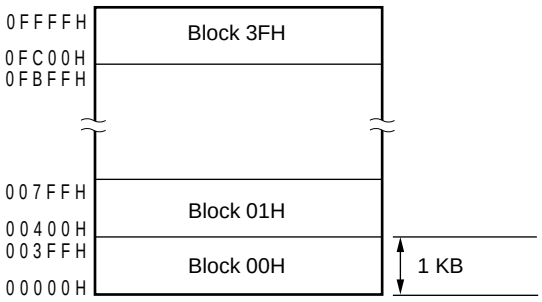
3. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

When boot swap is used: Set the option bytes to 000C0H to 000C3H and 010C0H to 010C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 010C4H to 010CDH.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **28.6 Security Settings**).

Caution While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed and the RAM area + 10 bytes when instructions are fetched from RAM areas, respectively. Reset signal generation sets RAM parity error resets to enabled (RPERDIS = 0). For details, see 25.3.3 RAM parity error detection function.

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



(R5F107AE, R5F107DE)

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

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

Address Value	Block Number	Address Value	Block Number
00000H to 003FFH	00H	08000H to 083FFH	20H
00400H to 007FFH	01H	08400H to 087FFH	21H
00800H to 00BFFH	02H	08800H to 08BFFH	22H
00C00H to 00FFFH	03H	08C00H to 08FFFH	23H
01000H to 013FFH	04H	09000H to 093FFH	24H
01400H to 017FFH	05H	09400H to 097FFH	25H
01800H to 01BFFH	06H	09800H to 09BFFH	26H
01C00H to 01FFFH	07H	09C00H to 09FFFH	27H
02000H to 023FFH	08H	0A000H to 0A3FFH	28H
02400H to 027FFH	09H	0A400H to 0A7FFH	29H
02800H to 02BFFH	0AH	0A800H to 0ABFFH	2AH
02C00H to 02FFFH	0BH	0AC00H to 0AFFFH	2BH
03000H to 033FFH	0CH	0B000H to 0B3FFH	2CH
03400H to 037FFH	0DH	0B400H to 0B7FFH	2DH
03800H to 03BFFH	0EH	0B800H to 0BBFFH	2EH
03C00H to 03FFFH	0FH	0BC00H to 0BFFFH	2FH
04000H to 043FFH	10H	0C000H to 0C3FFH	30H
04400H to 047FFH	11H	0C400H to 0C7FFH	31H
04800H to 04BFFH	12H	0C800H to 0CBFFH	32H
04C00H to 04FFFH	13H	0CC00H to 0CFFFH	33H
05000H to 053FFH	14H	0D000H to 0D3FFH	34H
05400H to 057FFH	15H	0D400H to 0D7FFH	35H
05800H to 05BFFH	16H	0D800H to 0DBFFH	36H
05C00H to 05FFFH	17H	0DC00H to 0DFFFH	37H
06000H to 063FFH	18H	0E000H to 0E3FFH	38H
06400H to 067FFH	19H	0E400H to 0E7FFH	39H
06800H to 06BFFH	1AH	0E800H to 0EBFFH	3AH
06C00H to 06FFFH	1BH	0EC00H to 0EFFFH	3BH
07000H to 073FFH	1CH	0F000H to 0F3FFH	3CH
07400H to 077FFH	1DH	0F400H to 0F7FFH	3DH
07800H to 07BFFH	1EH	0F800H to 0FBFFH	3EH
07C00H to 07FFFH	1FH	0FC00H to 0FFFFH	3FH

Remark R5F1076C, R5F107AC: Block number 00H to 1FH
R5F107AE, R5F107DE: Block number 00H to 3FH

3.1.1 Internal program memory space

The internal program memory space stores the program and table data.

The RL78/I1A products incorporate internal ROM (flash memory), as shown below.

Table 3-2. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
R5F1076C, R5F107AC	Flash memory	32768 × 8 bits (00000H to 07FFFH)
R5F107AE, R5F107DE		65536 × 8 bits (00000H to 0FFFFH)

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 01000H to 0107FH.

Table 3-3. Vector Table (1/2)

Vector Table Address	Interrupt Source	20-pin	30-pin	38-pin
0000H	RESET, POR, LVD, WDT, TRAP, IAW, RPE	√	√	√
0004H	INTWDTI	√	√	√
0006H	INTLVI	√	√	√
0008H	INTP0	√	√	√
000EH	INTP3	–	–	√
0010H	INTP4	–	√	√
001AH	INTDMA0	√	√	√
001CH	INTDMA1	√	√	√
001EH	INTST0	√	√	√
	INTCSI00	–	–	√
0020H	INTSR0	√	√	√
0022H	INTSRE0	√	√	√
	INTTM01H	√	√	√
0024H	INTST1	–	√	√
0026H	INTSR1	–	√	√
0028H	INTSRE1	–	√	√
	INTTM03H	√	√	√
002AH	INTIICA0	√	√	√
002CH	INTTM00	√	√	√
002EH	INTTM01	√	√	√
0030H	INTTM02	√	√	√
0032H	INTTM03	√	√	√
0034H	INTAD	√	√	√
0036H	INTRTC	√	√	√
0038H	INTIT	√	√	√
003CH	INTSTDL4	√	√	√
003EH	INTSRDL4	√	√	√
	INTSREDL4	√	√	√
0040H	INTP20	√	√	√
	INTP22	√	√	√

Table 3-3. Vector Table (2/2)

Vector Table Address	Interrupt Source	20-pin	30-pin	38-pin
0042H	INTTM04	√	√	√
0044H	INTTM05	√	√	√
0046H	INTTM06	√	√	√
0048H	INTTM07	√	√	√
004AH	INTCMP0	√	√	√
004CH	INTCMP1	√	√	√
004EH	INTCMP2	√	√	√
0050H	INTP9	–	–	√
	INTCMP3	√	√	√
0052H	INTP10	–	–	√
	INTCMP4	–	√	√
0054H	INTP11	–	√	√
	INTCMP5	–	√	√
0056H	INTTMKB0	√	√	√
0058H	INTTMKB1	√	√	√
005AH	INTTMKB2	–	√	√
005CH	INTTMKC0	√	√	√
005EH	INTMD	√	√	√
0060H	INTP21	√	√	√
	INTP23	–	√	√
0062H	INTFL	√	√	√
007EH	BRK	√	√	√

(2) CALLT instruction table area

The 64-byte area 00080H to 000BFH can store the subroutine entry address of a 2-byte call instruction (CALLT). Set the subroutine entry address to a value in a range of 00000H to 0FFFFH (because an address code is of 2 bytes).
To use the boot swap function, set a CALLT instruction table also at 01080H to 010BFH.

(3) Option byte area

A 4-byte area of 000C0H to 000C3H can be used as an option byte area. Set the option byte at 010C0H to 010C3H when the boot swap is used. For details, see **CHAPTER 27 OPTION BYTE**.

(4) On-chip debug security ID setting area

A 10-byte area of 000C4H to 000CDH and 010C4H to 010CDH 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 010C4H to 010CDH when the boot swap is used. For details, see **CHAPTER 29 ON-CHIP DEBUG FUNCTION**.

3.1.2 Mirror area

The RL78/I1A mirrors the code flash area of 00000H to 0FFFFH, to F0000H to FFFFFH (the code flash area to be mirrored is set by the processor mode control register (PMC)).

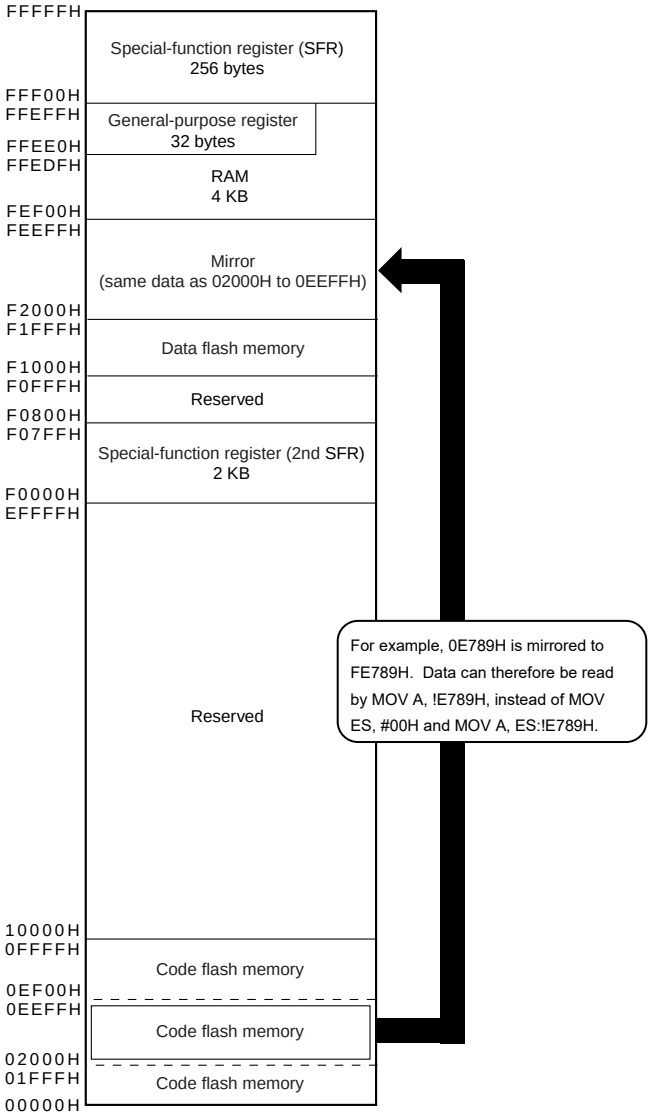
By reading data from F0000H to FFFFFH, an instruction that does not have the ES register as an operand can be used, and thus the contents of the code flash can be read with the shorter code. However, the code flash area is not mirrored to the SFR, extended SFR, RAM, and use prohibited areas.

See **3.1 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 R5F107AE, R5F107DE (Flash memory: 64 KB, RAM: 4 KB)



The PMC register is described as follows.

- **Processor mode control register (PMC)**

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

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

  Reset signal generation sets this register to 00H.

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

Address: FFFFEH After reset: 00H R/W

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

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

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

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

3.1.3 Internal data memory space

The RL78/I1A products incorporate the following RAMs.

Table 3-4. Internal RAM Capacity

Part Number	Internal RAM
R5F1076C, R5F107AC	2048 × 8 bits (FF700H to FFEFFH)
R5F107AE, R5F107DE	4096 × 8 bits (FEF00H to FFEFFH)

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

The internal RAM is used as a stack memory.

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

2. Do not allocate RAM addresses which are used as a stack area, a data buffer, a branch destination of vector interrupt processing, and a DMA transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

3. Use of the RAM areas of the following products is prohibited when performing self-programming and rewriting the data flash memory, because these areas are used for each library.
R5F107AE, R5F107DE: FEF00H to FF309H

3.1.4 Special function register (SFR) area

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

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

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

On-chip peripheral hardware special function registers (2nd SFRs) are allocated in the area F0000H to F07FFH (see **Table 3-6** in **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**).

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

Cautions 1. Do not access addresses to which extended SFRs are not assigned.

2. During access to the registers assigned to the addresses between F0500H and F06FFH in the extended SFR (2nd SFR) area, the CPU does not start the next instruction processing but becomes wait state. If this happens, the number of execution clocks of an instruction increases by the number of wait clocks. See CHAPTER 35 CAUTIONS FOR WAIT for the details.

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/I1A, based on operability and other considerations. In particular, special addressing methods designed for the functions of the special function registers (SFR) and general-purpose registers are available for use. Figures 3-4 and 3-5 show correspondence between data memory and addressing.

For details of each addressing, see **3.4 Addressing for Processing Data Addresses**.

Figure 3-4. Correspondence Between Data Memory and Addressing (R5F1076C, R5F107AC)

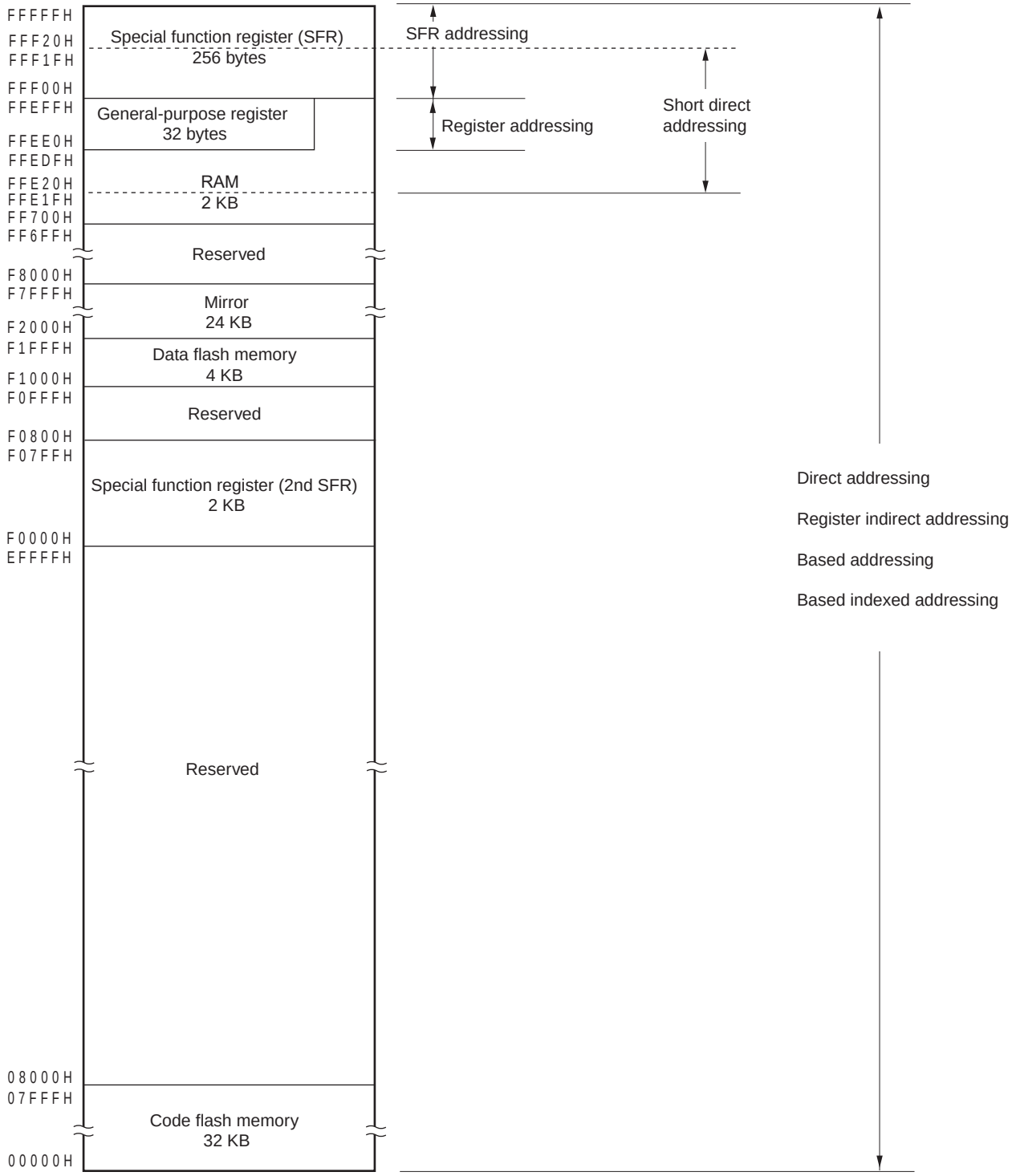
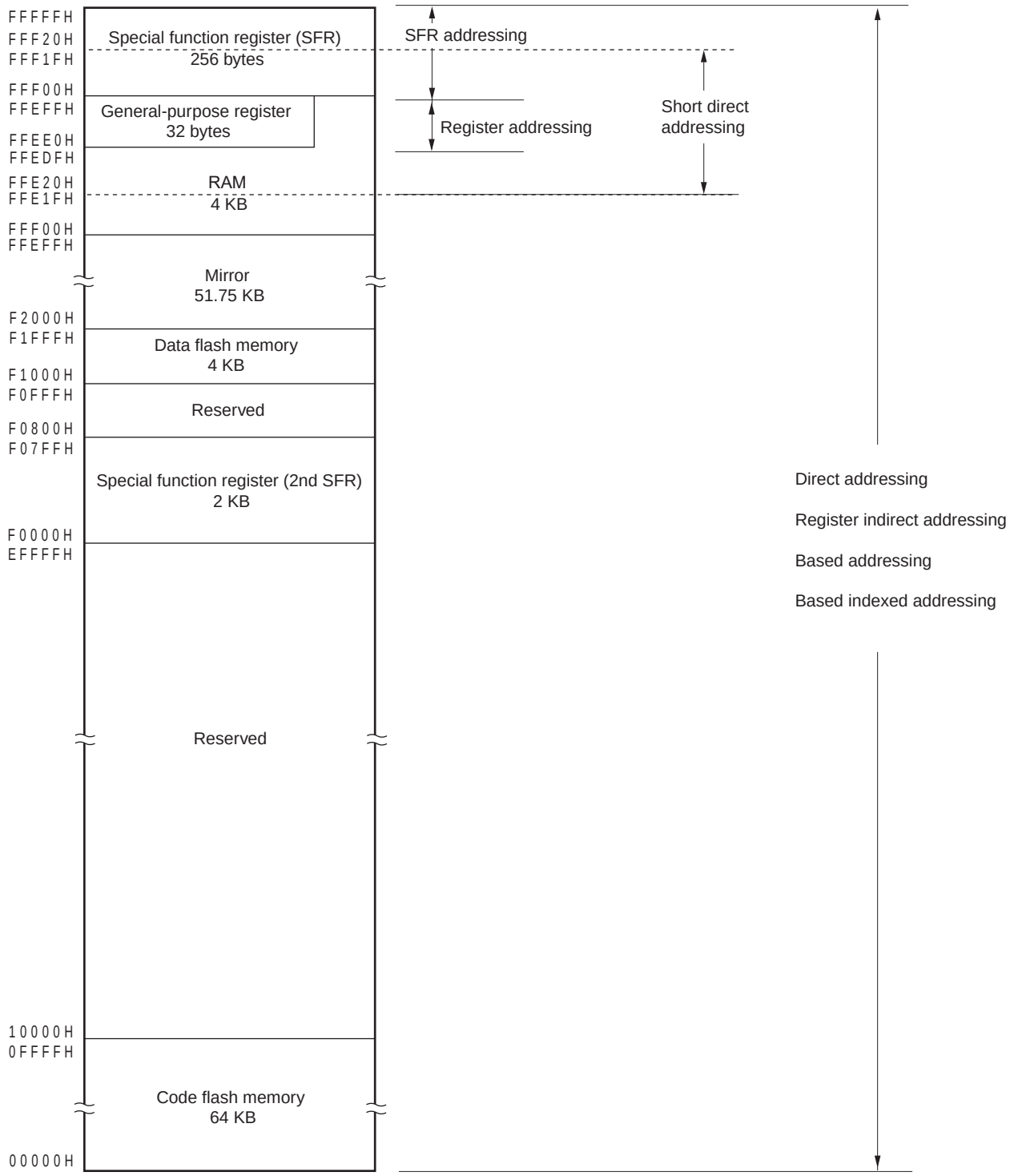


Figure 3-5. Correspondence Between Data Memory and Addressing (R5F107AE, R5F107DE)



3.2 Processor Registers

The RL78/I1A products incorporate the following processor registers.

3.2.1 Control registers

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

(1) Program counter (PC)

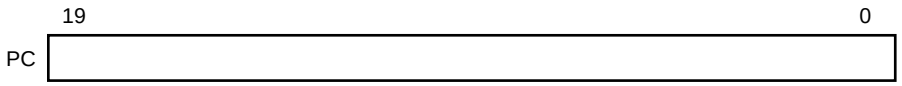
The program counter is a 20-bit register that holds the address information of the next program to be executed.

In normal operation, PC is automatically incremented according to the number of bytes of the instruction to be fetched.

When a branch instruction is executed, immediate data and register contents are set.

Reset signal generation sets the reset vector table values at addresses 0000H and 0001H to the 16 lower-order bits of the program counter. The four higher-order bits of the program counter are cleared to 0000.

Figure 3-6. Format of Program Counter



(2) Program status word (PSW)

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

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

Figure 3-7. Format of Program Status Word

	7							0
PSW	IE	Z	RBS1	AC	RBS0	ISP1	ISP0	CY

(a) Interrupt enable flag (IE)

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

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

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

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

(b) Zero flag (Z)

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

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

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

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

(d) Auxiliary carry flag (AC)

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

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

This flag manages the priority of acknowledgeable maskable vectored interrupts. Vectored interrupt requests specified lower than the value of ISP0 and ISP1 flags by the priority specification flag registers (PRn0L, PRn0H, PRn1L, PRn1H, PRn2L, PRn2H) (see **20.3.3**) can not be acknowledged. Actual vectored request acknowledgment is controlled by the interrupt enable flag (IE).

Remark n = 0, 1

(f) Carry flag (CY)

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

(3) Stack pointer (SP)

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

Figure 3-8. Format of Stack Pointer

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

In stack addressing through a stack pointer, the SP is decremented ahead of write (save) to the stack memory and is incremented after read (restore) from the stack memory.
Each stack operation saves data as shown in Figure 3-9.

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

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

3. Do not allocate RAM addresses which are used as a stack area, a data buffer, a branch destination of vector interrupt processing, and a DMA transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

4. Use of the RAM areas of the following products is prohibited when performing self-programming and rewriting the data flash memory, because these areas are used for each library.
R5F107AE, R5F107DE: FEF00H to FF309H

3.2.2 General-purpose registers

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

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

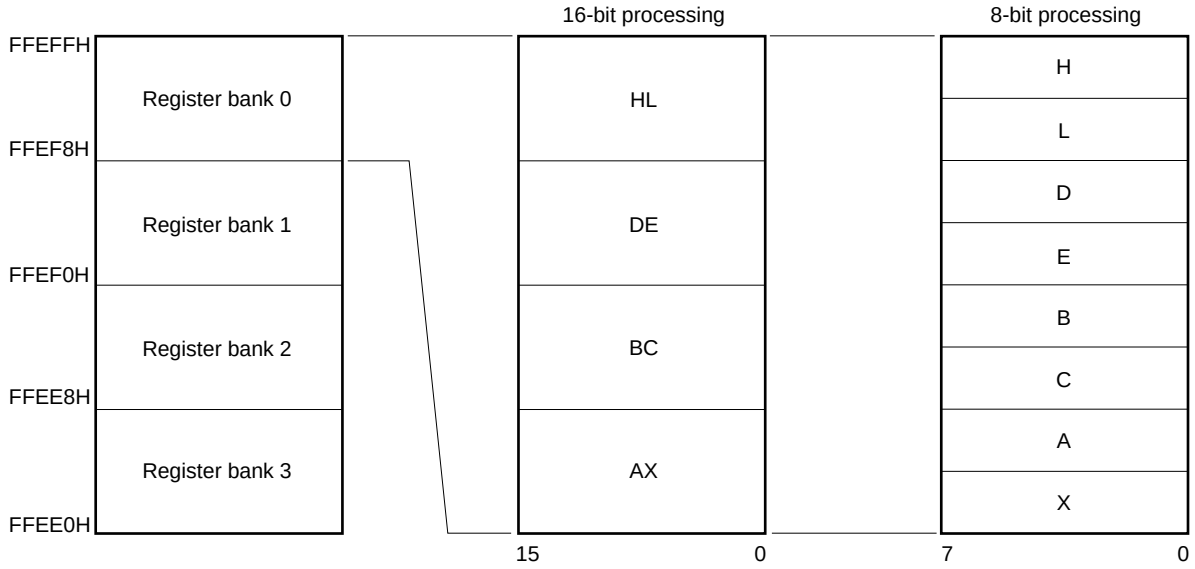
These registers can be described in terms of function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) and absolute names (R0 to R7 and RP0 to RP3).

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

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

Figure 3-9. Configuration of General-Purpose Registers

(a) Function name



3.2.3 ES and CS registers

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

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

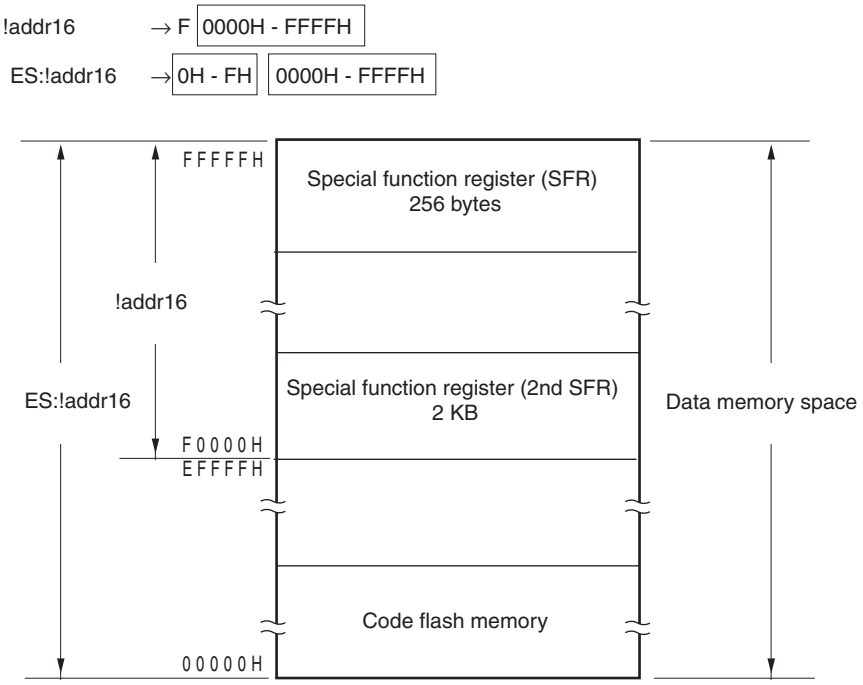
Figure 3-10. Configuration of ES and CS Registers

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

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

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

Figure 3-11. 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 a list of the SFRs. The meanings of items in the table are as follows.

- Symbol
  Symbol indicating 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
  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
  Indicates each register status upon reset signal generation.

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

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

Table 3-5. SFR List (1/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF00H	Port register 0	P0		R/W	√	√	–	00H
FFF01H	Port register 1	P1		R/W	√	√	–	00H
FFF02H	Port register 2	P2		R/W	√	√	–	00H
FFF03H	Port register 3	P3		R/W	√	√	–	00H
FFF04H	Port register 4	P4		R/W	√	√	–	00H
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	SDR01	R/W	–	√	√	0000H
FFF13H		–			–	–		
FFF18H	Timer data register 00	TDR00		R/W	–	–	√	0000H
FFF19H								
FFF1AH	Timer data register 01	TDR01L	TDR01	R/W	–	√	√	00H
FFF1BH		TDR01H			–	√		00H
FFF1EH	10-bit A/D conversion result register	ADCR		R	–	–	√	0000H
FFF1FH	8-bit A/D conversion result register	ADCRH		R	–	√	–	00H
FFF20H	Port mode register 0	PM0		R/W	√	√	–	FFH
FFF21H	Port mode register 1	PM1		R/W	√	√	–	FFH
FFF22H	Port mode register 2	PM2		R/W	√	√	–	FFH
FFF23H	Port mode register 3	PM3		R/W	√	√	–	FFH
FFF24H	Port mode register 4	PM4		R/W	√	√	–	FFH
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
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

Table 3-5. SFR List (2/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF44H	Serial data register 02	TXD1	SDR02	R/W	–	√	√	0000H
FFF45H		–			–	–		
FFF46H	Serial data register 03	RXD1	SDR03	R/W	–	√	√	0000H
FFF47H		–			–	–		
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
FFF64H	Timer data register 02	TDR02		R/W	–	–	√	0000H
FFF65H								
FFF66H	Timer data register 03	TDR03L	TDR03	R/W	–	√	√	00H
FFF67H		TDR03H			–	√		00H
FFF68H	Timer data register 04	TDR04		R/W	–	–	√	0000H
FFF69H								
FFF6AH	Timer data register 05	TDR05		R/W	–	–	√	0000H
FFF6BH								
FFF6CH	Timer data register 06	TDR06		R/W	–	–	√	0000H
FFF6DH								
FFF6EH	Timer data register 07	TDR07		R/W	–	–	√	0000H
FFF6FH								
FFF90H	Interval timer control register	ITMC		R/W	–	–	√	0FFFH
FFF91H								
FFF92H	Second count register	SEC		R/W	–	√	–	00H
FFF93H	Minute count register	MIN		R/W	–	√	–	00H
FFF94H	Hour count register	HOUR		R/W	–	√	–	12H[Note]
FFF95H	Week count register	WEEK		R/W	–	√	–	00H
FFF96H	Day count register	DAY		R/W	–	√	–	01H
FFF97H	Month count register	MONTH		R/W	–	√	–	01H
FFF98H	Year count register	YEAR		R/W	–	√	–	00H
FFF99H	Watch error correction register	SUBCUD		R/W	–	√	–	00H
FFF9AH	Alarm minute register	ALARMWM		R/W	–	√	–	00H
FFF9BH	Alarm hour register	ALARMWH		R/W	–	√	–	12H
FFF9CH	Alarm week register	ALARMWW		R/W	–	√	–	00H
FFF9DH	Real-time clock control register 0	RTCC0		R/W	√	√	–	00H
FFF9EH	Real-time clock control register 1	RTCC1		R/W	√	√	–	00H

Note The value of this register is 00H if the AMPM bit (bit 3 of real-time clock control register 0 (RTCC0)) is set to 1 after reset.

Table 3-5. SFR List (3/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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
FFFA8H	Reset control flag register	RESF		R	–	√	–	Undefined[Note 1]
FFFA9H	Voltage detection register	LVIM		R/W	√	√	–	00H[Note 1]
FFFAAH	Voltage detection level register	LVIS		R/W	√	√	–	00H/01H/81H[Note 1]
FFFABH	Watchdog timer enable register	WDTE		R/W	–	√	–	1AH/9AH[Note 2]
FFFACH	CRC input register	CRCIN		R/W	–	√	–	00H
FFFB0H	DMA SFR address register 0	DSA0		R/W	–	√	–	00H
FFFB1H	DMA SFR address register 1	DSA1		R/W	–	√	–	00H
FFFB2H	DMA RAM address register 0	DRA0L	DRA0	R/W	–	√	√	00H
FFFB3H		DRA0H		R/W	–	√		00H
FFFB4H	DMA RAM address register 1	DRA1L	DRA1	R/W	–	√	√	00H
FFFB5H		DRA1H		R/W	–	√		00H
FFFB6H	DMA byte count register 0	DBC0L	DBC0	R/W	–	√	√	00H
FFFB7H		DBC0H		R/W	–	√		00H
FFFB8H	DMA byte count register 1	DBC1L	DBC1	R/W	–	√	√	00H
FFFB9H		DBC1H		R/W	–	√		00H
FFFBAH	DMA mode control register 0	DMC0		R/W	√	√	–	00H
FFFBBH	DMA mode control register 1	DMC1		R/W	√	√	–	00H
FFFBCH	DMA operation control register 0	DRC0		R/W	√	√	–	00H
FFFBDH	DMA operation control register 1	DRC1		R/W	√	√	–	00H

Notes 1. These values vary depending on the reset source.

Register \ Reset Source		$\overline{\text{RESET}}$ Input	Reset by POR	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by RAM Parity Error	Reset by Illegal-memory Access	Reset by LVD
RESF	TRAP	Cleared (0)		Set (1)	Held			Held
	WDTRF			Held	Set (1)	Held		
	RPERF			Held		Set (1)	Held	
	IAWRF			Held			Set (1)	
	LVIRF			Held				Set (1)
LVIM	LVISEN	Cleared (0)						Held
	LVIOMSK	Held						
	LVIF							
LVIS		Cleared (00H/01H/81H)						

2. The reset values of the WDTE register is determined by the setting of the option byte.

Table 3-5. SFR List (4/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFFD0H	Interrupt request flag register 2	IF2L	IF2	R/W	√	√	√	00H
FFFD1H		IF2H		R/W	√	√		00H
FFFD4H	Interrupt mask flag register 2	MK2L	MK2	R/W	√	√	√	FFH
FFFD5H		MK2H		R/W	√	√		FFH
FFFD8H	Priority specification flag register 02	PR02L	PR02	R/W	√	√	√	FFH
FFFD9H		PR02H		R/W	√	√		FFH
FFFDCH	Priority specification flag register 12	PR12L	PR12	R/W	√	√	√	FFH
FFFDDH		PR12H		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/division data register A (L)	MDAL		R/W	–	–	√	0000H
FFFF1H								
FFFF2H	Multiplication/division data register A (H)	MDAH		R/W	–	–	√	0000H
FFFF3H								
FFFF4H	Multiplication/division data register B (H)	MDBH		R/W	–	–	√	0000H
FFFF5H								
FFFF6H	Multiplication/division data register B (L)	MDBL		R/W	–	–	√	0000H
FFFF7H								
FFFFEH	Processor mode control register	PMC		R/W	√	√	–	00H

Remark For extended SFRs (2nd SFRs), see **Table 3-6 Extended SFR (2nd SFR) List**.

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

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

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

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

Each manipulation bit unit can be specified as follows.

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

Table 3-6 gives a list of the extended SFRs. The meanings of items in the table are as follows.

- Symbol
  Symbol indicating 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
  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
  Indicates each register status upon reset signal generation.

Cautions 1. Do not access addresses to which extended SFRs (2nd SFRs) are not assigned.

2. During access to the registers assigned to the addresses between F0500H and F06FFH in the extended SFR (2nd SFR) area, the CPU does not start the next instruction processing but becomes wait state. If this happens, the number of execution clocks of an instruction increases by the number of wait clocks. See CHAPTER 35 CAUTIONS FOR WAIT for the details.

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

Table 3-6. Extended SFR (2nd SFR) List (1/12)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0010H	A/D converter mode register 2	ADM2	R/W	√	√	–	00H
F0011H	Conversion result comparison upper limit setting register	ADUL	R/W	–	√	–	FFH
F0012H	Conversion result comparison lower limit setting register	ADLL	R/W	–	√	–	00H
F0013H	A/D test register	ADTES	R/W	–	√	–	00H
F0030H	Pull-up resistor option register 0	PU0	R/W	√	√	–	00H
F0031H	Pull-up resistor option register 1	PU1	R/W	√	√	–	00H
F0033H	Pull-up resistor option register 3	PU3	R/W	√	√	–	00H
F0034H	Pull-up resistor option register 4	PU4	R/W	√	√	–	01H
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
F0050H	Port output mode register 0	POM0	R/W	√	√	–	00H
F0051H	Port output mode register 1	POM1	R/W	√	√	–	00H
F0060H	Port mode control register 0	PMC0	R/W	√	√	–	FFH
F006CH	Port mode control register 12	PMC12	R/W	√	√	–	FFH
F006EH	Port mode control register 14	PMC14	R/W	√	√	–	FFH
F0070H	Noise filter enable register 0	NFEN0	R/W	√	√	–	00H
F0071H	Noise filter enable register 1	NFEN1	R/W	√	√	–	00H
F0073H	Input switch control register	ISC	R/W	√	√	–	00H
F0074H	Timer input select register 0	TIS0	R/W	–	√	–	00H
F0076H	A/D port configuration register	ADPC	R/W	–	√	–	00H
F0078H	Invalid memory access detection control register 0	IAWCTL0	R/W	–	√	–	00H
F0090H	Data flash control register	DFLCTL	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]

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

2. The value after a reset is a value set in FRQSEL2 to FRQSEL0 of the option byte (000C2H).

Table 3-6. Extended SFR (2nd SFR) List (2/12)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F00E0H	Multiplication/division data register C (L)	MDCL		R/W	–	–	√	0000H
F00E2H	Multiplication/division data register C (H)	MDCH		R/W	–	–	√	0000H
F00E8H	Multiplication/division control register	MDUC		R/W	√	√	–	00H
F00F0H	Peripheral enable register 0	PER0		R/W	√	√	–	00H
F00F3H	Subsystem clock supply mode control register	OSMC		R/W	–	√	–	00H
F00F5H	RAM parity error control register	RPECTL		R/W	√	√	–	00H
F00FEH	BCD adjust result register	BCDADJ		R	–	√	–	Undefined
F0100H	Serial status register 00	SSR00L	SSR00	R	–	√	√	0000H
F0101H		–			–	–		
F0102H	Serial status register 01	SSR01L	SSR01	R	–	√	√	0000H
F0103H		–			–	–		
F0104H	Serial status register 02	SSR02L	SSR02	R	–	√	√	0000H
F0105H		–			–	–		
F0106H	Serial status register 03	SSR03L	SSR03	R	–	√	√	0000H
F0107H		–			–	–		
F0108H	Serial flag clear trigger register 00	SIR00L	SIR00	R/W	–	√	√	0000H
F0109H		–			–	–		
F010AH	Serial flag clear trigger register 01	SIR01L	SIR01	R/W	–	√	√	0000H
F010BH		–			–	–		
F010CH	Serial flag clear trigger register 02	SIR02L	SIR02	R/W	–	√	√	0000H
F010DH		–			–	–		
F010EH	Serial flag clear trigger register 03	SIR03L	SIR03	R/W	–	√	√	0000H
F010FH		–			–	–		
F0110H	Serial mode register 00	SMR00		R/W	–	–	√	0020H
F0111H								
F0112H	Serial mode register 01	SMR01		R/W	–	–	√	0020H
F0113H								
F0114H	Serial mode register 02	SMR02		R/W	–	–	√	0020H
F0115H								
F0116H	Serial mode register 03	SMR03		R/W	–	–	√	0020H
F0117H								
F0118H	Serial communication operation setting register 00	SCR00		R/W	–	–	√	0087H
F0119H								
F011AH	Serial communication operation setting register 01	SCR01		R/W	–	–	√	0087H
F011BH								
F011CH	Serial communication operation setting register 02	SCR02		R/W	–	–	√	0087H
F011DH								
F011EH	Serial communication operation setting register 03	SCR03		R/W	–	–	√	0087H
F011FH								

Table 3-6. Extended SFR (2nd SFR) List (3/12)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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
		–			–	–		
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								
F0188H	Timer counter register 04	TCR04		R	–	–	√	FFFFH
F0189H								
F018AH	Timer counter register 05	TCR05		R	–	–	√	FFFFH
F018BH								
F018CH	Timer counter register 06	TCR06		R	–	–	√	FFFFH
F018DH								
F018EH	Timer counter register 07	TCR07		R	–	–	√	FFFFH
F018FH								
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								

Table 3-6. Extended SFR (2nd SFR) List (4/12)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0196H	Timer mode register 03	TMR03		R/W	–	–	√	0000H
F0197H								
F0198H	Timer mode register 04	TMR04		R/W	–	–	√	0000H
F0199H								
F019AH	Timer mode register 05	TMR05		R/W	–	–	√	0000H
F019BH								
F019CH	Timer mode register 06	TMR06		R/W	–	–	√	0000H
F019DH								
F019EH	Timer mode register 07	TMR07		R/W	–	–	√	0000H
F019FH								
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		–			–	–		
F01A6H	Timer status register 03	TSR03L	TSR03	R	–	√	√	0000H
F01A7H		–			–	–		
F01A8H	Timer status register 04	TSR04L	TSR04	R	–	√	√	0000H
F01A9H		–			–	–		
F01AAH	Timer status register 05	TSR05L	TSR05	R	–	√	√	0000H
F01ABH		–			–	–		
F01ACH	Timer status register 06	TSR06L	TSR06	R	–	√	√	0000H
F01ADH		–			–	–		
F01AEH	Timer status register 07	TSR07L	TSR07	R	–	√	√	0000H
F01AFH		–			–	–		
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		–			–	–		

Table 3-6. Extended SFR (2nd SFR) List (5/12)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
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
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
F0500H	Port register 20	P20	R/W	√	√	–	00H
F0508H	Peripheral enable register 1	PER1	R/W	√	√	–	00H
F0509H	Peripheral enable register 2	PER2	R/W	√	√	–	00H
F050AH	PLL control register	PLLCTL	R/W	√	√	–	00H
F0510H	Port mode register 20	PM20	R/W	√	√	–	FFH
F0518H	External interrupt rising edge enable register 2	EGP2	R/W	√	√	–	00H
F0519H	External interrupt falling edge enable register 2	EGN2	R/W	√	√	–	00H
F0520H	Pull-up resistor option register 20	PU20	R/W	√	√	–	00H
F0530H	Port output mode register 20	POM20	R/W	√	√	–	00H
F0550H	Programmable gain amplifier control register	PGACTL	R/W	√	√	–	00H
F0551H	Programmable gain amplifier input channel select register	PGAINS	R/W	–	√	–	00H
F0552H	Comparator 0 control register	C0CTL	R/W	√	√	–	00H
F0553H	Comparator 1 control register	C1CTL	R/W	√	√	–	00H
F0554H	Comparator 2 control register	C2CTL	R/W	√	√	–	00H
F0555H	Comparator 3 control register	C3CTL	R/W	√	√	–	00H
F0556H	Comparator 4 control register	C4CTL	R/W	√	√	–	00H
F0557H	Comparator 5 control register	C5CTL	R/W	√	√	–	00H
F0558H	Comparator rising edge enable register	CMPEGP0	R/W	√	√	–	00H
F0559H	Comparator falling edge enable register	CMPEGN0	R/W	√	√	–	00H
F055AH	Comparator output monitor register	CMPMON	R	√	√	–	00H
F055BH	External interrupt control register	INTPCTL	R/W	√	√	–	00H
F0560H	Comparator and PGA internal reference voltage control register	CVRCTL	R/W	√	√	–	00H
F0561H	Comparator internal reference voltage select register 0	C0RVM	R/W	–	√	–	00H
F0562H	Comparator internal reference voltage select register 1	C1RVM	R/W	–	√	–	00H
F0563H	Comparator internal reference voltage select register 2	C2RVM	R/W	–	√	–	00H

Table 3-6. Extended SFR (2nd SFR) List (6/12)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0564H	Window comparator function setting register	CMPWDC		R/W	√	√	–	00H
F0565H	Comparator input switch control register	CMPSEL		R/W	–	√	–	00H
F0570H	Serial data register 40	TXD4	SDR40	R/W	–	√	√	0000H
F0571H		–			–	–		
F0572H	Serial data register 41	RXD4	SDR41	R/W	–	√	√	0000H
F0573H		–			–	–		
F0578H	DALI transmit data register L4	SDTL4		R/W	–	–	√	0000H
F0579H								
F057AH	DALI transmit data register H4	SDTH4		R/W	–	–	√	0000H
F057BH								
F057CH	DALI receive data register L4	SDCL4		R	–	–	√	0000H
F057DH								
F057EH	DALI receive data register H4	SDCH4		R	–	–	√	0000H
F057FH								
F0580H	Serial status register 40	SSR40L	SSR40	R	–	√	√	0000H
F0581H		–			–	–		
F0582H	Serial status register 41	SSR40L	SSR41	R	–	√	√	0000H
F0583H		–			–	–		
F0588H	Serial flag clear trigger register 40	SIR40L	SIR40	R/W	–	√	√	0000H
F0589H		–			–	–		
F058AH	Serial flag clear trigger register 41	SIR41L	SIR41	R/W	–	√	√	0000H
F058BH		–			–	–		
F0590H	Serial mode register 40	SMR40		R/W	–	–	√	0020H
F0591H								
F0592H	Serial mode register 41	SMR41		R/W	–	–	√	0087H
F0593H								
F0598H	Serial communication operation setting register 40	SCR40		R/W	–	–	√	0087H
F0599H								
F059AH	Serial communication operation setting register 41	SCR41		R/W	–	–	√	0020H
F059BH								
F05A0H	Serial channel enable status register 4	SE4L	SE4	R	√	√	√	0000H
F05A1H		–			–	–		
F05A2H	Serial channel start register 4	SS4L	SS4	R/W	√	√	√	0000H
F05A3H		–			–	–		
F05A4H	Serial channel stop register 4	ST4L	ST4	R/W	√	√	√	0000H
F05A5H		–			–	–		
F05A6H	Serial clock select register 4	SPS4L	SPS4	R/W	–	√	√	0000H
F05A7H		–			–	–		
F05A8H	Serial output register 40	SO4		R/W	–	–	√	0F0FH
F05A9H								

Table 3-6. Extended SFR (2nd SFR) List (7/12)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F05AAH	Serial output enable register 4	SOE4L	SOE4	R/W	√	√	√	0000H
F05ABH		–			–	–		
F05B4H	Serial output level register 4	SOL4L	SOL4	R/W	–	√	√	0000H
F05B5H		–			–	–		
F05B8H	Serial standby control register 4	SSC4L	SSC4	R/W	–	√	√	0000H
F05B9H		–			–	–		
F05BAH	Serial option control register 4	SOC4		R/W	–	–	√	0000H
F05BBH								
F05C0H	Peripheral I/O redirection register 1	PIOR1		R/W	–	√	–	00H
F05C1H	Noise filter enable register 3	NFEN3		R/W	√	√	–	00H
F05C2H	Interrupt mask flag register 0	INTMK0		R/W	√	√	–	FFH
F05C3H	Interrupt monitor flag register 0	INTMF0		R/W[Note]	√	√	–	00H
F05C4H	Invalid memory access detection control register 1	IAWCTL1		R/W	–	√	–	00H
F05C5H	Timer clock select register 2	TPS2		R/W	–	√	–	00H
F05C6H	Peripheral function switch register 0	PFSEL0		R/W	√	√	–	00H
F05C7H	Single-wire UART control register	SUCTL		R/W	√	√	–	00H
F05C8H	16-bit timer KC output pin control register	TOETKC0		R/W	√	√	–	00H
F0600H	16-bit timer KB compare register 00	TKBCR00		R/W	–	–	√	0000H
F0601H								
F0602H	16-bit timer KB compare register 01	TKBCR01		R/W	–	–	√	0000H
F0603H								
F0604H	16-bit timer KB compare register 02	TKBCR02		R/W	–	–	√	0000H
F0605H								
F0606H	16-bit timer KB compare register 03	TKBCR03		R/W	–	–	√	0000H
F0607H								
F0608H	16-bit timer KB trigger compare register 0	TKBTGCR0		R/W	–	–	√	0000H
F0609H								
F060AH	16-bit timer KB smooth start initial duty register 00	TKBSIR00		R/W	–	–	√	0000H
F060BH								
F060CH	16-bit timer KB smooth start initial duty register 01	TKBSIR01		R/W	–	–	√	0000H
F060DH								
F060EH	16-bit timer KB dithering count register 00	TKBDNR00		R/W	–	√	–	00H
F060FH	16-bit timer KB smooth start step width register 00	TKBSSR00		R/W	–	√	–	00H

Note An 8-bit memory manipulation instruction can only be read.

Table 3-6. Extended SFR (2nd SFR) List (8/12)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0610H	16-bit timer KB dithering count register 01	TKBDNR01	R/W	–	√	–	00H
F0611H	16-bit timer KB smooth start step width register 01	TKBSSR01	R/W	–	√	–	00H
F0612H	16-bit timer KB trigger register 0	TKBTRG0	W	√	√	–	00H
F0613H	16-bit timer KB flag register 0	TKBFLG0	R	√	√	–	00H
F0614H	16-bit timer KB compare 1L & dithering count register 00	TKBCRLD00	R/W	–	–	√	0000H
F0615H							
F0616H	16-bit timer KB compare 3L & dithering count register 01	TKBCRLD01	R/W	–	–	√	0000H
F0617H							
F0620H	16-bit timer counter KB0	TKBCNT0	R	–	–	√	FFFFH
F0621H							
F0622H	16-bit timer KB operation control register 00	TKBCTL00	R/W	–	–	√	0000H
F0623H							
F0624H	16-bit timer KB maximum frequency limit setting register 0	TKBMFR0	R/W	–	–	√	0000H
F0625H							
F0626H	16-bit timer KB output control register 00	TKBIOC00	R/W	√	√	–	00H
F0627H	16-bit timer KB flag clear trigger register 0	TKBCLR0	W	√	√	–	00H
F0628H	16-bit timer KB output control register 01	TKBIOC01	R/W	√	√	–	00H
F0629H	16-bit timer KB operation control register 01	TKBCTL01	R/W	√	√	–	00H
F0630H	Forced output stop function control register 00	TKBPACTL00	R/W	–	–	√	0000H
F0631H							
F0632H	Forced output stop function control register 01	TKBPACTL01	R/W	–	–	√	0000H
F0633H							
F0634H	Forced output stop function start trigger register 0	TKBPAHFS0	W	√	√	–	00H
F0635H	Forced output stop function stop trigger register 0	TKBPAHFT0	W	√	√	–	00H
F0636H	Forced output stop function flag register 0	TKBPAFLG0	R	√	√	–	00H
F0637H	Forced output stop function control register 02	TKBPACTL02	R/W	√	√	–	00H

Table 3-6. Extended SFR (2nd SFR) List (9/12)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0640H	16-bit timer KB compare register 10	TKBCR10	R/W	–	–	√	0000H
F0641H							
F0642H	16-bit timer KB compare register 11	TKBCR11	R/W	–	–	√	0000H
F0643H							
F0644H	16-bit timer KB compare register 12	TKBCR12	R/W	–	–	√	0000H
F0645H							
F0646H	16-bit timer KB compare register 13	TKBCR13	R/W	–	–	√	0000H
F0647H							
F0648H	16-bit timer KB trigger compare register 1	TKBTGCR1	R/W	–	–	√	0000H
F0649H							
F064AH	16-bit timer KB smooth start initial duty register 10	TKBSIR10	R/W	–	–	√	0000H
F064BH							
F064CH	16-bit timer KB smooth start initial duty register 11	TKBSIR11	R/W	–	–	√	0000H
F064DH							
F064EH	16-bit timer KB dithering count register 10	TKBDNR10	R/W	–	√	–	00H
F064FH	16-bit timer KB smooth start step width register 10	TKBSSR10	R/W	–	√	–	00H
F0650H	16-bit timer KB dithering count register 11	TKBDNR11	R/W	–	√	–	00H
F0651H	16-bit timer KB smooth start step width register 11	TKBSSR11	R/W	–	√	–	00H
F0652H	16-bit timer KB trigger register 1	TKBTRG1	W	√	√	–	00H
F0653H	16-bit timer KB flag register 1	TKBFLG1	R	√	√	–	00H
F0654H	16-bit timer KB compare 1L & dithering count register 10	TKBCRLD10	R/W	–	–	√	0000H
F0655H							
F0656H	16-bit timer KB compare 1L & dithering count register 11	TKBCRLD11	R/W	–	–	√	0000H
F0657H							
F0660H	16-bit timer counter KB1	TKBCNT1	R	–	–	√	FFFFH
F0661H							
F0662H	16-bit timer KB operation control register 10	TKBCTL10	R/W	–	–	√	0000H
F0663H							
F0664H	16-bit timer KB maximum frequency limit setting register 1	TKBMFR1	R/W	–	–	√	0000H
F0665H							
F0666H	16-bit timer KB output control register 10	TKBIOC10	R/W	√	√	–	00H
F0667H	16-bit timer KB flag clear trigger register 1	TKBCLR1	W	√	√	–	00H

Table 3-6. Extended SFR (2nd SFR) List (10/12)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0668H	16-bit timer KB output control register 11	TKBIOC11	R/W	√	√	–	00H
F0669H	16-bit timer KB operation control register 11	TKBCTL11	R/W	√	√	–	00H
F0670H	Forced output stop function control register 10	TKBPACTL10	R/W	–	–	√	0000H
F0671H							
F0672H	Forced output stop function control register 11	TKBPACTL11	R/W	–	–	√	0000H
F0673H							
F0674H	Forced output stop function start trigger register 1	TKBPAHFS1	W	√	√	–	00H
F0675H	Forced output stop function stop trigger register 1	TKBPAHFT1	W	√	√	–	00H
F0676H	Forced output stop function flag register 1	TKBPAFLG1	R	√	√	–	00H
F0677H	Forced output stop function control register 12	TKBPACTL12	R/W	√	√	–	00H
F0680H	16-bit timer KB compare register 20	TKBCR20	R/W	–	–	√	0000H
F0681H							
F0682H	16-bit timer KB compare register 21	TKBCR21	R/W	–	–	√	0000H
F0683H							
F0684H	16-bit timer KB compare register 22	TKBCR22	R/W	–	–	√	0000H
F0685H							
F0686H	16-bit timer KB compare register 23	TKBCR23	R/W	–	–	√	0000H
F0687H							
F0688H	16-bit timer KB trigger compare register 2	TKBTGCR2	R/W	–	–	√	0000H
F0689H							
F068AH	16-bit timer KB smooth start initial duty register 20	TKBSIR20	R/W	–	–	√	0000H
F068BH							
F068CH	16-bit timer KB smooth start initial duty register 21	TKBSIR21	R/W	–	–	√	0000H
F068DH							
F068EH	16-bit timer KB dithering count register 20	TKBDNR20	R/W	–	√	–	00H
F068FH	16-bit timer KB smooth start step width register 20	TKBSSR20	R/W	–	√	–	00H
F0690H	16-bit timer KB dithering count register 21	TKBDNR21	R/W	–	√	–	00H
F0691H	16-bit timer KB smooth start step width register 21	TKBSSR21	R/W	–	√	–	00H

Table 3-6. Extended SFR (2nd SFR) List (11/12)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0692H	16-bit timer KB trigger register 2	TKBTRG2	W	√	√	–	00H
F0693H	16-bit timer KB flag register 2	TKBFLG2	R	√	√	–	00H
F0694H F0695H	16-bit timer KB compare 1L & dithering count register 20	TKBCRLD20	R/W	–	–	√	0000H
F0696H F0697H	16-bit timer KB compare 1L & dithering count register 21	TKBCRLD21	R/W	–	–	√	0000H
F06A0H F06A1H	16-bit timer counter KB2	TKBCNT2	R	–	–	√	FFFFH
F06A2H F06A3H	16-bit timer KB operation control register 20	TKBCTL20	R/W	–	–	√	0000H
F06A4H F06A5H	16-bit timer KB maximum frequency limit setting register 2	TKBMFR2	R/W	–	–	√	0000H
F06A6H	16-bit timer KB output control register 20	TKBIOC20	R/W	√	√	–	00H
F06A7H	16-bit timer KB flag clear trigger register 2	TKBCLR2	W	√	√	–	00H
F06A8H	16-bit timer KB output control register 21	TKBIOC21	R/W	√	√	–	00H
F06A9H	16-bit timer KB operation control register 21	TKBCTL21	R/W	√	√	–	00H
F06B0H F06B1H	Forced output stop function control register 20	TKBPACTL20	R/W	–	–	√	0000H
F06B2H F06B3H	Forced output stop function control register 21	TKBPACTL21	R/W	–	–	√	0000H
F06B4H	Forced output stop function start trigger register 2	TKBPAHFS2	W	√	√	–	00H
F06B5H	Forced output stop function stop trigger register 2	TKBPAHFT2	W	√	√	–	00H
F06B6H	Forced output stop function flag register 2	TKBPAFLG2	R	√	√	–	00H
F06B7H	Forced output stop function control register 22	TKBPACTL22	R/W	√	√	–	00H

Table 3-6. Extended SFR (2nd SFR) List (12/12)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F06D0H	16-bit timer KC compare register 0	TKCCR0	R/W	–	–	√	0000H
F06D2H	16-bit timer KC duty compare register 00	TKCDUTY00	R/W	–	–	√	0000H
F06D4H	16-bit timer KC duty compare register 01	TKCDUTY01	R/W	–	–	√	0000H
F06D6H	16-bit timer KC duty compare register 02	TKCDUTY02	R/W	–	–	√	0000H
F06D8H	16-bit timer KC duty compare register 03	TKCDUTY03	R/W	–	–	√	0000H
F06DAH	16-bit timer KC duty compare register 04	TKCDUTY04	R/W	–	–	√	0000H
F06DCH	16-bit timer KC duty compare register 05	TKCDUTY05	R/W	–	–	√	0000H
F06DEH	16-bit timer KC trigger register 0	TKCTRG0	W	√	√	–	00H
F06DFH	16-bit timer KC flag register 0	TKCFLG0	R	√	√	–	00H
F06E2H	16-bit timer KC output control register 00	TKCIOC00	R/W	–	–	√	0000H
F06E4H	16-bit timer KC output control register 01	TKCIOC01	R/W	√	√	–	00H
F06E5H	16-bit timer KC operation control register 0	TKCCTL0	R/W	√	√	–	00H
F06E6H	16-bit timer KC output flag register 0	TKCTOF0	R	√	√	–	00H
F06F0H	16-bit timer counter KC0	TKCCNT0	R	–	–	√	FFFFH

Remark For SFRs in SFR area, see **Table 3-5 SFR List**.

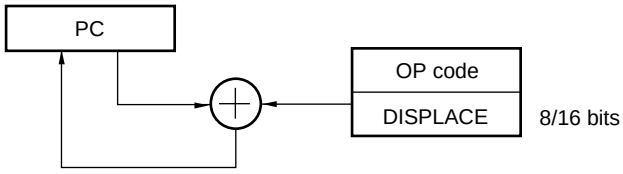
3.3 Instruction Address Addressing

3.3.1 Relative addressing

[Function]

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

Figure 3-12. 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 3-13. Example of CALL !!addr20/BR !!addr20

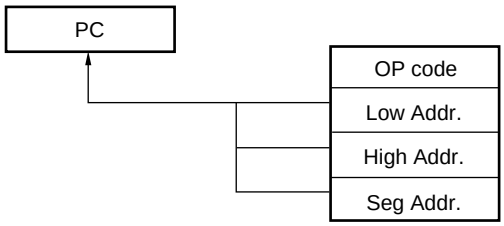
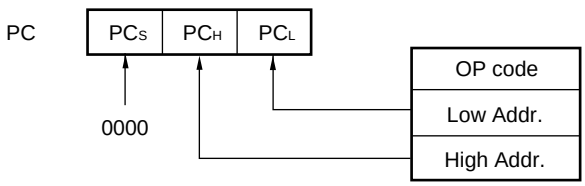


Figure 3-14. Example of CALL !addr16/BR !addr16



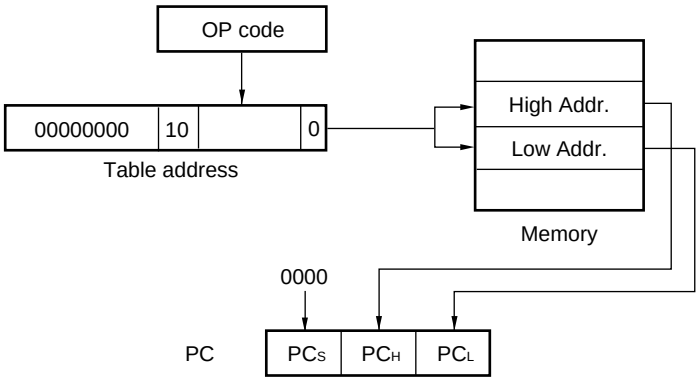
3.3.3 Table indirect addressing

[Function]

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

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

Figure 3-15. Outline of Table Indirect Addressing

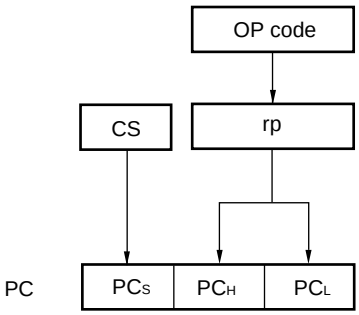


3.3.4 Register indirect addressing

[Function]

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

Figure 3-16. Outline of Register Indirect Addressing



3.4 Addressing for Processing Data Addresses

3.4.1 Implied addressing

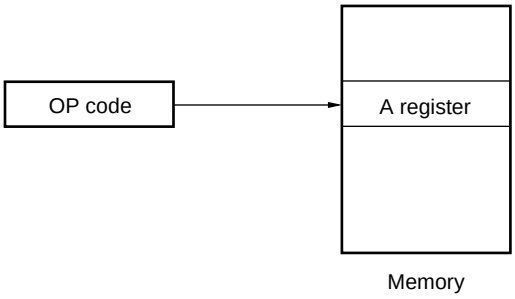
[Function]

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 3-17. Outline of Implied Addressing



3.4.2 Register addressing

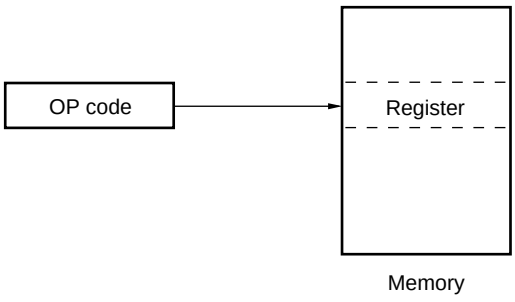
[Function]

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 3-18. Outline of Register Addressing



3.4.3 Direct addressing

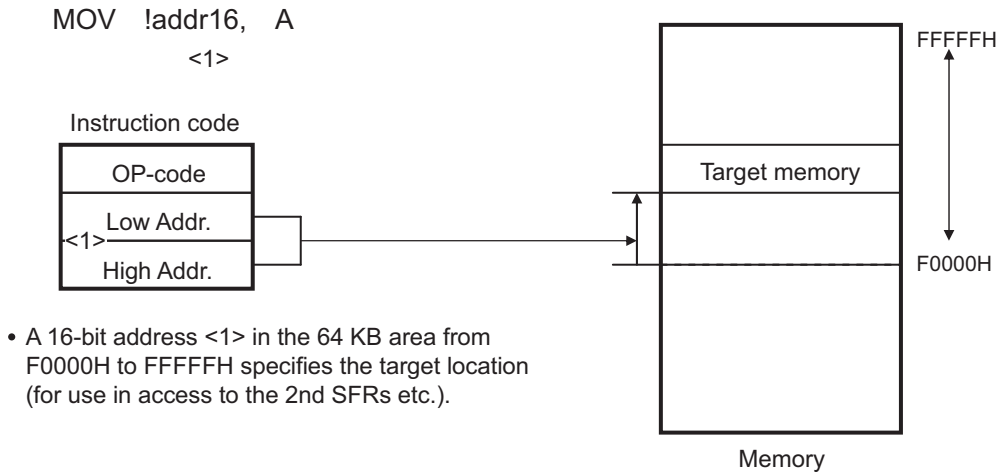
[Function]

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

[Operand format]

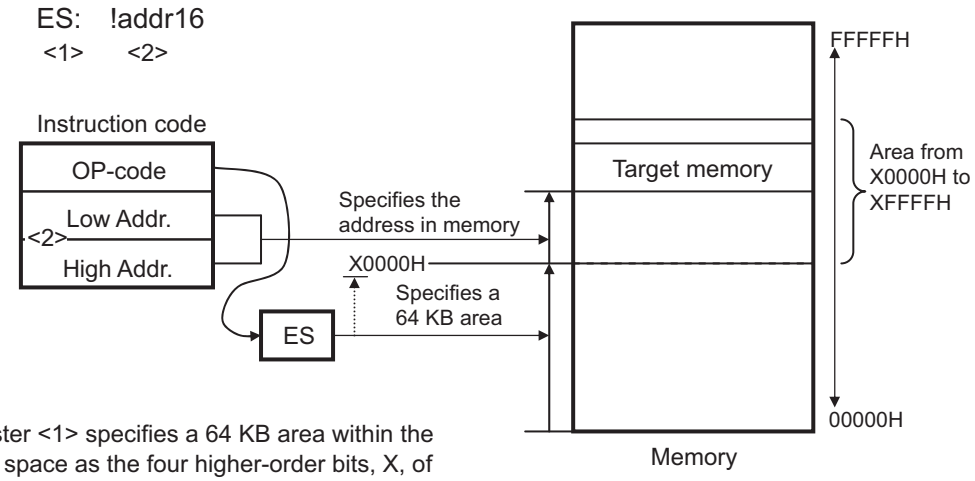
Identifier	Description
!addr16	Label or 16-bit immediate data (only the space from F0000H to FFFFFH is specifiable)
ES:!addr16	Label or 16-bit immediate data (higher 4-bit addresses are specified by the ES register)

Figure 3-19. Example of !addr16



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

Figure 3-20. Example of ES:!addr16



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

3.4.4 Short direct addressing

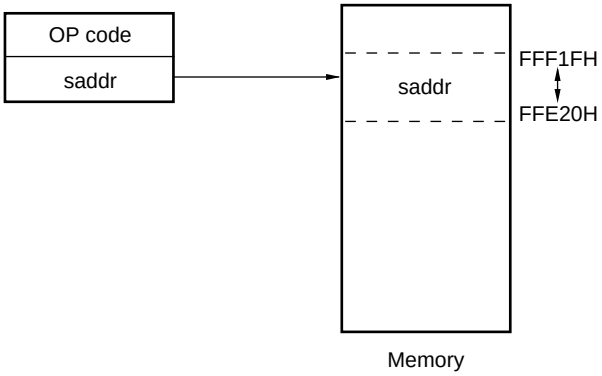
[Function]

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

[Operand format]

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

Figure 3-21. 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

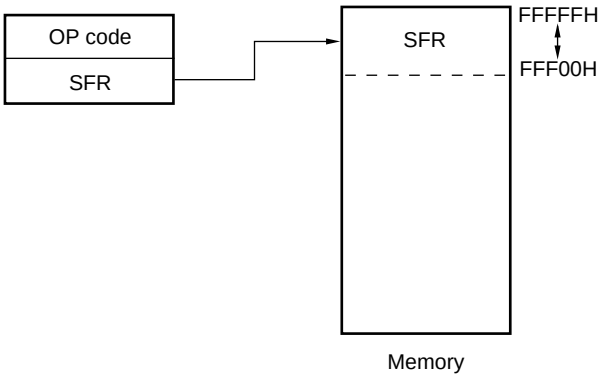
[Function]

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 3-22. Outline of SFR Addressing



3.4.6 Register indirect addressing

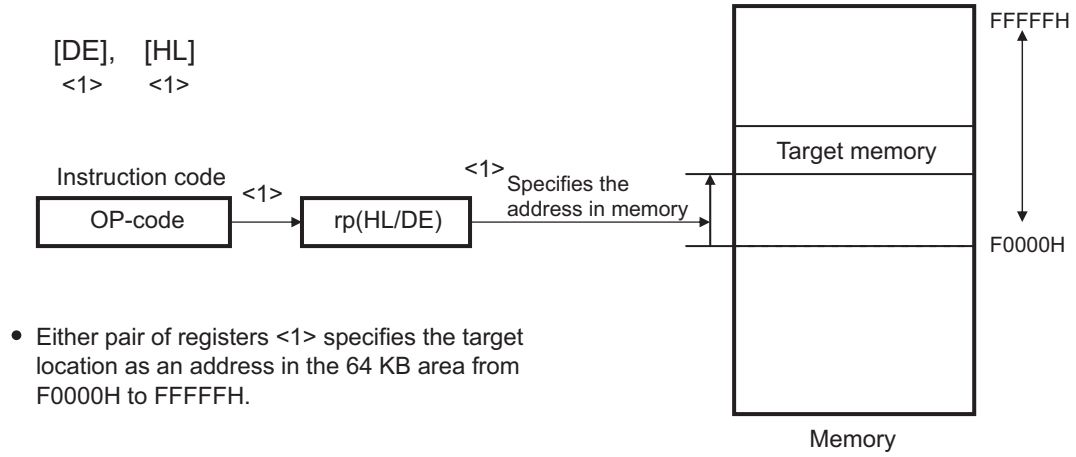
[Function]

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]

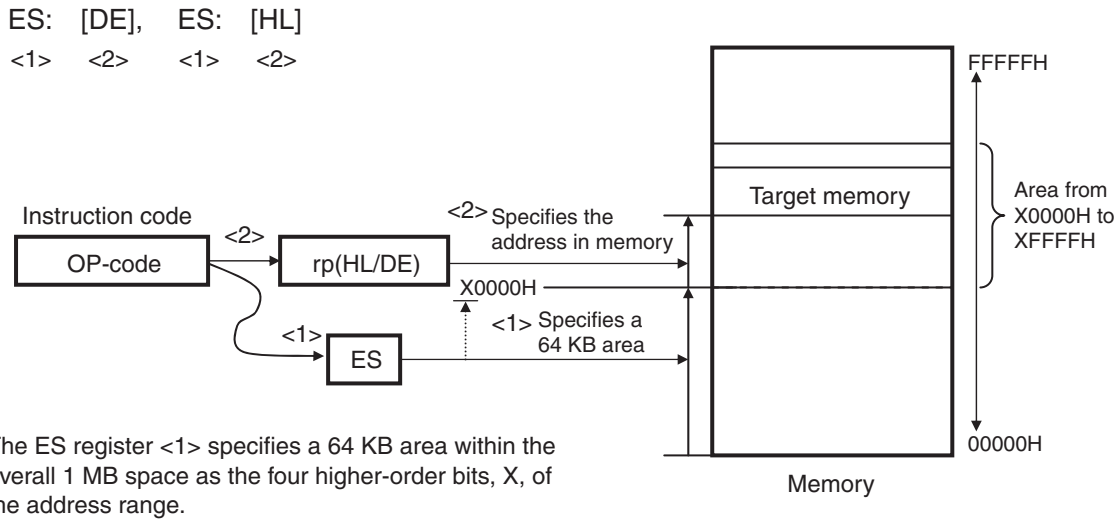
Identifier	Description
–	[DE], [HL] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[DE], ES:[HL] (higher 4-bit addresses are specified by the ES register)

Figure 3-23. Example of [DE], [HL]



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

Figure 3-24. Example of ES:[DE], ES:[HL]



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

3.4.7 Based addressing

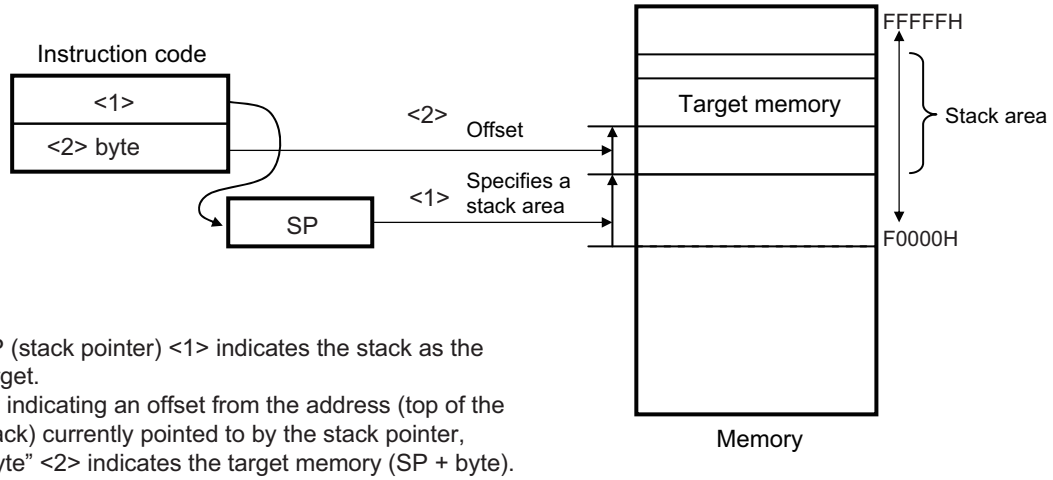
[Function]

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

[Operand format]

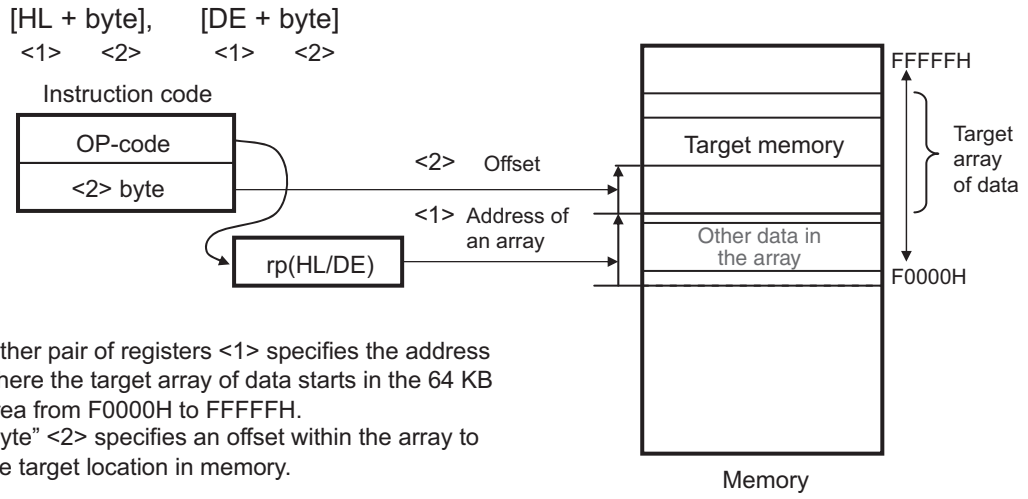
Identifier	Description
–	[HL + byte], [DE + byte], [SP + byte] (only the space from F0000H to FFFFFH is specifiable)
–	word[B], word[C] (only the space from F0000H to FFFFFH is specifiable)
–	word[BC] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[HL + byte], ES:[DE + byte] (higher 4-bit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

Figure 3-25. Example of [SP+byte]



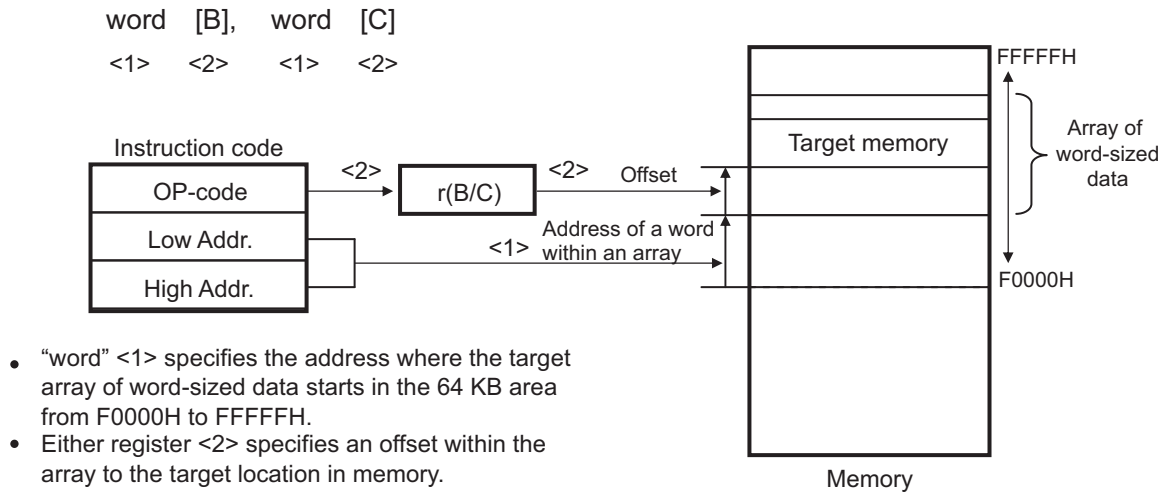
- SP (stack pointer) <1> indicates the stack as the target.
- By indicating an offset from the address (top of the stack) currently pointed to by the stack pointer, “byte” <2> indicates the target memory (SP + byte).

Figure 3-26. Example of [HL + byte], [DE + byte]



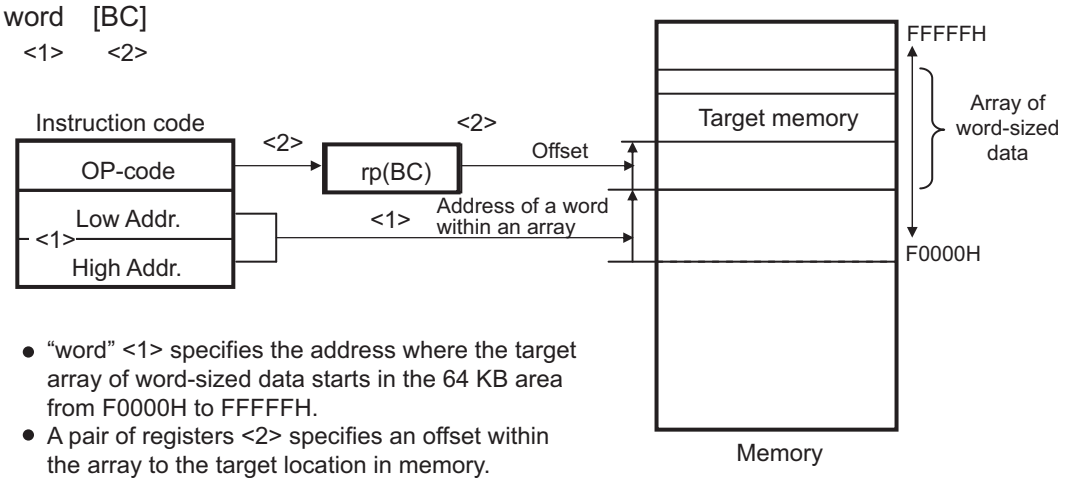
- Either pair of registers <1> specifies the address where the target array of data starts in the 64 KB area from F0000H to FFFFFH.
- "byte" <2> specifies an offset within the array to the target location in memory.

Figure 3-27. Example of word[B], word[C]



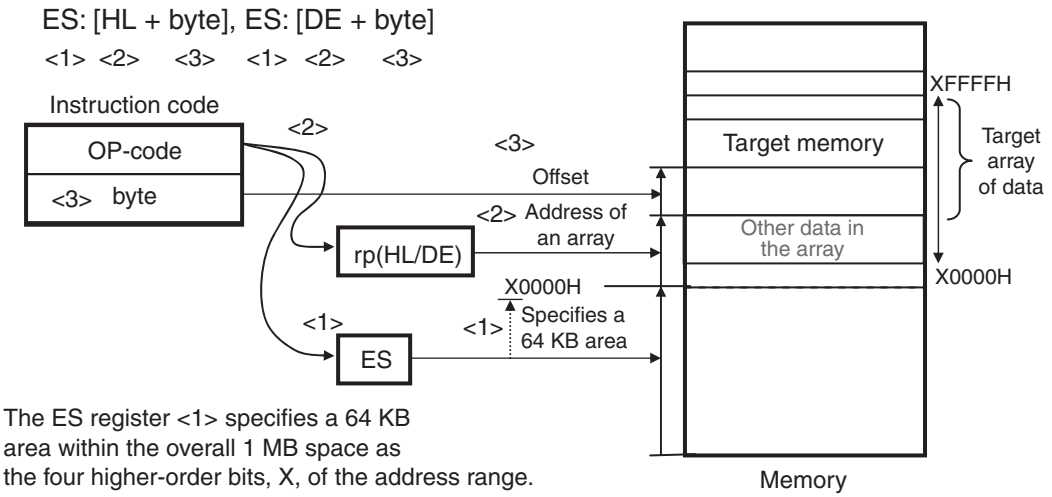
- "word" <1> specifies the address where the target array of word-sized data starts in the 64 KB area from F0000H to FFFFFH.
- Either register <2> specifies an offset within the array to the target location in memory.

Figure 3-28. Example of word[BC]



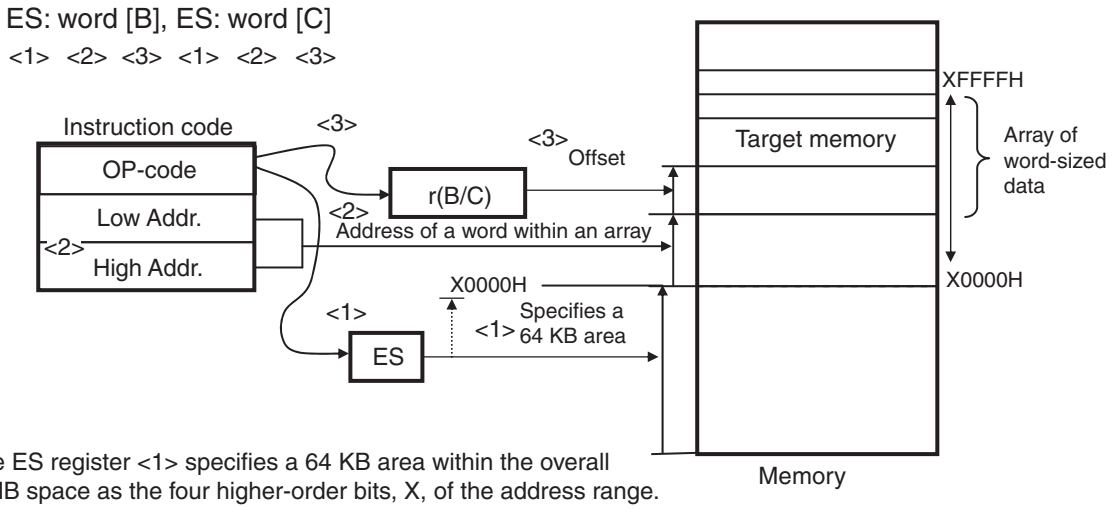
- "word" <1> specifies the address where the target array of word-sized data starts in the 64 KB area from F0000H to FFFFFH.
- A pair of registers <2> specifies an offset within the array to the target location in memory.

Figure 3-29. Example of ES:[HL + byte], ES:[DE + byte]



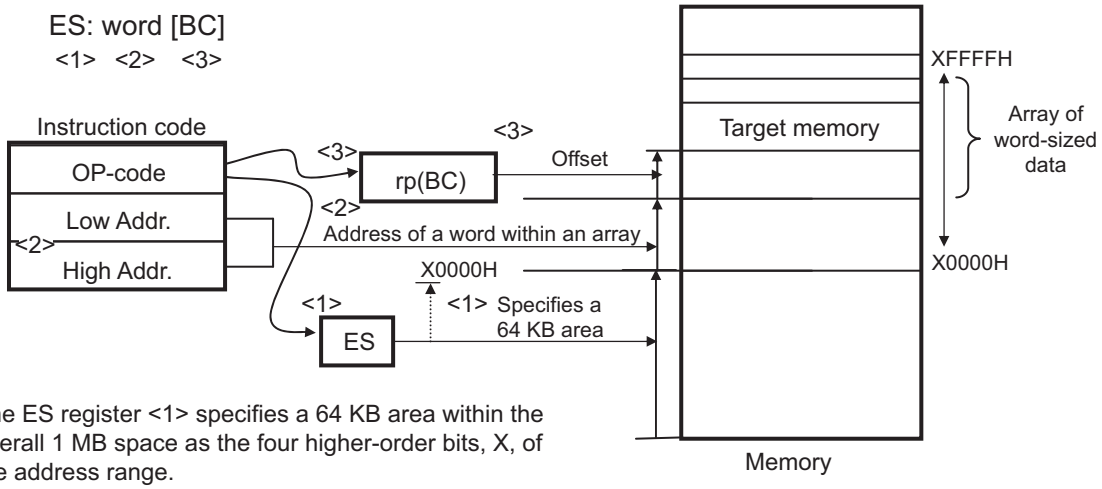
- The ES register <1> specifies a 64 KB area within the overall 1 MB space as the four higher-order bits, X, of the address range.
- Either pair of registers <2> specifies the address where the target array of data starts in the 64 KB area specified in the ES register <1>.
- “byte” <3> specifies an offset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

Figure 3-30. Example of ES:word[B], ES:word[C]



- The ES register <1> specifies a 64 KB area within the overall 1 MB space as the four higher-order bits, X, of the address range.
- “word” <2> specifies the address where the target array of word-sizeddata starts in the 64 KB area specified in the ES register <1>.
- Either register <3> specifies an offset within the array tothe target location in memory.

Figure 3-31. Example of ES:word[BC]



- The ES register <1> specifies a 64 KB area within the overall 1 MB 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 KB 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

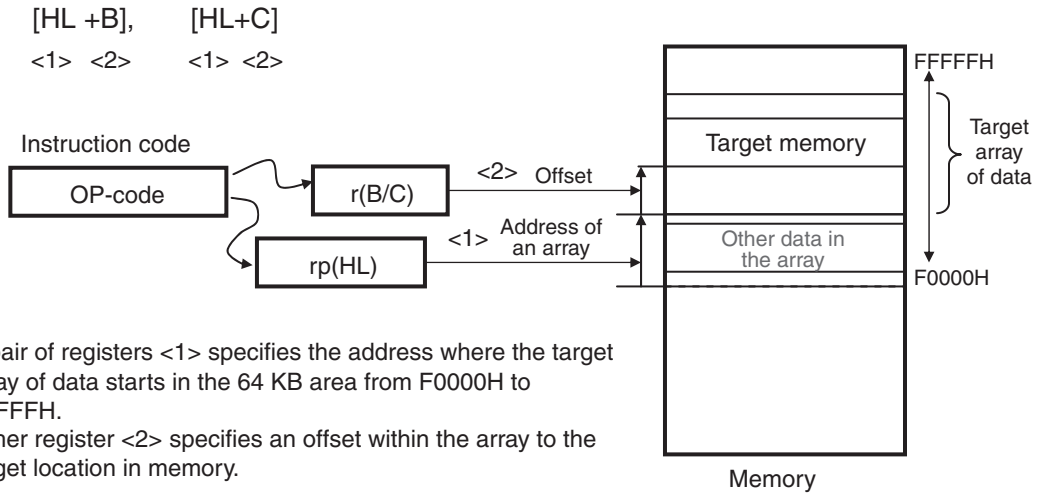
[Function]

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]

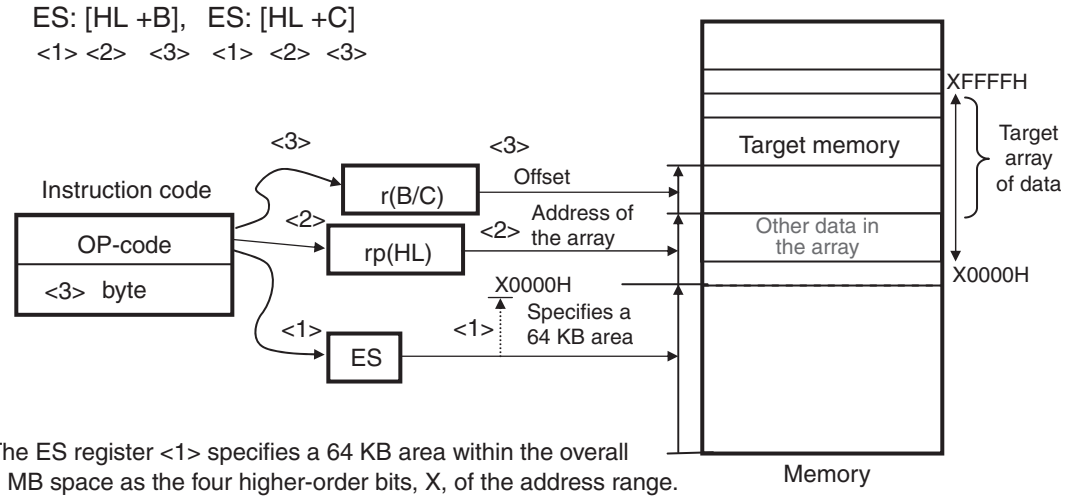
Identifier	Description
–	[HL+B], [HL+C] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[HL+B], ES:[HL+C] (higher 4-bit addresses are specified by the ES register)

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



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

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



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

3.4.9 Stack addressing

[Function]

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

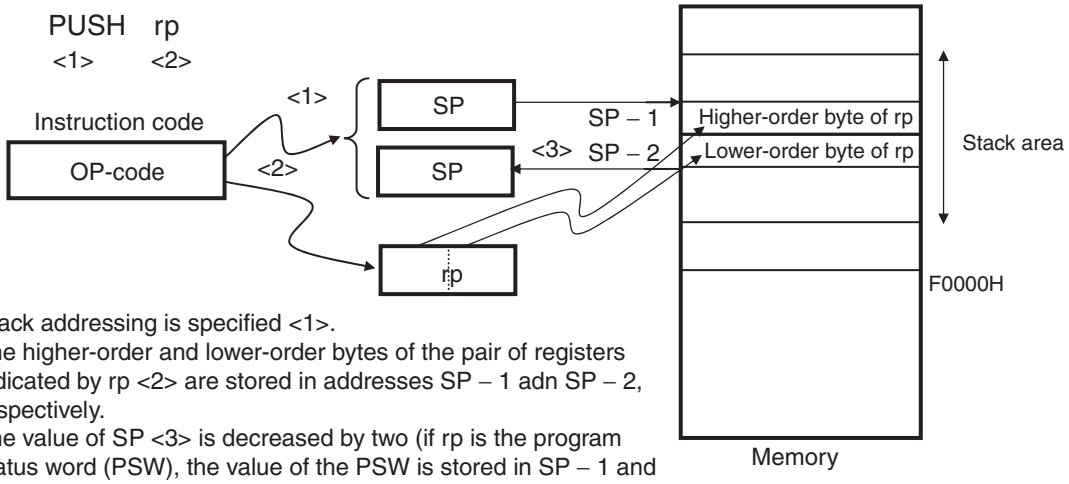
Only the internal RAM area can be set as the stack area.

[Operand format]

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

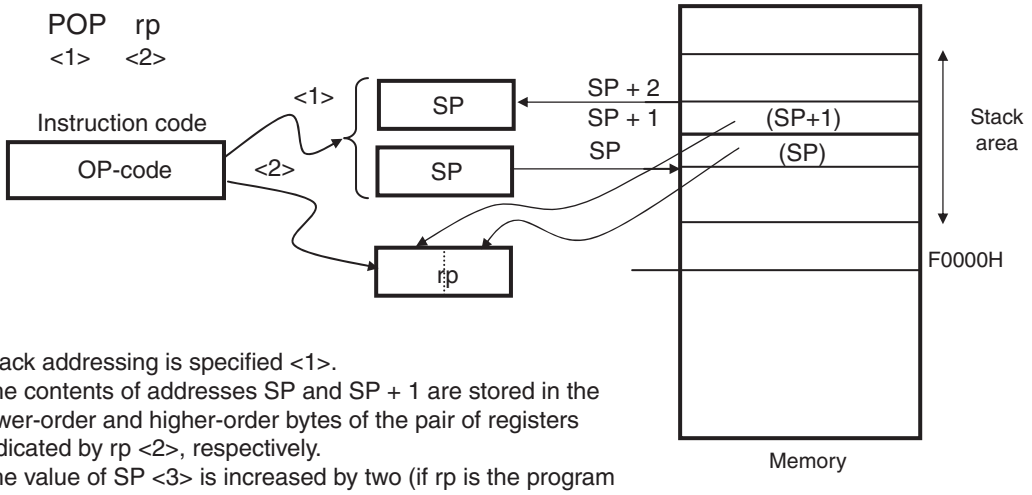
Each stack operation saves or restores data as shown in Figures 3-34 to 3-39.

Figure 3-34. Example of PUSH rp



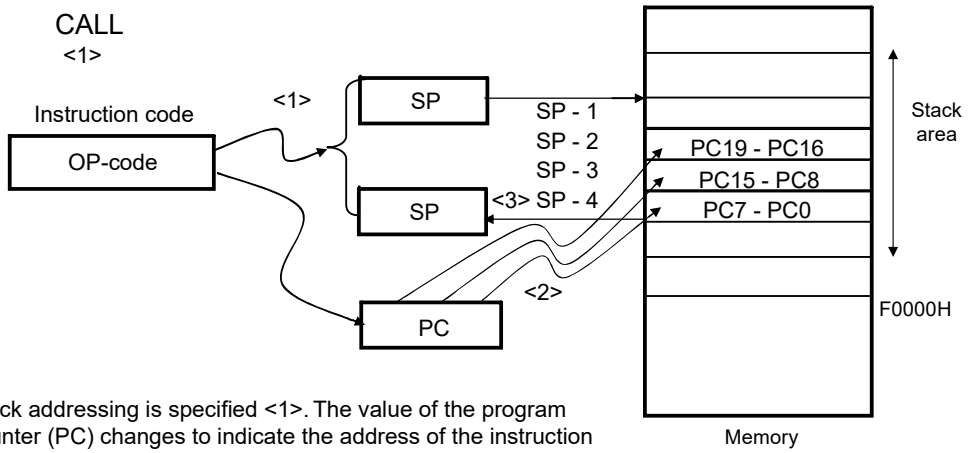
- Stack addressing is specified <1>.
- The higher-order and lower-order bytes of the pair of registers indicated by rp <2> are stored in addresses SP – 1 adn SP – 2, respectively.
- The value of SP <3> is decreased by two (if rp is the program status word (PSW), the value of the PSW is stored in SP –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

Figure 3-35. Example of POP



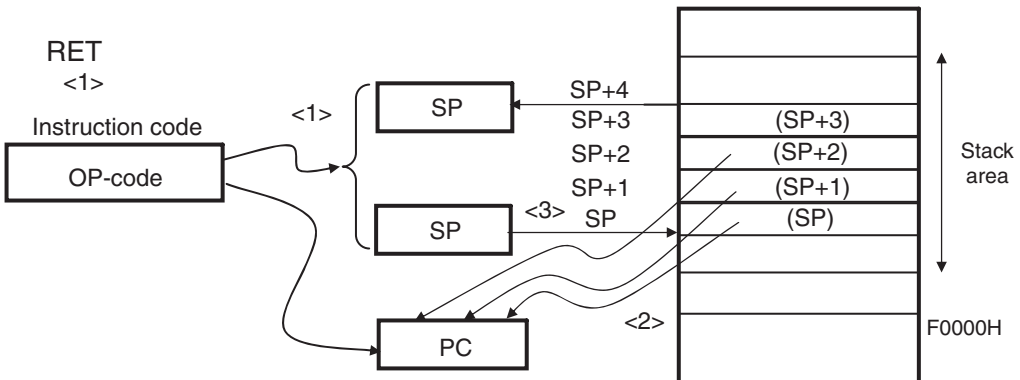
- Stack addressing is specified <1>.
- The contents of addresses SP and SP + 1 are stored in the lower-order and higher-order bytes of the pair of registers indicated by rp <2>, respectively.
- The value of SP <3> is increased by two (if rp is the program status word (PSW), the content of address SP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

Figure 3-36. Example of CALL, CALLT



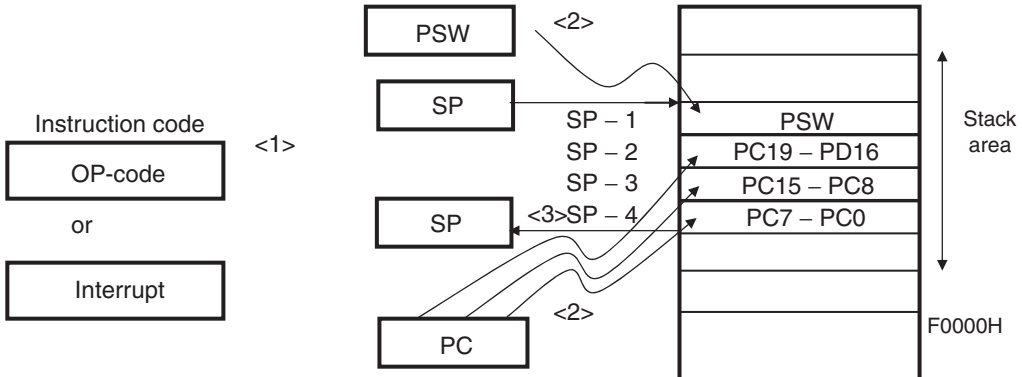
- Stack addressing is specified <1>. The value of the program counter (PC) changes to indicate the address of the instruction following the CALL instruction.
- The values of PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP - 2, SP - 3, and SP - 4, respectively <2>.
- The value of the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

Figure 3-37. Example of RET



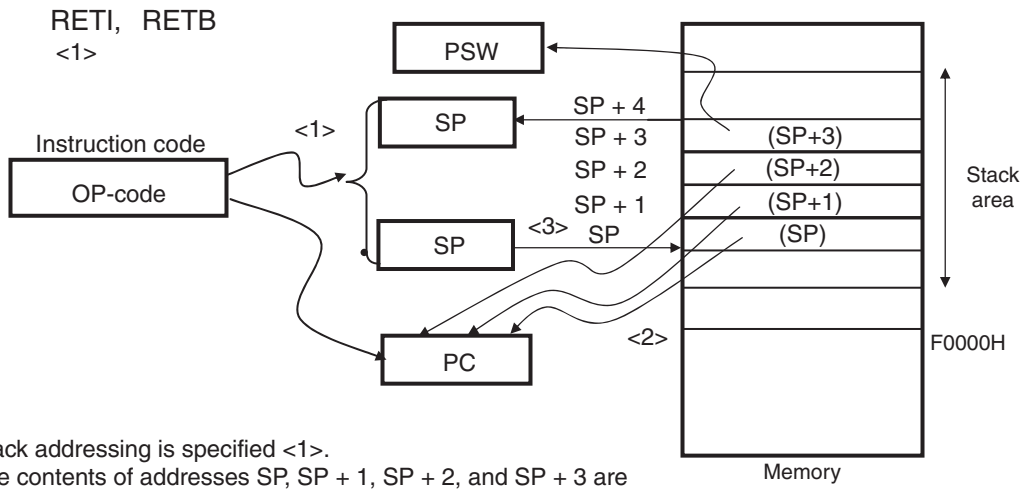
- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, and SP + 2 are stored in PC bits 7 to 0, 15 to 8, and 19 to 16, respectively <2>.
- The value of SP <3> is increased by four.

Figure 3-38. 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 3-39. Example of RETI, RETB



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

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The RL78/I1A microcontrollers are provided with digital I/O ports, which enable variety of control operations.

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

4.2 Port Configuration

Ports include the following hardware.

Table 4-1. Port Configuration

Item	Configuration
Control registers	Port mode registers (PM0 to PM4, PM7, PM12, PM14, PM20) Port registers (P0 to P4, P7, P12 to P14, P20) Pull-up resistor option registers (PU0, PU1, PU3, PU4, PU7, PU12, PU14, PU20) Port input mode registers (PIM0, PIM1) Port output mode registers (POM0, POM1, POM20) Port mode control registers (PMC0, PMC12, PMC14) A/D port configuration register (ADPC) Peripheral I/O redirection register (PIOR1)
Port	• 20-pin products Total: 16 (CMOS I/O: 13, CMOS input: 3) • 30-pin products Total: 26 (CMOS I/O: 23, CMOS input: 3) • 38-pin products Total: 34 (CMOS I/O: 29, CMOS input: 5)
Pull-up resistor	• 20-pin products Total: 8 • 30-pin products Total: 16 • 38-pin products Total: 22

4.2.1 Port 0

Port 0 is an I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P02, P03, P05, 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 a P03 pin can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 0 (PIM0).

Output from a P02 pin can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 0 (POM0).

To use the P02 and P03 pins as digital I/O port pins, set them in the digital I/O mode by using port mode control register 0 (PMC0) (can be specified in 1-bit units).

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

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

- P02 and P03 pins of the 30 and 38-pin products ⋯ Analog input
- P05 and P06 pins of the 38-pin products ⋯ Input mode

4.2.2 Port 1

Port 1 is an I/O port with an output latch. Port 1 can be set to the input mode or output mode in 1-bit units using port mode register 1 (PM1). When the P10 to P12 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 1 (PU1).

Input to the P10 and P11 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 1 (PIM1).

Output from the P10 to P12 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 1 (POM1).

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

Reset signal generation sets port 1 to input mode.

4.2.3 Port 2

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

This port can also be used for A/D converter analog input, internal reference voltage inputs of A/D converter and comparator, reference voltage input of PGA, and analog input of the comparator.

To use P20/ANI0 to P22/ANI2, P24/ANI4 to P27/ANI7 as digital I/O pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC). Use these pins starting from the upper bit.

To use P20/ANI0 to P22/ANI2, P24/ANI4 to P27/ANI7 as analog input pins, set them in the analog input mode by using the A/D port configuration register (ADPC) and in the input mode by using the PM2 register. Use these pins starting from the lower bit.

Table 4-2. Setting Functions of P20/ANI0 to P22/ANI2, P24/ANI4 to P27/ANI7 Pins

ADPC Register	PM2 Register	ADS Register	P20/ANI0 to P22/ANI2, P24/ANI4 to P27/ANI7 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

All P20/ANI0 to P22/ANI2, P24/ANI4 to P27/ANI7 are set in the analog input mode when the reset signal is generated.

4.2.4 Port 3

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

This port can be used for external interrupt request input, real-time clock correction clock output, and timer I/O.

Reset signal generation sets P30 and P31 to input mode.

4.2.5 Port 4

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

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

Reset signal generation sets port 4 to input mode.

4.2.6 Port 7

Port 7 is an I/O port with an output latch. Port 7 can be set to the input mode or output mode in 1-bit units using port mode register 7 (PM7). When the P75 to P77 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 7 (PU7).

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

Reset signal generation sets port 7 to input mode.

4.2.7 Port 12

P120 is an I/O port with an output latch. Port 12 can be set to the input mode or output mode in 1-bit units using port mode register 12 (PM12). When the P120 pin 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).

P121 to P124 are 4-bit input only ports.

Digital input/output or analog input can be specified for the P120 pin, using port mode control register 12 (PMC12).

This port can also be used for A/D converter analog input, connecting resonator for main system clock, connecting resonator for subsystem clock, external clock input for main system clock, and external clock input for subsystem clock.

Reset signal generation sets P120 to analog input, and sets P121 to P124 to input mode.

4.2.8 Port 13

Port 13 is a 1-bit input-only port.

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

4.2.9 Port 14

Port 14 is an I/O port with an output latch. Port 14 can be set to the input mode or output mode in 1-bit units using port mode register 14 (PM14). When the P147 pin 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 14 (PU14).

Digital input/output or analog input can be specified for the P147 pin, using port mode control register 14 (PMC14).

This port can also be used for A/D converter analog input, and comparator external reference voltage input.

Reset signal generation sets port 14 to analog mode.

4.2.10 Port 20

Port 20 is an I/O port with an output latch. Port 20 can be set to the input mode or output mode in 1-bit units using port mode register 20 (PM20). When the P200 to P206 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 20 (PU20).

Output from the P200 to P206 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 20 (POM20).

This port can also be used for serial interface data I/O, timer I/O, and external interrupt request.

Reset signal generation sets port 20 to input mode.

4.3 Registers Controlling Port Function

Port functions are controlled by the following registers.

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

Caution Which registers and bits are included depends on the product. For registers and bits mounted on each product, see Table 4-3. Be sure to set bits that are not mounted to their initial values.
Note, however, that this does not apply to the bits having undefined values which are specified in the caution of Figure 4-1 Format of Port Mode Register.

Table 4-3. PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx Registers and Bits Mounted on Each Product (1/3)

Port		Bit Name						20 Pin	30 Pin	38 Pin
		PMxx Register	Pxx Register	PUxx Register	PIMxx Register	POMxx Register	PMCxx Register			
Port 0	0	–	–	–	–	–	–	–	–	–
	1	–	–	–	–	–	–	–	–	–
	2	PM02	P02	PU02	–	POM02	PMC02	–	√	√
	3	PM03	P03	PU03	PIM03	–	PMC03	–	√	√
	4	–	–	–	–	–	–	–	–	–
	5	PM05	P05	PU05	–	–	–	–	–	√
	6	PM06	P06	PU06	–	–	–	–	–	√
	7	–	–	–	–	–	–	–	–	–
Port 1	0	PM10	P10	PU10	PIM10	POM10	–	√	√	√
	1	PM11	P11	PU11	PIM11	POM11	–	√	√	√
	2	PM12	P12	PU12	–	POM12	–	–	–	√
	3	–	–	–	–	–	–	–	–	–
	4	–	–	–	–	–	–	–	–	–
	5	–	–	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–	–	–
Port 2	0	PM20	P20	–	–	–	–	√	√	√
	1	PM21	P21	–	–	–	–	√	√	√
	2	PM22	P22	–	–	–	–	√	√	√
	3	–	–	–	–	–	–	–	–	–
	4	PM24	P24	–	–	–	–	√	√	√
	5	PM25	P25	–	–	–	–	√	√	√
	6	PM26	P26	–	–	–	–	–	√	√
	7	PM27	P27	–	–	–	–	–	√	√

Table 4-3. PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx Registers and Bits Mounted on Each Product (2/3)

Port		Bit Name						20 Pin	30 Pin	38 Pin
		PMxx Register	Pxx Register	PUxx Register	PIMxx Register	POMxx Register	PMCxx Register			
Port 3	0	PM30	P30	PU30	–	–	–	–	–	√
	1	PM31	P31	PU31	–	–	–	–	√	√
	2	–	–	–	–	–	–	–	–	–
	3	–	–	–	–	–	–	–	–	–
	4	–	–	–	–	–	–	–	–	–
	5	–	–	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–	–	–
Port 4	0	PM40	P40	PU40	–	–	–	√	√	√
	1	–	–	–	–	–	–	–	–	–
	2	–	–	–	–	–	–	–	–	–
	3	–	–	–	–	–	–	–	–	–
	4	–	–	–	–	–	–	–	–	–
	5	–	–	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–	–	–
Port 7	0	–	–	–	–	–	–	–	–	–
	1	–	–	–	–	–	–	–	–	–
	2	–	–	–	–	–	–	–	–	–
	3	–	–	–	–	–	–	–	–	–
	4	–	–	–	–	–	–	–	–	–
	5	PM75	P75	PU75	–	–	–	–	–	√
	6	PM76	P76	PU76	–	–	–	–	–	√
	7	PM77	P77	PU77	–	–	–	–	√	√
Port 12	0	PM120	P120	PU120	–	–	PMC120	–	√	√
	1	–	P121	–	–	–	–	√	√	√
	2	–	P122	–	–	–	–	√	√	√
	3	–	P123	–	–	–	–	–	–	√
	4	–	P124	–	–	–	–	–	–	√
	5	–	–	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–	–	–
Port 13	0	–	–	–	–	–	–	–	–	–
	1	–	–	–	–	–	–	–	–	–
	2	–	–	–	–	–	–	–	–	–
	3	–	–	–	–	–	–	–	–	–
	4	–	–	–	–	–	–	–	–	–
	5	–	–	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–	–	–
	7	–	P137	–	–	–	–	√	√	√

Table 4-3. PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx Registers and Bits Mounted on Each Product (3/3)

Port		Bit Name						20 Pin	30 Pin	38 Pin
		PMxx Register	Pxx Register	PUxx Register	PIMxx Register	POMxx Register	PMCxx Register			
Port 14	0	–	–	–	–	–	–	–	–	–
	1	–	–	–	–	–	–	–	–	–
	2	–	–	–	–	–	–	–	–	–
	3	–	–	–	–	–	–	–	–	–
	4	–	–	–	–	–	–	–	–	–
	5	–	–	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–	–	–
	7	PM147	P147	PU147	–	–	PMC147	√	√	√
Port 20	0	PM200	P200	PU200	–	POM200	–	√	√	√
	1	PM201	P201	PU201	–	POM201	–	√	√	√
	2	PM202	P202	PU202	–	POM202	–	√	√	√
	3	PM203	P203	PU203	–	POM203	–	√	√	√
	4	PM204	P204	PU204	–	POM204	–	–	√	√
	5	PM205	P205	PU205	–	POM205	–	–	√	√
	6	PM206	P206	PU206	–	POM206	–	–	√	√
	7	–	–	–	–	–	–	–	–	–

4.3.1 Port mode registers (PMxx)

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

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

Reset signal generation sets these registers to FFH.

When port pins are used as alternate-function pins, set the port mode register by referencing **4.5 Register Settings When Using Alternate Function**.

Figure 4-1. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	PM06	PM05	1	PM03	PM02	1	1	FFF20H	FFH	R/W
PM1	1	1	1	1	1	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	1	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	1	1	1	1	1	1	PM31	PM30	FFF23H	FFH	R/W
PM4	1	1	1	1	1	1	1	PM40	FFF24H	FFH	R/W
PM7	PM77	PM76	PM75	1	1	1	1	1	FFF27H	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	FFF2CH	FFH	R/W
PM14	PM147	1	1	1	1	1	1	1	FFF2EH	FFH	R/W
PM20	1	PM206	PM205	PM204	PM203	PM202	PM201	PM200	F0510H	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 4, 7, 12, 14, 20; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set the following bits to 1.

Bits 0, 1, 4, and 7 of the PM0 register, bits 3 to 7 of the PM1 register, bit 3 of the PM2 register, bits 2 to 7 of the PM3 register, bits 1 to 7 of the PM4 register, bits 0 to 4 of the PM7 register, bits 1 to 7 of the PM12 register, bits 0 to 6 of the PM14 register, and bit 7 of the PM20 register.

For 30- and 20-pin products, the following bits must be set for the output mode (by setting the port registers and port mode registers to 0) by software after release from the reset state.

30-pin products: Bits 5 and 6 of the PM0 register, bit 2 of the PM1 register, bit 0 of the PM3 register, and bits 5 and 6 of the PM7 register

20-pin products: Bits 2, 3, 5 and 6 of the PM0 register, bit 2 of the PM1 register, bits 6 and 7 of the PM2 register, bits 0 and 1 of the PM3 register, bits 5 to 7 of the PM7 register, bit 0 of the PM12 register, and bits 4 to 6 of the PM20 register

4.3.2 Port registers (Pxx)

These registers set the output latch value of a port.

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

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

Reset signal generation clears these registers to 00H.

Note If P02, P03, P20 to P22, P24 to P27, P120, and P147 are set up as analog inputs of the A/D converter, when a port is read while in the input mode, 0 is always returned, not the pin level.

Figure 4-2. Format of Port Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	P06	P05	0	P03	P02	0	0	FFF00H	00H (output latch)	R/W
P1	0	0	0	0	0	P12	P11	P10	FFF01H	00H (output latch)	R/W
P2	P27	P26	P25	P24	0	P22	P21	P20	FFF02H	00H (output latch)	R/W
P3	0	0	0	0	0	0	P31	P30	FFF03H	00H (output latch)	R/W
P4	0	0	0	0	0	0	0	P40	FFF04H	00H (output latch)	R/W
P7	P77	P76	P75	0	0	0	0	0	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	0	FFF0DH	**Note 2**	R
P14	P147	0	0	0	0	0	0	0	FFF0EH	00H (output latch)	R/W
P20	0	P206	P205	P204	P203	P202	P201	P200	F0500H	00H (output latch)	R/W

Pmn	m = 0 to 4, 7, 12 to 14, 20; n = 0 to 7	
	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

Notes 1. P121 to P124 and P137 are read-only.

2. P137: Undefined

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

4.3.3 Pull-up resistor option registers (PUxx)

These registers specify whether the on-chip pull-up resistors are to be used or not. On-chip pull-up resistors can be used in 1-bit units only for the bits set to 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 cannot be connected to bits set to output mode and bits used as alternate-function output pins and analog setting (PMC = 1, ADPC = 1), regardless of the settings of these registers.

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

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

Caution When a port with the PIMn register is input from different potential device to TTL buffer, pull up to the power supply of the different potential device via a external pull-up resistor by setting PUmn = 0.

Figure 4-3. Format of Pull-up Resistor Option Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	PU06	PU05	0	PU03	PU02	0	0	F0030H	00H	R/W
PU1	0	0	0	0	0	PU12	PU11	PU10	F0031H	00H	R/W
PU3	0	0	0	0	0	0	PU31	PU30	F0033H	00H	R/W
PU4	0	0	0	0	0	0	0	PU40	F0034H	01H	R/W
PU7	PU77	PU76	PU75	0	0	0	0	0	F0037H	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	F003CH	00H	R/W
PU14	PU147	0	0	0	0	0	0	0	F003EH	00H	R/W
PU20	0	PU206	PU205	PU204	PU203	PU202	PU201	PU200	F0520H	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 1, 3, 4, 7, 12, 14, 20; 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 mounted to their initial values.

4.3.4 Port input mode registers (PIMxx)

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

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

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

Reset signal generation clears these registers to 00H.

Figure 4-4. Format of Port Input Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM0	0	0	0	0	PIM03	0	0	0	F0040H	00H	R/W
PIM1	0	0	0	0	0	0	PIM11	PIM10	F0041H	00H	R/W

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

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

4.3.5 Port output mode registers (POMxx)

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

N-ch open drain output (V_{DD} tolerance) mode can be selected during serial communication with an external device of the different potential, and for the SDAA0 pin during IICA communication with an external device of the same potential.

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

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

Reset signal generation clears these registers to 00H.

Caution An on-chip pull-up resistor is not connected to a bit for which N-ch open drain output (V_{DD} tolerance) mode (POMmn = 1) is set.

Figure 4-5. Format of Port Output Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM0	0	0	0	0	0	POM02	0	0	F0050H	00H	R/W
POM1	0	0	0	0	0	POM12	POM11	POM10	F0051H	00H	R/W
POM20	0	POM206	POM205	POM204	POM203	POM202	POM201	POM200	F0530H	00H	R/W

POMmn	Pmn pin output mode selection (m = 0, 1, 20; n = 0 to 6)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

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

4.3.6 Port mode control registers (PMCxx)

These registers set the digital I/O/analog input in 1-bit units.

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

Reset signal generation clears these registers to FFH.

Figure 4-6. Format of Port Mode Control Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMC0	1	1	1	1	PMC03	PMC02	1	1	F0060H	FFH	R/W
PMC12	1	1	1	1	1	1	1	PMC120	F006CH	FFH	R/W
PMC14	PMC147	1	1	1	1	1	1	1	F006EH	FFH	R/W

PMCmn	Pmn pin digital I/O/analog input selection (m = 0, 12, 14; n = 0, 2, 3, 7)
0	Digital I/O (alternate function other than analog input)
1	Analog input

Cautions 1. Select input mode by using port mode registers 0, 12, 14 (PM0, PM12, PM14) for the ports which are set by the PMCxx register as analog input.

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

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

4.3.7 A/D port configuration register (ADPC)

This register switches the ANI0/P20, ANI1/P21, ANI2/CMP0P/P22, ANI4/CMP1P/P24 to ANI7/CMP4P/P27 pins, and PGAOUT pin (internal pin) to digital I/O of port or analog input of A/D converter, programmable gain amplifier, or comparator.

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

Reset signal generation sets this register to 00H.

Figure 4-7. Format of A/D Port Configuration Register (ADPC)

Address: F0076H After reset: 00H R/W

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

ADPC3	ADPC2	ADPC1	ADPC0	Analog input (A)/digital I/O (D) switching							
				ANI7/ CMP4P/P27	ANI6/ CMP3P/P26	ANI5/ CMP2P/P25	ANI4/ CMP1P/P24	PGAOUT[Note]	ANI2/ CMP0P/P22	ANI1/P21	ANI0/P20
0	0	0	0	A	A	A	A	A	A	A	A
0	0	0	1	D	D	D	D	D	D	D	D
0	0	1	0	D	D	D	D	D	D	D	A
0	0	1	1	D	D	D	D	D	D	A	A
0	1	0	0	D	D	D	D	D	A	A	A
0	1	0	1	D	D	D	D	A	A	A	A
0	1	1	0	D	D	D	A	A	A	A	A
0	1	1	1	D	D	A	A	A	A	A	A
1	0	0	0	D	A	A	A	A	A	A	A
1	0	0	1	A	A	A	A	A	A	A	A
1	1	1	1	A	A	A	A	A	A	A	A
Other than above				Setting prohibited							

Note This is an internal output pin for the programmable gain amplifier. When output signals from the programmable gain amplifier is used as an analog input channel for the A/D converter, set ADPC as 0101B or above.

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

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

3. When using AV_{REFP} and AV_{REFM}, set ANI0 and ANI1 to analog input and set the port mode register to the input mode.

4.3.8 Peripheral I/O redirection register (PIOR1)

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

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

Use the PIOR1 register to assign a port to the function to redirect and enable the function.

In addition, can be changed the settings for redirection until its function enable operation.

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

Reset signal generation sets this register to 00H.

Figure 4-8. Format of Peripheral I/O Redirection Register (PIOR1)

Address: F05C0H After reset: 00H R/W

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

Bit	Alternate Function	20-pin		30-pin		38-pin	
		Setting value		Setting value		Setting value	
		0	1	0	1	0	1
PIOR10	TKCO03	This area cannot be used as alternate function. Set this bit to 0 (default value).		P204	–	P204	P12
PIOR11	DALITxD4/ DALIRxD4	–	P10/ P11	P205/ P206	P10/ P11	P205/ P206	P10/ P11
PIOR12	INTP20	P10	P203	P10	P203	P10	P203
PIOR13	INTP21	P11	P202	P11	P202	P11	P202

4.4 Port Function Operations

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

4.4.1 Writing to I/O port

(1) Output mode

A value is written to the output latch by a transfer instruction, and the output latch contents are output from the pin.
Once data is written to the output latch, it is retained until data is written to the output latch again.
The data of the output latch is cleared when a reset signal is generated.

(2) Input mode

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

4.4.2 Reading from I/O port

(1) Output mode

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

(2) Input mode

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

4.4.3 Operations on I/O port

(1) Output mode

An operation is performed on the output latch contents, and the result is written to the output latch. The output latch contents are output from the pins.
Once data is written to the output latch, it is retained until data is written to the output latch again.
The data of the output latch is cleared when a reset signal is generated.

(2) Input mode

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

4.4.4 Handling different potential (2.5 V, 3 V)

When connecting an external device operating on a different potential (2.5 V or 3 V), it is possible to connect the I/O pins of general ports by changing V_{DD} to accord with the power supply of the connected device.

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

It is possible to connect an external device operating on a different potential (2.5 V, or 3 V) by switching I/O buffers with port input mode registers 0 and 1 (PIM0 and PIM1) and port output mode registers 0 and 1 (POM0 and POM1).

When receiving input from an external device with a different potential (2.5 V, or 3 V), set port input mode registers 0 and 1 (PIM0 and PIM1) on a bit-by-bit basis to enable normal input (CMOS)/TTL input buffer switching.

When outputting data to an external device with a different potential (2.5 V, or 3 V), set port output mode registers 0 and 1 (POM0 and POM1) on a bit-by-bit basis to enable normal output (CMOS)/N-ch open drain (V_{DD} tolerance) switching.

Following, describes the connection of a serial interface.

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

In case of UART0: P11
In case of UART1: P03
In case of CSI00: P11, P12

<1> Using an external resistor, pull up 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 and PIM1 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.

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

In case of UART0: P10
In case of UART1: P02
In case of CSI00: P10, P12

<1> Using an external resistor, pull up 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 port mode changes to the input mode (Hi-Z).
<3> Set the output latch of the corresponding port to 1.
<4> Set the corresponding bit of the POM0 and POM1 registers to 1 to set the N-ch open drain output (V_{DD} withstand voltage) mode.
<5> Enable the operation of the serial array unit and set the mode to the UART/Simplified SPI (CSI) mode.
<6> Set the output mode by manipulating the PM0 and PM1 registers.
At this time, the output data is high level, so the pin is in the Hi-Z state.

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.

(3) Setting procedure when using I/O pins of IICA0 functions

<1> Externally pull up the P10 and P11 pins to be used (on-chip pull-up resistor cannot be used).

<2> After reset release, the port mode is the input mode (Hi-Z).

<3> Set the output latch of the corresponding port to 1.

<4> Set the corresponding bit of the POM1 register to 1 to set the N-ch open drain output (V_{DD} withstand voltage) mode.

<5> Set the corresponding bit of the PM1 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.

<6> Enable the operation of the IICA0.

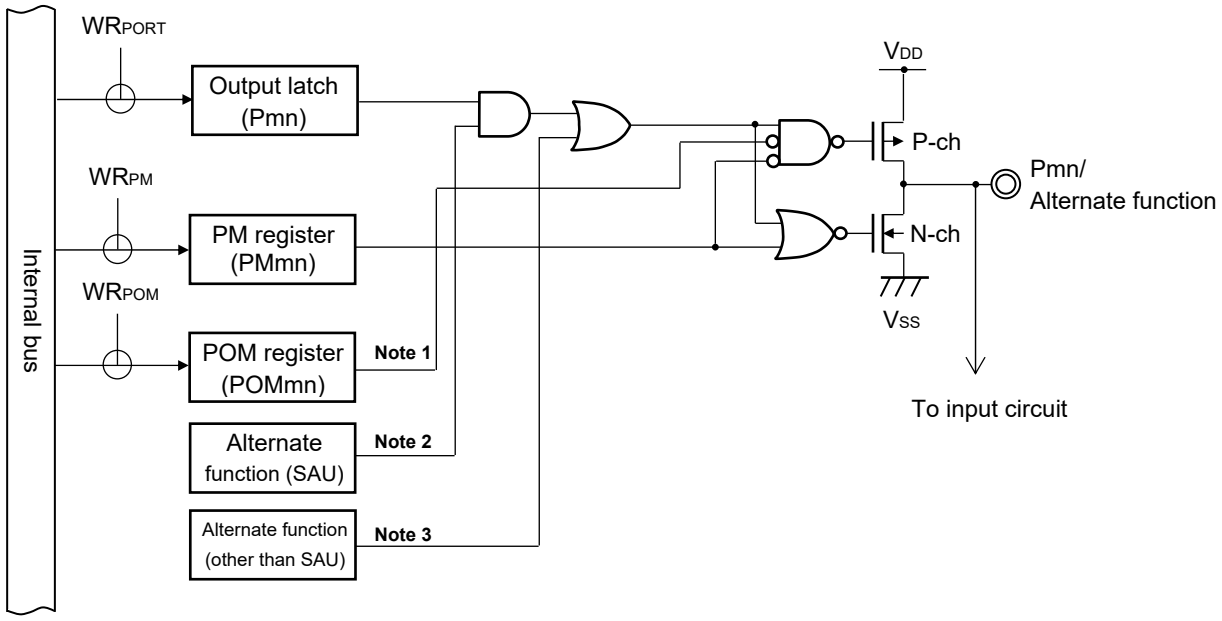
4.5 Register Settings When Using Alternate Function

4.5.1 Basic concept when using alternate function

In the beginning, for a pin also assigned to be used for analog input, use the ADPC register or port mode control register (PMCxx) to specify whether to use the pin for analog input or digital input/output.

Figure 4-9 shows the basic configuration of an output circuit for pins used for digital input/output. The output of the output latch for the port and the output of the alternate SAU function are input to an AND gate. The output of the AND gate is input to an OR gate. The output of an alternate function other than SAU (TAU, RTC, IICA, etc.) is connected to the other input pin of the OR gate. When such kind of pins are used by the port function or an alternate function, the unused alternate function must not hinder the output of the function to be used. An idea of basic settings for this kind of case is shown in Table 4-4.

Figure 4-9. Basic Configuration of Output Circuit for Pins



Notes 1. When there is no POM register, this signal should be considered to be low level (0).

2. When there is no alternate function, this signal should be considered to be high level (1).

3. When there is no alternate function, this signal should be considered to be low level (0).

Remark m: Port number (m = 0 to 15); n: Bit number (n = 0 to 7)

Table 4-4. Concept of Basic Settings

Output Function of Used Pin	Output Settings of Unused Alternate Function		
	Port Function	Output Function for SAU	Output Function for other than SAU
Output function for port	–	Output is high (1)	Output is low (0)
Output function for SAU	High (1)	–	Output is low (0)
Output function for other than SAU	Low (0)	Output is high (1)	Output is low (0)Note

Note Since more than one output function other than SAU may be assigned to a single pin, the output of an unused alternate function must be set to low level (0). For details on the setting method, see **4.5.2 Register settings for alternate function whose output function is not used**.

4.5.2 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 the target, the output can be switched to another pin by setting the peripheral I/O redirection register (PIOR1). 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). 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). 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 bit in timer output register 0 (TO0) to 0 (low). These are the same settings as the initial state.

(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.

4.5.3 Register setting examples for used port and alternate functions

Register setting examples for used port and alternate functions are shown in Table 4-5. The registers used to control the port functions should be set as shown in Table 4-5. See the following remark for legends used in Table 4-5.

Remark –: Not supported
×: don't care
PIOR×: Peripheral I/O redirection register
POM××: Port output mode register
PMC××: Port mode control register
PM××: Port mode register
P××: Port output latch
Functions in parentheses can be assigned via settings in the peripheral I/O redirection register (PIOR1).

Table 4-5. Setting Examples of Registers and Output Latches When Using Pin Function (1/7)

Pin Name	Used Function		PIOR×	POM××	PMC××	PM××	P××	Alternate Function Output		20-pin	30-pin	38-pin
	Function Name	I/O						SAU Output Function	Other than SAU			
P02	P02	Input	–	×	0	1	×	×	×	×	√	√
		Output	–	0	0	0	0/1	TxD1 = 1	×			
		N-ch open drain output	–	1	0	0	0/1					
	ANI17	Analog input	–	–	1	1	×	×	×	×	√	√
	TxD1	Output	–	0/1	0	0	1	×	×	×	√	√
P03	P03	Input	–	×	0	1	×	×	×	×	√	√
		Output	–	0	0	0	0/1	×	×			
		N-ch open drain output	–	1	0	0	0/1					
	ANI16	Analog input	–	–	1	1	×	×	×	×	√	√
	CM5P	Input	–	–	1	1	×	×	×	×	√	√
	RxD1	Input	–	–	0	1	×	×	×	×	√	√
P05	P05	Input	–	–	–	1	×	×	×	×	√	√
		Output	–	–	–	0	0/1	×	TO05 = 0	×	×	√
	TI05	Input	–	–	–	1	×	×	×	×	×	√
	TO05	Output	–	–	–	0	0	×	×	×	×	√
P06	P06	Input	–	–	–	1	×	×	×	×	×	√
		Output	–	–	–	0	0/1	×	TO06 = 0	×	×	√
	TI06	Input	–	–	–	1	×	×	×	×	×	√
	TO06	Output	–	–	–	0	0	×	×	×	×	√

Table 4-5. Setting Examples of Registers and Output Latches When Using Pin Function (2/7)

Pin Name	Used Function		PIOR×	POM××	PMC××	PM××	P××	Alternate Function Output		20-pin	30-pin	38-pin
	Function Name	I/O						SAU Output Function	Other than SAU			
P10	P10	Input	–	×	–	1	×	×	×			
		Output	–	0	–	0	0/1	SO00/TxD0 = 1 SCLA0 = 0 (DALITxD4 = 1)	TKCO00 = 0	√	√	√
		N-ch open drain output	–	1	–	0	0/1					
	SO00	Output	×	0/1	–	0	1	×	×	√	√	√
	TxD0	Output	×	0/1	–	0	1	×	×	√	√	√
	TKC00	Output	×	0	–	0	0	×	×	√	√	√
	INTP20	Input	PIOR12 = 0	×	–	1	×	×	×	√	√	√
	SCLA0	I/O	×	1	–	0	0	×	×	√	√	√
	(DALITxD4)	Output	PIOR11 = 1	0/1	–	0	1	×	×	√	√	√
P11	P11	Input	–	×	–	1	×	×	×			
		Output	–	0	–	0	0/1	SDAA0 = 0 (TxRx4 = 1)	TKCO01 = 0	√	√	√
		N-ch open drain output	–	1	–	0	0/1					
	SI00	Input	–	×	–	1	×	×	×	√	√	√
	RxD0	Input	–	×	–	1	×	×	×	√	√	√
	TKCO01	Output	–	0	–	0	0	×	×	√	√	√
	INTP21	Input	PIOR13 = 0	×	–	1	×	×	×	√	√	√
	SDAA0	I/O	–	1	–	0	0	×	×	√	√	√
	(TI07)	Input	–	×	–	1	×	×	×	√	√	√
	(DALIRxD4)	Input	PIOR11 = 1	×	–	1	×	×	×	√	√	√
	(TxRx4)	I/O	–	0/1	–	0	1	×	×	√	√	√
P12	P12	Input	–	×	–	1	×	×	×	√	√	√
		Output	–	0	–	0	0/1	SCK00 = 1	(TKCO03 = 0)	×	×	√
		N-ch open drain output	–	1	–	0	0/1	×	×	×	×	√
	SCK00	Input	–	×	–	1	×	×	×	×	×	√
		Output	–	0/1	–	0	1	×	×	×	×	√
	(TCKO03)	Output	PIOR10 = 1	0	–	0	0	×	×	×	×	√

Table 4-5. Setting Examples of Registers and Output Latches When Using Pin Function (3/7)

Pin Name	Used Function		ADPC	ADM2	PM××	P××	20-pin	30-pin	38-pin
	Function Name	I/O							
P20	P20	Input	ADPC = 01H	×	1	×	√	√	√
		Output	ADPC = 01H	×	0	0/1			
	ANI0	Analog input	ADPC = 00H/02H to 0FH	00x0xx0x, 10x0xx0x	1	×	√	√	√
	AV_{REFP}	Reference voltage	ADPC = 00H/02H to 0FH	01x0xx0x	1	×	√	√	√
P21	P21	Input	ADPC = 01H/02H	×	1	×	√	√	√
		Output	ADPC = 01H/02H	×	0	0/1			
	ANI1	Analog input	ADPC = 00H/3FH to 0FH	xx00xx0x	1	×	√	√	√
	AV_{REFM}	Reference voltage	ADPC = 00H/3FH to 0FH	xx10xx0x	1	×	√	√	√
P22	P22	Input	ADPC = 01H to 03H	×	1	×	√	√	√
		Output	ADPC = 01H to 03H	×	0	0/1			
	ANI2	Analog input	ADPC = 00H/04H to 0FH	xxx0xx0x	1	×	√	√	√
	CMP0P	Analog input	ADPC = 00H/04H to 0FH	xxx0xx0x	1	×	√	√	√
P24	P24	Input	ADPC = 01H to 05H	×	1	×	√	√	√
		Output	ADPC = 01H to 05H	×	0	0/1			
	ANI4	Analog input	ADPC = 00H/06H to 0FH	xxx0xx0x	1	×	√	√	√
	CMP1P	Analog input	ADPC = 00H/06H to 0FH	xxx0xx0x	1	×	√	√	√
P25	P25	Input	ADPC = 01H to 06H	×	1	×	√	√	√
		Output	ADPC = 01H to 06H	×	0	0/1			
	ANI5	Analog input	ADPC = 00H/07H to 0FH	xxx0xx0x	1	×	√	√	√
	CMP2P	Analog input	ADPC = 00H/07H to 0FH	xxx0xx0x	1	×	√	√	√
P26	P26	Input	ADPC = 01H to 07H	×	1	×	×	√	√
		Output	ADPC = 01H to 07H	×	0	0/1			
	ANI6	Analog input	ADPC = 00H/08H to 0FH	xxx0xx0x	1	×	×	√	√
	CMP3P	Analog input	ADPC = 00H/08H to 0FH	xxx0xx0x	1	×	×	√	√
P27	P27	Input	ADPC = 01H to 08H	×	1	×	×	√	√
		Output	ADPC = 01H to 08H	×	0	0/1			
	ANI7	Analog input	ADPC = 00H/09H to 0FH	xxx0xx0x	1	×	×	√	√
	CMP4P	Analog input	ADPC = 00H/09H to 0FH	xxx0xx0x	1	×	×	√	√

Table 4-5. Setting Examples of Registers and Output Latches When Using Pin Function (4/7)

Pin Name	Used Function		PIOR×	POM××	PMC××	PM××	P××	Alternate Function Output		20-pin	30-pin	38-pin
	Function Name	I/O						SAU Output Function	Other than SAU			
P30	P30	Input	–	–	–	1	×	×	×	×	×	√
		Output	–	–	–	0	0/1	×	RTC1HZ = 0Note 3			
	INTP3	Input	–	–	–	1	×	×	×	×	×	√
	RTC1HZ	Output	–	–	–	0	0	SCK11/ SCL11 = 1 Note 5	×	×	×	√
P31	P31	Input	–	–	–	1	×	×	×	×	√	√
		Output	–	–	–	0	0/1	×	TO03 = 0			
	TI03	Input	×	–	–	1	×	×	×	×	√	√
	TO03	Output	×	–	–	0	0	×	×	×	√	√
	INTP4	Input	×	–	–	1	×	×	×	×	√	√
P40	P40	Input	–	–	–	1	×	×	×	√	√	√
	TOOL0	I/O	×	–	–	×	×	×	×	√	√	√
P75	P75	Input	–	–	–	1	×	×	×	×	×	√
		Output	–	–	–	0	0/1	×	×	×	×	√
	INTP9	Input	×	–	–	1	×	×	×	×	×	√
P76	P76	Input	–	–	–	1	×	×	×	×	×	√
		Output	–	–	–	0	0/1	×	×	×	×	√
	INTP10	Input	×	–	–	1	×	×	×	×	×	√
P77	P77	Input	–	–	–	1	×	×	×	×	√	√
		Output	–	–	–	0	0/1	×	×	×	√	√
	INTP11	Input	×	–	–	1	×	×	×	×	√	√
P120	P120	Input	–	–	0	1	×	×	×	×	√	√
		Output	–	–	0	0	0/1	×	×	×	√	√
	ANI19	Analog input	×	–	1	1	×	×	×	×	√	√

Table 4-5. Setting Examples of Registers and Output Latches When Using Pin Function (5/7)

Pin Name	Used Function		CMC (EXCLK,OSCSEL, EXCLKS, OSCSELS)	P××	20-pin	30-pin	38-pin
	Function Name	I/O					
P121	P121	Input	00xx/10 xx/11 xx	×	√	√	√
	X1	–	01 xx	–	√	√	√
P122	P122	Input	00 xx/10 xx	×	√	√	√
	X2	–	01 xx	–	√	√	√
	EXCLK	Input	11 xx	–	√	√	√
P123	P123	Input	xx 00/xx 10/xx11	×	×	×	√
	XT1	–	xx 01	–	×	×	√
P124	P124	Input	xx 00/xx 10	×	×	×	√
	XT2	–	xx 01	–	×	×	√
	EXCLKS	Input	xx 11	–	×	×	√

Table 4-5. Setting Examples of Registers and Output Latches When Using Pin Function (6/7)

Pin Name	Used Function		PIOR×	POM××	PMC××	PM××	P××	Alternate Function Output		20-pin	30-pin	38-pin
	Function Name	I/O						SAU Output Function	Other than SAU			
P137	P137	Input	–	–	–	–	×	×	×	√	√	√
	INTP0	Input	×	–	–	1	×	×	×	√	√	√
P147	P147	Input	–	–	0	1	×	×	×	√	√	√
		Output	–	–	0	0	0/1	×	×	√	√	√
	ANI18	Analog input	–	–	1	1	×	×	×	√	√	√
	CMPCOM	Input	×	–	1	1	×	×	×	√	√	√

Table 4-5. Setting Examples of Registers and Output Latches When Using Pin Function (7/7)

Pin Name	Used Function		PIOR×	POM××	PMC××	PM××	P××	Alternate Function Output		20-pin	30-pin	38-pin
	Function Name	I/O						SAU Output Function	Other than SAU			
P200	P200	Input	–	×	–	1	0/1	×	×	√	√	√
		Output	–	0	–	0	0	×	TKBO00 = 0	√	√	√
		N-ch open drain output	–	1	–	0	0	×		√	√	√
	TKBO00	Output	×	0	–	0	0	×	×	√	√	√
	INTP22	Input	×	×	–	1	×	×	×	√	√	√
P201	P201	Input	–	×	–	1	0/1	×	×	√	√	√
		Output	–	0	–	0	0	×	TKBO01 = 0	√	√	√
		N-ch open drain output	–	1	–	0	0	×		√	√	√
	TKBO01	Output	×	0	–	0	0	×	×	√	√	√
P202	P202	Input	–	×	–	1	0/1	×	×	√	√	√
		Output	–	0	–	0	0	×	TKBO10 = 0	√	√	√
		N-ch open drain output	–	1	–	0	0	×		√	√	√
	TKBO10	Output	×	0	–	0	0	×	×	√	√	√
	(INTP21)	Input	PIOR13 = 1	×	–	1	×	×	×	√	√	√
P203	P203	Input	–	×	–	1	0/1	×	×	√	√	√
		Output	–	0	–	0	0	×	TKBO11 = 0 TKCO02 = 0	√	√	√
		N-ch open drain output	–	1	–	0	0	×		√	√	√
	TKBO11	Output	×	0	–	0	0	×	×	√	√	√
	TKC02	Output	×	0	–	0	0	×	×	√	√	√
	(INTP21)	Input	PIOR12 = 1	×	–	1	×	×	×	√	√	√
P204	P204	Input	–	×	–	1	0/1	×	×	×	√	√
		Output	–	0	–	0	0	×	TKBO20 = 0 TKCO03 = 0	×	√	√
		N-ch open drain output	–	1	–	0	0	×		×	√	√
	TKBO20	Output	×	0	–	0	0	×	×	×	√	√
	TKC03	Output	PIOR10 = 0	0	–	0	0	×	×	×	√	√
P205	P205	Input	–	×	–	1	0/1	×	×	×	√	√
		Output	–	0	–	0	0	DALITxD4 = 1	TKBO21 = 0 TKCO04 = 0	×	√	√
		N-ch open drain output	–	1	–	0	0			×	√	√
	TKBO21	Output	×	0	–	0	0	×	×	×	√	√
	TKC04	Output	×	0	–	0	0	×	×	×	√	√
	DALITxD4	Output	PIOR11 = 0	0/1	–	0	1	×	×	×	√	√
P206	P206	Input	–	×	–	1	0/1	×	×	×	√	√
		Output	–	0	–	0	0	TxRx4 = 1	TKCO05 = 0	×	√	√
		N-ch open drain output	–	1	–	0	0			×	√	√
	TKCO05	Output	×	0	–	0	0	×	×	×	√	√
	DALIRxD4	Input	PIOR11 = 0	×	–	1	×	×	×	×	√	√
	TxRx4	I/O	×	0/1	–	0	1	×	×	×	√	√
	INTP23	Input	×	×	–	1	×	×	×	×	√	√

4.6 Cautions When Using Port Function

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

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

Therefore, it is recommended to rewrite the output latch when switching a port from input mode to output mode.

<Example> When P200 is an output port, P201 to P206 are input ports (all pin statuses are high level), and the port latch value of port 20 is 00H, if the output of output port P200 is changed from low level to high level via a 1-bit manipulation instruction, the output latch value of port 20 is 7FH.

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

A 1-bit manipulation instruction is executed in the following order in the RL78/I1A.

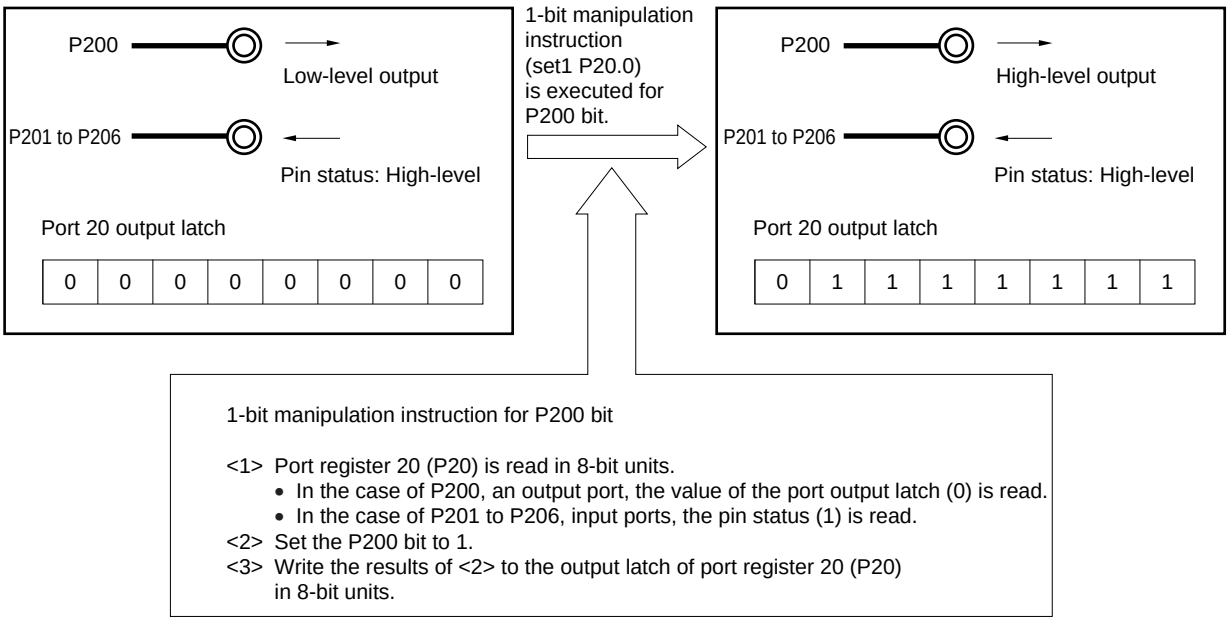
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

In step <1>, the output latch value (0) of P200, which is an output port, is read, while the pin statuses of P201 to P206, which are input ports, are read. If the pin statuses of P201 to P206 are high level at this time, the read value is 7EH.

The value is changed to 7FH by the manipulation in <2>.

7FH is written to the output latch by the manipulation in <3>.

Figure 4-10. Bit Manipulation Instruction (P200)



4.6.2 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 its initial state so as to prevent conflicting outputs. This also applies to the functions assigned by using the peripheral I/O redirection register (PIOR1). For details about the alternate function output, see **4.5 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).

CHAPTER 5 CLOCK GENERATOR

The presence or absence of connecting resonator pin for main system clock, connecting resonator pin for subsystem clock, external clock input pin for main system clock, and external clock input pin for subsystem clock, depends on the product.

Output pin	20, 30-pin	38-pin
X1, X2 pins	√	√
EXCLK pin	√	√
XT1, XT2 pins	–	√
EXCLKS pin	–	√

Note The 20 and 30-pin products don't have the subsystem clock.

5.1 Functions of Clock Generator

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

(1) Main system clock

<1> X1 oscillator

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

<2> High-speed on-chip oscillator

The frequency at which to oscillate can be selected from among f_{IH} = 32, 24, 16, 12, 8, 6, 4, 3, 2, or 1 MHz (typ.) by using the option byte (000C2H). After a reset release, the CPU always starts operating with this high-speed on-chip oscillator clock. Oscillation can be stopped by executing the STOP instruction or setting the HIOSTOP bit (bit 0 of the CSC register).
The frequency specified by using an option byte can be changed by using the high-speed on-chip oscillator frequency select register (HOCODIV). For details about the frequency, see **Figure 5-12 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)[Note]									
	1	2	3	4	6	8	12	16	24	32
2.7 V ≤ V_{DD} ≤ 5.5 V	√	√	√	√	√	√	√	√	√	√

Note The frequency at which the RL78/I1A can operate depends on the operating ambient temperature. For details, see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T_A = –40 to +105°C)** and **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T_A = –40 to +125°C).**

An external main system clock (f_{EX} = 1 to 20 MHz) can also be supplied from the EXCLK/X2/P122 pin. An external main system clock input can be disabled by executing the STOP instruction or setting of the MSTOP bit.
As the main system clock, a high-speed system clock (X1 clock or external main system clock) or high-speed on-chip oscillator clock can be selected by setting of the MCM0 bit (bit 4 of the system clock control register (CKC)).

The main system clock, the range of frequencies that can be used by the power supply voltage is different. Thus, depending on whether the option byte (000C2H) is COMODE0 or MODE1, it is necessary to set the flash operation voltage mode. For details, see CHAPTER 27 VOLTAGE DETECTOR.

<3> External main system clock

An external main system clock (f_{EX} = 1 to 20 MHz) can also be supplied from the EXCLK/X2/P122 pin. An external main system clock input can be disabled by executing the STOP instruction or setting of the MSTOP bit.

<4> Multiplication function using PLL (phase locked loop)

If 4 MHz is selected for the high-speed system clock or high-speed on-chip oscillator clock, the PLL mode can be used. In this mode, a clock of sixteen times the high-speed system clock or high-speed on-chip oscillator clock (64 MHz) can be supplied as the count clock of 16-bit timers KB0 to KB2, KC0, and a comparator, programmable gain amplifier, and a clock of sixteen times the high-speed system clock or high-speed on-chip oscillator clock times 1/2 (32 MHz) or clock of sixteen times the high-speed system clock or high-speed on-chip oscillator clock times 1/4 (16 MHz) can be supplied for all other clocks. Oscillation can be stopped by setting the PLLON bit (bit 0 of the PLLCTL register)

Caution When PLL output has been selected for the main system clock, STOP mode cannot be set. After stopping the PLL function (SELPLL = 0 → PLLON = 0 in the PLL control register (PLLCTL)), select the high-speed on-chip oscillator clock (f_{IH}) or the high-speed system clock (f_{MX}) as the main system clocks, then execute the STOP instruction.

(2) Subsystem clock

- **XT1 clock oscillator**

This circuit oscillates a clock of f_{XT} = 32.768 kHz by connecting a 32.768 kHz resonator to the XT1 and XT2 pins. 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. An external subsystem clock input can be disabled by setting of the XTSTOP bit.

(3) Low-speed on-chip oscillator clock

This circuit oscillates a clock of f_{IL} = 15 kHz (TYP.).

The low-speed on-chip oscillator clock cannot be used as the CPU clock.

Only the following peripheral hardware runs on the low-speed on-chip oscillator clock.

- Watchdog timer
- Real-time clock
- 12-bit interval timer

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

However, when WDTON = 1, WUTMMCK0 = 0, and bit 0 (WDSTBYON) of the option byte (000C0H) is 0, oscillation of the low-speed on-chip oscillator stops if the HALT or STOP instruction is executed.

Caution The low-speed on-chip oscillator clock (f_{IL}) can only be selected as the real-time clock count clock when the fixed-cycle interrupt function is used.

Remark f_X: X1 clock oscillation frequency
f_{IH}: High-speed on-chip oscillator clock frequency
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

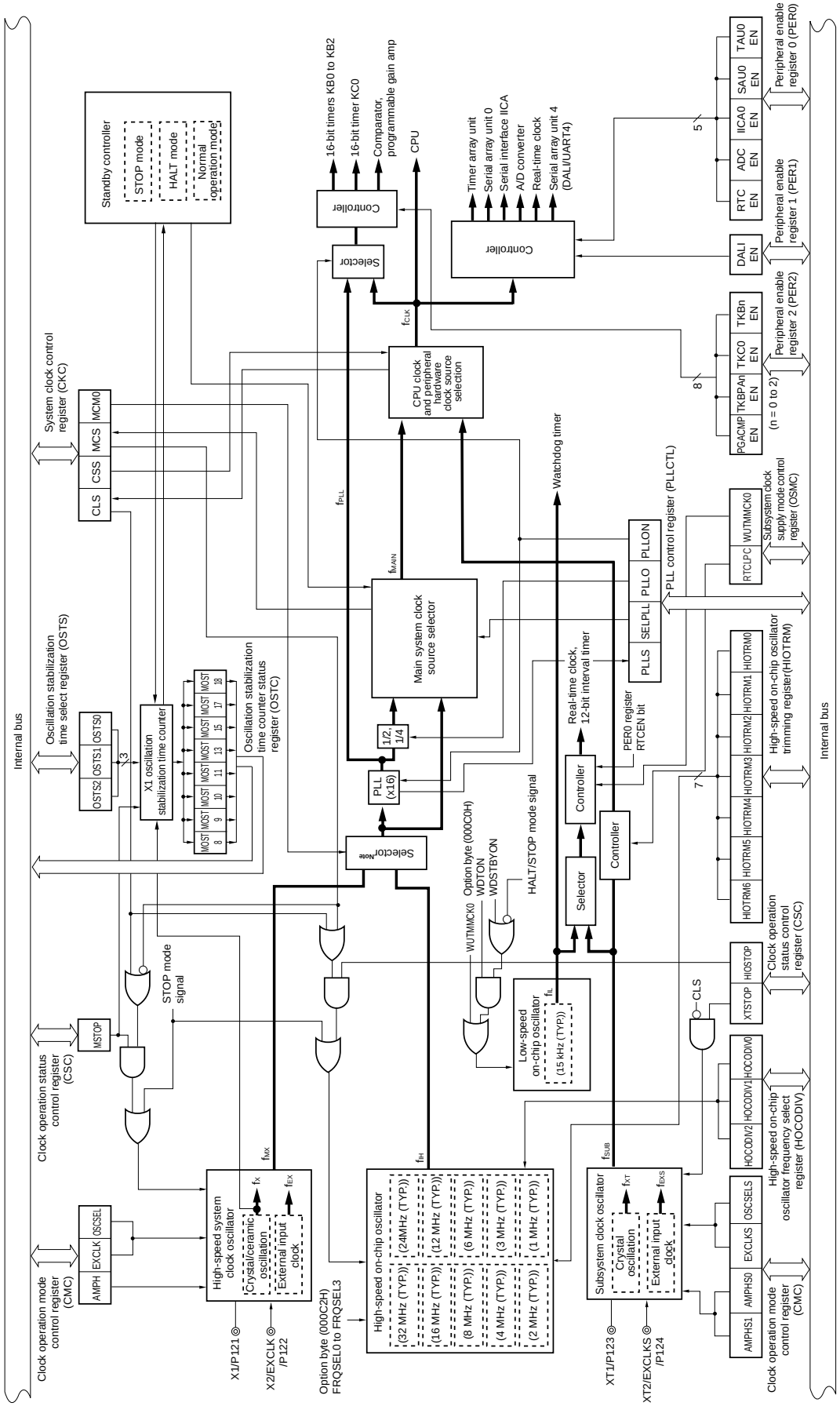
5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control registers	Clock operation mode control register (CMC) System clock control register (CKC) Clock operation status control register (CSC) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS) PLL control register (PLLCTL) Peripheral enable registers 0 to 2 (PER0 to PER2) Subsystem clock supply mode control register (OSMC) High-speed on-chip oscillator frequency select register (HOCODIV) High-speed on-chip oscillator trimming register (HIOTRM)
Oscillators	X1 oscillator XT1 oscillator High-speed on-chip oscillator Low-speed on-chip oscillator

Figure 5-1. Block Diagram of Clock Generator



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

Caution When using the PLL output function, only 4 MHz can be selected as the oscillation frequency.

Remark f_X: X1 clock oscillation frequency
f_{IH}: High-speed on-chip oscillator clock frequency
f_{EX}: External main system clock frequency
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency
f_{XT}: XT1 clock oscillation frequency
f_{EXS}: External subsystem clock frequency
f_{SUB}: Subsystem clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency
f_{PLL}: PLL output clock frequency

5.3 Registers Controlling Clock Generator

The following ten 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)
- PLL control register (PLLCTL)
- Peripheral enable registers 0 to 2 (PER0 to PER2)
- Subsystem clock supply mode control register (OSMC)
- High-speed on-chip oscillator frequency select register (HOCODIV)
- High-speed on-chip oscillator trimming register (HIOTRM)

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

5.3.1 Clock operation mode control register (CMC)

This register is used to set the operation mode of the X1/P121, X2/EXCLK/P122, XT1/P123, and XT2/EXCLKS/P124 pins, and to select a gain of the oscillator.

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

Reset signal generation clears this register to 00H.

Figure 5-2. Format of Clock Operation Mode Control Register (CMC)

Address: FFFA0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS	OSCSELS	0	AMPHS1	AMPHS0	AMPH

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

EXCLKS	OSCSELS	Subsystem clock pin operation mode	XT1/P123 pin	XT2/EXCLKS/P124 pin
0	0	Input port mode	Input port	
0	1	XT1 oscillation mode	Crystal resonator connection	
1	0	Input port mode	Input port	
1	1	External clock input mode	Input port	External clock input

AMPHS1	AMPHS0	XT1 oscillator oscillation mode selection
0	0	Low power consumption oscillation (default)
0	1	Normal oscillation
1	0	Ultra-low power consumption oscillation
1	1	Setting prohibited

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

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

2. After reset release, set the CMC register before X1 or XT1 oscillation is started as set by the clock operation status control register (CSC).

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

4. Specify the settings for the AMPH, AMPHS1, and AMPHS0 bits while f_{IH} is selected as f_{CLK} after a reset ends (before f_{CLK} is switched to f_{MX} or f_{SUB}).

5. Oscillation stabilization time of f_{XT}, counting on the software.

(**Cautions** and **Remark** are given on the next page.)

Cautions 6. Although the maximum system clock frequency is 32 MHz, the maximum frequency of the X1 oscillator is 20 MHz.

7. 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.**
- **Make the wiring between the XT1 and XT2 pins and the resonators as short as possible, and minimize the parasitic capacitance and wiring resistance. Note this particularly when the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) is selected.**
- **Configure the circuit of the circuit board, using material with little wiring resistance.**
- **Place a ground pattern that has the same potential 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 resonators do not cross with the other signal lines. Do not route the wiring near a signal line through which a high 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.**

Remark fx: X1 clock frequency

5.3.2 System clock control register (CKC)

This register is used to select a CPU/peripheral hardware clock and a main system clock.

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

Reset signal generation sets this register to 00H.

Figure 5-3. Format of System Clock Control Register (CKC)

Address: FFFA4H After reset: 00H R/W[Note 1]

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

CLS	Status of CPU/peripheral hardware clock (f_{CLK})
0	Main system clock (f_{MAIN})
1	Subsystem clock (f_{SUB})

CSS	Selection of CPU/peripheral hardware clock (f_{CLK})
0	Main system clock (f_{MAIN})
1	Subsystem clock (f_{SUB})

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

MCM0[Note 2]	Selection of source clock of PLL output clock (f_{PLL}) or main system clock (f_{MAIN})
0	Selects the high-speed on-chip oscillator clock (f_{IH})
1	Selects the high-speed system clock (f_{MX})

Notes 1. Bits 7 and 5 are read-only.

2. Changing the value of the MCM0 bit is prohibited while the CSS bit is set to 1.

Remark f_{IH}: High-speed on-chip oscillator clock frequency

f_{MX}: High-speed system clock frequency

f_{MAIN}: Main system clock frequency

f_{SUB}: Subsystem clock frequency

Cautions 1. Be sure to set bit 3 to "0".

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 real-time clock, 12-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.

3. If the subsystem clock is used as the peripheral hardware clock, the operations of the A/D converter and IICA are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 32 ELECTRICAL SPECIFICATIONS (T_A = –40 to +105°C) and CHAPTER 33 ELECTRICAL SPECIFICATIONS (T_A = –40 to +125°C) .

5.3.3 Clock operation status control register (CSC)

This register is used to control the operations of the high-speed system clock, high-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.

Reset signal generation sets this register to C0H.

Figure 5-4. Format of Clock Operation Status Control Register (CSC)

Address: FFFA1H After reset: C0H R/W

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

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

XTSTOP	Subsystem clock operation control		
	XT1 oscillation mode	External clock input mode	Input port mode
0	XT1 oscillator operating	External clock from EXCLKS pin is valid	Input port
1	XT1 oscillator stopped	External clock from EXCLKS pin is invalid	

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

Cautions

1. **After reset release, set the clock operation mode control register (CMC) before setting the CSC register.**
2. **Set the oscillation stabilization time select register (OSTS) before setting the MSTOP bit to 0 after releasing reset. Note that if the OSTS register is being used with its default settings, the OSTS register is not required to be set here.**
3. **To start X1 oscillation as set by the MSTOP bit, check the oscillation stabilization time of the X1 clock by using the oscillation stabilization time counter status register (OSTC).**
4. **When starting XT1 oscillation by setting the XTSTOP bit, wait for oscillation of the subsystem clock to stabilize by setting a wait time using software.**
5. **Do not stop the clock selected for the CPU peripheral hardware clock (f_{CLK}) with the OSC register.**
6. **The setting of the flags of the register to stop clock oscillation (invalidate the external clock input) and the condition before clock oscillation is to be stopped are as Table 5-2.**

   Before stopping the clock oscillation, check the conditions before the clock oscillation is stopped.

Table 5-2. Stopping Clock Method

Clock	Condition Before Stopping Clock (Invalidating External Clock Input)	Setting of CSC Register Flags
X1 clock External main system clock	CPU and peripheral hardware clocks operate with a clock other than the high-speed system clock. (CLS = 0 and MCS = 0, or CLS = 1)	MSTOP = 1
XT1 clock External subsystem clock	CPU and peripheral hardware clocks operate with a clock other than the subsystem clock. (CLS = 0)	XTSTOP = 1
High-speed on-chip oscillator clock	CPU and peripheral hardware clocks operate with a clock other than the high-speed on-chip oscillator clock. (CLS = 0 and MCS = 1, or CLS = 1)	HIOSTOP = 1

5.3.4 Oscillation stabilization time counter status register (OSTC)

This is the register that indicates the count status of the X1 clock oscillation stabilization time counter.

The X1 clock oscillation stabilization time can be checked in the following case,

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

The OSTC register can be read by a 1-bit or 8-bit memory manipulation instruction.

When reset signal is generated, the STOP instruction and MSTOP (bit 7 of clock operation status control register (CSC)) = 1 clear the OSTC register to 00H.

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

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

Figure 5-5. Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFFA2H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
OSTC	MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18

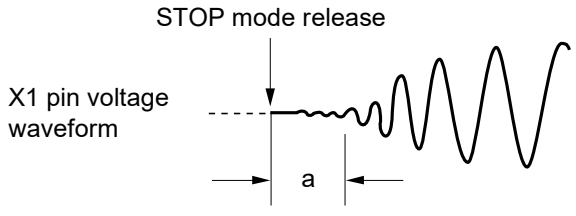
MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status		
									f_X = 10 MHz	f_X = 20 MHz
0	0	0	0	0	0	0	0	$2^8/f_X$ max.	25.6 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	$2^8/f_X$ min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	$2^9/f_X$ min.	51.2 μs min.	25.6 μs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

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

2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by the oscillation stabilization time select register (OSTS).
In the following cases, set the oscillation stabilization time of the OSTS register to the value greater than the count value which is to be checked by the OSTC register.

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

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



Remark f_X: X1 clock oscillation frequency

5.3.5 Oscillation stabilization time select register (OSTS)

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

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

When switching the CPU clock from the high-speed on-chip oscillator clock or the subsystem clock to the X1 clock, and when using the high-speed on-chip oscillator clock for switching the X1 clock from the oscillating state to STOP mode, use the oscillation stabilization time counter status register (OSTC) to confirm that the desired oscillation stabilization time has elapsed after release from the STOP mode. That is, use the OSTC register to check that the oscillation stabilization time corresponding to its setting has been reached.

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

Reset signal generation sets the OSTS register to 07H.

Figure 5-6. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFFA3H After reset: 07H R/W

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

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection		
				f_X = 10 MHz	f_X = 20 MHz
0	0	0	$2^8/f_X$	25.6 μs	12.8 μs
0	0	1	$2^9/f_X$	51.2 μs	25.6 μs
0	1	0	$2^{10}/f_X$	102 μs	51.2 μs
0	1	1	$2^{11}/f_X$	204 μs	102 μs
1	0	0	$2^{13}/f_X$	819 μs	409 μs
1	0	1	$2^{15}/f_X$	3.27 ms	1.64 ms
1	1	0	$2^{17}/f_X$	13.1 ms	6.55 ms
1	1	1	$2^{18}/f_X$	26.2 ms	13.1 ms

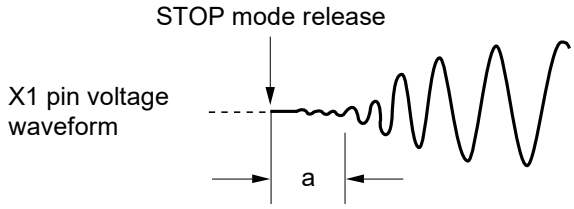
Cautions 1. Change the setting of the OSTS register before setting the MSTOP bit of the clock operation status control register (CSC) to 0.

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

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

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

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



Remark f_X: X1 clock oscillation frequency

5.3.6 PLL control register (PLLCTL)

This register controls the PLL function.

When the PLL function is used, select only 4 MHz for the high-speed system clock or high-speed on-chip oscillator clock.

Stop PLL function when the regulator is in the low-consumption current mode

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

Reset signal generation clears this register to 00H.

Figure 5-7. Format of PLL Control Register (PLLCTL)

Address: F050AH After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	<1>	<0>
PLLCTL	0	0	0	0	PLLS[Note]	SELPLL	PLLO	PLLON

PLLS	PLL output clock supply status flag
0	Not supplied
1	Supplied (The CPU/peripheral hardware clock (f_{CLK}) operates on 16 MHz (PLLO = 1) or 32 MHz (PLLO = 0).)

SELPLL	Selection of PLL output clock for CPU/peripheral hardware clock (f_{CLK})
0	Does not select PLL output clock (clock selected by the system clock control register (CKC) is supplied to f_{CLK})
1	Selects PLL output clock (16 MHz (PLLO = 1) or 32 MHz (PLLO = 0) is supplied to f_{CLK})

PLLO	PLL output division bit
0	Clock divided by 2 (32 MHz)
1	Clock divided by 4 (16 MHz)

PLLON	Operating or stopping PLL function
0	Stopped
1	Operating

Note Bit 3 is read-only.

Cautions 1. The main system clock must be operated (MSTOP = 0 or HIOSTOP = 0) when PLLON = 1.

2. Do not access the following peripheral function registers or the PER2 register until the CPU/peripheral hardware clock switches to PLL output (PLLS = 1), which occurs when SELPLL is set to 1 after the PLL circuit starts operating (PLLON = 1).

- **16-bit timer KB0 to KB2**
- **16-bit timer KC0**
- **Comparator**
- **Programmable gain amplifier**

5.3.7 Peripheral enable registers 0 to 2 (PER0 to PER2)

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

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

- Real-time clock, 12-interval timer
- A/D converter
- Serial interface IICA
- Serial array unit 0
- Serial array unit 4 (DALI/UART4)
- Timer array unit 0
- 16-bit timers KB0 to KB2
- Forced output stop function by the 16-bit timers KB0 to KB2
- 16-bit timer KC0
- Comparator/programmable gain amplifier

The PER0 to PER2 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 5-8. Format of Peripheral Enable Register 0 (PER0) (1/2)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	3	<2>	1	<0>
PER0	RTCEN	0	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

RTCEN	Control of real-time clock (RTC) and 12-interval timer input clock supply
0	Stops input clock supply. • SFR used by the real-time clock (RTC) and 12-interval timer cannot be written. • The real-time clock (RTC) and 12-bit interval timer are in the reset status.
1	Enables input clock supply. • SFR used by the real-time clock (RTC) and 12-interval timer can be read and written.

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

Caution Be sure to clear bits 1, 3, and 6 to “0”.

Figure 5-8. Format of Peripheral Enable Register 0 (PER0) (2/2)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	3	<2>	1	<0>
PER0	RTCEN	0	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

IICA0EN	Control of serial interface IICA input clock supply
0	Stops input clock supply. • SFR used by the serial interface IICA cannot be written. • The serial interface IICA is in the reset status.
1	Enables input clock supply. • SFR used by the serial interface IICA can be read and written.

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

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

Caution Be sure to clear bits 1, 3, and 6 to “0”.

Figure 5-9. Format of Peripheral Enable Register 1 (PER1)

Address: F0508H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
PER1	0	0	0	0	0	0	0	DALIEN

DALIEN	Control of serial array unit 4 (DALI/UART4) input clock supply
0	Stops input clock supply. • SFR used by the serial array unit 4 (DALI/UART4) cannot be written. • The serial array unit 4 (DALI/UART4) is in the reset status.
1	Enables input clock supply. • SFR used by the serial array unit 4 (DALI/UART4) can be read and written.

Caution Be sure to clear bits 1 to 7 to “0”.

Figure 5-10. Format of Peripheral Enable Register 2 (PER2)

Address: F0509H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER2	PGACMPEN	TKBPA2EN	TKBPA1EN	TKBPA0EN	TKC0EN	TKB2EN[Note]	TKB1EN	TKB0EN

PGACMPEN	Control of comparator/programmable gain amplifier input clock supply
0	Stops input clock supply. • SFR used by the comparator/programmable gain amplifier cannot be written. • The comparator/programmable gain amplifier is in the reset status.
1	Enables input clock supply. • SFR used by the comparator/programmable gain amplifier can be read and written.

TKBPAnEN	Control of forced output stop function of 16-bit timer KBn input clock supply (n = 0 to 2)
0	Stops input clock supply. • SFR used by the forced output stop function of the 16-bit timer KBn cannot be written. • The forced output stop function of the 16-bit timer KBn is in the reset status.
1	Enables input clock supply. • SFR used by the forced output stop function of the 16-bit timer KBn can be read and written.

TKC0EN	Control of timer KC0 input clock
0	Stops supply of input clock. • SFR used by timer KC0 cannot be written. • Timer KC0 is in the reset status.
1	Supplies input clock. • SFR used by timer KC0 can be read/written.

TKBnEN	Control of 16-bit timer KBn input clock supply (n = 0 to 2)
0	Stops input clock supply. • SFR used by the 16-bit timer KBn cannot be written. • The 16-bit timer KBn is in the reset status.
1	Enables input clock supply. • SFR used by the 16-bit timer KBn can be read and written.

Note 30-pin and 38-pin products only.

5.3.8 Subsystem clock supply mode control register (OSMC)

This 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 real-time clock and 12-bit interval timer, is stopped in STOP mode or HALT mode while subsystem clock is selected as CPU clock.

In addition, the OSMC register can be used to select the count clock of the real-time clock and 12-bit interval timer.

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

Reset signal generation clears this register to 00H.

Figure 5-11. Format of Subsystem Clock Supply Mode Control Register (OSMC)

Address: F00F3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
OSMC	RTCLPC	0	0	WUTMMCK0	0	0	0	0

RTCLPC	Setting in STOP mode or HALT mode while subsystem clock is selected as CPU clock
0	Enables supply of subsystem clock to peripheral functions (See **Table 21-1** for peripheral functions whose operations are enabled.)
1	Stops supply of subsystem clock to peripheral functions other than real-time clock and 12-bit interval timer.

WUTMMCK0	Selection of count clock for real-time clock and 12-bit interval timer.
0	Subsystem clock (f_{SUB})
1	Low-speed on-chip oscillator clock (f_{IL})

Caution The subsystem clock (f_{SUB}) can be selected as the operating clock of the real-time clock and the 12-bit interval timer with WUTMMCK0 bit only for 38-pin products.

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

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

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

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

Figure 5-12. Format of High-speed On-chip Oscillator Frequency Select Register (HOCODIV)

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

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

HOCODIV2	HOCODIV1	HOCODIV0	High-speed On-chip Oscillator Clock Frequency	
			FRQSEL3 Bit is 0	FRQSEL3 Bit of is 1
0	0	0	24 MHz	32 MHz
0	0	1	12 MHz	16 MHz
0	1	0	6 MHz	8 MHz
0	1	1	3 MHz	4 MHz
1	0	0	Setting prohibited	2 MHz
1	0	1	Setting prohibited	1 MHz
Other than above			Setting prohibited	

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

Option Byte (000C2H) Value		Flash Operation Mode	Operating Frequency Range	Operating Voltage Range
CMODE1	CMODE0			
1	0	LS (low-speed main) mode	1 to 8 MHz	2.7 to 5.5 V
1	1	HS (high-speed main) mode	1 to 32 MHz	

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

3. After the frequency is changed with the HOCODIV register, the frequency is switched after the following transition time has elapsed.

- **Operation for up to three clocks at the pre-change frequency**
- **CPU/peripheral hardware clock wait at the post-change frequency for up to three clocks**

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

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

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

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

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

Figure 5-13. Format of High-speed On-chip Oscillator Trimming Register (HIOTRM)

Address: F00A0H After reset: Undefined[Note] R/W

Symbol	7	6	5	4	3	2	1	0
HIOTRM	0	HIOTRM6	HIOTRM5	HIOTRM4	HIOTRM3	HIOTRM2	HIOTRM1	HIOTRM0

HIOTRM 6	HIOTRM 5	HIOTRM 4	HIOTRM 3	HIOTRM 2	HIOTRM 1	HIOTRM 0	High-speed on-chip oscillator
0	0	0	0	0	0	0	Minimum speed
0	0	0	0	0	0	1	
0	0	0	0	0	1	0	
0	0	0	0	0	1	1	
0	0	0	0	1	0	0	
•	•	•	•	•	•	•	
1	1	1	1	1	1	0	
1	1	1	1	1	1	1	Maximum speed

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

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

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

5.4 System Clock Oscillator

5.4.1 X1 oscillator

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

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

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

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

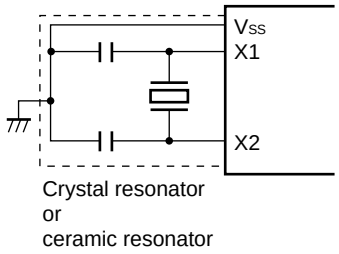
When the X1 oscillator is not used, set the input port mode (EXCLK, OSCSEL = 0, 0).

When the pins are not used as input port pins, either, see **Table 2-2 Connection of Unused Pins (38-pin Products)**.

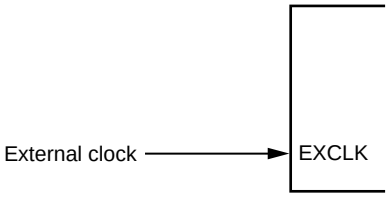
Figure 5-14 shows an example of the external circuit of the X1 oscillator.

Figure 5-14. Example of External Circuit of X1 Oscillator

(a) Crystal or ceramic oscillation



(b) External clock



Caution is listed on the next page.

5.4.2 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 the pins are not used as input port pins, either, see **Table 2-2 Connection of Unused Pins (38-pin Products)**.

Figure 5-15 shows an example of the external circuit of the XT1 oscillator.

Figure 5-15. Example of External Circuit of XT1 Oscillator

(a) Crystal oscillation **(b) External clock**

Caution When using the X1 oscillator and XT1 oscillator, wire as follows in the area enclosed by the broken lines in the Figures 5-14 and 5-15 to avoid an adverse effect from wiring capacitance.

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

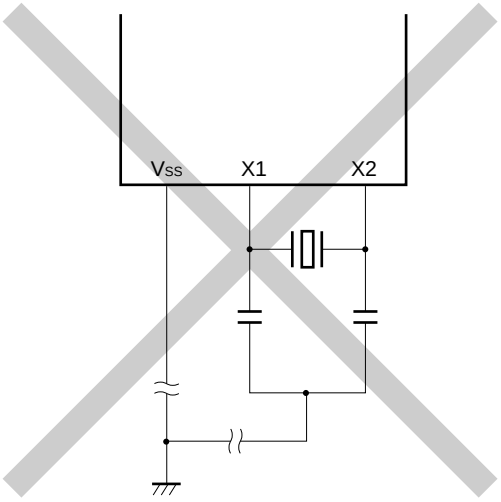
The XT1 oscillator is a circuit with low amplification in order to achieve low-power consumption. Note the following points when designing the circuit.

- **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.**
- **Make the wiring between the XT1 and XT2 pins and the resonators as short as possible, and minimize the parasitic capacitance and wiring resistance. Note this particularly when the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) is selected.**
- **Configure the circuit of the circuit board, using material with little wiring resistance.**
- **Place a ground pattern that has the same potential as Vss as much as possible near the XT1 oscillator.**
- **Be sure that the signal lines between the XT1 and XT2 pins, and the resonators do not cross with the other signal lines. Do not route the wiring near a signal line through which a high 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.**

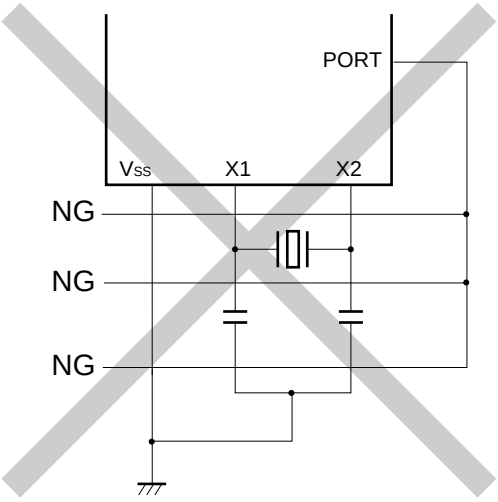
Figure 5-15 shows examples of incorrect resonator connection.

Figure 5-16. Examples of Incorrect Resonator Connection (1/2)

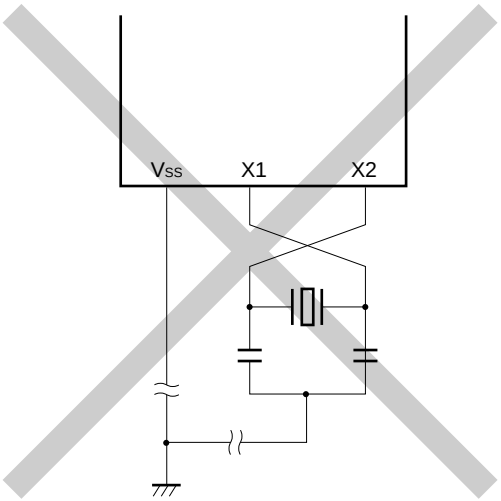
(a) Too long wiring



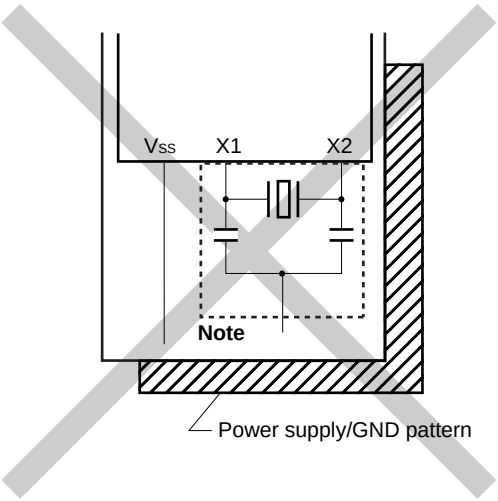
(b) Crossed signal line



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



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

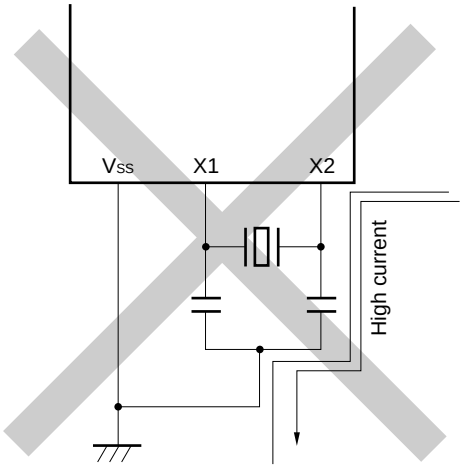


Note Do not place a power supply/GND pattern under the wiring section (section indicated by a broken line in the figure) of the X1 and X2 pins and the resonators in a multi-layer board or double-sided board.
Do not configure a layout that will cause capacitance elements and affect the oscillation characteristics.

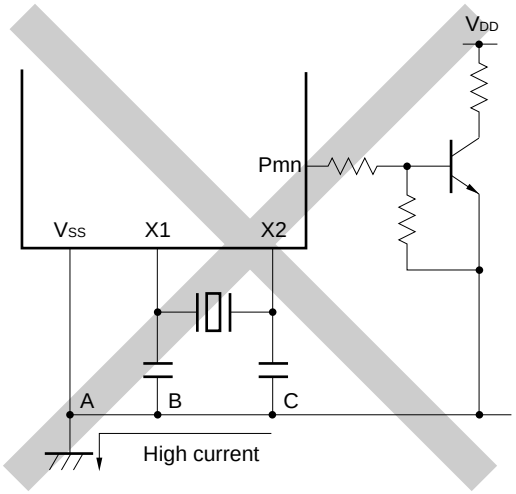
Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

Figure 5-16. Examples of Incorrect Resonator Connection (2/2)

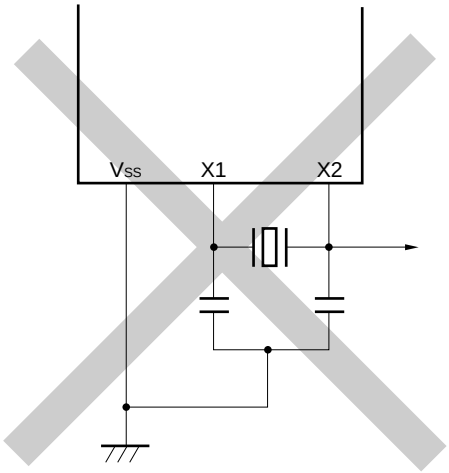
(e) Wiring near high alternating current



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



(g) Signals are fetched



Caution When X2 and XT1 are wired in parallel, the crosstalk noise of X2 may increase with XT1, resulting in malfunctioning.

Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

5.4.3 High-speed on-chip oscillator

The high-speed on-chip oscillator is incorporated in the RL78/I1A. The frequency can be selected from among 32, 24, 16, 12, 8, 6, 4, 3, 2, or 1 MHz by using the 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 reset release.

5.4.4 Low-speed on-chip oscillator

The low-speed on-chip oscillator is incorporated in the RL78/I1A.

The low-speed on-chip oscillator runs while the watchdog timer is operating or when the setting of the following bits is 1: bit 4 (WUTMMCK0) in the subsystem clock supply mode control register (OSMC).

The low-speed on-chip oscillator is stopped when the watchdog timer is stopped and WUTTMMCK0 is set to 0.

5.4.5 PLL (Phase Locked Loop)

The PLL circuit is incorporated in the RL78/I1A.

The PLL can be used to multiply the high-speed on-chip oscillator clock or high-speed system clock.

Operation of the PLL circuit can be controlled by using bit 0 (PLLON) of the PLL control register (PLLCTL).

When the PLL function is used, select only 4 MHz for the high-speed system clock or high-speed on-chip oscillator clock.

Cautions 1. When switching from PLL mode to the high-speed on-chip oscillator clock and the high-speed system clock, stop the functions (timers KB0 to KB2, KC0, and comparator/programmable gain amplifier) that provide the PLL output clock (f_{PLL}).

2. PLL operations cannot be performed while the subsystem clock is operating.

3. When switching to STOP mode, after stopping the PLL function (SELPLL = 0 → PLLON = 0 in the PLL control register (PLLCTL)), select the high-speed on-chip oscillator clock (f_{IH}) or the high-speed system clock (f_{MX}) as the main system clocks, then execute the STOP instruction.

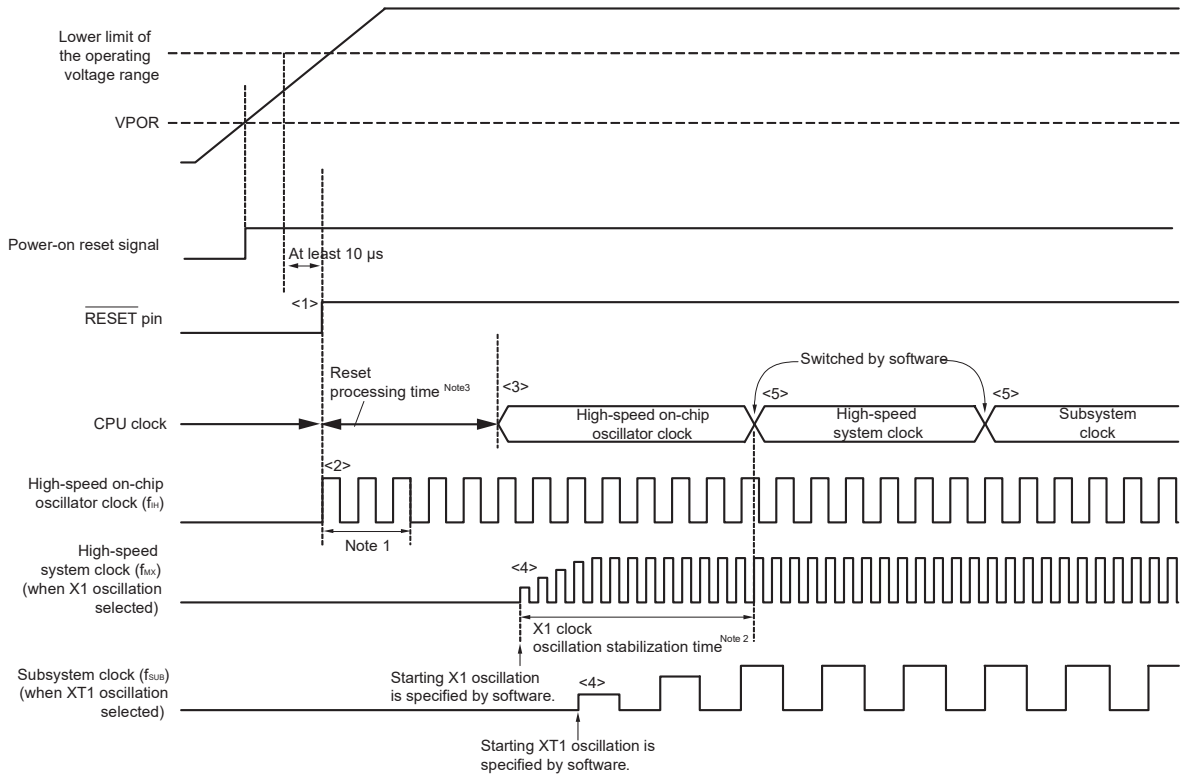
5.5 Clock Generator Operation

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

- Main system clock f_{MAIN}
  - High-speed system clock f_{MX}
    X1 clock f_X
    External main system clock f_{EX}
  - High-speed on-chip oscillator clock f_{IH}
- Subsystem clock 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}
- PLL output clock f_{PLL}

The CPU starts operation when the high-speed on-chip oscillator starts outputting after a reset release in the RL78/I1A. When the power supply voltage is turned on, the clock generator operation is shown in Figure 5-17.

Figure 5-17. 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 **32.4 AC Characteristics** or **33.4 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 on the high-speed on-chip oscillator clock after waiting for the voltage to stabilize and a reset processing have been performed after reset release.

<4> Set the start of oscillation of the X1 or XT1 clock via software (see **5.6.2 Example of setting X1 oscillation clock** and **5.6.3 Example of setting XT1 oscillation clock**).

<5> When switching the CPU clock to the X1 or XT1 clock, wait for the clock oscillation to stabilize, and then set switching via software (see **5.6.2 Example of setting X1 oscillation clock** and **5.6.3 Example of setting XT1 oscillation clock**).

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

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

3. For the reset processing time, see **CHAPTER 23 POWER-ON-RESET CIRCUIT**.

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

Remark See **5.6.4 Example of setting PLL circuit** when using the PLL.

5.6 Controlling Clock

5.6.1 Example of setting high-speed on-chip oscillator

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. The frequency of the high-speed on-chip oscillator can be selected from 32, 24, 16, 12, 8, 6, 4, 3, 2, and 1 MHz by using FRQSEL0 to FRQSEL3 of the option byte (000C2H). The frequency can be changed by using the high-speed on-chip oscillator frequency select register (HOCODIV).

[Option byte setting]

Address: 000C2H

Option byte (000C2H)

7	6	5	4	3	2	1	0
CMODE1	CMODE0			FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0
1	0/1	1	0	0/1	0/1	0/1	0/1

CMODE1	CMODE0	Setting of flash operation mode
1	0	LS (low speed main) mode V_{DD} = 2.7 V to 5.5 V @ 1 MHz to 8 MHz
1	1	HS (high speed main) mode V_{DD} = 2.7 V to 5.5 V @ 1 MHz to 32 MHz
Other than above		Setting prohibited

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Clock frequency of the high-speed on-chip oscillator
1	0	0	0	32 MHz
0	0	0	0	24 MHz
1	0	0	1	16 MHz
0	0	0	1	12 MHz
1	0	1	0	8 MHz
0	0	1	0	6 MHz
1	0	1	1	4 MHz
0	0	1	1	3 MHz
1	1	0	0	2 MHz
1	1	0	1	1 MHz
Other than above				Setting prohibited

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

Address: F00A8H

HOCODIV

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

HOCODIV2	HOCODIV1	HOCODIV0	Selection of high-speed on-chip oscillator clock frequency	
			FRQSEL3 Bit is 0	FRQSEL3 Bit of is 1
0	0	0	24 MHz	32 MHz
0	0	1	12 MHz	16 MHz
0	1	0	6 MHz	8 MHz
0	1	1	3 MHz	4 MHz
1	0	0	Setting prohibited	2 MHz
1	0	1	Setting prohibited	1 MHz
Other than above			Setting prohibited	

5.6.2 Example of setting X1 oscillation clock

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

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

<1> To operate the X1 oscillator, set (1) the OSCSEL bit of the CMC register, and when f_X > 10 MH, set (1) the AMPH bit.

CMC

7	6	5	4	3	2	1	0
EXCLK	OSCSEL	EXCLKS	OSCSELS		AMPHS1	AMPHS0	AMPH
0	1	0	0	0	0	0	0/1

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

<2> Using the OSTS register, select the oscillation stabilization time of the X1 oscillator at releasing of the STOP mode.

Example: Setting values when a wait of at least 102 μs is set based on a 10 MHz resonator.

OSTS

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

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

CSC

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

<4> Use the OSTC register to wait for oscillation of the X1 oscillator to stabilize.

Example: Wait until the bits reach the following values when a wait of at least 102 μs is set based on a 10 MHz resonator.

OSTC

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

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

CKC

7	6	5	4	3	2	1	0
CLS	CSS	MCS	MCM0				
0	0	0	1	0	0	0	0

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

Option Byte (000C2H) Value		Flash Operation Mode	Operating Frequency Range	Operating Voltage Range
CMODE1	CMODE0			
1	0	LS (low-speed main) mode	1 to 8 MHz	2.7 to 5.5 V
1	1	HS (high-speed main) mode	1 to 32 MHz	

5.6.3 Example of setting XT1 oscillation clock

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. To subsequently change the clock to the XT1 oscillation clock, set the oscillator and start oscillation by using the subsystem clock supply mode control register (OSMC), clock operation mode control register (CMC), and clock operation status control register (CSC), set the XT1 oscillation clock to f_{CLK} by using the system clock control register (CKC).

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

<1> Set the RTCLPC bit to 1 to run only the real-time clock and 12-bit interval timer on the subsystem clock (ultra-low current consumption) when in the STOP mode or HALT mode during CPU operation on the subsystem clock.

	7	6	5	4	3	2	1	0
OSMC	RTCLPC			WUTMMCK0				
	0/1	0	0	0	0	0	0	0

<2> Set (1) the OSCSELS bit of the CMC register to operate the XT1 oscillator.

	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS	OSCSELS		AMPHS1	AMPHS0	AMPH
	0	0	0	1	0	0/1	0/1	0

AMPHS0 and AMPHS1 bits: These bits are used to specify the oscillation mode of the XT1 oscillator.

<3> Clear (0) the XTSTOP bit of the CSC register to start oscillating the XT1 oscillator.

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

<4> Use the timer function or another function to wait for oscillation of the subsystem clock to stabilize by using software.

<5> Use the CSS bit of the CKC register to specify the XT1 oscillation clock as the CPU/peripheral hardware clock.

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

5.6.4 Example of setting PLL circuit

After setting the high-speed system clock and high-speed on-chip oscillator clock (see **5.6.1 Example of setting high-speed on-chip oscillator** and **5.6.2 Example of setting X1 oscillation clock**), use the PLL control register (PLLCTL) to control the PLL circuit.

When the PLL function is used, select only 4 MHz for the high-speed system clock or high-speed on-chip oscillator clock.

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

<1> Set the PLLO bit in the PLLCTL register so that f_{PLL} = 64 MHz is set for the 16-bit timers KB0 to KB2 and KC0, the comparator, and the programmable gain amplifier, then 16 MHz (PLLO = 1) or 32 MHz (PLLO = 0) is set for supply to other hardware (CPU and peripherals).

	7	6	5	4	3	2	1	0
PLLCCTL					PLLS	SELPLL	PLLO	PLLON
	0	0	0	0	0	0	0/1	0

Caution Do not access the following peripheral function registers or the PER2 register until the CPU/peripheral hardware clock switches to PLL output (PLLS = 1), which occurs when SELPLL is set to 1 after the PLL circuit starts operating (PLLON = 1).

- **16-bit timer KB0 to KB2**
- **16-bit timer KC0**
- **Comparator**
- **Programmable gain amplifier**

<2> Set (1) the PLLON bit of the PLLCTL register to operate the PLL circuit.

	7	6	5	4	3	2	1	0
PLLCTL					PLLS	SELPLL	PLLO	PLLON
	0	0	0	0	0	0	0/1	1

<3> Wait for 40 μs by using software.

<4> Set the SELPLL bit of the PLLCTL register to 1 to select PLL output for the CPU and peripheral hardware clocks (f_{CLK}).

	7	6	5	4	3	2	1	0
PLLCTL					PLLS	SELPLL	PLLO	PLLON
	0	0	0	0	0	1	0/1	1

<5> Wait until the PLLS bit changes to 1.

5.6.5 CPU clock status transition diagram

Figure 5-18 shows the CPU clock status transition diagram of this product.

Figure 5-18. CPU Clock Status Transition Diagram

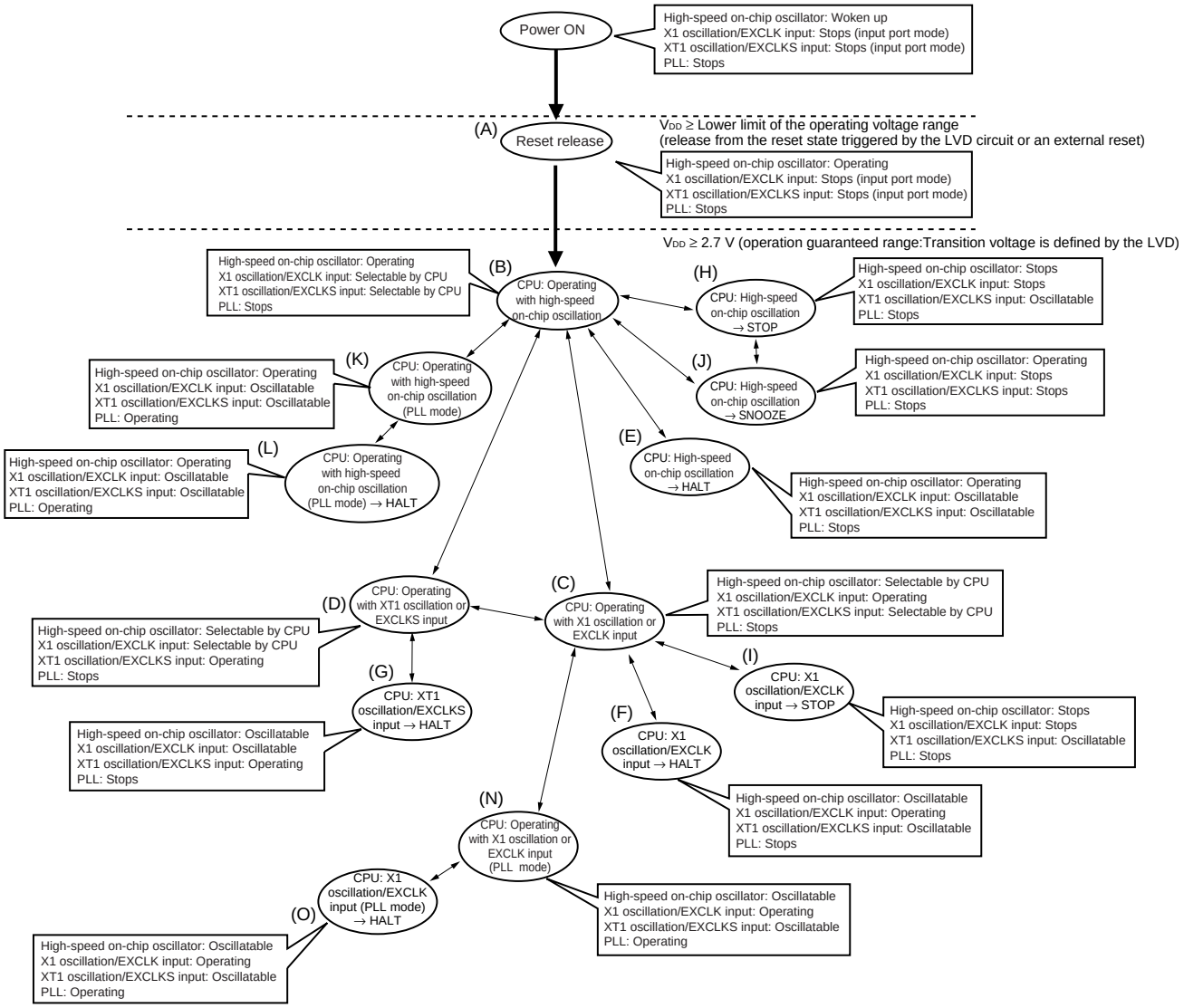


Table 5-3 shows transition of the CPU clock and examples of setting the SFR registers.

Table 5-3. CPU Clock Transition and SFR Register Setting Examples (1/6)

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

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

(2) CPU operating with high-speed system clock (C) after reset release (A)

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

(Setting sequence of SFR registers) →

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

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

2. Set the oscillation stabilization time as follows.

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

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 32 ELECTRICAL SPECIFICATIONS (T_A = –40 to +105°C) and CHAPTER 33 ELECTRICAL SPECIFICATIONS (T_A = –40 to +125°C).

(3) CPU operating with subsystem clock (D) after reset release (A)

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

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register[Note]				CSC Register	Waiting for Oscillation Stabilization	CKC Register
	EXCLKS	OSCSELS	AMPHS1	AMPHS0	XTSTOP		CSS
(A) → (B) → (D) (XT1 clock)	0	1	0/1	0/1	0	Necessary	1
(A) → (B) → (D) (external sub clock)	1	1	×	×	0	Necessary	1

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

Remarks 1. ×: don't care

2. (A) to (O) in Table 5-3 correspond to (A) to (O) in Figure 5-18.

Table 5-3. CPU Clock Transition and SFR Register Setting Examples (2/6)

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

(Setting sequence of SFR registers) →

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

Unnecessary if these registers are already set

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

Notes 1. The clock operation mode control register (CMC) can be written only once by an 8-bit memory manipulation instruction after reset release. This setting is not necessary if it has already been set.

2. Set the oscillation stabilization time as follows.

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

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 32 ELECTRICAL SPECIFICATIONS (T_A = –40 to +105°C) and CHAPTER 33 ELECTRICAL SPECIFICATIONS (T_A = –40 to +125°C).

(5) CPU clock changing from high-speed on-chip oscillator clock (B) to subsystem clock (D)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register[Note]			CSC Register	Waiting for Oscillation Stabilization	CKC Register
	EXCLKS	OSCSELS	AMPHS1, AMPHS0	XTSTOP		CSS
(B) → (D) (XT1 clock)	0	1	00: Low power consumption oscillation 01: Normal oscillation 10: Ultra-low power consumption oscillation	0	Necessary	1
(B) → (D) (external sub clock)	1	1	×	0	Necessary	1

Unnecessary if these registers are already set

Unnecessary if the CPU is operating with the subsystem clock

Note The clock operation mode control register (CMC) can be written only once by an 8-bit memory manipulation instruction after reset release. This setting is not necessary if it has already been set.

Remarks 1. ×: don't care

2. (A) to (O) in Table 5-3 correspond to (A) to (O) in Figure 5-18.

Table 5-3. CPU Clock Transition and SFR Register Setting Examples (3/6)

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

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register	Oscillation Accuracy Stabilization Time	CKC Register
	HIOSTOP		MCM0
(C) → (B)	0	18 μs to 65 μs	0

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

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

(7) CPU clock changing from high-speed system clock (C) to subsystem clock (D)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register	Waiting for Oscillation Stabilization	CKC Register
	XTSTOP		CSS
(C) → (D)	0	Necessary	1

Unnecessary if the CPU is operating with the subsystem clock

(8) CPU clock changing from subsystem clock (D) to high-speed on-chip oscillator clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register	Oscillation Accuracy Stabilization Time	CKC Register
	HIOSTOP		CSS
(D) → (B)	0	18 μs to 65 μs	0

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

Remarks 1. (A) to (O) in Table 5-3 correspond to (A) to (O) in Figure 5-18.

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

Table 5-3. CPU Clock Transition and SFR Register Setting Examples (4/6)

(9) CPU clock changing from subsystem clock (D) to high-speed system clock (C)

(Setting sequence of SFR registers) ⟶

Setting Flag of SFR Register / Status Transition	OSTS Register	CSC Register	OSTC Register	CKC Register
		MSTOP		CSS
(D) → (C) (X1 clock: 1 MHz ≤ f_X ≤ 10 MHz)	**Note**	0	Must be checked	0
(D) → (C) (X1 clock: 10 MHz < f_X ≤ 20 MHz)	**Note**	0	Must be checked	0
(D) → (C) (external main clock)	**Note**	0	Must not be checked	0

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

Note Set the oscillation stabilization time as follows.

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

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 32 ELECTRICAL SPECIFICATIONS (T_A = –40 to +105°C) and CHAPTER 33 ELECTRICAL SPECIFICATIONS (T_A = –40 to +125°C).

Remark (A) to (O) in Table 5-3 correspond to (A) to (O) in Figure 5-18.

Table 5-3. CPU Clock Transition and SFR Register Setting Examples (5/6)

(10)
- **CPU clock changing from high-speed on-chip oscillator clock (B) to high-speed on-chip oscillator clock (PLL mode) (K)**
- **CPU clock changing from high-speed system clock (C) to high-speed system clock (PLL mode) (N)**

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	PLLCTL Register			
	PLLO	PLLON	Waiting for Oscillation Stabilization	SELPLL
(B) → (K) (C) → (N)	0/1	1	40 μs	1

(11)
- **CPU clock changing from high-speed on-chip oscillator clock (PLL mode) (K) to high-speed on-chip oscillator clock (B)**
- **CPU clock changing from high-speed system clock (PLL mode) (N) to high-speed system clock (N)**

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	PLLCTL Register		
	SELPLL	PLLS	PLLON
(K) → (B) (N) → (C)	0	0	0

Caution When switching from PLL mode to the high-speed on-chip oscillator clock and the high-speed system clock, stop the functions (timers KB0 to KB2, KC0, and comparator/programmable gain amplifier) that provide the PLL output clock (PLL).

(12)
- **HALT mode (E) set while CPU is operating with high-speed on-chip oscillator clock (B)**
- **HALT mode (F) set while CPU is operating with high-speed system clock (C)**
- **HALT mode (G) set while CPU is operating with subsystem clock (D)**
- **HALT mode (L) set while CPU is operating with high-speed on-chip oscillator clock (PLL mode) (K)**
- **HALT mode (O) set while CPU is operating with high-speed system clock (PLL mode) (N)**

Status Transition	Setting
(B) → (E) (C) → (F) (D) → (G) (K) → (L) (N) → (O)	Executing HALT instruction

Remark (A) to (O) in Table 5-3 correspond to (A) to (O) in Figure 5-18.

Table 5-3. CPU Clock Transition and SFR Register Setting Examples (6/6)

(13) • STOP mode (H) set while CPU is operating with high-speed on-chip oscillator clock (B)
• STOP mode (I) set while CPU is operating with high-speed system clock (C)

(Setting sequence) ⟶

Status Transition		Setting		
(B) → (H)		Stopping peripheral functions that are disabled in STOP mode	–	Executing STOP instruction
(C) → (I)	In X1 oscillation		Sets the OSTS register	
	External main system clock		–	

(14) CPU changing from STOP mode (H) to SNOOZE mode (J)

See the following sections with regard to settings used with functions that support SNOOZE mode to switch from STOP mode to SNOOZE mode.

A/D converter	→ **12.8 SNOOZE Mode Function**
CSI00 of serial array unit 0	→ **15.5.7 SNOOZE mode function**
UART0 of serial array unit 0	→ **15.6.3 SNOOZE mode function**
UART4 of serial array unit 4	→ **16.5.3 SNOOZE mode function**

(15) • STOP mode (H) set while CPU is operating with high-speed on-chip oscillator clock (PLL mode) (K)
• STOP mode (I) set while CPU is operating with high-speed system clock (PLL mode) (N)

Switch from PLL mode operation to high-speed on-chip oscillator clock and high-speed system clock operations (see **(11) in 5.6.5**), then execute the STOP instruction.

Caution When switching from PLL mode to the high-speed on-chip oscillator clock and the high-speed system clock, stop the functions (timers KB0 to KB2, KC0, and comparator/programmable gain amplifier) that provide the PLL output clock (f_{PLL}).

Remark (A) to (O) in Table 5-3 correspond to (A) to (O) in Figure 5-18.

5.6.6 Condition before changing CPU clock and processing after changing CPU clock

Condition before changing the CPU clock and processing after changing the CPU clock are shown below.

Table 5-4. Changing CPU Clock (1/2)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
High-speed on-chip oscillator clock	X1 clock	Stabilization of X1 oscillation • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time	The operating current can be reduced by stopping the high-speed on-chip oscillator (HIOSTOP = 1) after checking that the CPU clock is changed.
	External main system clock	Enabling input of external clock from the EXCLK pin • OSCSEL = 1, EXCLK = 1, MSTOP = 0	
	XT1 clock	Stabilization of XT1 oscillation • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • After elapse of oscillation stabilization time	
	External subsystem clock	Enabling input of external clock from the EXCLKS pin • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
X1 clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • After elapse of oscillation stabilization time	X1 oscillation can be stopped (MSTOP = 1) after checking that the CPU clock is changed.
	External main system clock	Transition not possible (To change the clock, set it again after executing reset once.)	–
	XT1 clock	Stabilization of XT1 oscillation • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • After elapse of oscillation stabilization time	X1 oscillation can be stopped (MSTOP = 1) after checking that the CPU clock is changed.
	External subsystem clock	Enabling input of external clock from the EXCLKS pin • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	X1 oscillation can be stopped (MSTOP = 1) after checking that the CPU clock is changed.
External main system clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • After elapse of oscillation stabilization time	External main system clock input can be disabled (MSTOP = 1) after checking that the CPU clock is changed.
	X1 clock	Transition not possible	–
	XT1 clock	Stabilization of XT1 oscillation • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • After elapse of oscillation stabilization time	External main system clock input can be disabled (MSTOP = 1) after checking that the CPU clock is changed.
	External subsystem clock	Enabling input of external clock from the EXCLKS pin • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	External main system clock input can be disabled (MSTOP = 1) after checking that the CPU clock is changed.

Table 5-4. Changing CPU Clock (2/2)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
XT1 clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator and selection of high-speed on-chip oscillator clock as main system clock • HIOSTOP = 0, MCS = 0	XT1 oscillation can be stopped (XTSTOP = 1) after checking that the CPU clock is changed.
	X1 clock	Stabilization of X1 oscillation and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time • MCS = 1	
	External main system clock	Enabling input of external clock from the EXCLK pin and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	External subsystem clock	Transition not possible	–
External subsystem clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator and selection of high-speed on-chip oscillator clock as main system clock • HIOSTOP = 0, MCS = 0	External subsystem clock input can be disabled (XTSTOP = 1) after checking that the CPU clock is changed.
	X1 clock	Stabilization of X1 oscillation and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time • MCS = 1	
	External main system clock	Enabling input of external clock from the EXCLK pin and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	XT1 clock	Transition not possible (To change the clock, set it again after executing reset once.)	–

5.6.7 Time required for switchover of CPU clock and system clock

By setting bits 4 and 6 (MCM0, CSS) of the system clock control register (CKC), the CPU clock can be switched (between the main system clock and the subsystem clock), and main system clock can be switched (between the high-speed on-chip oscillator clock and the high-speed system clock).

The actual switchover operation is not performed immediately after rewriting to the CKC register; operation continues on the pre-switchover clock for several clocks (see **Tables 5-5** to **5-7**).

Whether the CPU is operating on the main system clock or the subsystem clock can be ascertained using bit 7 (CLS) of the CKC register. Whether the main system clock is operating on the high-speed system clock or high-speed on-chip oscillator clock can be ascertained using bit 5 (MCS) of the CKC register.

When the CPU clock is switched, the peripheral hardware clock is also switched.

Table 5-5. Maximum Time Required for System Clock Switchover

Clock A	Switching directions	Clock B	Remark
f_{IH}	←→	f_{MX}	See **Table 5-6**
f_{MAIN}	←→	f_{SUB}	See **Table 5-7**

Table 5-6. Maximum Number of Clocks Required for f_{IH} ↔ f_{MX}

Set Value Before Switchover		Set Value After Switchover	
MCM0		MCM0	
		0 ($f_{MAIN} = f_{IH}$)	1 ($f_{MAIN} = f_{MX}$)
0 ($f_{MAIN} = f_{IH}$)	$f_{MX} \geq f_{IH}$		2 clock
	$f_{MX} < f_{IH}$		$2f_{IH}/f_{MX}$ clock
1 ($f_{MAIN} = f_{MX}$)	$f_{MX} \geq f_{IH}$	$2f_{MX}/f_{IH}$ clock	
	$f_{MX} < f_{IH}$	2 clock	

Table 5-7. Maximum Number of Clocks 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 ($f_{CLK} = f_{MAIN}$)		$1 + 2f_{MAIN}/f_{SUB}$ clock
1 ($f_{CLK} = f_{SUB}$)	3 clock	

Remarks 1. The number of clocks listed in Tables 5-6 and 5-7 is the number of CPU clocks before switchover.

2. Calculate the number of clocks in **Tables 5-6** and **5-7** by removing the decimal portion.

Example When switching the main system clock from the high-speed system clock to the high-speed on-chip oscillator clock (@ oscillation with f_{IH} = 8 MHz, f_{MX} = 10 MHz)

$2f_{MX}/f_{IH} = 2\ (10/8) = 2.5 \rightarrow 3$ clocks

5.6.8 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.

Before stopping the clock oscillation, check the conditions before the clock oscillation is stopped.

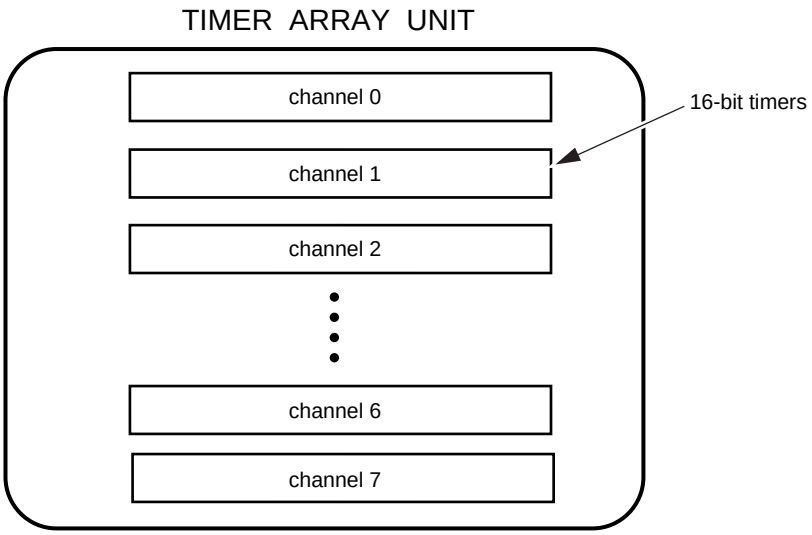
Table 5-7. Conditions Before the Clock Oscillation Is Stopped and Flag Settings

Clock	Conditions Before Clock Oscillation Is Stopped (External Clock Input Disabled)	Flag Settings of SFR Register
High-speed on-chip oscillator clock	MCS = 1 or CLS = 1 (The CPU is operating on a clock other than the high-speed on-chip oscillator clock.)	HIOSTOP = 1
X1 clock External main system clock	MCS = 0 or CLS = 1 (The CPU is operating on a clock other than the high-speed system clock.)	MSTOP = 1
XT1 clock External subsystem clock	CLS = 0 (The CPU is operating on a clock other than the subsystem clock.)	XTSTOP = 1

CHAPTER 6 TIMER ARRAY UNIT

The timer array unit has eight 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 (→ see **6.8.1**) • Square wave output (→see **6.8.1**) • External event counter (→see **6.8.2**) • Input pulse interval measurement (→see **6.8.3**) • Measurement of high-/low-level width of input signal (→see **6.8.4**) • Delay counter (→see **6.8.5**)	• One-shot pulse output(→see **6.9.1**) • PWM output(→see **6.9.2**) • Multiple PWM output(→see **6.9.3**)

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 (higher 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 7 can be used to realize LIN-bus and DMX512 reception processing in combination with UART0 of the serial array unit 0.

6.1 Functions of Timer Array Unit

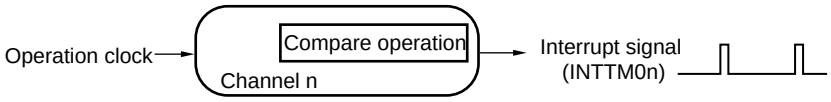
Timer array unit has the following functions.

6.1.1 Independent channel operation function

By operating a channel independently, it can be used for the following purposes without being affected by the operation mode of other channels.

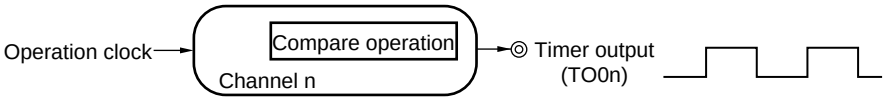
(1) Interval timer

Each timer of a unit can be used as a reference timer that generates an interrupt (INTTM0n) at fixed intervals.



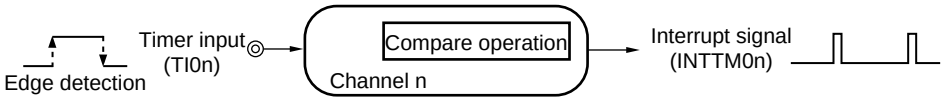
(2) Square wave output

A toggle operation is performed each time INTTM0n interrupt is generated and a square wave with a duty factor of 50% is output from a timer output pin (TO0n).



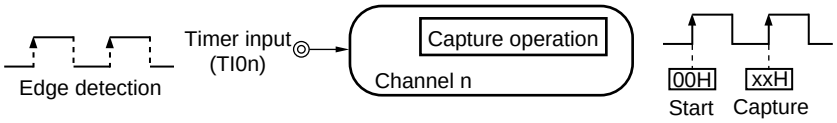
(3) External event counter

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 (TI0n) has reached a specific value.



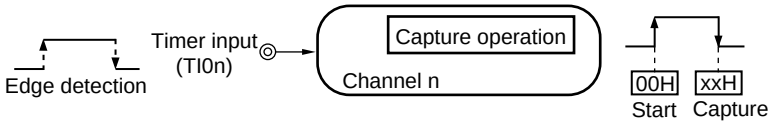
(4) Input pulse interval measurement

Counting is started by the valid edge of a pulse signal input to a timer input pin (TI0n). 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.



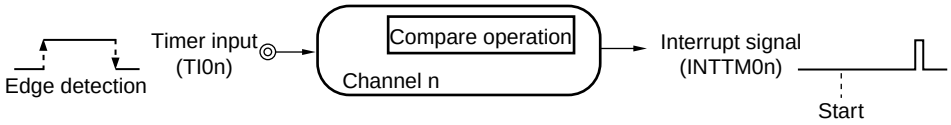
(5) Measurement of high-/low-level width of input signal

Counting is started by a single edge of the signal input to the timer input pin (TI0n), 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.



(6) Delay counter

Counting is started at the valid edge of the signal input to the timer input pin (TI0n), and an interrupt is generated after any delay period.



Remarks 1. n: Channel number (n = 0 to 7)

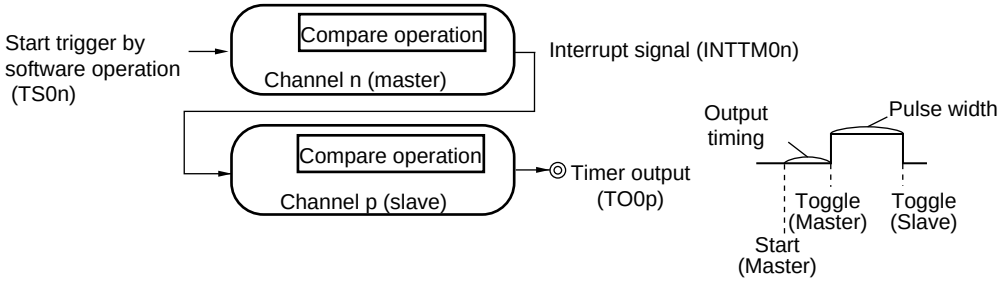
2. The presence or absence of timer I/O pins of channel 0 to 7 depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product** for details.

6.1.2 Simultaneous channel operation function

By using the combination of a master channel (a reference timer mainly controlling the cycle) and slave channels (timers operating according to the master channel), channels can be used for the following purposes.

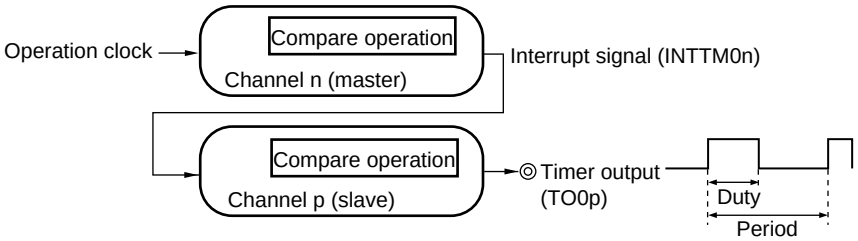
(1) One-shot pulse output

Two channels are used as a set to generate a one-shot pulse with a specified output timing and a specified pulse width.



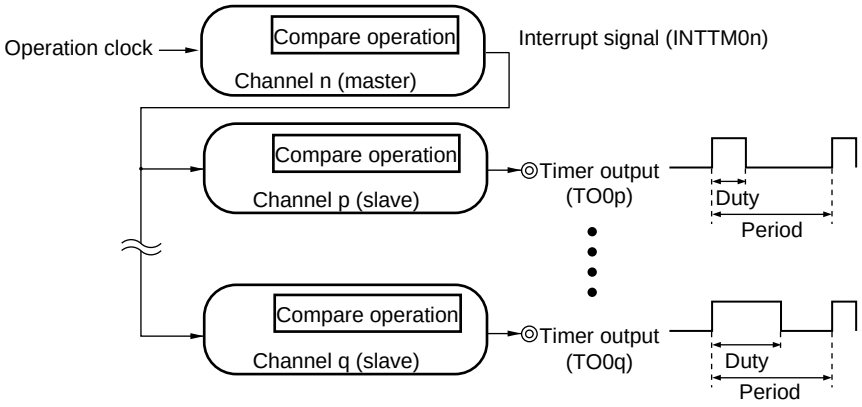
(2) PWM (Pulse Width Modulation) output

Two channels are used as a set to generate a pulse with a specified period and a specified duty factor.



(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 seven 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 6.4.1 Basic rules of simultaneous channel operation function.

6.1.3 LIN-bus supporting function (channel 7 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 sync 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 **6.3.13 Input switch control register (ISC)** and **6.8.4 Operation as input signal high-/low-level width measurement**.

6.1.4 DMX512 supporting function (channel 7 only)

Timer array unit is used to check whether signals received in DMX512 communication match the DMX512 communication format.

(1) Detection of BREAK 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 BREAK signal.

(2) Measurement of signal width

The low-level and high-level width of the signal input to the serial data input pin (RxD0) of UART0 are measured.

Remark For details about setting up the operations used to implement the DMX512, see **6.3.13 Input switch control register (ISC)** and **6.8.4 Operation as input signal high-/low-level width measurement**.

6.2 Configuration of Timer Array Unit

Timer array unit includes the following hardware.

Table 6-1. Configuration of Timer Array Unit

Item	Configuration
Timer/counter	Timer/counter register 0n (TCR0n)
Register	Timer data register 0n (TDR0n)
Timer input	TI03, TI05, TI06 pins[Note 1], TI07/RxD0 pin (for LIN-bus, DMX512)[Note 2]
Timer output	TO03, TO05, TO06 pins[Note 1], output controller
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Timer clock select register 0 (TPS0) • Timer channel enable status register 0 (TE0) • Timer channel start register 0 (TS0) • Timer channel stop register 0 (TT0) • Timer input select register 0 (TIS0) • Timer output enable register 0 (TOE0) • Timer output register 0 (TO0) • Timer output level register 0 (TOL0) • Timer output mode register 0 (TOM0)
	<Registers of each channel> • Timer mode register 0n (TMR0n) • Timer status register 0n (TSR0n) • Input switch control register (ISC) • Noise filter enable register 1 (NFEN1) • Port mode register (PM0, PM3)[Note 2] • Port register (P0, P3)[Note 2]

Notes 1. The presence or absence of timer I/O pins of channel 0 to 7 depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product** for details.

2. The TI07 pin can be used for P11/RxD0 pin by setting bit 1 (ISC1) in the input switch control register (ISC) to 1. For details, see **6.3.13 Input switch control register (ISC)**.

3. The port mode registers (PM0, PM3) and port registers (P0, P3) to be set differ depending on the product. For details, see **4.5.3 Register setting examples for used port and alternate functions.**

Remark n: Channel number (n = 0 to 7)

The presence or absence of timer I/O pins in each timer array unit channel depends on the product.

Table 6-2. Timer I/O Pins provided in Each Product

Timer array unit channels		I/O Pins of Each Product		
		38-pin	30-pin	20-pin
Unit 0	Channel 0	–	–	–
	Channel 1	–	–	–
	Channel 2	–	–	–
	Channel 3	P31/TI03/TO03	P31/TI03/TO03	–
	Channel 4	–	–	–
	Channel 5	P05/TI05/TO05	–	–
	Channel 6	P06/TI06/TO06	–	–
	Channel 7	P11/RxD0/(TI07) Note		

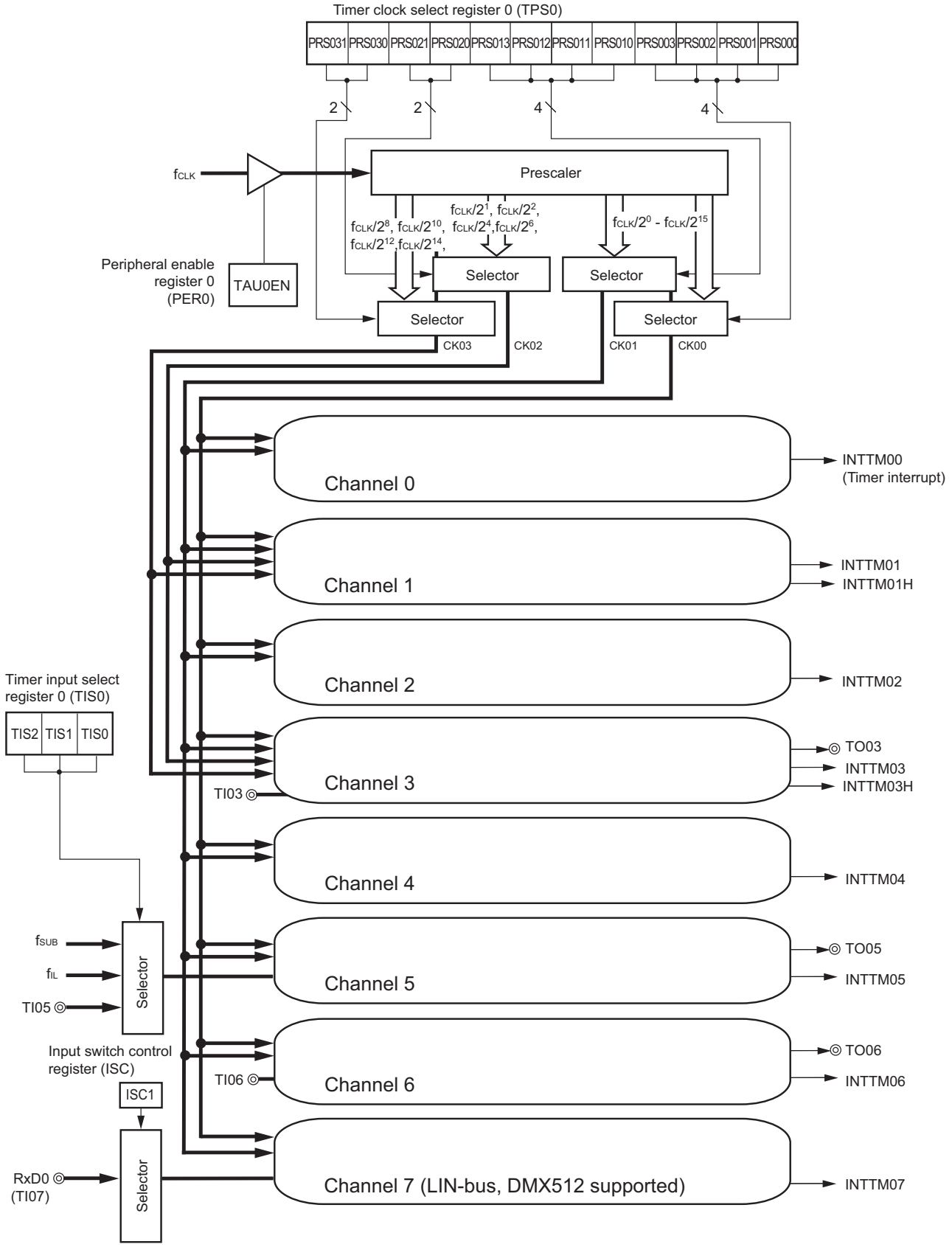
Note The TI07 pin can only be used by setting bit 1 (ISC1) in the input switch control register (ISC) to 1. For details, see **6.3.13 Input switch control register (ISC)**.

Remarks 1. Since timer input and timer output are shared by the same pin, either only timer input or only timer output can be used.

2. –: There is no timer I/O pin, but the channel is available. (However, the channel can only be used as an interval timer.)

Figures 6-1 and 6-2 show the block diagrams of the timer array unit.

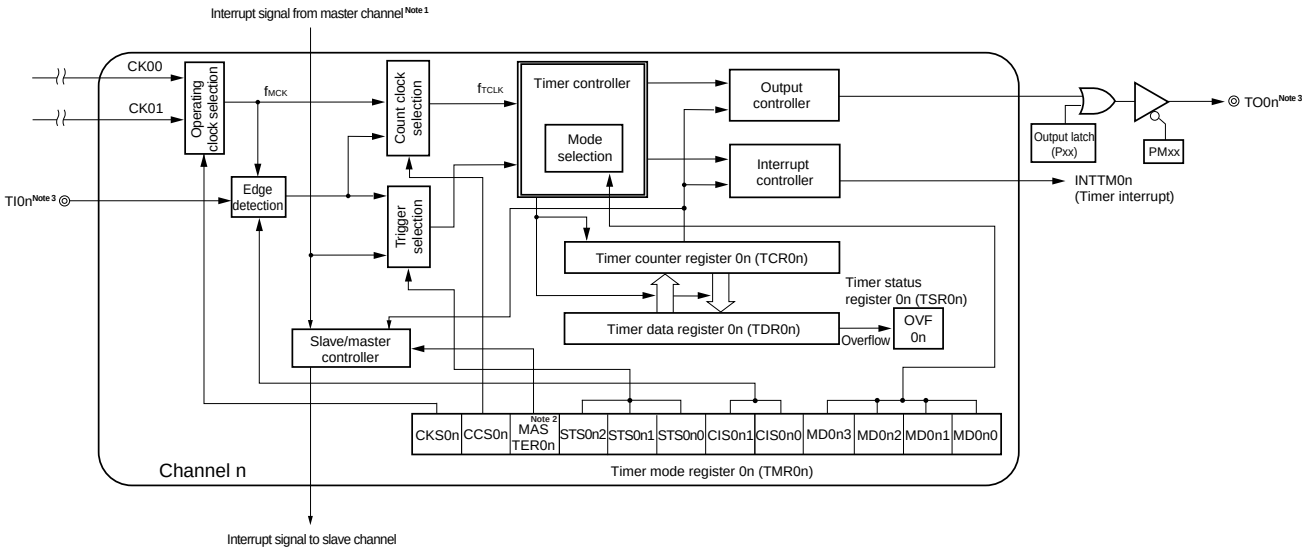
Figure 6-1. Entire Configuration of Timer Array Unit (Example: 38-pin Products)



Remark f_{SUB}: Subsystem clock frequency

f_{IL}: Internal Low-speed on-chip oscillator clock frequency

Figure 6-2. Internal Block Diagram of Channels 0, 2, 4, 6 of Timer Array Unit



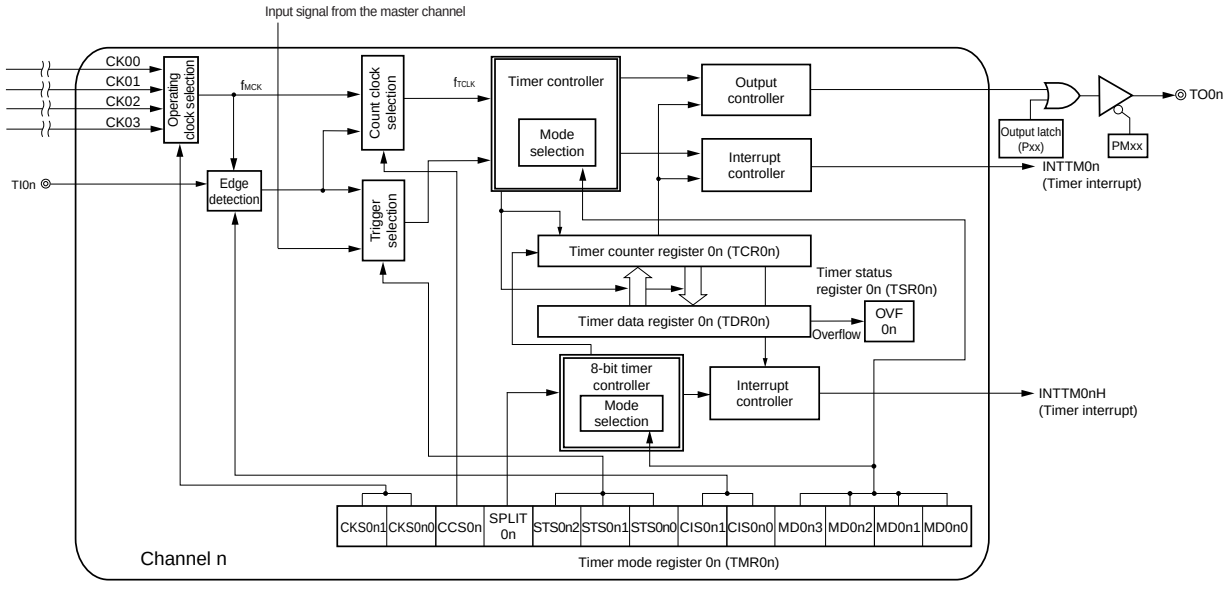
Notes 1. Channels 2, 4, and 6 only

2. Channels 2, 4, and 6 only for MASTER0n

3. Channel 6 only for TI0n,TO0n

Remark n = 0, 2, 4, 6

Figure 6-3. Internal Block Diagram of Channel 1, 3 of Timer Array Unit



Note Channel 3 only for TI0n,TO0n

Remark n = 1, 3

Figure 6-4. Internal Block Diagram of Channel 5 of Timer Array Unit

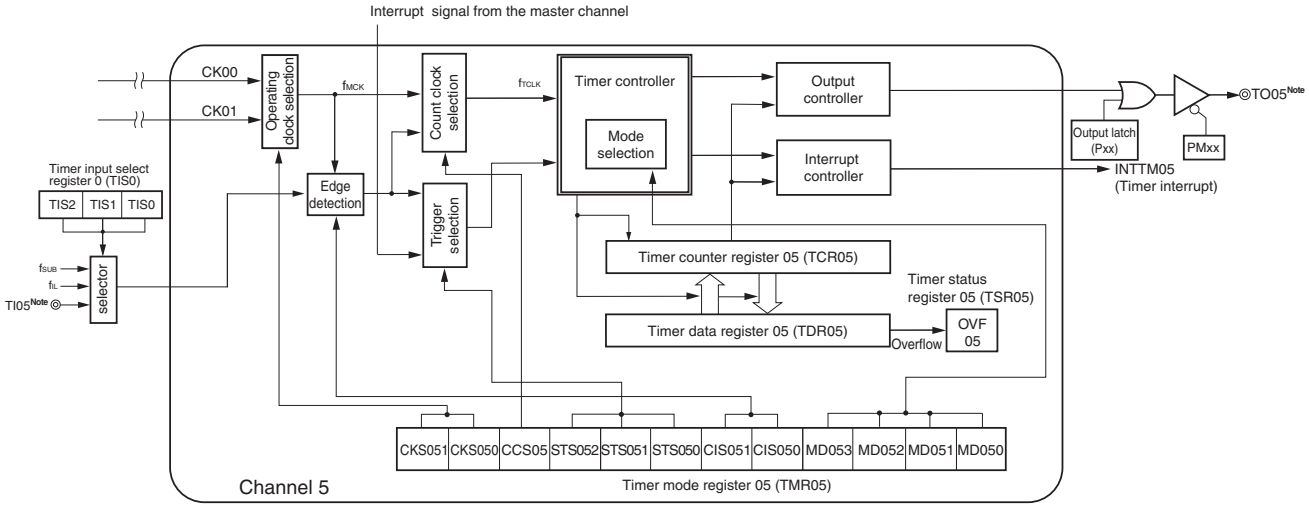
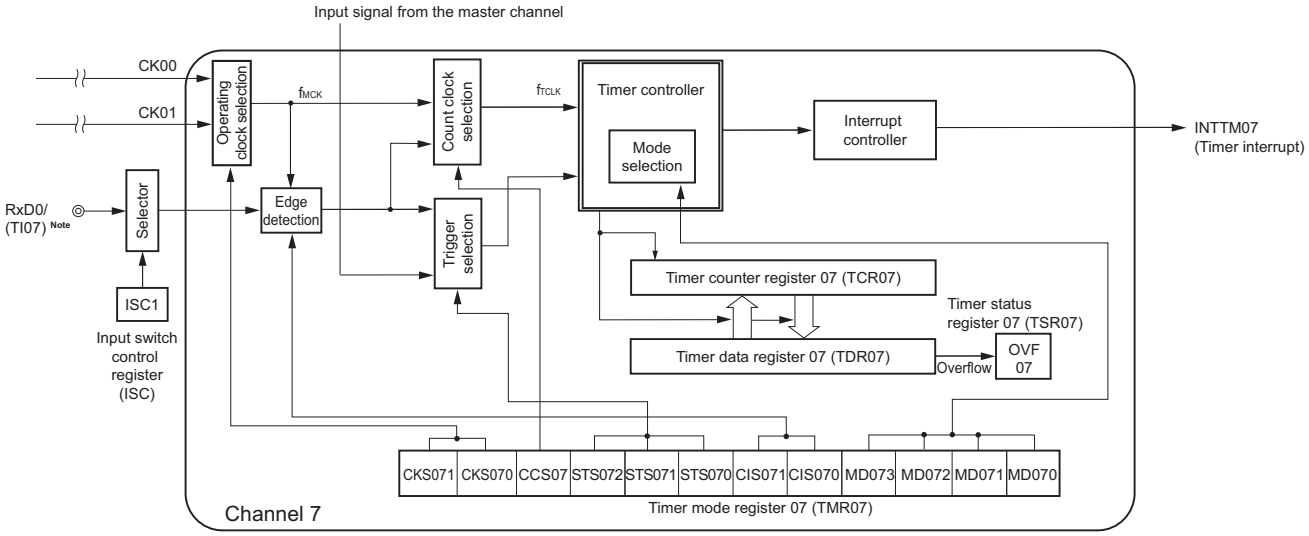


Figure 6-5. Internal Block Diagram of Channel 7 of Timer Array Unit



Note The TI07 pin can only be used by setting bit 1 (ISC1) in the input switch control register (ISC) to 1. For details, see 6.3.13 Input switch control register (ISC).

6.2.1 Timer/counter register 0n (TCR0n)

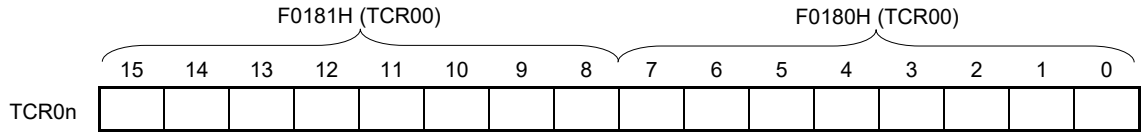
The TCR0n register is a 16-bit read-only register and is used to count clocks.

The value of this counter is incremented or decremented in synchronization with the rising edge of a count clock.

Whether the counter is incremented or decremented depends on the operation mode that is selected by the MD0n3 to MD0n0 bits of timer mode register 0n (TMR0n) (see **6.3.3 Timer mode register 0n (TMR0n)**).

Figure 6-6. Format of Timer/Counter Register 0n (TCR0n)

Address: F0180H, F0181H (TCR00) to F018EH, F018FH (TCR07) After reset: FFFFH R



Remark n: Channel number (n = 0 to 7)

The count value can be read by reading timer/counter register 0n (TCR0n).

The count value is set to FFFFH in the following cases.

- When the reset signal is generated
- When the TAU0EN bit of peripheral enable register 0 (PER0) is cleared
- When counting of the slave channel has been completed in the PWM output mode
- When counting of the slave channel has been completed in the delay count mode
- When counting of the master/slave channel has been completed in the one-shot pulse output mode
- When counting of the slave channel has been completed in the multiple PWM output mode

The count value is cleared to 0000H in the following cases.

- When the start trigger is input in the capture mode
- When capturing has been completed in the capture mode

Caution The count value is not captured to timer data register 0n (TDR0n) even when the TCR0n register is read.

The TCR0n register read value differs as follows according to operation mode changes and the operating status.

Table 6-3. Timer/counter Register 0n (TCR0n) Read Value in Various Operation Modes

Operation Mode	Count Mode	Timer/counter Register 0n (TCR0n) Read Value[Note]			
		Value if the operation mode was changed after releasing reset	Value if the operation was restarted after count operation paused (TT0n = 1)	Value if the operation mode was changed after count operation paused (TT0n = 1)	Value when waiting for a start trigger after one count
Interval timer mode	Count down	FFFFH	Value if stop	Undefined	–
Capture mode	Count up	0000H	Value if stop	Undefined	–
Event counter mode	Count down	FFFFH	Value if stop	Undefined	–
One-count mode	Count down	FFFFH	Value if stop	Undefined	FFFFH
Capture & one-count mode	Count up	0000H	Value if stop	Undefined	Capture value of TDR0n register + 1

Note This indicates the value read from the TCR0n register when channel n has stopped operating as a timer (TE0n = 0) and has been enabled to operate as a counter (TS0n = 1). The read value is held in the TCR0n register until the count operation starts.

Remark n: Channel number (n = 0 to 7)

6.2.2 Timer data register 0n (TDR0n)

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 MD0n3 to MD0n0 bits of timer mode register 0n (TMR0n).

The value of the TDR0n register can be changed at any time.

This register can be read or written in 16-bit units.

In addition, for the TDR01 and TDR03 registers, while in the 8-bit timer mode (when the SPLIT 01,SPLIT03 bits of timer mode registers 01 and 03 (TMR01, TMR03) are 1), it is possible to read and write the data in 8-bit units, with TDR01H and TDR03H used as the higher 8 bits, and TDR01L and TDR03L used as the lower 8 bits.

Reset signal generation clears this register to 0000H.

Figure 6-7. Format of Timer Data Register 0n (TDR0n) (n = 0, 2, 4 to 7)

Address: FFF18H, FFF19H (TDR00), FFF64H, FFF65H (TDR02), FFF68H, FFF69H (TDR04) to FFF6EH, FFF6FH (TDR07) After reset: 0000H R/W

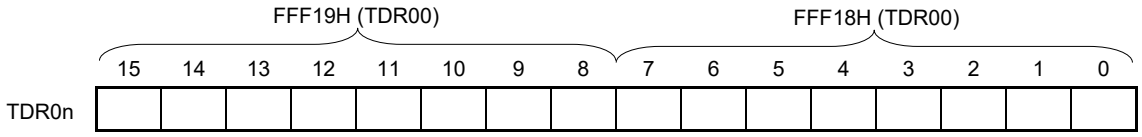
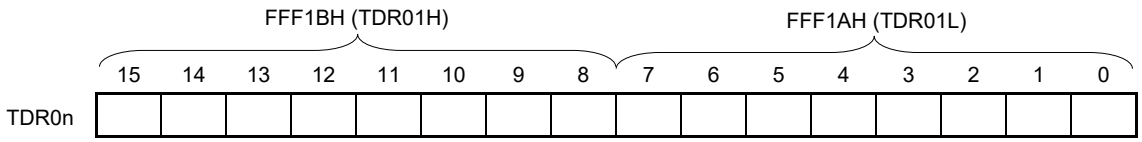


Figure 6-8. Format of Timer Data Register 0n (TDR0n) (n = 1, 3)

Address: FFF1AH, FFF1BH (TDR01), FFF66H, FFF67H (TDR03) After reset: 0000H R/W



(i) When timer data register 0n (TDR0n) is used as compare register

Counting down is started from the value set to the TDR0n register. When the count value reaches 0000H, an interrupt signal (INTTM0n) is generated. The TDR0n register holds its value until it is rewritten.

Caution The TDR0n register does not perform a capture operation even if a capture trigger is input, when it is set to the compare function.

(ii) When timer data register 0n (TDR0n) is used as capture register

The count value of timer/counter register 0n (TCR0n) is captured to the TDR0n register when the capture trigger is input.

A valid edge of the TI0n pin can be selected as the capture trigger. This selection is made by timer mode register 0n (TMR0n).

Remark n: Channel number (n = 3, 5, 6, 7)

6.3 Registers Controlling Timer Array Unit

Timer array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Timer clock select register 0 (TPS0)
- Timer mode register 0n (TMR0n)
- Timer status register 0n (TSR0n)
- Timer channel enable status register 0 (TE0)
- Timer channel start register 0 (TS0)
- Timer channel stop register 0 (TT0)
- Timer input select register 0 (TIS0)
- Timer output enable register 0 (TOE0)
- Timer output register 0 (TO0)
- Timer output level register 0 (TOL0)
- Timer output mode register 0 (TOM0)
- Input switch control register (ISC)
- Noise filter enable register 1 (NFEN1)
- Port mode register (PM0, PM03)[Note]
- Port register (P0, P3)[Note]

Note Which registers and bits are included depends on the product. Be sure to set bits that are not mounted to their initial values. Note, however, that this does not apply to the bits having undefined values which are specified in the caution of Figure 4-1 Format of Port Mode Register.

Remark n: Channel number (n = 0 to 7)

6.3.1 Peripheral enable register 0 (PER0)

This registers is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the timer array unit is used, be sure to set bit 0 (TAU0EN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Figure 6-9. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	3	<2>	1	<0>
PER0	RTCEN	0	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

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

Cautions 1. When setting the timer array unit, be sure to set the following registers first while the TAU0EN bit is set to 1. If TAU0EN = 0, the values of the registers which control the timer array unit are cleared to their initial values and writing to them is ignored (except for the timer input select register 0 (TIS0), input switch control register (ISC), noise filter enable register 1 (NFEN1), port mode registers 0, 3 (PM0, PM3), and port registers 0, 3 (P0, P3)).

- **Timer clock select register 0 (TPS0)**
- **Timer mode register 0n (TMR0n)**
- **Timer status register 0n (TSR0n)**
- **Timer channel enable status register 0 (TE0)**
- **Timer channel start register 0 (TS0)**
- **Timer channel stop register 0 (TT0)**
- **Timer output enable register 0 (TOE0)**
- **Timer output register 0 (TO0)**
- **Timer output level register 0 (TOL0)**
- **Timer output mode register 0 (TOM0)**

2. Be sure to clear the bits 1, 3, and 6 to “0”.

6.3.2 Timer clock select register 0 (TPS0)

The TPS0 register is a 16-bit register that is used to select two types or four types of operation clocks (CK00, CK01, CK02, CK03) that are commonly supplied to each channel. CK00 is selected by using bits 3 to 0 of the TPS0 register, and CK01 is selected by using bits 7 to 4 of the TPS0 register. In addition, only for channels 1 and 3, CK02 and CK03 can be also selected. CK02 is selected by using bits 9 and 8 of the TPS0 register, and CK03 is selected by using bits 13 and 12 of the TPS0 register.

Rewriting of the TPS0 register during timer operation is possible only in the following cases.

If the PRS000 to PRS003 bits can be rewritten (n = 0 to 7):
All channels for which CK00 is selected as the operation clock (CKS0n1, CKS0n0 = 0, 0) are stopped (TE0n = 0).
If the PRS010 to PRS013 bits can be rewritten (n = 0 to 7):
All channels for which CK01 is selected as the operation clock (CKS0n1, CKS0n0 = 0, 1) are stopped (TE0n = 0).
If the PRS020 and PRS021 bits can be rewritten (n = 1, 3):
All channels for which CK02 is selected as the operation clock (CKS0n1, CKS0n0 = 1, 0) are stopped (TE0n = 0).
If the PRS030 and PRS031 bits can be rewritten (n = 1, 3):
All channels for which CK03 is selected as the operation clock (CKS0n1, CKS0n0 = 1, 1) are stopped (TE0n = 0).

The TPS0 register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 6-10. Format of Timer Clock Select register 0 (TPS0) (1/2)

Address: F01B6H, F01B7H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPS0	0	0	PRS 031	PRS 030	0	0	PRS 021	PRS 020	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000

PRS 0k3	PRS 0k2	PRS 0k1	PRS 0k0	Selection of operation clock (CK0k)[Note] (k = 0, 1)					
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 32 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz	32 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	4 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz	2 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz	1 MHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	250 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	62.5 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	15.6 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz	3.91 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61.0 Hz	153 Hz	305 Hz	610 Hz	977 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit (TT0 = 00FFH).

Cautions 1. Be sure to clear bits 15, 14, 11, 10 to "0".

2. If f_{CLK} (undivided) is selected as the operation clock (CKmk) and TDRnm is set to 0000H (n = 0 or 1, m = 0 to 7), interrupt requests output from timer array units cannot be used.

Remarks 1. f_{CLK}: CPU/peripheral hardware clock frequency

2. Waveform of the clock to be selected in the TPSm register which becomes high level for one period of f_{CLK} from its rising edge (m = 1 to 15). For details, see **6.5.1 Count clock (f_{TCLK}).**

Figure 6-10. Format of Timer Clock Select register 0 (TPS0) (2/2)

Address: F01B6H, F01B7H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPS0	0	0	PRS 031	PRS 030	0	0	PRS 021	PRS 020	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000

PRS 021	PRS 020	Selection of operation clock (CK02)[Note]					
			f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 32 MHz
0	0	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz
0	1	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz
1	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 MHz	1.25 MHz	2 MHz
1	1	$f_{CLK}/2^6$	31.3 kHZ	78.1 kHz	156 kHz	313 kHz	500 kHZ

PRS 031	PRS 030	Selection of operation clock (CK03)[Note]					
			f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 32 MHz
0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz
0	1	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz
1	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz
1	1	$f_{CLK}/2^{14}$	122 HZ	305 Hz	610 Hz	1.22 kHz	1.95 kHZ

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit (TT0 = 00FFH).
The timer array unit must also be stopped if the operating clock (f_{MCK}) or the valid edge of the signal input from the TI0n pin is selected.

Caution Be sure to clear bits 15, 14, 11, 10 to "0".

Remark f_{CLK}: CPU/peripheral hardware clock frequency

By using channels 1 and 3 in the 8-bit timer mode and specifying CK02 or CK03 as the operation clock, the interval times shown in Table 6-4 can be achieved by using the interval timer function.

Table 6-4. Interval Times Available for Operation Clock CKS02 or CKS03

Clock		Interval Time[Note] (f_{CLK} = 32 MHz)			
		10 μs	100 μs	1 ms	10 ms
CK02	$f_{CLK}/2$	√	–	–	–
	$f_{CLK}/2^2$	√	–	–	–
	$f_{CLK}/2^4$	√	√	–	–
	$f_{CLK}/2^6$	√	√	–	–
CK03	$f_{CLK}/2^8$	–	√	√	–
	$f_{CLK}/2^{10}$	–	√	√	–
	$f_{CLK}/2^{12}$	–	–	√	√
	$f_{CLK}/2^{14}$	–	–	√	√

Note The margin is within 5 %.

Remarks 1. f_{CLK}: CPU/peripheral hardware clock frequency

2. For details of a signal of $f_{CLK}/2^j$ selected with the TPS0 register, see **6.5.1 Count clock (f_{TCLK}).**

6.3.3 Timer mode register 0n (TMR0n)

The TMR0n register sets an operation mode of channel n. This register is used to select the operation clock (f_{MCK}), select the count clock, select the master/slave, select the 16 or 8-bit timer (only for channels 1 and 3), specify the start trigger and capture trigger, select the valid edge of the timer input, and specify the operation mode (interval, capture, event counter, one-count, or capture and one-count).

Rewriting the TMR0n register is prohibited when the register is in operation (when TE0n = 1). However, bits 7 and 6 (CIS0n1, CIS0n0) can be rewritten even while the register is operating with some functions (when TE0n = 1) (for details, see **6.8 Independent Channel Operation Function of Timer Array Unit** and **6.9 Simultaneous Channel Operation Function of Timer Array Unit**.

The TMR0n register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Caution The bits mounted depend on the channels in the bit 11 of TMR0n register.
TMR02, TMR04, TMR06: MASTER0n bit (n = 2, 4, 6)
TMR01, TMR03: SPLIT0n bit (n = 1, 3)
TMR00, TMR05, TMR07: Fixed to 0

Figure 6-11. Format of Timer Mode Register 0n (TMR0n) (1/4)

Address: F0190H, F0191H (TMR00)) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 2, 4, 6)	CKS 0n1	CKS 0n0	0	CCS 0n	MAST ER0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 1, 3)	CKS 0n1	CKS 0n0	0	CCS 0n	SPLIT 0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 0, 5, 7)	CKS 0n1	CKS 0n0	0	CCS 0n	0Note	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

CKS 0n1	CKS 0n0	Selection of operation clock (f_{MCK}) of channel n
0	0	Operation clock CK00 set by timer clock select register 0 (TPS0)
0	1	Operation clock CK02 set by timer clock select register 0 (TPS0)
1	0	Operation clock CK01 set by timer clock select register 0 (TPS0)
1	1	Operation clock CK03 set by timer clock select register 0 (TPS0)
Operation clock (f_{MCK}) is used by the edge detector. A count clock (f_{TCLK}) and a sampling clock are generated depending on the setting of the CCS0n bit. The operation clocks CK02 and CK03 can only be selected for channels 1 and 3.		

CCS 0n	Selection of count clock (f_{TCLK}) of channel n
0	Operation clock (f_{MCK}) specified by the CKS0n0 and CKS0n1 bits
1	Valid edge of input signal input from the TI0n pin In channel 5, Valid edge of input signal selected by TIS0 In channel 7, Valid edge of input signal selected by ISC
Count clock (f_{TCLK}) is used for the counter, output controller, and interrupt controller.	

Note Bit 11 is fixed at 0 of read only, write is ignored.

Cautions 1. Be sure to clear bits 13, 5, and 4 to "0".

2. The timer array unit must be stopped (TT0 = 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 CKS0n0 and CKS0n1 bits (f_{MCK}) or the valid edge of the signal input from the TI0n pin is selected as the count clock (f_{TCLK}).

Remark n: Channel number (n = 0 to 7)

Figure 6-11. Format of Timer Mode Register 0n (TMR0n) (2/4)

Address: F0190H, F0191H (TMR00)) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 2, 4, 6)	CKS 0n1	CKS 0n0	0	CCS 0n	MASTER0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 1, 3)	CKS 0n1	CKS 0n0	0	CCS 0n	SPLIT 0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 0, 5, 7)	CKS 0n1	CKS 0n0	0	CCS 0n	0Note	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

(Bit 11 of TMR0n (n = 2, 4, 6))

MASTER 0n	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 the channel 2, 4, 6 can be set as a master channel (MASTER0n = 1). Be sure to use channel 0, 5, 7 are fixed to 0 (Regardless of the bit setting, channel 0 operates as master, because it is the highest channel). Clear the MASTER0n bit to 0 for a channel that is used with the independent channel operation function.	

(Bit 11 of TMR0n (n = 1, 3))

SPLIT0n	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.

STS 0n2	STS 0n1	STS 0n0	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 TI0n pin input is used as both the start trigger and capture trigger.
0	1	0	Both the edges of the TI0n pin input are used as a start trigger and a capture trigger.
1	0	0	Interrupt signal of the master channel is used (when the channel is used as a slave channel with the simultaneous channel operation function).
Other than above			Setting prohibited

Note Bit 11 is fixed at 0 of read only, write is ignored.

Remark n: Channel number (n = 0 to 7)

Figure 6-11. Format of Timer Mode Register 0n (TMR0n) (3/4)

Address: F0190H, F0191H (TMR00)) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 2, 4, 6)	CKS 0n1	CKS 0n0	0	CCS 0n	MASTER0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 1, 3)	CKS 0n1	CKS 0n0	0	CCS 0n	SPLIT 0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 0, 5, 7)	CKS 0n1	CKS 0n0	0	CCS 0n	0	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

CIS 0n1	CIS 0n0	Selection of TI0n 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 STS0n2 to STS0n0 bits is other than 010B, set the CIS0n1 to CIS0n0 bits to 10B.		

MD 0n3	MD 0n2	MD 0n1	Operation mode of channel n	Corresponding function	Count operation of TCR
0	0	0	Interval timer mode	Interval timer/Square wave output/ Divider function/PWM output (master)	Counting down
0	1	0	Capture mode	Input pulse interval measurement	Counting up
0	1	1	Event counter mode	External event counter	Counting down
1	0	0	One-count mode	Delay counter/One-shot pulse output/ PWM output (slave)	Counting down
1	1	0	Capture & one-count mode	Measurement of high-/low-level width of input signal	Counting up
Other than above			Setting prohibited		
The operation of each mode varies depending on MD0n0 bit (see the table on the next page).					

Note Bit 11 is fixed at 0 of read only, write is ignored.

Remark n: Channel number (n = 0 to 7)

Figure 6-11. Format of Timer Mode Register 0n (TMR0n) (4/4)

Address: F0190H, F0191H (TMR00)) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 2, 4, 6)	CKS 0n1	CKS 0n0	0	CCS 0n	MAST ER0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 1, 3)	CKS 0n1	CKS 0n0	0	CCS 0n	SPLIT 0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 0, 5, 7)	CKS 0n1	CKS 0n0	0	CCS 0n	0[Note 1]	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Operation mode (Value set by the MD0n3 to MD0n1 bits (see the table on the previous page)	MD 0n0	Setting of starting counting and interrupt
• Interval timer mode (0, 0, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• Capture mode (0, 1, 0)	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

Notes 1. Bit 11 is fixed at 0 of read only, write is ignored.

2. In one-count mode, interrupt output (INTTM0n) when starting a count operation and TO0n output are not controlled.

3. If the start trigger (TS0n = 1) is issued during operation, the counter is initialized, an interrupt is generated, and recounting is started (does not occur the interrupt request).

Remark n: Channel number (n = 0 to 7)

6.3.4 Timer status register 0n (TSR0n)

The TSR0n register indicates the overflow status of the counter of channel n.

The TSR0n register is valid only in the capture mode (MD0n3 to MD0n1 = 010B) and capture & one-count mode (MD0n3 to MD0n1 = 110B). See **Table 6-5** for the operation of the OVF bit in each operation mode and set/clear conditions.

The TSR0n register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TSR0n register can be read with an 8-bit memory manipulation instruction with TSR0nL.

Reset signal generation clears this register to 0000H.

Figure 6-12. Format of Timer Status Register 0n (TSR0n)

Address: F01A0H, F01A1H (TSR00) to F01AEH, F01AFH (TSR07) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSR0n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OVF

OVF	Counter overflow status of channel n
0	Overflow does not occur.
1	Overflow occurs.
When OVF = 1, this flag is cleared (OVF = 0) when the next value is captured without overflow.	

Remark n: Channel number (n = 0 to 7)

Table 6-5. OVF Bit Operation and Set/Clear Conditions in Each Operation Mode

Timer Operation Mode	OVF Bit	Set/Clear Conditions
• Capture mode • Capture & one-count mode	clear	When no overflow has occurred upon capturing
	set	When an overflow has occurred upon capturing
• Interval timer mode • Event counter mode • One-count mode	clear	– (Use prohibited)
	set	

Remark The OVF bit does not change immediately after the counter has overflowed, but changes upon the subsequent capture.

6.3.5 Timer channel enable status register 0 (TE0)

The TE0 register is used to enable or stop the timer operation of each channel.

Each bit of the TE0 register corresponds to each bit of the timer channel start register 0 (TS0) and the timer channel stop register 0 (TT0). When a bit of the TS0 register is set to 1, the corresponding bit of this register is set to 1. When a bit of the TT0 register is set to 1, the corresponding bit of this register is cleared to 0.

The TE0 register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TE0 register can be read with a 1-bit or 8-bit memory manipulation instruction with TE0L.

Reset signal generation clears this register to 0000H.

Figure 6-13. Format of Timer Channel Enable Status register 0 (TE0)

Address: F01B0H, F01B1H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TE0	0	0	0	0	TEH03	0	TEH01	0	TE07	TE06	TE05	TE04	TE03	TE02	TE01	TE00

TEH03	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.

TEH01	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.

TE0n	Indication of operation enable/stop status of channel n
0	Operation is stopped.
1	Operation is enabled.
This bit displays whether operation of the lower 8-bit timer for TE01 and TE03 is enabled or stopped when channel 1 or 3 is in the 8-bit timer mode.	

Remark n: Channel number (n = 0 to 7)

6.3.6 Timer channel start register 0 (TS0)

The TS0 register is a trigger register that is used to clear timer/counter register 0n (TCR0n) 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 0 (TE0) is set to 1. The TS0n, TSH01, TSH03 bits are immediately cleared when operation is enabled (TE0n, TEH01, TEH03 = 1), because they are trigger bits.

The TS0 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TS0 register can be set with a 1-bit or 8-bit memory manipulation instruction with TS0L.

Reset signal generation clears this register to 0000H.

Figure 6-14. Format of Timer Channel Start register 0 (TS0)

Address: F01B2H, F01B3H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS0	0	0	0	0	TSH03	0	TSH01	0	TS07	TS06	TS05	TS04	TS03	TS02	TS01	TS00

TSH03	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 TEH03 bit is set to 1 and the count operation becomes enabled. The TCR03 register count operation start in the interval timer mode in the count operation enabled state (see **Table 6-6**).

TSH01	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 TEH01 bit is set to 1 and the count operation becomes enabled. The TCR01 register count operation start in the interval timer mode in the count operation enabled state (see **Table 6-6**).

TS0n	Operation enable (start) trigger of channel n
0	No trigger operation
1	The TE0n bit is set to 1 and the count operation becomes enabled. The TCR0n register count operation start in the count operation enabled state varies depending on each operation mode (see **Table 6-6**). This bit is the trigger to enable operation (start operation) of the lower 8-bit timer for TS01 and TS03 when channel 1 or 3 is in the 8-bit timer mode.

Cautions 1. Be sure to clear bits 15 to 12, 10, 8 to "0"

2. When switching from a function that does not use TI0n pin input to one that does, the following wait period is required from when timer mode register 0n (TMR0n) is set until the TS0n (TSH01, TSH03) bit is set to 1.

When the TI0n pin noise filter is enabled (TNFEN0n = 1): Four cycles of the operation clock (f_{MCK})

When the TI0n pin noise filter is disabled (TNFEN0n = 0): Two cycles of the operation clock (f_{MCK})

Remarks 1. When the TS0 register is read, 0 is always read.

2. n: Channel number (n = 0 to 7)

6.3.7 Timer channel stop register 0 (TT0)

The TT0 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 0 (TE0) is cleared to 0. The TT0n, TTH01, TTH03 bits are immediately cleared when operation is stopped (TE0n, TEH01, TEH03 = 0), because they are trigger bits.

The TT0 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TT0 register can be set with a 1-bit or 8-bit memory manipulation instruction with TT0L.

Reset signal generation clears this register to 0000H.

Figure 6-15. Format of Timer Channel Stop register 0 (TT0)

Address: F01B4H, F01B5H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TT0	0	0	0	0	TTH03	0	TTH01	0	TT07	TT06	TT05	TT04	TT03	TT02	TT01	TT00

TTH03	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	TEH03 bit is cleared to 0 and the count operation is stopped.

TTH01	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	TEH01 bit is cleared to 0 and the count operation is stopped.

TT0n	Operation stop trigger of channel n
0	No trigger operation
1	TE0n 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 TT01 and TT03 when channel 1 or 3 is in the 8-bit timer mode.

Caution Be sure to clear bits 15 to 12, 10, 8 of the TT0 register to “0”.

Remarks 1. When the TT0 register is read, 0 is always read.

2. n: Channel number (n = 0 to 7)

6.3.8 Timer input select register 0 (TIS0)

The TIS0 register is used to select the channel 5 timer input.

The TIS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-16. Format of Timer Input Select register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	0	0	TIS02	TIS01	TIS00

TIS02	TIS01	TIS00	Selection of timer input used with channel 5
0	0	0	Input signal of timer input pin (TI05)
0	0	1	
0	1	0	
0	1	1	
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
1	0	1	Subsystem clock (f_{SUB})
Other than above			Setting prohibited

Caution High-level width, low-level width of timer input is selected, will require more than 1/f_{MCK} +10 ns. Therefore, when selecting f_{SUB} to f_{CLK} (CSS bit of CKC register = 1), can not TIS02 bit set to 1.

6.3.9 Timer output enable register 0 (TOE0)

The TOE0 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 TO0n bit of timer output register 0 (TO0) 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 (TO0n).

The TOE0 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOE0 register can be set with a 1-bit or 8-bit memory manipulation instruction with TOE0L.

Reset signal generation clears this register to 0000H.

Figure 6-17. Format of Timer Output Enable register 0 (TOE0)

Address: F01BAH, F01BBH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOE0	0	0	0	0	0	0	0	0	0	TOE06	TOE05	0	TOE03	0	0	0

TOE0n	Timer output enable/disable of channel n
0	Disable output of timer. Without reflecting on TO0n bit timer operation, to fixed the output. Writing to the TO0n bit is enabled and the level set in the TO0n bit is output from the TO0n pin.
1	Enable output of timer. Reflected in the TO0n bit timer operation, to generate the output waveform. Writing to the TO0n bit is disabled (writing is ignored).

Caution Be sure to clear bits 15 to 7, 4, and 2 to 0 to “0”.

Remark n: Channel number (n = 3, 5, 6)

6.3.10 Timer output register 0 (TO0)

The TO0 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 (TO0n) of each channel.

The TO0n bit of this register can be rewritten by software only when timer output is disabled (TOE0n = 0). When timer output is enabled (TOE0n = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.

To use the P31/TI03/TO03, P05/TI05/TO05, or P06/TI06/TO06 pin as a port function pin, set the corresponding TO0n bit to “0”.

The TO0 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TO0 register can be set with an 8-bit memory manipulation instruction with TO0L.

Reset signal generation clears this register to 0000H.

Figure 6-18. Format of Timer Output register 0 (TO0)

Address: F01B8H, F01B9H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO0	0	0	0	0	0	0	0	0	0	TO06	TO05	0	TO03	0	0	0

TO0n	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 7, 4, and 2 to 0 to “0”.

Remark n: Channel number (n = 3, 5, 6)

6.3.11 Timer output level register 0 (TOL0)

The TOL0 register is a register that controls the timer output level of each channel.

The setting of the inverted output of channel n by this register is reflected at the timing of set or reset of the timer output signal while the timer output is enabled (TOE0n = 1) in the Slave channel output mode (TOM0n = 1). In the master channel output mode (TOM0n = 0), this register setting is invalid.

The TOL0 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOL0 register can be set with an 8-bit memory manipulation instruction with TOL0L.

Reset signal generation clears this register to 0000H.

Figure 6-19. Format of Timer Output Level register 0 (TOL0)

Address: F01BCH, F01BDH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOL0	0	0	0	0	0	0	0	0	0	TOL06	TOL05	0	TOL03	0	0	0

TOL0n	Control of timer output level of channel n
0	Positive logic output (active-high)
1	Inverted output (active-low)

Caution Be sure to clear bits 15 to 7, 4, and 2 to 0 to "0".

Remarks 1. If the value of this register is rewritten during timer operation, the timer output is inverted when the timer output signal changes next, instead of immediately after the register value is rewritten.

2. n: Channel number (n = 3, 5, 6)

6.3.12 Timer output mode register 0 (TOM0)

The TOM0 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 (TOE0n = 1).

The TOM0 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOM0 register can be set with an 8-bit memory manipulation instruction with TOM0L.

Reset signal generation clears this register to 0000H.

Figure 6-20. Format of Timer Output Mode register 0 (TOM0)

Address: F01BEH, F01BFH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOM0	0	0	0	0	0	0	0	0	0	TOM06	TOM05	0	TOM03	0	0	0

TOM0n	Control of timer output mode of channel n
0	Master channel output mode (to produce toggle output by timer interrupt request signal (INTTM0n))
1	Slave channel output mode (output is set by the timer interrupt request signal (INTTM0n) 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 7, 4, and 2 to 0 to "0".

Remark n: Master channel number (n = 0, 2, 4)
p: Slave channel number (n < p ≤ 6)
(For details of the relation between the master channel and slave channel, see **6.4.1 Basic rules of simultaneous channel operation function**.)

6.3.13 Input switch control register (ISC)

The ISC1 and ISC0 bits of the ISC register are used to set whether to assign TI07 and INTP0 pins to P11/RxD0 pin.

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

Reset signal generation clears this register to 00H.

Figure 6-21. Format of Input Switch Control Register (ISC)

Address: F0073H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	ISC1	ISC0

ISC1	Switching channel 7 input of timer array unit
0	Do not use a timer input signal for channel 7.
1	Input signal of the P11/RxD0 pin is used as timer input. (for LIN-bus to detect the wakeup signal and to measure the low width of the break field and the pulse width of the sync field, and for DMX512 to measure each signal width).

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 P11/RxD0 pin as an external interrupt. (for LIN-bus to detect wakeup signal, and for DMX512 to detect BREAK signal).

Caution Be sure to clear bits 7 to 2 to “0”.

Remark When the LIN-bus and DMX512 communication function is used, select the input signal of the RxD0 pin by setting ISC1 to 1.

6.3.14 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.

The NFEN1 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-22. Format of Noise Filter Enable Register 1 (NFEN1)

Address: F0071H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN1	TNFEN07	TNFEN06	TNFEN05	0	TNFEN03	0	0	0

TNFEN07	Enable/disable using noise filter of TI07 (P11/RxD0) pin input signal[Note]
0	Noise filter OFF
1	Noise filter ON

TNFEN06	Enable/disable using noise filter of TI06 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN05	Enable/disable using noise filter of TI05 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN03	Enable/disable using noise filter of TI03 pin input signal
0	Noise filter OFF
1	Noise filter ON

Note When P11/RxD0 pin is made to become applicable, set the ISC1 bit of the input switch control register (ISC) to 1.

Remark The presence or absence of timer I/O pins of channel 0 to 7 depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product** for details.

6.3.15 Registers controlling port functions of pins to be used for timer I/O

Using port pins for the timer array unit functions requires setting of the registers that control the port functions multiplexed on the target pins (port mode register (PMxx) and port register (Pxx)). For details, see **4.3.1 Port mode registers (PMxx)** and **4.3.2 Port registers (Pxx)**.

The port mode register (PMxx) and port register (Pxx) to be set depend on the product. For details, see **4.5.3 Register setting examples for used port and alternate functions**.

When using the ports (P31/TI03/TO03, P05/TI05/TO05 and P06/TI06/TO06) to be shared with the timer output pin for timer output, set the port mode register (PMxx) bit and port register (Pxx) bit corresponding to each port to 0.

Example: When using P31/TI03/TO03 for timer output
Set the PM31 bit of port mode register 3 to 0.
Set the P31 bit of port register 3 to 0.

When using the ports (P31/TI03/TO03, P05/TI05/TO05 and P06/TI06/TO06) to be shared with the timer input pin for timer input, set the port mode register (PMxx) bit corresponding to each port to 1. At this time, the port register (Pxx) bit may be 0 or 1.

Example: When using P31/TI03/TO03 for timer output
Set the PM31 bit of port mode register 3 to 1.
Set the P31 bit of port register 3 to 1.

6.4 Basic Rules of Timer Array Unit

6.4.1 Basic rules of simultaneous channel operation function

When simultaneously using multiple channels, namely, a combination of a master channel (a reference timer mainly counting the cycle) and slave channels (timers operating according to the master channel), the following rules apply.

(1) Only an even channel (channels 0, 2, 4, etc.) can be set as a master channel.
(2) Any channel, except channel 0, can be set as a slave channel.
(3) Only channels whose number is greater than the master channel can be set as a slave channel.
Example: If channel 2 is set as a master channel, channel 3 can be set as a slave channel. If channel 4 is set as a master channel, channels 5 and 6 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 2 and 4 are set as master channels, channel 3 can be set as the slave channels of master channel 2. Channels 5 and 6 cannot be set as the slave channels of master channel 2.

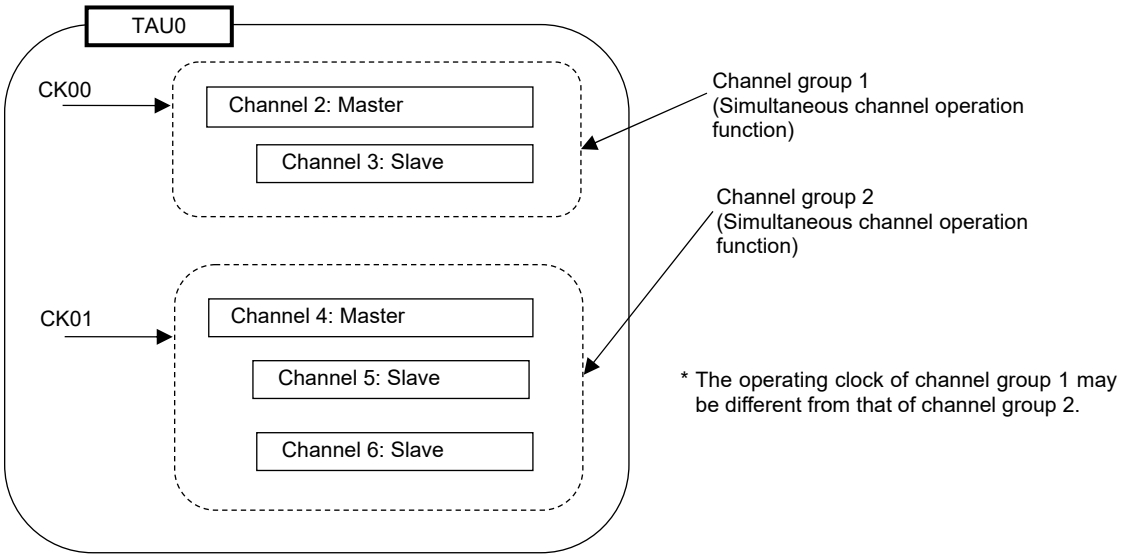
(6) The operating clock for a slave channel in combination with a master channel must be the same as that of the master channel. The CKS0n0, CKS0n1 bits (bit 15, 14 of timer mode register 0n (TMR0n)) of the slave channel that operates in combination with the master channel must be the same value as that of the master channel.
(7) A master channel can transmit INTTM0n (interrupt), start software trigger, and count clock to the lower channels.
(8) A slave channel can use INTTM0n (interrupt), a start software trigger, or the count clock of the master channel as a source clock, but cannot transmit its own INTTM0n (interrupt), start software trigger, or count clock to channels with lower channel numbers.
(9) A master channel cannot use INTTM0n (interrupt), a start software trigger, or the count clock from the other higher master channel as a source clock.
(10) To simultaneously start channels that operate in combination, the channel start trigger bit (TS0n) of the channels in combination must be set at the same time.
(11) During the counting operation, a TS0n bit of a master channel or TS0n bits of all channels which are operating simultaneously can be set. It cannot be applied to TS0n bits of slave channels alone.
(12) To stop the channels in combination simultaneously, the channel stop trigger bit (TT0n) of the channels in combination must be set at the same time.
(13) CK02/CK03 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 00 (TMR00) has no master bit (it is fixed as “0”). However, as channel 0 is the highest channel, it can be used as a master channel during simultaneous operation.

The rules of the simultaneous channel operation function are applied in a channel group (a master channel and slave channels forming one simultaneous channel operation function).

If two or more channel groups that do not operate in combination are specified, the basic rules of the simultaneous channel operation function in this section do not apply to the channel groups.

Remark n: Channel number (n = 0 to 7)

Example


TAU0
CK00
Channel 2: Master
Channel 3: Slave
CK01
Channel 4: Master
Channel 5: Slave
Channel 6: Slave
Channel group 1
(Simultaneous channel operation function)
Channel group 2
(Simultaneous channel operation function)
* The operating clock of channel group 1 may be different from that of channel group 2.

6.4.2 8-bit timer operation function overview (only channels 1 and 3)

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 SPLIT0n bit of timer mode register 0n (TMR0n) 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 INTTM01H/INTTM03H (an interrupt) (which is the same operation performed when MD0n0 is set to 1).
(5) The operation clock of the higher 8 bits is selected according to the CKS0n1 and CKS0n0 bits of the lower-bit TMR0n register.
(6) For the higher 8 bits, the TSH10/TSH03 bit is manipulated to start channel operation and the TTH01/TTH03 bit is manipulated to stop channel operation. The channel status can be checked using the TEH01/TEH03 bit.
(7) The lower 8 bits operate according to the TMR0n register settings. The following three functions support operation of the lower 8 bits:
    Interval timer function/Square Wave Output Function
    External event counter function (channel 3 only)
    Delay count function (channel 3 only)

(8) For the lower 8 bits, the TS01/TS03 bit is manipulated to start channel operation and the TT01/TT03 bit is manipulated to stop channel operation. The channel status can be checked using the TE01/TE03 bit.
(9) During 16-bit operation, manipulating the TSH01, TSH03, TTH01, and TTH03 bits is invalid. The TS01, TS03, TT01, and TT03 bits are manipulated to operate channels 1 and 3. The TEH03 and TEH01 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 n: Channel number (n = 1, 3)

6.5 Operation of Counter

6.5.1 Count clock (f_{TCLK})

The count clock (f_{TCLK}) of the timer array unit can be selected between following by CCS0n bit of timer mode register 0n (TMR0n).

- Operation clock (f_{MCK}) specified by the CKS0n0 and CKS0n1 bits
- Valid edge of input signal input from the TI0n pin

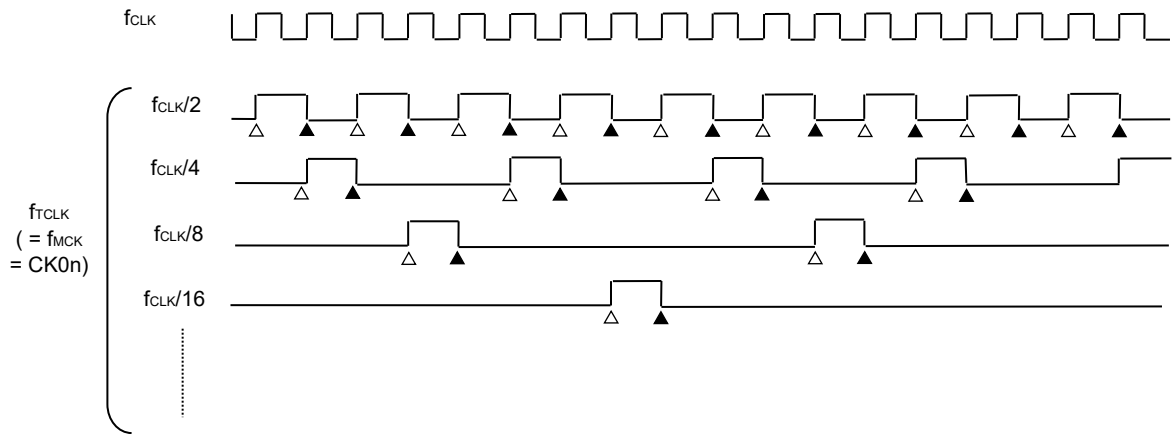
Because the timer array unit is designed to operate in synchronization with f_{CLK}, the timings of the count clock (f_{TCLK}) are shown below.

(1) When operation clock (f_{MCK}) specified by the CKS0n0 and CKS0n1 bits is selected (CCS0n = 0)

The count clock (f_{TCLK}) is between f_{CLK} to $f_{CLK}/2^{15}$ by setting of timer clock select register m (TPSm). When a divided f_{CLK} is selected, however, the clock selected in TPS0n register, but a signal which becomes high level for one period of f_{CLK} from its rising edge. When a f_{CLK} is selected, fixed to high level.

Counting of timer count register 0n (TCR0n) delayed by one period of f_{CLK} from rising edge of the count clock, because of synchronization with f_{CLK}. But, this is described as “counting at rising edge of the count clock”, as a matter of convenience.

Figure 6-23. Timing of f_{CLK} and Count Clock (f_{TCLK}) (When CCS0n = 0)



Remarks 1. △: Rising edge of the count clock

▲: Synchronization, increment/decrement of counter

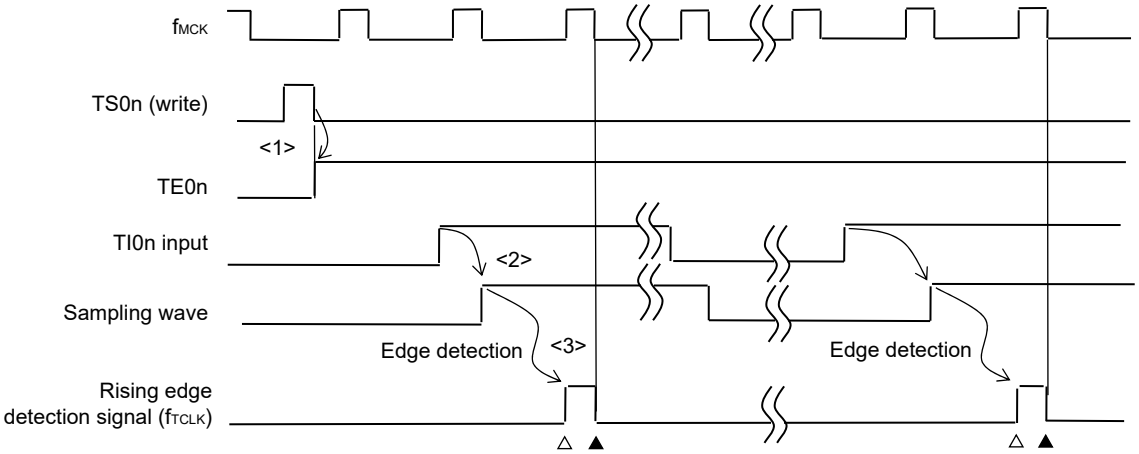
2. f_{CLK}: CPU/peripheral hardware clock

(2) When valid edge of input signal via the TI0n pin is selected (CCS0n = 1)

The count clock (f_{TCLK}) becomes the signal that detects valid edge of input signal via the TI0n pin and synchronizes next rising f_{MCK}. The count clock (f_{TCLK}) is delayed for 1 to 2 period of f_{MCK} from the input signal via the TI0n pin (when a noise filter is used, the delay becomes 3 to 4 clock).

Counting of timer count register 0n (TCR0n) delayed by one period of f_{CLK} from rising edge of the count clock, because of synchronization with f_{CLK}. But, this is described as “counting at valid edge of input signal via the TI0n pin”, as a matter of convenience.

Figure 6-24. Timing of f_{CLK} and Count Clock (f_{TCLK}) (When CCS0n = 1, Noise Filter Unused)



<1> Setting TS0n bit to 1 enables the timer to be started and to become wait state for valid edge of input signal via the TI0n pin.

<2> The rise of input signal via the TI0n pin is sampled by f_{MCK}.

<3> The edge is detected by the rising of the sampled signal and the detection signal (count clock) is output.

Remarks 1. △: Rising edge of the count clock
▲: Synchronization, increment/decrement of counter

2. f_{CLK}: CPU/peripheral hardware clock
f_{MCK}: Operation clock of channel n

3. The waveform of the input signal of the input pulse interval measurement, the measurement of high/low width of input signal, and the delay counter are the same.

6.5.2 Start timing of counter

Timer count register 0n (TCR0n) becomes enabled to operation by setting of TS0n bit of timer channel start register 0 (TS0).

Operations from count operation enabled state to timer count Register 0n (TCR0n) count start is shown in Table 6-6.

Table 6-6. Operations from Count Operation Enabled State to Timer count Register 0n (TCR0n) Count Start

Timer Operation Mode	Operation When TS0n = 1 Is Set
• Interval timer mode	No operation is carried out from start trigger detection (TS0n=1) until count clock generation. The first count clock loads the value of the TDR0n register to the TCR0n register and the subsequent count clock performs count down operation (see **6.5.3 (1) Operation of interval timer mode**).
• Event counter mode	Writing 1 to the TS0n bit loads the value of the TDR0n register to the TCR0n register. If detect edge of TI0n input. The subsequent count clock performs count down operation (see **6.5.3 (2) Operation of event counter mode**).
• Capture mode	No operation is carried out from start trigger detection (TS0n = 1) until count clock generation. The first count clock loads 0000H to the TCR0n register and the subsequent count clock performs count up operation (see **6.5.3 (3) Operation of capture mode (input pulse interval measurement)**).
• One-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TS0n bit while the timer is stopped (TE0n = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads the value of the TDR0n register to the TCR0n register and the subsequent count clock performs count down operation (see **6.5.3 (4) Operation of one-count mode**).
• Capture & one-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TS0n bit while the timer is stopped (TE0n = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to the TCR0n register and the subsequent count clock performs count up operation (see **6.5.3 (5) Operation of capture & one-count mode (high-level width measurement)**).

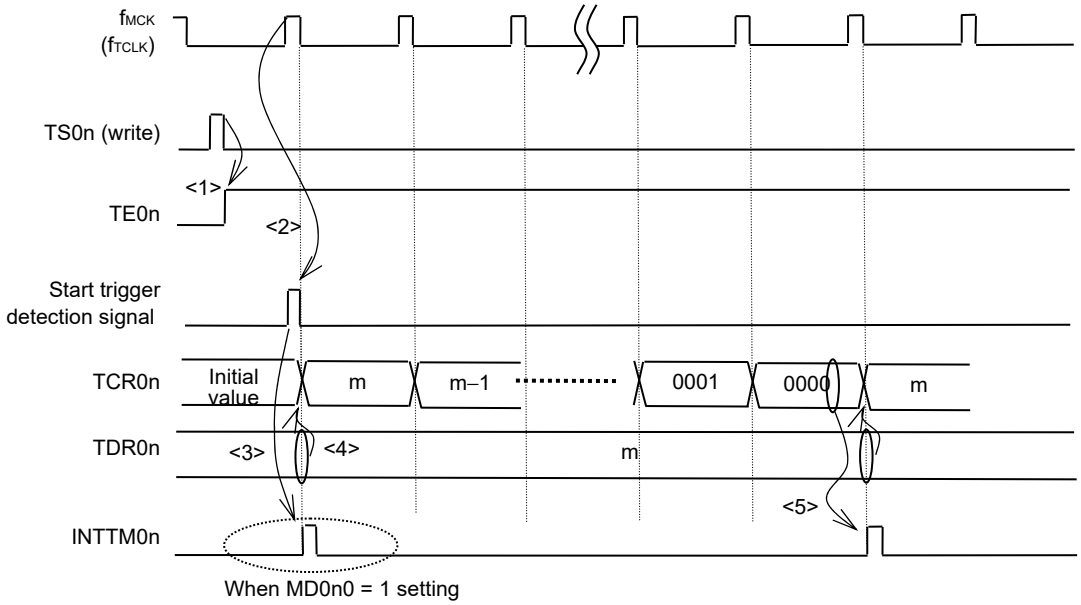
6.5.3 Operation of counter

Here, the counter operation in each mode is explained.

(1) Operation of interval timer mode

<1> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit. Timer count register 0n (TCR0n) holds the initial value until count clock generation.
<2> A start trigger is generated at the first count clock after operation is enabled.
<3> When the MD0n0 bit is set to 1, INTTM0n is generated by the start trigger.
<4> By the first count clock after the operation enable, the value of timer data register 0n (TDR0n) is loaded to the TCR0n register and counting starts in the interval timer mode.
<5> When the TCR0n register counts down and its count value is 0000H, INTTM0n is generated and the value of timer data register 0n (TDR0n) is loaded to the TCR0n register and counting keeps on.

Figure 6-25. Operation Timing (In Interval Timer Mode)



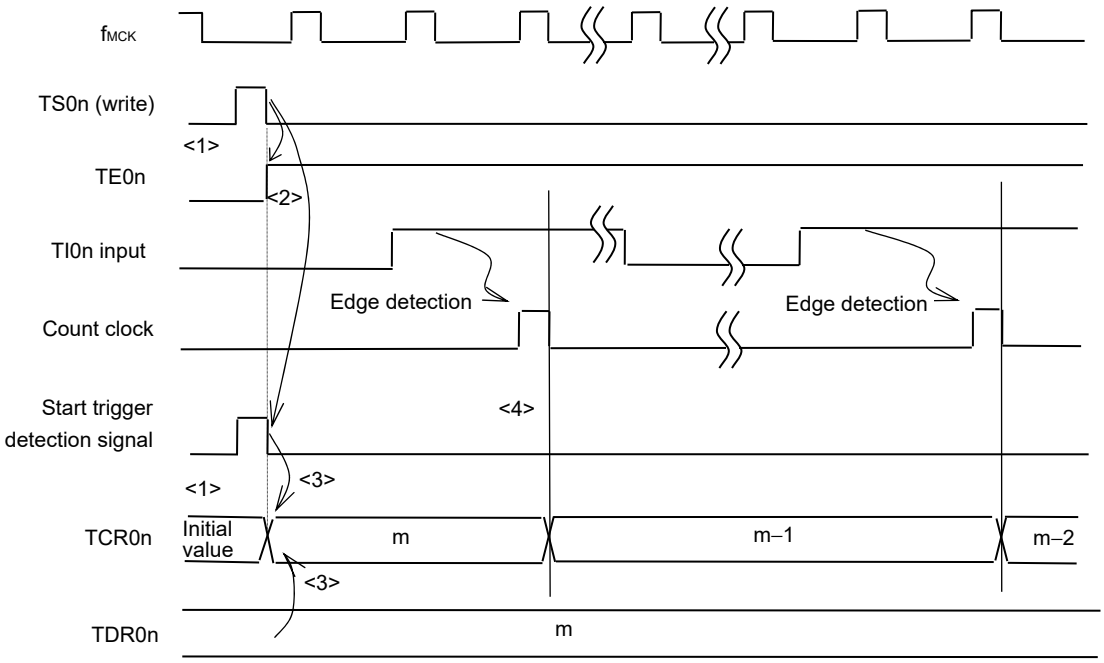
Caution In the first cycle operation of count clock after writing the TS0n bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MD0n0 = 1.

Remark f_{MCK}, the start trigger detection signal, and INTTM0n become active between one clock in synchronization with f_{CLK}.

(2) Operation of event counter mode

<1> Timer count register 0n (TCR0n) holds its initial value while operation is stopped (TE0n = 0).
<2> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit.
<3> As soon as 1 has been written to the TS0n bit and 1 has been set to the TE0n bit, the value of timer data register 0n (TDR0n) is loaded to the TCR0n register to start counting.
<4> After that, the TCR0n register value is counted down according to the count clock of the valid edge of the TI0n input .

Figure 6-26. Operation Timing (In Event Counter Mode)

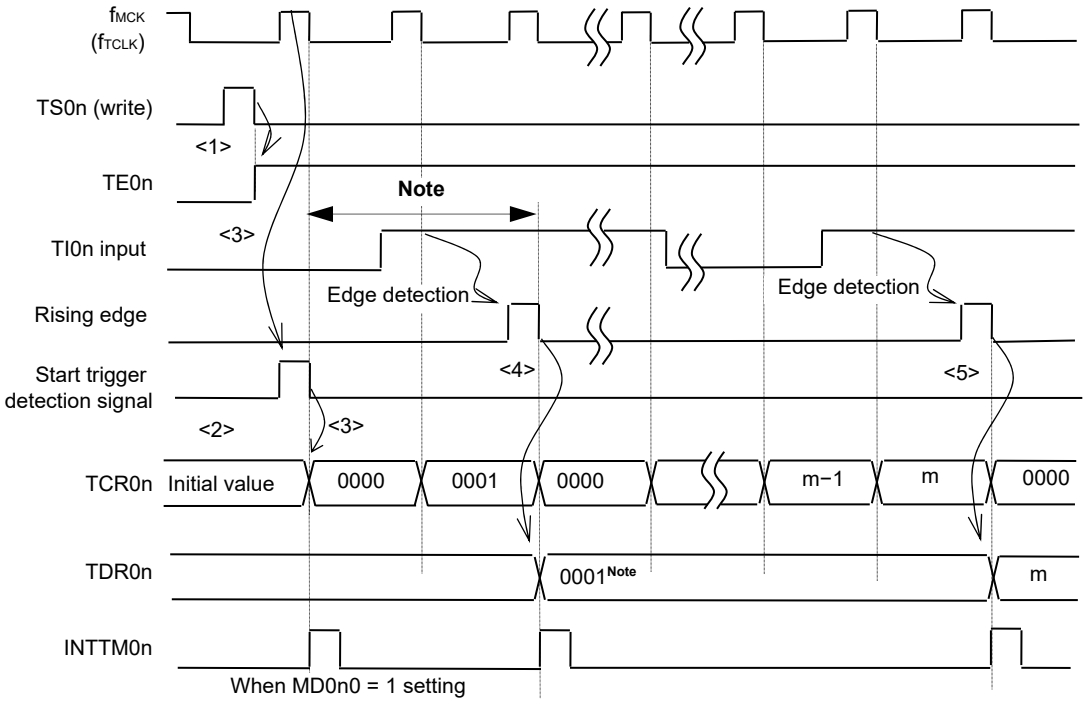


Remark The above figure shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TI0n input. The error per one period occurs by the asynchronous between the period of the TI0n input and that of the count clock (f_{MCK}).

(3) Operation of capture mode (input pulse interval measurement)

<1> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit.
<2> Timer count register 0n (TCR0n) holds the initial value until count clock generation.
<3> A start trigger is generated at the first count clock after operation is enabled. And the value of 0000H is loaded to the TCR0n register and counting starts in the capture mode. (When the MD0n0 bit is set to 1, INTTM0n is generated by the start trigger.)
<4> On detection of the valid edge of the TI0n input, the value of the TCR0n register is captured to timer data register 0n (TDR0n) and INTTM0n is generated. However, this capture value is nomeaning. The TCR0n register keeps on counting from 0000H.
<5> On next detection of the valid edge of the TI0n input, the value of the TCR0n register is captured to timer data register 0n (TDR0n) and INTTM0n is generated.

Figure 6-27. Operation Timing (In Capture Mode: Input Pulse Interval Measurement)



Note If a clock has been input to TI0n (the trigger exists) when capturing starts, counting starts when a trigger is detected, even if no edge is detected. Therefore, the first captured value (<4>) does not determine a pulse interval (in the above figure, 0001 just indicates two clock cycles but does not determine the pulse interval) and so the user can ignore it.

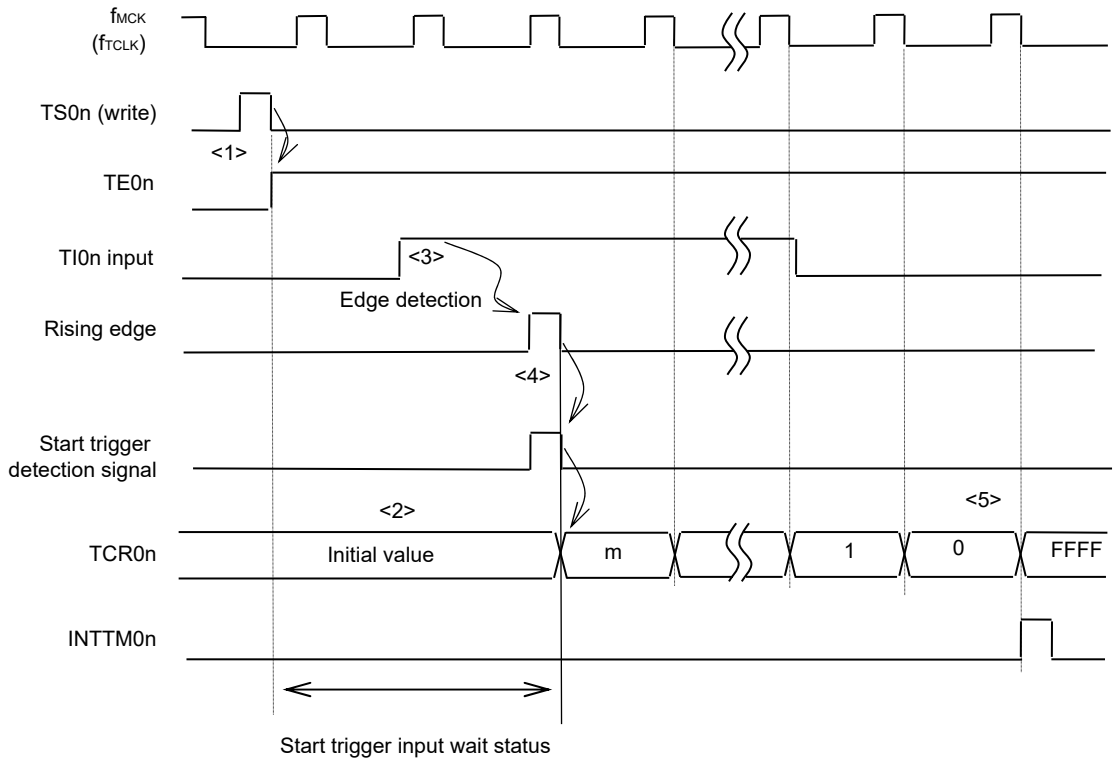
Caution In the first cycle operation of count clock after writing the TS0n bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MD0n0 = 1.

Remark The above figure 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 TI0n input. The error per one period occurs by the asynchronous between the period of the TI0n input and that of the count clock (fMCK).

(4) Operation of one-count mode

<1> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit.
<2> Timer count register 0n (TCR0n) holds the initial value until start trigger generation.
<3> Rising edge of the TI0n input is detected.
<4> On start trigger detection, the value of timer data register 0n (TDR0n) is loaded to the TCR0n register and count starts.
<5> When the TCR0n register counts down and its count value is 0000H, INTTM0n is generated and the value of the TCR0n register becomes FFFFH and counting stops.

Figure 6-28. Operation Timing (In One-count Mode)

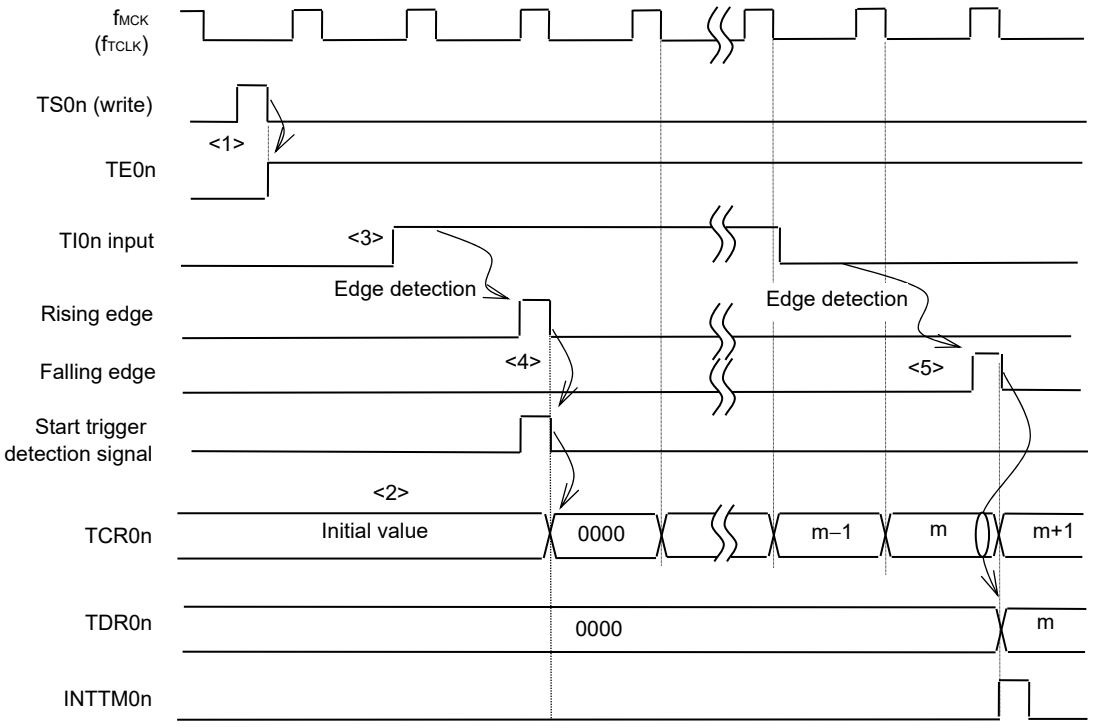


Remark The above figure shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TI0n input. The error per one period occurs by the asynchronous between the period of the TI0n input and that of the count clock (fMCK).

(5) Operation of capture & one-count mode (high-level width measurement)

<1> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit of timer channel start register 0 (TS0).
<2> Timer count register 0n (TCR0n) holds the initial value until start trigger generation.
<3> Rising edge of the TI0n input is detected.
<4> On start trigger detection, the value of 0000H is loaded to the TCR0n register and count starts.
<5> On detection of the falling edge of the TI0n input, the value of the TCR0n register is captured to timer data register 0n (TDR0n) and INTTM0n is generated.

Figure 6-29. Operation Timing (In Capture & One-count Mode: High-level Width Measurement)

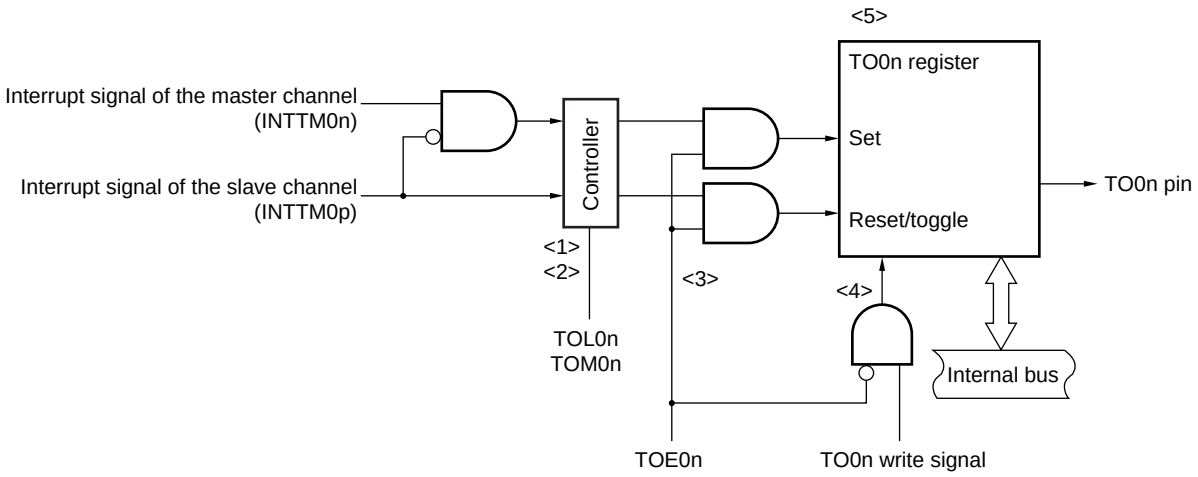


Remark The above figure shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TI0n input. The error per one period occurs by the asynchronous between the period of the TI0n input and that of the count clock (fMCK).

6.6 Channel Output (TO0n pin) Control

6.6.1 TO0n pin output circuit configuration

Figure 6-30. Output Circuit Configuration



The following describes the TO0n pin output circuit.

<1> When TOM0n = 0 (master channel output mode), the set value of timer output level register 0 (TOL0) is ignored and only INTTM0p (slave channel timer interrupt) is transmitted to timer output register 0 (TO0).

<2> When TOM0n = 1 (slave channel output mode), both INTTM0n (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TO0 register.
At this time, the TOL0 register becomes valid and the signals are controlled as follows:

When TOL0n = 0: Positive logic output (INTTM0n → set, INTTM0p → reset)
When TOL0n = 1: Negative logic output (INTTM0n → reset, INTTM0p → set)

When INTTM0n and INTTM0p are simultaneously generated, (0% output of PWM), INTTM0p (reset signal) takes priority, and INTTM0n (set signal) is masked.

<3> While timer output is enabled (TOE0n = 1), INTTM0n (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TO0 register. Writing to the TO0 register (TO0n write signal) becomes invalid.
When TOE0n = 1, the TO0n pin output never changes with signals other than interrupt signals.
To initialize the TO0n pin output level, it is necessary to set timer operation is stopped (TOE0n = 0) and to write a value to the TO0 register.

<4> While timer output is disabled (TOE0n = 0), writing to the TO0n bit to the target channel (TO0n write signal) becomes valid. When timer output is disabled (TOE0n = 0), neither INTTM0n (master channel timer interrupt) nor INTTM0p (slave channel timer interrupt) is transmitted to the TO0 register.

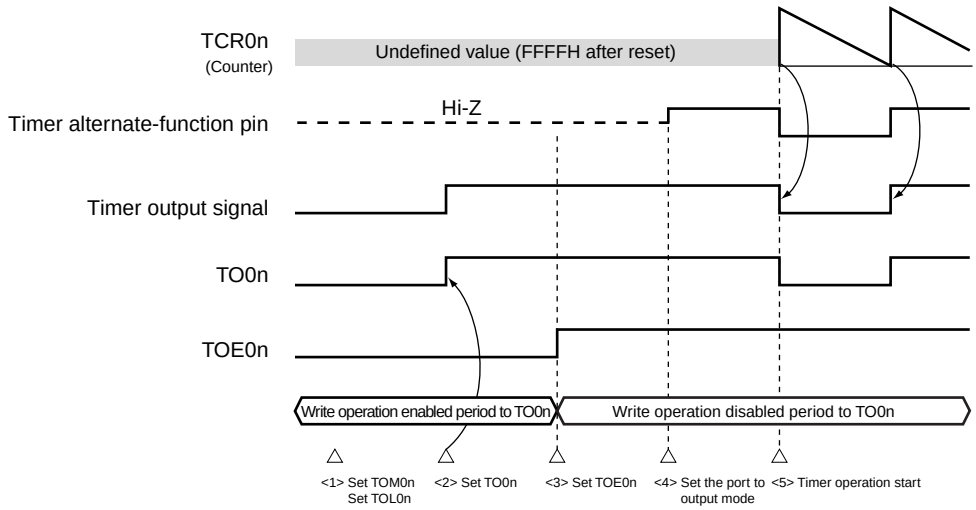
<5> The TO0 register can always be read, and the TO0n pin output level can be checked.

Remark n: Master channel number (n = 0, 2, 4)
p: Slave channel number (n < p ≤ 6)

6.6.2 TO0n pin output setting

The following figure shows the procedure and status transition of the TO0n output pin from initial setting to timer operation start.

Figure 6-31. Status Transition from Timer Output Setting to Operation Start



<1> The operation mode of timer output is set.

- TOM0n bit (0: Master channel output mode, 1: Slave channel output mode)
- TOL0n bit (0: Forward output, 1: Reverse output)

<2> The timer output signal is set to the initial status by setting timer output register 0 (TO0).
<3> The timer output operation is enabled by writing 1 to the TOE0n bit (writing to the TO0 register is disabled).
<4> The port is set to digital I/O by port mode control register (PMCxx) (see **6.3.15 Registers controlling port functions of pins to be used for timer I/O**).
<5> The port I/O setting is set to output (see **6.3.15 Registers controlling port functions of pins to be used for timer I/O**).
<6> The timer operation is enabled (TS0n = 1).

Remark n: Channel number (n = 3, 5, 6)

6.6.3 Cautions on channel output operation

(1) Changing values set in the registers TO0, TOE0, and TOL0 during timer operation

Since the timer operations (operations of timer/counter register 0n (TCR0n) and timer data register 0n (TDR0n)) are independent of the TO0n output circuit and changing the values set in timer output register 0 (TO0), timer output enable register 0 (TOE0), and timer output level register 0 (TOL0) does not affect the timer operation, the values can be changed during timer operation. To output an expected waveform from the TO0n pin by timer operation, however, set the TO0, TOE0, TOL0, and TOM0 registers to the values stated in the register setting example of each operation shown by 6.7 and 6.8.

When the values set to the TOE0, and TOL0 registers (but not the TO0 register) are changed close to the occurrence of the timer interrupt (INTTM0n) of each channel, the waveform output to the TO0n pin might differ, depending on whether the values are changed immediately before or immediately after the timer interrupt (INTTM0n) occurs.

Remark n: Channel number (n = 3, 5, 6)

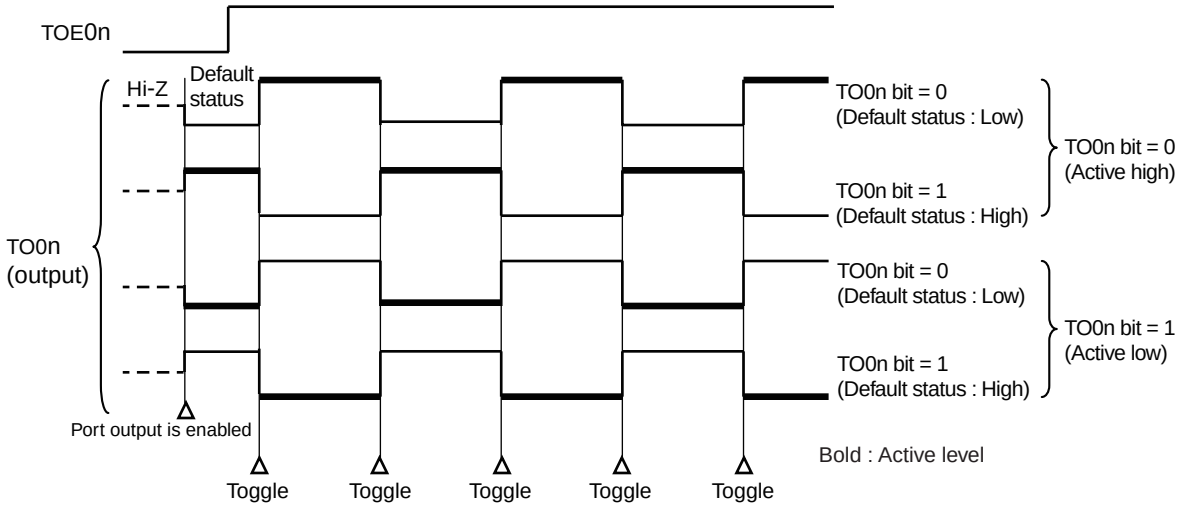
(2) Default level of TO0n pin and output level after timer operation start

The change in the output level of the TO0n pin when timer output register 0 (TO0) is written while timer output is disabled (TOE0n = 0), the initial level is changed, and then timer output is enabled (TOE0n = 1) before port output is enabled, is shown below.

(a) When operation starts with master channel output mode (TOM0n = 0) setting

The setting of timer output level register 0 (TOL0) is invalid when master channel output mode (TOM0n = 0). When the timer operation starts after setting the default level, the toggle signal is generated and the output level of the TO0n pin is reversed.

Figure 6-32. TO0n Pin Output Status at Toggle Output (TOM0n = 0)



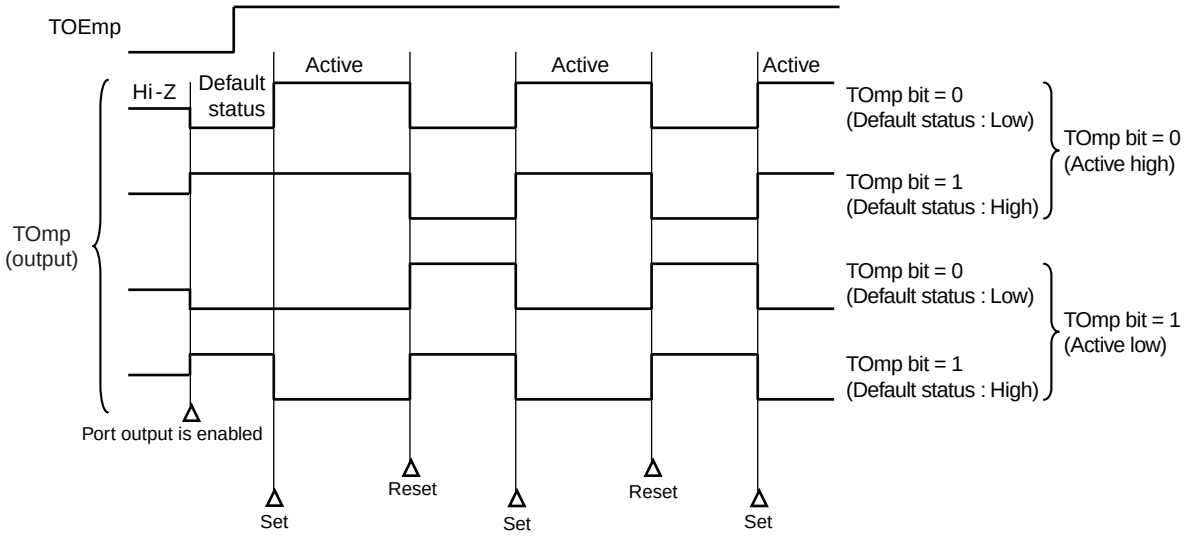
Remarks 1. Toggle: Reverse TO0n pin output status

2. n: Channel number (n = 3, 5, 6)

(b) When operation starts with slave channel output mode (TOM0p = 1) setting (PWM output))

When slave channel output mode (TOM0p = 1), the active level is determined by timer output level register 0 (TOL0) setting.

Figure 6-33. TO0p Pin Output Status at PWM Output (TOM0p = 1)



Remarks 1. Set: The output signal of the TO0p pin changes from inactive level to active level.
Reset: The output signal of the TO0p pin changes from active level to inactive level.

2. p: Channel number (p = 3, 5, 6)

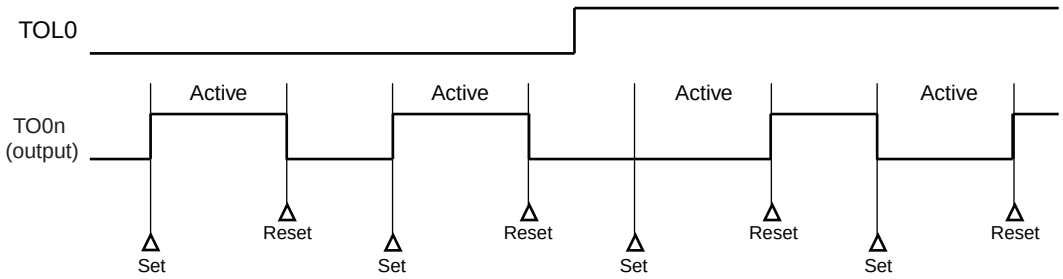
(3) Operation of TO0n pin in slave channel output mode (TOM0n = 1)

(a) When timer output level register 0 (TOL0) setting has been changed during timer operation

When the TOL0 register setting has been changed during timer operation, the setting becomes valid at the generation timing of the TO0n pin change condition. Rewriting the TOL0 register does not change the output level of the TO0n pin.

The operation when TOM0n is set to 1 and the value of the TOL0 register is changed while the timer is operating (TE0n = 1) is shown below.

Figure 6-34. Operation When TOL0 Register Has Been Changed During Timer Operation



Remarks 1. Set: The output signal of the TO0n pin changes from inactive level to active level.
Reset: The output signal of the TO0n pin changes from active level to inactive level.

2. n: Channel number (n = 3, 5, 6)

(b) Set/reset timing

To realize 0%/100% output at PWM output, the TO0n pin/TO0n bit set timing at master channel timer interrupt (INTTM0n) generation is delayed by 1 count clock by the slave channel.

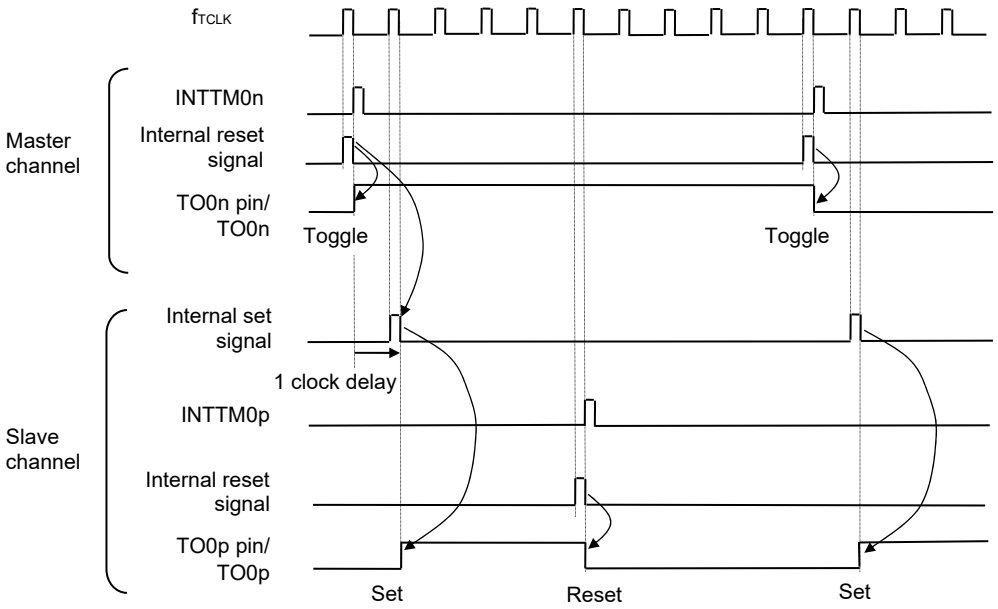
If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

Figure 6-32 shows the set/reset operating statuses where the master/slave channels are set as follows.

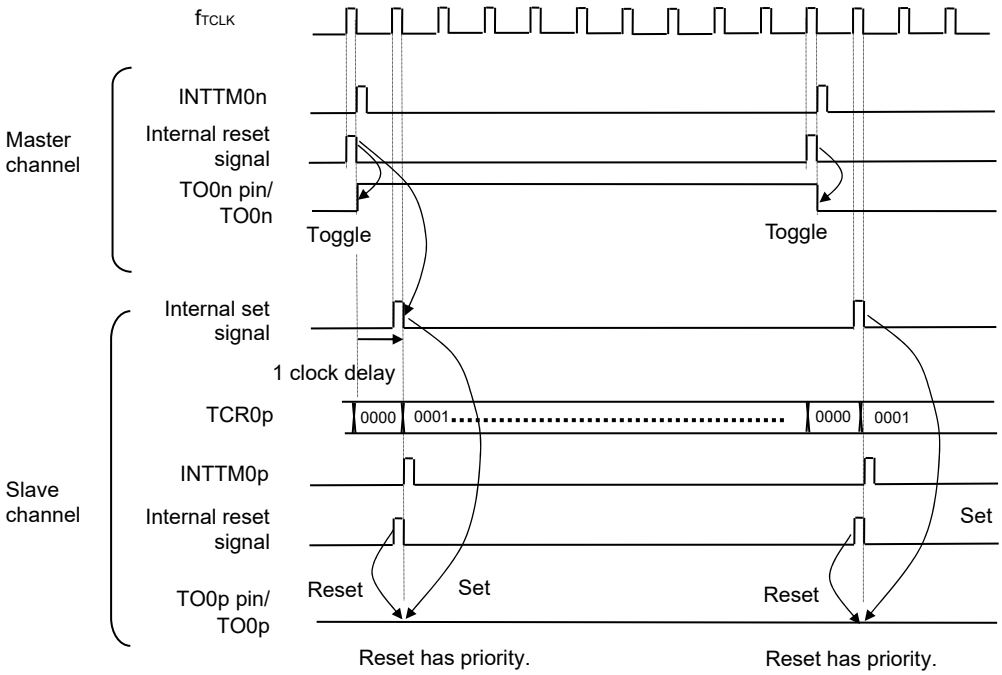
Master channel: TOE0n = 1, TOM0n = 0, TOL0n = 0
Slave channel: TOE0p = 1, TOM0p = 1, TOL0p = 0

Figure 6-35. Set/Reset Timing Operating Statuses

(1) Basic operation timing



(2) Operation timing when 0 % duty



Remarks 1. Internal reset signal: TO0n pin reset/toggle signal
Internal set signal: TO0n pin set signal

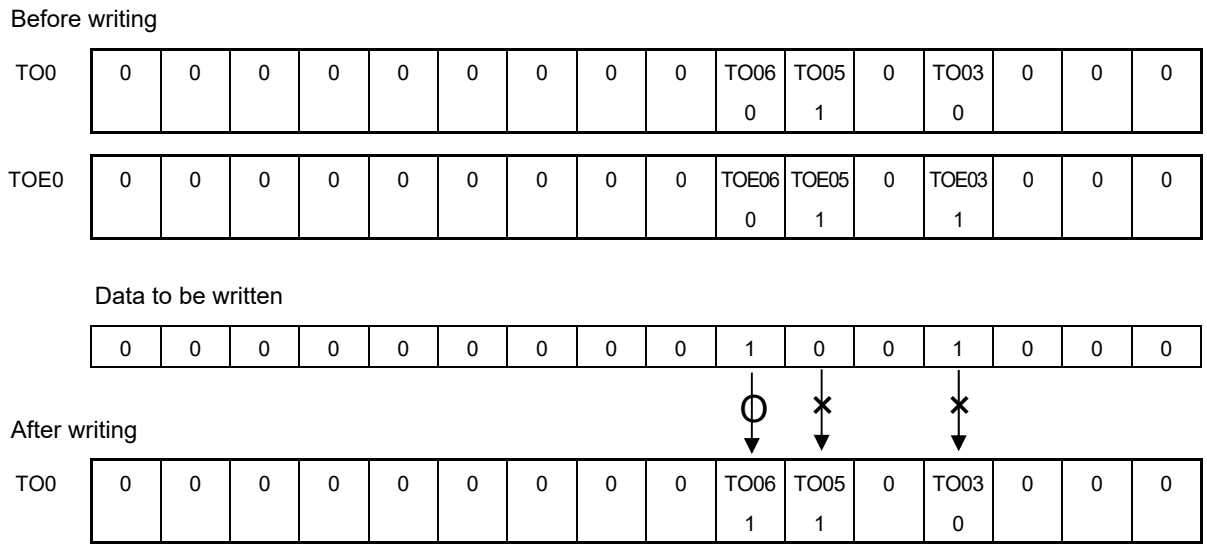
2. n: Master channel number (n = 0, 2, 4)
p: Slave channel number (n < p ≤ 6)

6.6.4 Collective manipulation of TO0n bit

In timer output register 0 (TO0), the setting bits for all the channels are located in one register in the same way as timer channel start register 0 (TS0). Therefore, the TO0n bit of all the channels can be manipulated collectively.

Only the desired bits can also be manipulated by enabling writing only to the TO0n bits (TOE0n = 0) that correspond to the relevant bits of the channel used to perform output (TO0n).

Figure 6-36. Example of TO0n Bit Collective Manipulation



Writing is done only to the TO0n bit with TOE0n = 0, and writing to the TO0n bit with TOE0n = 1 is ignored.

TO0n (channel output) to which TOE0n = 1 is set is not affected by the write operation. Even if the write operation is done to the TO0n bit, it is ignored and the output change by timer operation is normally done.

Remark n: Channel number (n = 3, 5, 6)

6.6.5 Timer interrupt and TO0n pin output at operation start

In the interval timer mode or capture mode, the MD0n0 bit in timer mode register 0n (TMR0n) sets whether or not to generate a timer interrupt at count start.

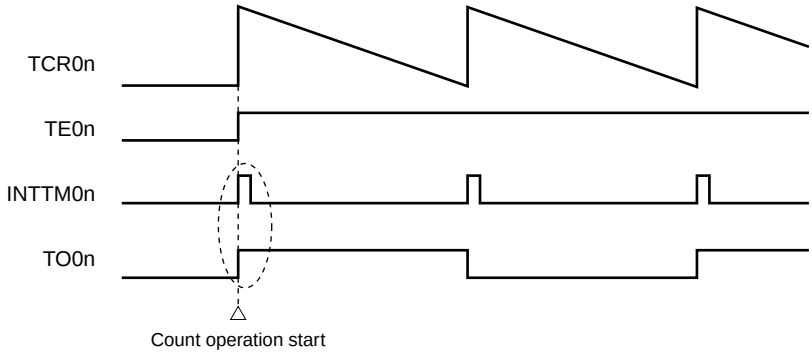
When MD0n0 is set to 1, the count operation start timing can be known by the timer interrupt (INTTM0n) generation.

In the other modes, neither timer interrupt at count operation start nor TO0n output is controlled.

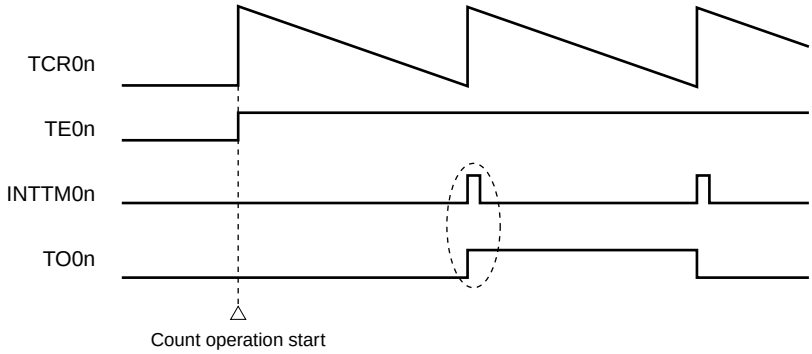
Figure 6-34 shows operation examples when the interval timer mode (TOE0n = 1, TOM0n = 0) is set.

Figure 6-37. Operation Examples of Timer Interrupt at Count Operation Start and TO0n Output

(a) When MD0n0 is set to 1



(b) When MD0n0 is set to 0



When MD0n0 is set to 1, a timer interrupt (INTTM0n) is output at count operation start, and TO0n performs a toggle operation.

When MD0n0 is set to 0, a timer interrupt (INTTM0n) is not output at count operation start, and TO0n does not change either. After counting one cycle, INTTM0n is output and TO0n performs a toggle operation.

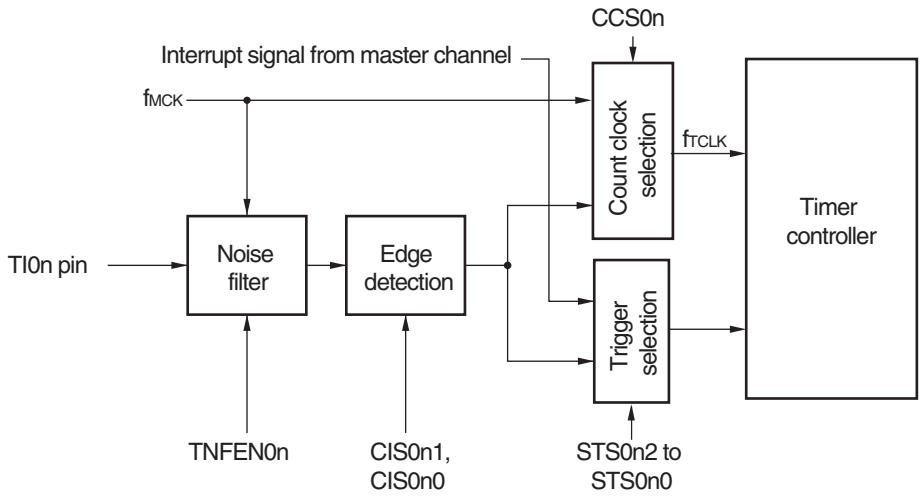
Remark n: Channel number (n = 3, 5, 6)

6.7 Timer Input (TI0n) Control

6.7.1 TI0n input circuit configuration

A signal is input from a timer input pin, goes through a noise filter and an edge detector, and is sent to a timer controller. Enable the noise filter for the pin in need of noise removal. The following shows the configuration of the input circuit.

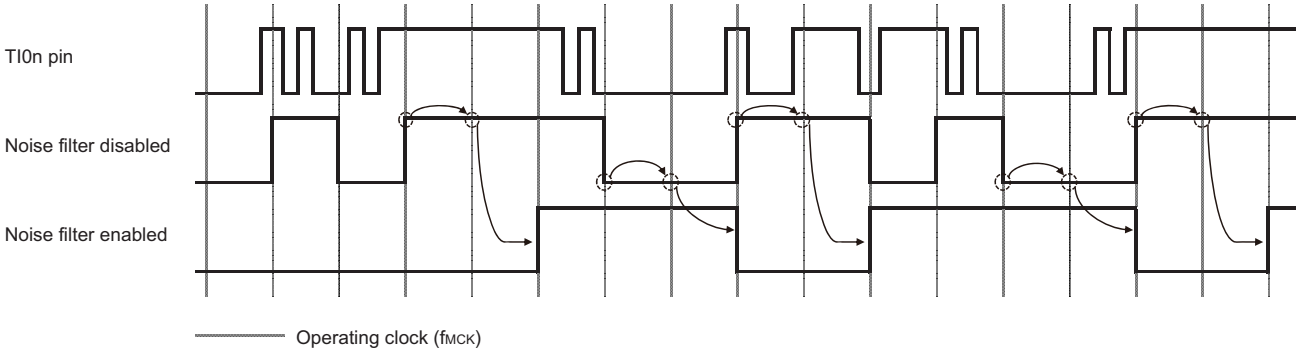
Figure 6-38. Input Circuit Configuration



6.7.2 Noise filter

When the noise filter is disabled, the input signal is only synchronized with the operating clock (f_{MCK}) for channel n. When the noise filter is enabled, after synchronization with the operating clock (f_{MCK}) for channel n, whether the signal keeps the same value for two clock cycles is detected. The following shows differences in waveforms output from the noise filter between when the noise filter is enabled and disabled.

Figure 6-39. Sampling Waveforms through TI0n Input Pin with Noise Filter Enabled and Disabled



6.7.3 Cautions on channel input operation

When a timer input pin is set as unused, the operating clock is not supplied to the noise filter. Therefore, after settings are made to use the timer input pin, the following wait time is necessary before a trigger is specified to enable operation of the channel corresponding to the timer input pin.

(1) Noise filter is disabled

When bits 12 (CCS0n), 9 (STS0n1), and 8 (STS0n0) in the timer mode register 0n (TMR0n) are 0 and then one of them is set to 1, wait for at least two cycles of the operating clock (f_{MCK}), and then set the operation enable trigger bit in the timer channel start register (TS0).

(2) Noise filter is enabled

When bits 12 (CCS0n), 9 (STS0n1), and 8 (STS0n0) in the timer mode register 0n (TMR0n) are all 0 and then one of them is set to 1, wait for at least four cycles of the operating clock (f_{MCK}), and then set the operation enable trigger bit in the timer channel start register (TS0).

6.8 Independent Channel Operation Function of Timer Array Unit

6.8.1 Operation as interval timer/square wave output

(1) Interval timer

The timer array unit can be used as a reference timer that generates INTTM0n (timer interrupt) at fixed intervals. The interrupt generation period can be calculated by the following expression.

Generation period of INTTM0n (timer interrupt) = Period of count clock × (Set value of TDR0n + 1)

(2) Operation as square wave output

TO0n performs a toggle operation as soon as INTTM0n has been generated, and outputs a square wave with a duty factor of 50%.

The period and frequency for outputting a square wave from TO0n can be calculated by the following expressions.

- Period of square wave output from TO0n = Period of count clock × (Set value of TDR0n + 1) × 2
- Frequency of square wave output from TO0n = Frequency of count clock/{(Set value of TDR0n + 1) × 2}

Timer/counter register 0n (TCR0n) operates as a down counter in the interval timer mode.

The TCR0n register loads the value of timer data register 0n (TDR0n) at the first count clock after the channel start trigger bit (TS0n, TSH01, TSH03) of timer channel start register 0 (TS0) is set to 1. If the MD0n0 bit of timer mode register 0n (TMR0n) is 0 at this time, INTTM0n is not output and TO0n is not toggled. If the MD0n0 bit of the TMR0n register is 1, INTTM0n is output and TO0n is toggled.

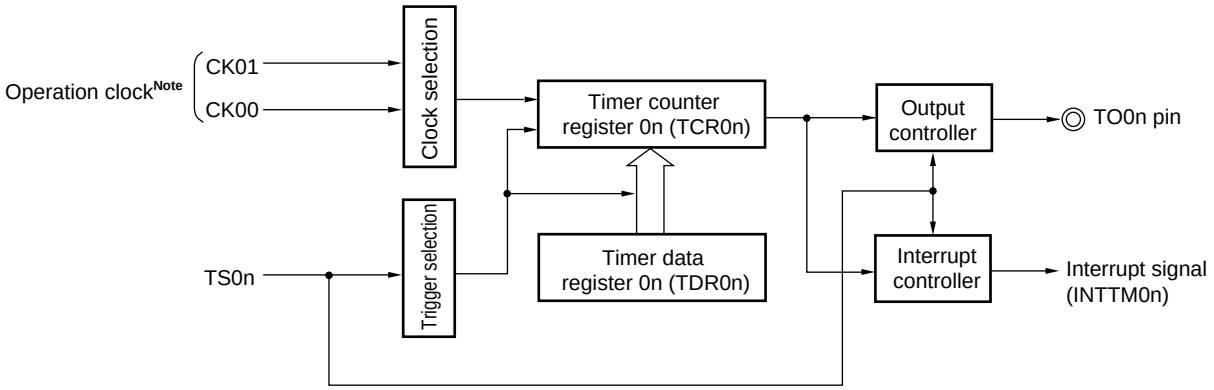
After that, the TCR0n register count down in synchronization with the count clock.

When TCR0n = 0000H, INTTM0n is output and TO0n is toggled at the next count clock. At the same time, the TCR0n register loads the value of the TDR0n register again. After that, the same operation is repeated.

The TDR0n register can be rewritten at any time. The new value of the TDR0n register becomes valid from the next period.

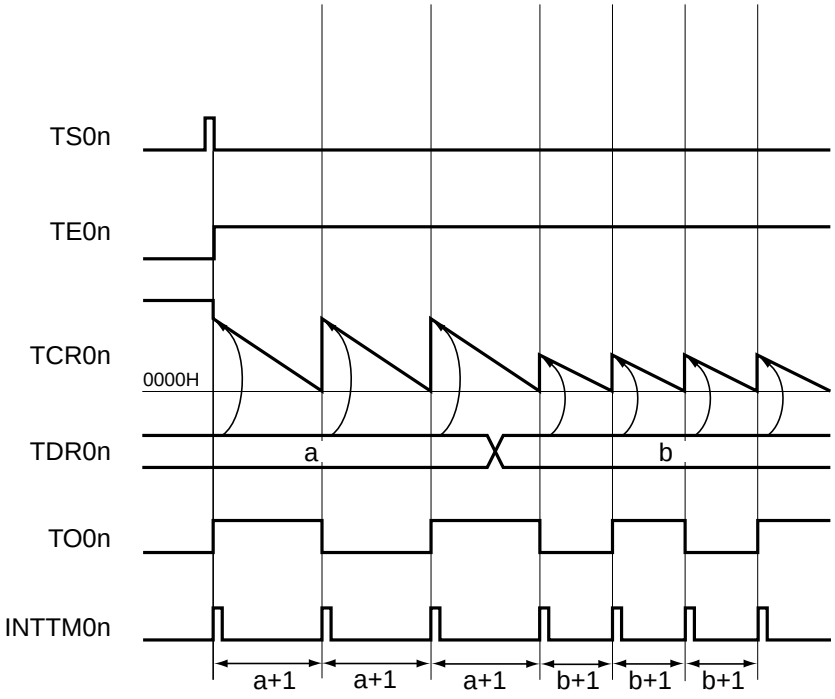
Remark n: Channel number (n = 0 to 7)

Figure 6-40. Block Diagram of Operation as Interval Timer/Square Wave Output



Note When channels 1 and 3, the clock can be selected from CK00, CK01, CK02 and CK03.

Figure 6-41. Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MD0n0 = 1)

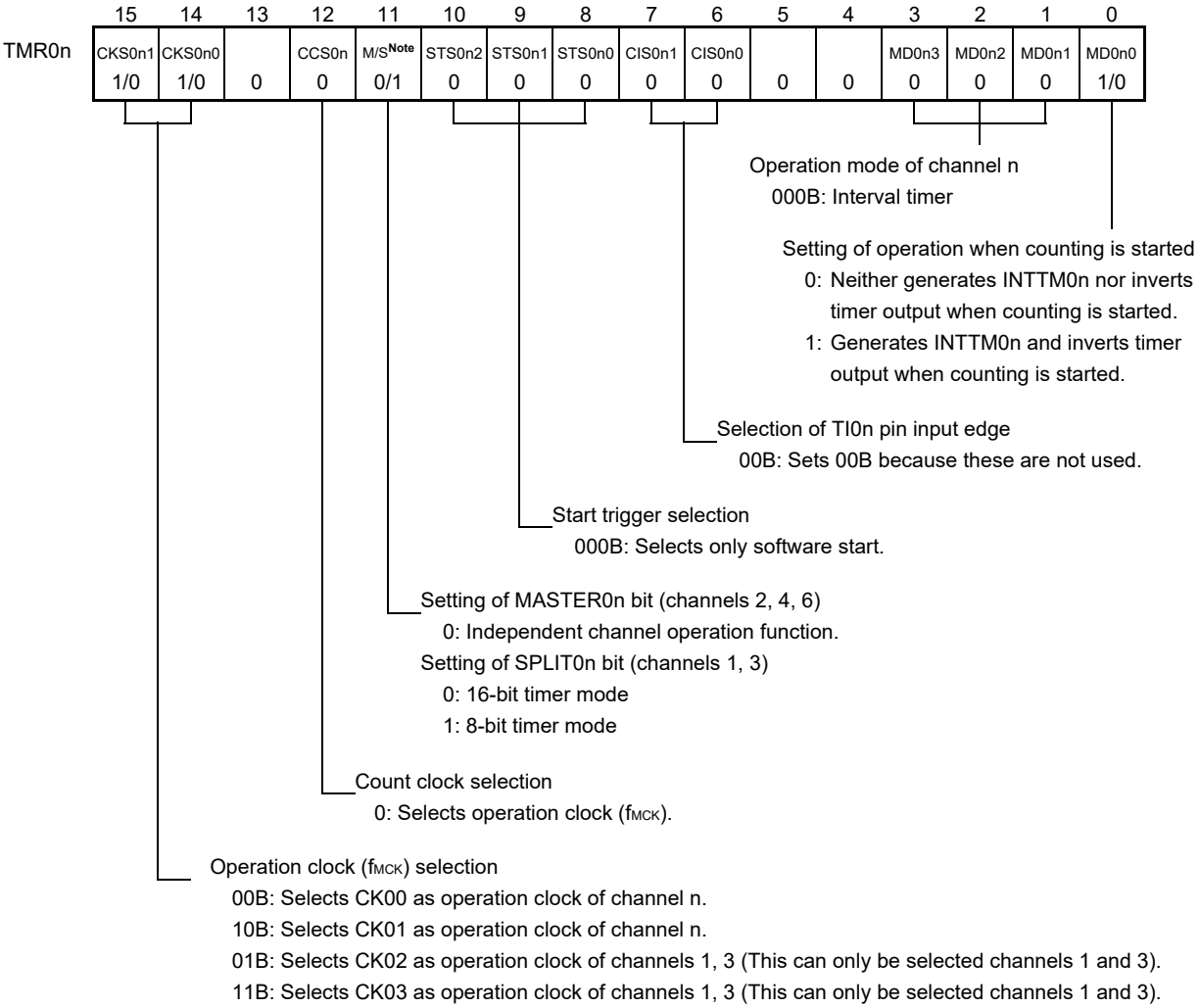


Remarks 1. n: Channel number (n = 0 to 7)

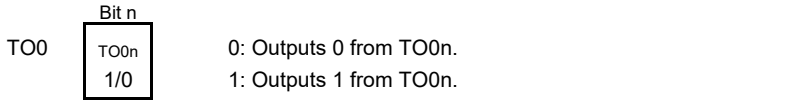
2. TS0n: Bit n of timer channel start register 0 (TS0)
TE0n: Bit n of timer channel enable status register 0 (TE0)
TCR0n: Timer/counter register 0n (TCR0n)
TDR0n: Timer data register 0n (TDR0n)
TO0n: TO0n pin output signal

Figure 6-42. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (1/2)

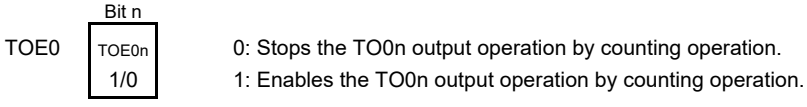
(a) Timer mode register 0n (TMR0n)



(b) Timer output register 0 (TO0)



(c) Timer output enable register 0 (TOE0)



Note TMR02, TMR04, TMR06: MASTER0n bit
TMR01, TMR03: SPLIT0n bit
TMR00, TMR05, TMR07: Fixed to 0

Remark n: Channel number (n = 0 to 7)

Figure 6-42. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (2/2)

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when TOM0n = 0 (master channel output mode)

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n 0	0: Sets master channel output mode.

Remark n: Channel number (n = 0 to 7)

Figure 6-43. Operation Procedure of Interval Timer/Square Wave Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (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 status. 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 timer mode register 0n (TMR0n) (determines operation mode of channel). Sets interval (period) value to timer data register 0n (TDR0n).	Channel stops operating. (Clock is supplied and some power is consumed.)
	To use the TO0n output Clears the TOM0n bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the TOL0n bit to 0. Sets the TO0n bit and determines default level of the TO0n output. →	The TO0n pin goes into Hi-Z output state. The TO0n default setting level is output when the port mode register is in the output mode and the port register is 0.
	Sets the TOE0n bit to 1 and enables operation of TO0n. →	TO0n does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO0n pin outputs the TO0n set level.
Operation start	(Sets the TOE0n bit to 1 only if using TO0n output and resuming operation.). Sets the TS0n (TSH01, TSH03) bit to 1. → The TS0n (TSH01, TSH03) bit automatically returns to 0 because it is a trigger bit.	TE0n (TEH01, TEH03) = 1, and count operation starts. Value of the TDR0n register is loaded to timer/counter register 0n (TCR0n). INTTM0n is generated and TO0n performs toggle operation if the MD0n0 bit of the TMR0n register is 1.
During operation	Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used. Set values of the TO0 and TOE0 registers can be changed. Set values of the TMR0n register, TOM0n, and TOL0n bits cannot be changed.	Counter (TCR0n) counts down. When count value reaches 0000H, the value of the TDR0n register is loaded to the TCR0n register again and the count operation is continued. By detecting TCR0n = 0000H, INTTM0n is generated and TO0n performs toggle operation. After that, the above operation is repeated.
Operation stop	The TT0n (TTH01, TTH03) bit is set to 1. → The TT0n (TTH01, TTH03) bit automatically returns to 0 because it is a trigger bit.	TE0n (TEH01, TEH03), and count operation stops. The TCR0n register holds count value and stops. The TO0n output is not initialized but holds current status.
	The TOE0n bit is cleared to 0 and value is set to the TO0n bit. →	The TO0n pin outputs the TO0n bit set level.

Operation is resumed.

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

Figure 6-43. Operation Procedure of Interval Timer/Square Wave Output Function (2/2)

	Software Operation	Hardware Status
TAU stop	To hold the TO0n pin output level Clears the TO0n bit to 0 after the value to be held is set to the port register. →	The TO0n pin output level is held by port function.
	When holding the TO0n pin output level is not necessary Switches the port mode register to input mode. →	The TO0n pin output level goes into Hi-Z output state.
	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0n bit is cleared to 0 and the TO0n pin is set to port mode.)

Remark n: Channel number (n = 0 to 7)

6.8.2 Operation as external event counter

The timer array unit can be used as an external event counter that counts the number of times the valid input edge (external event) is detected in the TI0n 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 TDR0n + 1

Timer/counter register 0n (TCR0n) operates as a down counter in the event counter mode.

The TCR0n register loads the value of timer data register 0n (TDR0n) by setting any channel start trigger bit (TS0n) of timer channel start register 0 (TS0) to 1.

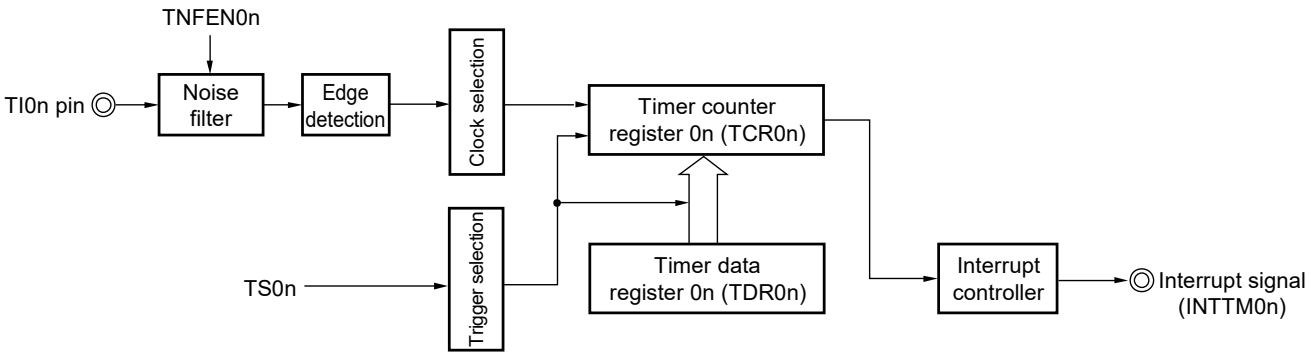
The TCR0n register counts down each time the valid input edge of the TI0n pin has been detected. When TCR0n = 0000H, the TCR0n register loads the value of the TDR0n register again, and outputs INTTM0n.

After that, the above operation is repeated.

An irregular waveform that depends on external events is output from the TO0n pin. Stop the output by setting the TOE0n bit of timer output enable register 0 (TOE0) to 0.

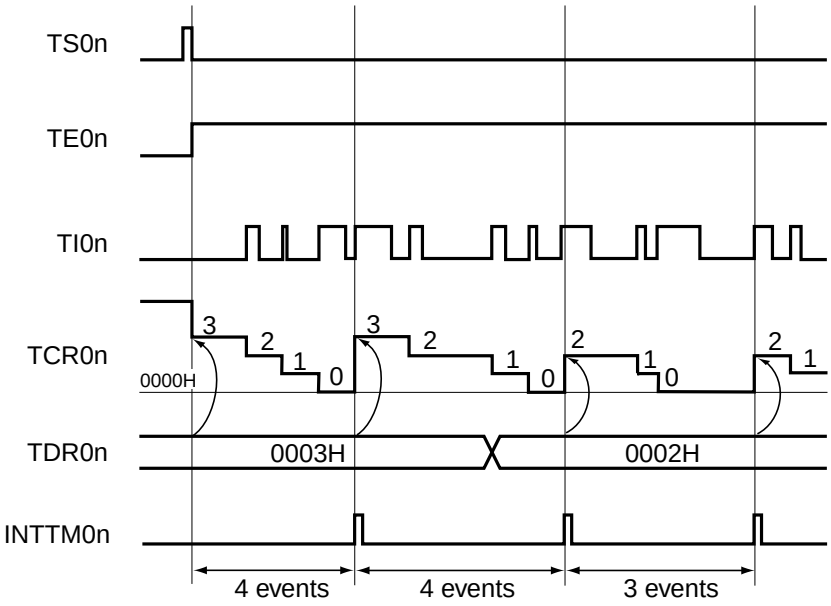
The TDR0n register can be rewritten at any time. The new value of the TDR0n register becomes valid during the next count period.

Figure 6-44. Block Diagram of Operation as External Event Counter



Remark n: Channel number (n = 3, 5, 6, 7)
The TI07 pin can be used for P11/RxD0 pin by setting bit 1 (ISC1) in the input switch control register (ISC) to 1. For details, see **6.3.13 Input switch control register (ISC)**.

Figure 6-45. Example of Basic Timing of Operation as External Event Counter

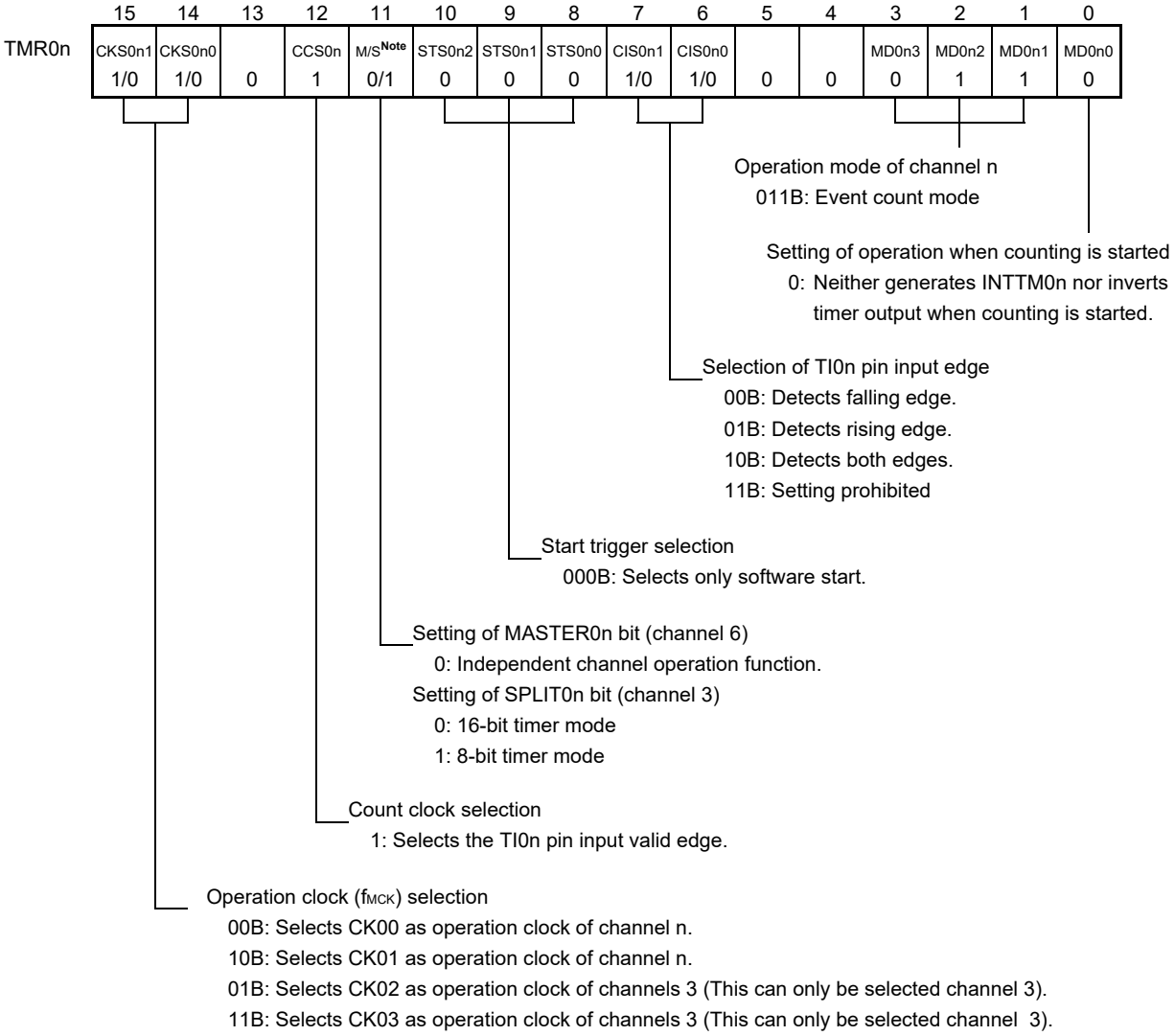


Remarks 1. n: Channel number (n = 3, 5, 6, 7)
The TI07 pin can be used for P11/RxD0 pin by setting bit 1 (ISC1) in the input switch control register (ISC) to 1. For details, see **6.3.13 Input switch control register (ISC)**.

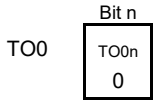
2. TS0n: Bit n of timer channel start register 0 (TS0)
TE0n: Bit n of timer channel enable status register 0 (TE0)
TI0n: TI0n pin input signal
TCR0n: Timer/counter register 0n (TCR0n)
TDR0n: Timer data register 0n (TDR0n)

Figure 6-46. Example of Set Contents of Registers in External Event Counter Mode (1/2)

(a) Timer mode register 0n (TMR0n)

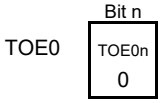


(b) Timer output register 0 (TO0)



0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)



0: Stops the TO0n output operation by counting operation.

Note TMR06: MASTER06 bit
TMR03: SPLIT03 bit
TMR05, TMR07: Fixed to 0

Remark n: Channel number (n = 3, 5, 6, 7)
The TI07 pin can be used for P11/RxD0 pin by setting bit 1 (ISC1) in the input switch control register (ISC) to 1. For details, see **6.3.13 Input switch control register (ISC)**.

Figure 6-46. Example of Set Contents of Registers in External Event Counter Mode (2/2)

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when TOM0n = 0 (master channel output mode).

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n 0	0: Sets master channel output mode.

Remark n: Channel number (n = 3, 5, 6, 7)
The TI07 pin can be used for P11/RxD0 pin by setting bit 1 (ISC1) in the input switch control register (ISC) to 1. For details, see **6.3.13 Input switch control register (ISC)**.

Figure 6-47. Operation Procedure When External Event Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (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 status. 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). Sets number of counts to timer data register 0n (TDR0n). Clears the TOE0n bit of timer output enable register 0 (TOE0) to 0.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TS0n (TSH03) bit to 1. → The TS0n (TSH03) bit automatically returns to 0 because it is a trigger bit.	TE0n (TEH03) = 1, and count operation starts. Value of the TDR0n register is loaded to timer/counter register 0n (TCR0n) and detection of the TI0n pin input edge is awaited.
During operation	Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used. Set values of the TMR0n register, TOM0n, TOL0n, TO0n, and TOE0n bits cannot be changed.	Counter (TCR0n) counts down each time input edge of the TI0n pin has been detected. When count value reaches 0000H, the value of the TDR0n register is loaded to the TCR0n register again, and the count operation is continued. By detecting TCR0n = 0000H, the INTTM0n output is generated. After that, the above operation is repeated.
Operation stop	The TT0n (TTH03) bit is set to 1. → The TT0n (TTH03) bit automatically returns to 0 because it is a trigger bit.	TE0n (TEH03) = 0, and count operation stops. The TCR0n register holds count value and stops.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark n: Channel number (n = 3, 5, 6, 7)
The TI07 pin can be used for P11/RxD0 pin by setting bit 1 (ISC1) in the input switch control register (ISC) to 1. For details, see **6.3.13 Input switch control register (ISC)**.

6.8.3 Operation as input pulse interval measurement

The count value can be captured at the TI0n valid edge and the interval of the pulse input to TI0n can be measured. In addition, the count value can be captured by using software operation (TS0n = 1) as a capture trigger while the TE0n bit is set to 1.

The pulse interval can be calculated by the following expression.

TI0n input pulse interval = Period of count clock × ((10000H × TSR0n: OVF) + (Capture value of TDR0n + 1))

Caution The TI0n pin input is sampled using the operating clock selected with the CKS0n bit of timer mode register 0n (TMR0n), so an error of up to one operating clock cycle occurs.

Timer/counter register 0n (TCR0n) operates as an up counter in the capture mode.

When the channel start trigger bit (TS0n) of timer channel start register 0 (TS0) is set to 1, the TCR0n register counts up from 0000H in synchronization with the count clock.

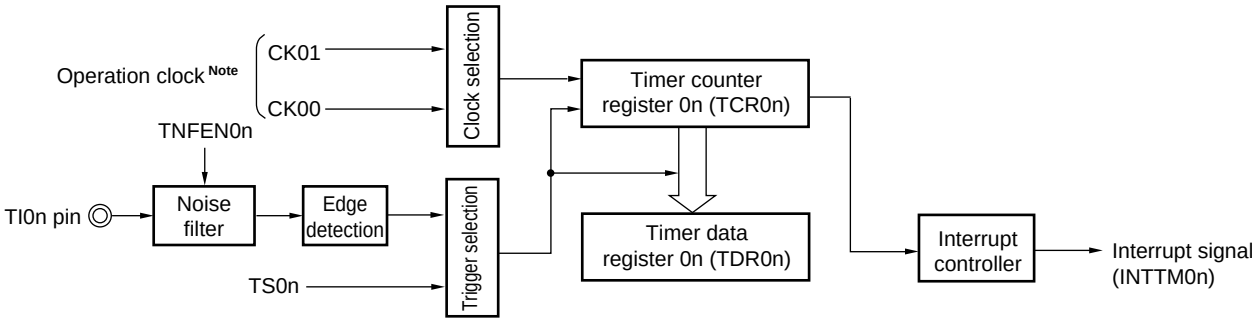
When the TI0n pin input valid edge is detected, the count value of the TCR0n register is transferred (captured) to timer data register 0n (TDR0n) and, at the same time, the TCR0n register is cleared to 0000H, and the INTTM0n is output. If the counter overflows at this time, the OVF bit of timer status register 0n (TSR0n) 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 TDR0n register, the OVF bit of the TSR0n register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSR0n 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 STS0n2 to STS0n0 bits of the TMR0n register to 001B to use the valid edges of TI0n as a start trigger and a capture trigger.

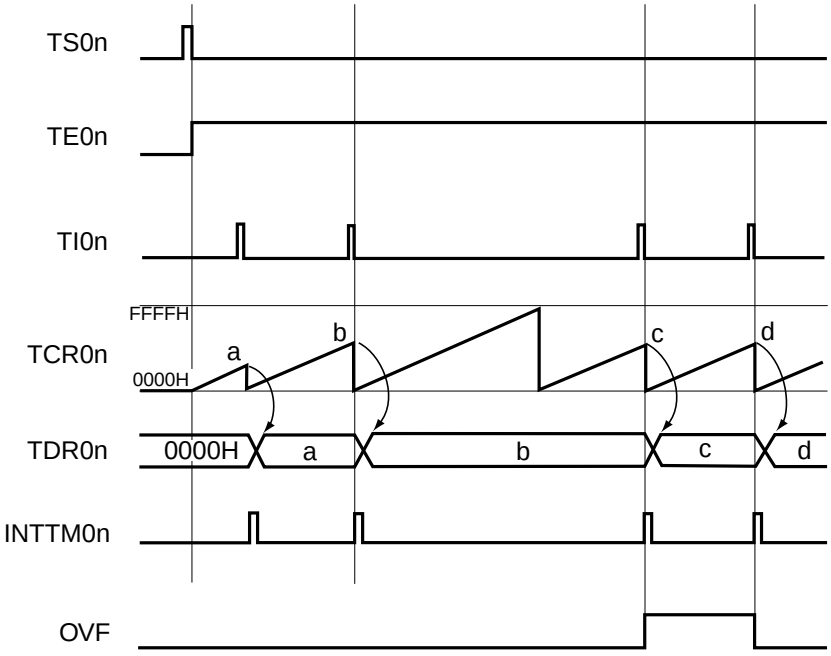
Figure 6-48. Block Diagram of Operation as Input Pulse Interval Measurement



Note When channel 3, the clock can be selected from CK00, CK01, CK02 and CK03.

Remark n: Channel number (n = 3, 5, 6, 7)
The TI07 pin can be used for P11/RxD0 pin by setting bit 1 (ISC1) in the input switch control register (ISC) to 1. For details, see **6.3.13 Input switch control register (ISC)**.

Figure 6-49. Example of Basic Timing of Operation as Input Pulse Interval Measurement (MD0n0 = 0)

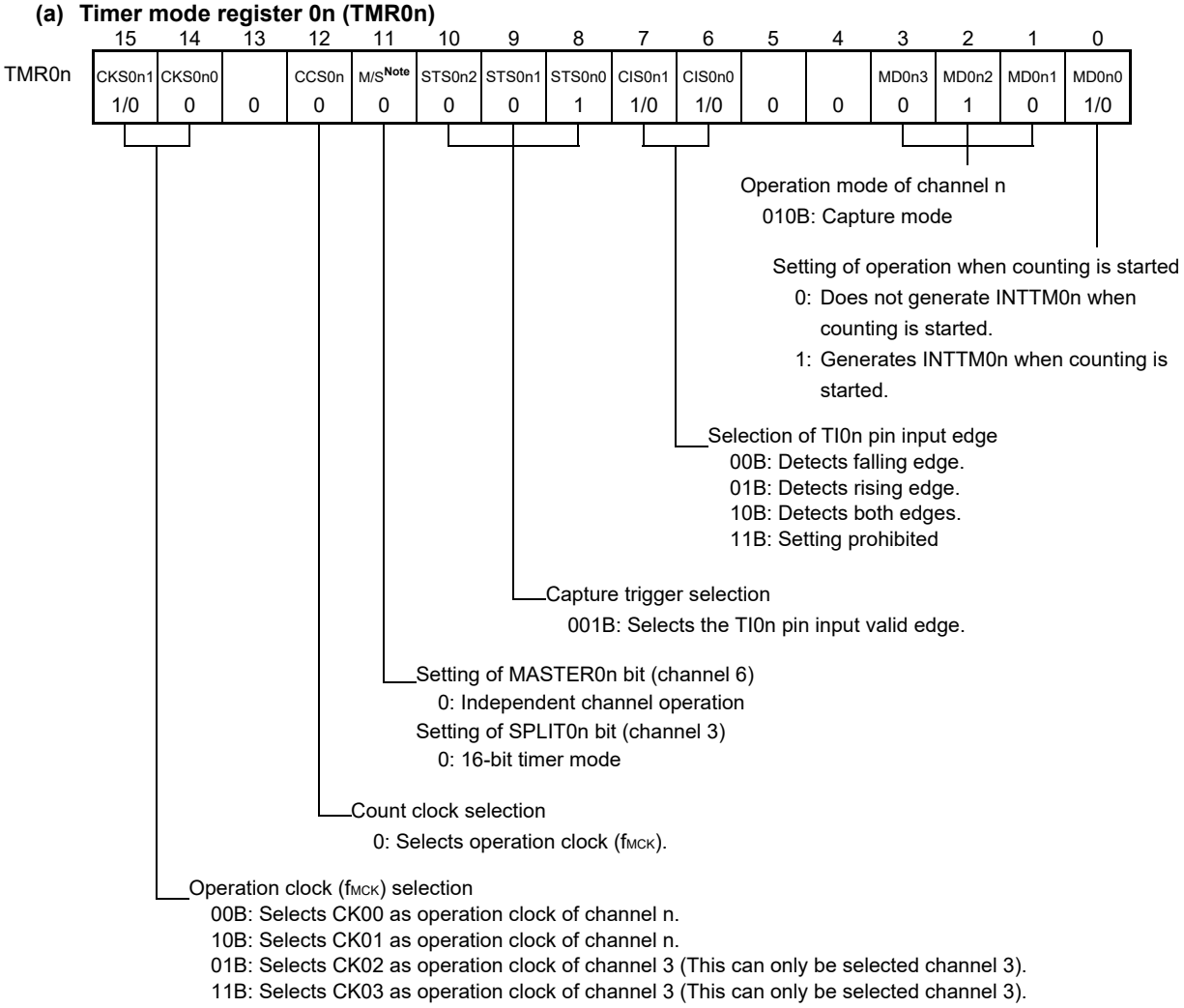


Remarks 1. n: Channel number (n = 3, 5, 6, 7)
The TI07 pin can be used for P11/RxD0 pin by setting bit 1 (ISC1) in the input switch control register (ISC) to 1. For details, see **6.3.13 Input switch control register (ISC)**.

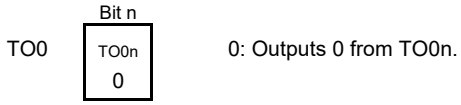
2. TS0n: Bit n of timer channel start register 0 (TS0)
TE0n: Bit n of timer channel enable status register 0 (TE0)
TI0n: TI0n pin input signal
TCR0n: Timer/counter register 0n (TCR0n)
TDR0n: Timer data register 0n (TDR0n)
OVF: Bit 0 of timer status register 0n (TSR0n)

Figure 6-50. Example of Set Contents of Registers to Measure Input Pulse Interval (1/2)

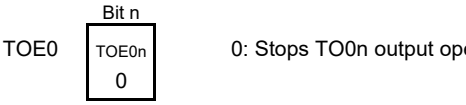
(a) Timer mode register 0n (TMR0n)



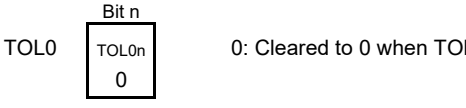
(b) Timer output register 0 (TO0)



(c) Timer output enable register 0 (TOE0)



(d) Timer output level register 0 (TOL0)



Note TMR06: MASTER0n bit
TMR03: SPLIT0n bit
TMR05, TMR07: Fixed to 0

Remark n: Channel number (n = 3, 5, 6, 7)
The TI07 pin can be used for P11/RxD0 pin by setting bit 1 (ISC1) in the input switch control register (ISC) to 1. For details, see **6.3.13 Input switch control register (ISC)**.

Figure 6-50. Example of Set Contents of Registers to Measure Input Pulse Interval (2/2)

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n 0	0: Sets master channel output mode.

Note TMR06: MASTER0n bit
TMR03: SPLIT0n bit
TMR05, TMR07: Fixed to 0

Remark n: Channel number (n = 3, 5, 6, 7)
The TI07 pin can be used for P11/RxD0 pin by setting bit 1 (ISC1) in the input switch control register (ISC) to 1. For details, see **6.3.13 Input switch control register (ISC)**.

Figure 6-51. Operation Procedure When Input Pulse Interval Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (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 status. 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).	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets TS0n bit to 1. → The TS0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 1, and count operation starts. Timer/counter register 0n (TCR0n) is cleared to 0000H. When the MD0n0 bit of the TMR0n register is 1, INTTM0n is generated.
During operation	Set values of only the CIS0n1 and CIS0n0 bits of the TMR0n register can be changed. The TDR0n register can always be read. The TCR0n register can always be read. The TSR0n register can always be read. Set values of the TOM0n, TOL0n, TO0n, and TOE0n bits cannot be changed.	Counter (TCR0n) counts up from 0000H. When the TI0n pin input valid edge is detected or the TS0n bit is set to 1, the count value is transferred (captured) to timer data register 0n (TDR0n). At the same time, the TCR0n register is cleared to 0000H, and the INTTM0n signal is generated. If an overflow occurs at this time, the OVF bit of timer status register 0n (TSR0n) is set; if an overflow does not occur, the OVF bit is cleared. After that, the above operation is repeated.
Operation stop	The TT0n bit is set to 1. → The TT0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 0, and count operation stops. The TCR0n register holds count value and stops. The OVF bit of the TSR0n register is also held.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark n: Channel number (n = 3, 5, 6, 7)
The TI07 pin can be used for P11/RxD0 pin by setting bit 1 (ISC1) in the input switch control register (ISC) to 1. For details, see **6.3.13 Input switch control register (ISC)**.

6.8.4 Operation as input signal high-/low-level width measurement

Caution When using a channel to implement the LIN-bus and DMX512, read TI0n as RxD0 in the following descriptions.

By starting counting at one edge of the TI0n pin input and capturing the number of counts at another edge, the signal width (high-level width/low-level width) of TI0n can be measured. The signal width of TI0n can be calculated by the following expression.

Signal width of TI0n input = Period of count clock × ((10000H × TSR0n: OVF) + (Capture value of TDR0n + 1))

Caution The TI0n pin input is sampled using the operating clock selected with the CKS0n bit of timer mode register 0n (TMR0n), so an error equivalent to one operation clock occurs.

Timer/counter register 0n (TCR0n) operates as an up counter in the capture & one-count mode.

When the channel start trigger bit (TS0n) of timer channel start register 0 (TS0) is set to 1, the TE0n bit is set to 1 and the TI0n pin start edge detection wait status is set.

When the TI0n pin input start edge (rising edge of the TI0n pin input when the high-level width is to be measured) is detected, the counter counts up from 0000H in synchronization with the count clock. When the valid capture edge (falling edge of the TI0n pin input when the high-level width is to be measured) is detected later, the count value is transferred to timer data register 0n (TDR0n) and, at the same time, INTTM0n is output. If the counter overflows at this time, the OVF bit of timer status register 0n (TSR0n) is set to 1. If the counter does not overflow, the OVF bit is cleared. The TCR0n register stops at the value “value transferred to the TDR0n register + 1”, and the TI0n pin start edge detection wait status is set. After that, the above operation is repeated.

As soon as the count value has been captured to the TDR0n register, the OVF bit of the TSR0n register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSR0n register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

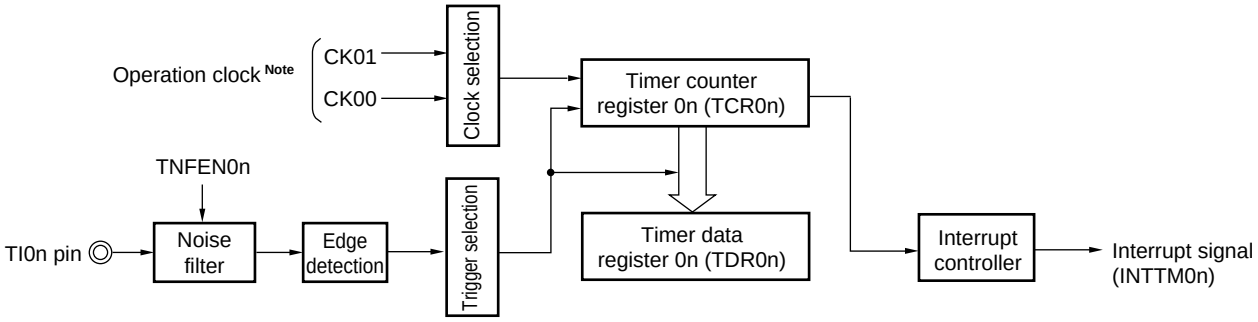
Whether the high-level width or low-level width of the TI0n pin is to be measured can be selected by using the CIS0n1 and CIS0n0 bits of the TMR0n register.

Because this function is used to measure the signal width of the TI0n pin input, the TS0n bit cannot be set to 1 while the TE0n bit is 1.

CIS0n1, CIS0n0 of TMR0n register = 10B: Low-level width is measured.

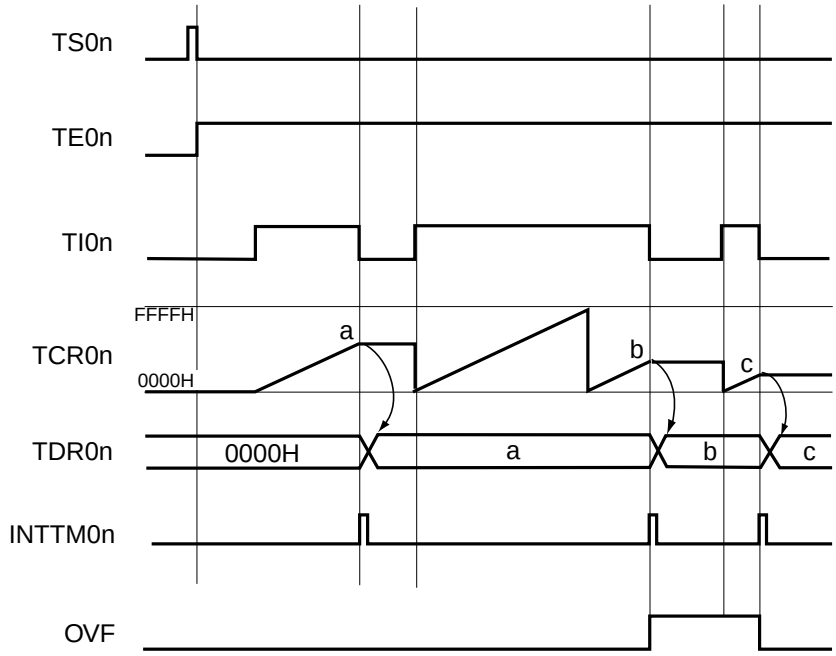
CIS0n1, CIS0n0 of TMR0n register = 11B: High-level width is measured.

Figure 6-52. Block Diagram of Operation as Input Signal High-/Low-Level Width Measurement



Note For channel 3, the clock can be selected from CK00, CK01, CK02 and CK03.

Figure 6-53. Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement

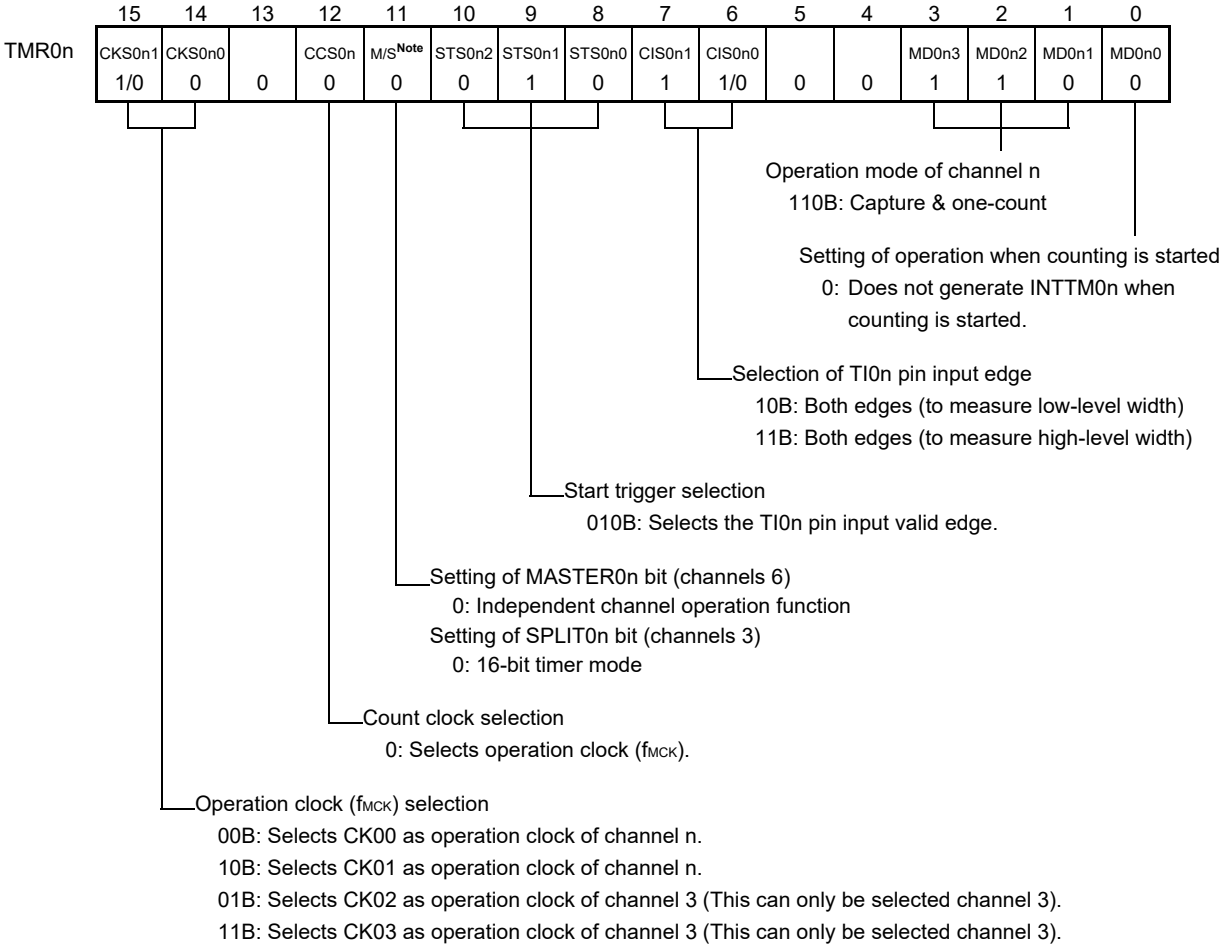


Remarks 1. n: Channel number (n = 3, 5, 6, 7)
The TI07 pin can be used for P11/RxD0 pin by setting bit 1 (ISC1) in the input switch control register (ISC) to 1. For details, see **6.3.13 Input switch control register (ISC)**.

2. TS0n: Bit n of timer channel start register 0 (TS0)
TE0n: Bit n of timer channel enable status register 0 (TE0)
TI0n: TI0n pin input signal
TCR0n: Timer/counter register 0n (TCR0n)
TDR0n: Timer data register 0n (TDR0n)
OVF: Bit 0 of timer status register 0n (TSR0n)

Figure 6-54. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width (1/2)

(a) Timer mode register 0n (TMR0n)



(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n 0	0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n 0	0: Stops the TO0n output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when TOM0n = 0 (master channel output mode).

Note TMR06: MASTER06 bit
TMR03: SPLIT03 bit
TMR05, TMR07: Fixed to 0

Remark n: Channel number (n = 3, 5, 6, 7)
The TI07 pin can be used for P11/RxD0 pin by setting bit 1 (ISC1) in the input switch control register (ISC) to 1. For details, see **6.3.13 Input switch control register (ISC)**.

Figure 6-54. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width (2/2)

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n 0	0: Sets master channel output mode.

Note TMR06: MASTER06 bit
TMR03: SPLIT03 bit
TMR05, TMR07: Fixed to 0

Remark n: Channel number (n = 3, 5, 6, 7)
The TI07 pin can be used for P11/RxD0 pin by setting bit 1 (ISC1) in the input switch control register (ISC) to 1. For details, see **6.3.13 Input switch control register (ISC)**.

Figure 6-55. Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (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 status. 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). Clears the TOE0n bit to 0 and stops operation of TO0n.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TS0n bit to 1. → The TS0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 1, and the TI0n pin start edge detection wait status is set.
	Detects the TI0n pin input count start valid edge. →	Clears timer/counter register 0n (TCR0n) to 0000H and starts counting up.
During operation	Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used. Set values of the TMR0n register, TOM0n, TOL0n, TO0n, and TOE0n bits cannot be changed.	When the TI0n pin start edge is detected, the counter (TCR0n) counts up from 0000H. If a capture edge of the TI0n pin is detected, the count value is transferred to timer data register 0n (TDR0n) and INTTM0n is generated. If an overflow occurs at this time, the OVF bit of timer status register 0n (TSR0n) is set; if an overflow does not occur, the OVF bit is cleared. The TCR0n register stops the count operation until the next TI0n pin start edge is detected.
Operation stop	The TT0n bit is set to 1. → The TT0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 0, and count operation stops. The TCR0n register holds count value and stops. The OVF bit of the TSR0n register is also held.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark n: Channel number (n = 3, 5, 6, 7)
The TI07 pin can be used for P11/RxD0 pin by setting bit 1 (ISC1) in the input switch control register (ISC) to 1. For details, see **6.3.13 Input switch control register (ISC)**.

6.8.5 Operation as delay counter

It is possible to start counting down when the valid edge of the TI0n pin input is detected (an external event), and then generates INTTM0n (a timer interrupt) after any specified interval.

It is also possible to start counting down and generate INTTM0n (timer interrupt) at any interval by setting TS0n to 1 by software while TE0n = 1.

The interrupt generation period can be calculated by the following expression.

Generation period of INTTM0n (timer interrupt) = Period of count clock × (Set value of TDR0n + 1)

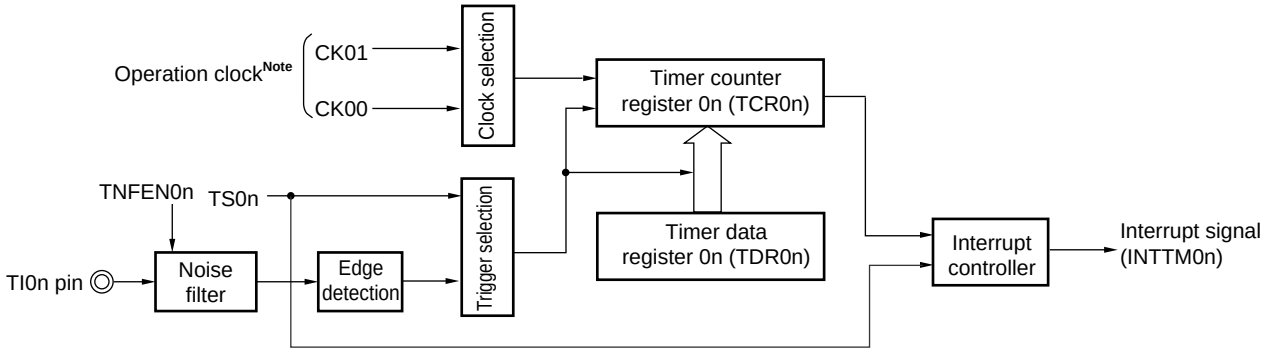
Timer/counter register 0n (TCR0n) operates as a down counter in the one-count mode.

When the channel start trigger bit (TS0n, TSH03) of timer channel start register 0 (TS0) is set to 1, the TE0n, TEH03 bits are set to 1 and the TI0n pin input valid edge detection wait status is set.

Timer/counter register 0n (TCR0n) starts operating upon TI0n pin input valid edge detection and loads the value of timer data register 0n (TDR0n). The TCR0n register counts down from the value of the TDR0n register it has loaded, in synchronization with the count clock. When TCR0n = 0000H, it outputs INTTM0n and stops counting until the next TI0n pin input valid edge is detected.

The TDR0n register can be rewritten at any time. The new value of the TDR0n register becomes valid from the next period.

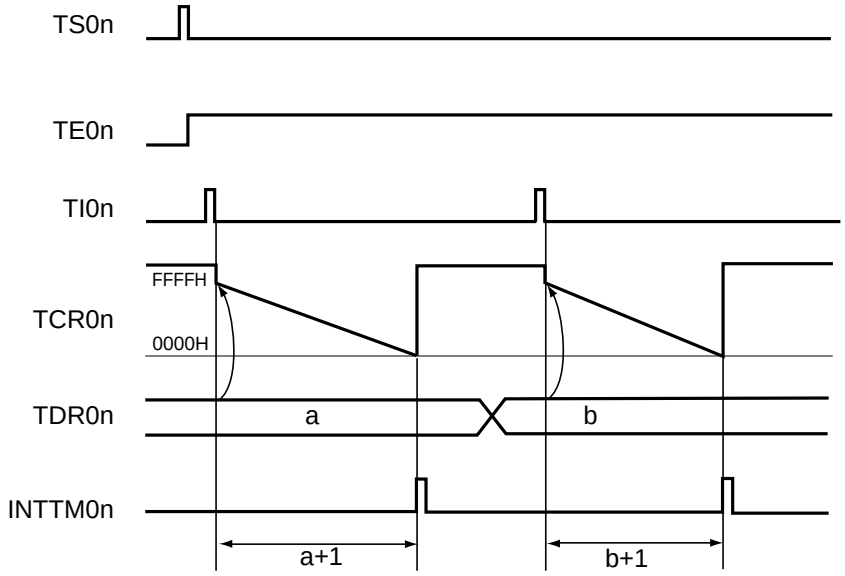
Figure 6-56. Block Diagram of Operation as Delay Counter



Note For using channel 3, the clock can be selected from CK00, CK01, CK02 and CK03.

Remark n: Channel number (n = 3, 5, 6, 7)
The TI07 pin can be used for P11/RxD0 pin by setting bit 1 (ISC1) in the input switch control register (ISC) to 1. For details, see **6.3.13 Input switch control register (ISC)**.

Figure 6-57. Example of Basic Timing of Operation as Delay Counter

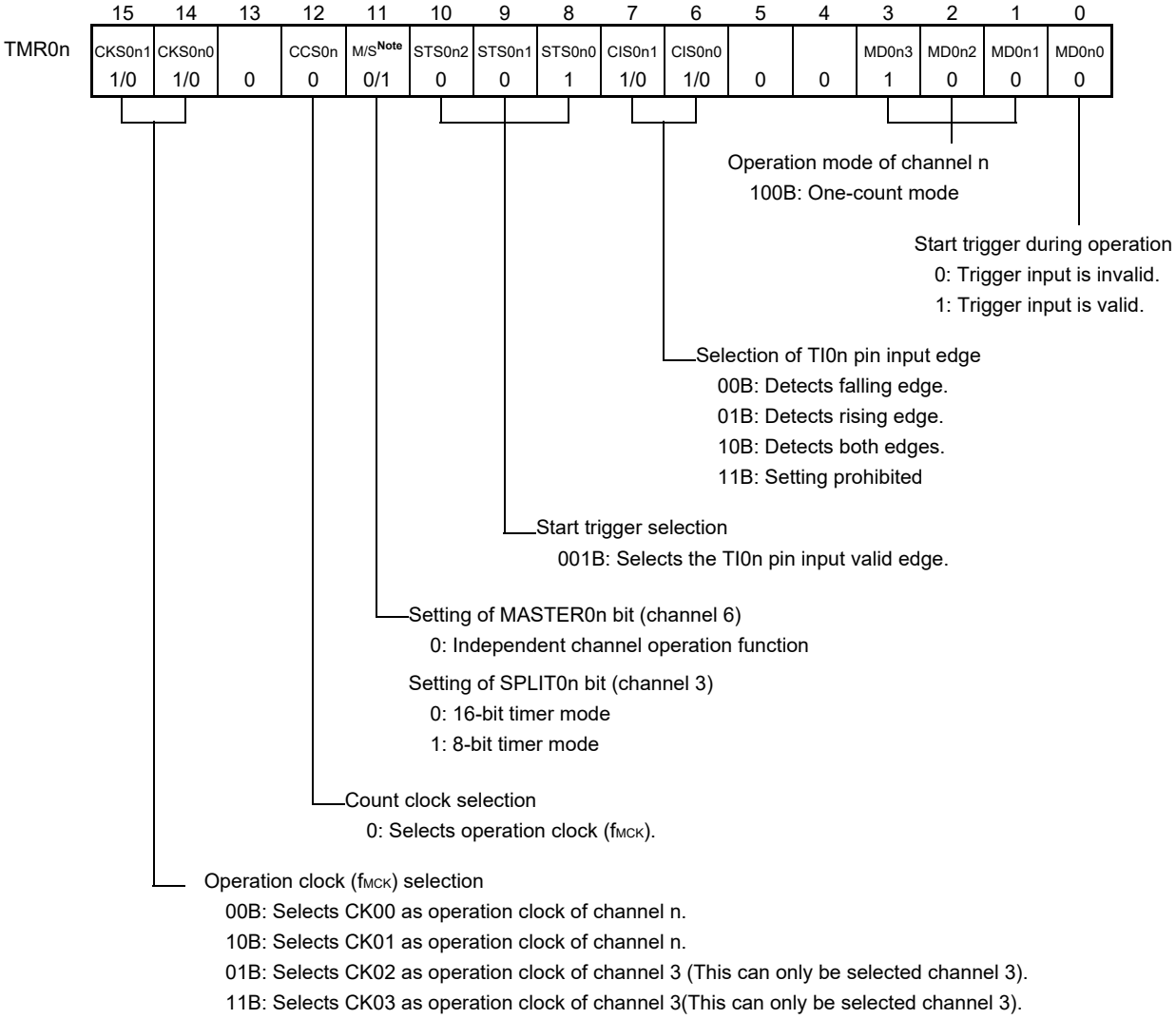


Remarks 1. n: Channel number (n = 3, 5, 6, 7)
The TI07 pin can be used for P11/RxD0 pin by setting bit 1 (ISC1) in the input switch control register (ISC) to 1. For details, see **6.3.13 Input switch control register (ISC)**.

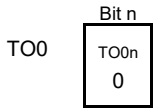
2. TS0n: Bit n of timer channel start register 0 (TS0)
TE0n: Bit n of timer channel enable status register 0 (TE0)
TI0n: TI0n pin input signal
TCR0n: Timer/counter register 0n (TCR0n)
TDR0n: Timer data register 0n (TDR0n)

Figure 6-58. Example of Set Contents of Registers to Delay Counter (1/2)

(a) Timer mode register 0n (TMR0n)

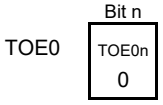


(b) Timer output register 0 (TO0)



0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)



0: Stops the TO0n output operation by counting operation.

Note TMR06: MASTER06 bit
TMR03: SPLIT03 bit
TMR05, TMR07: Fixed to 0

Remark n: Channel number (n = 3, 5, 6, 7)
The TI07 pin can be used for P11/RxD0 pin by setting bit 1 (ISC1) in the input switch control register (ISC) to 1. For details, see **6.3.13 Input switch control register (ISC)**.

Figure 6-58. Example of Set Contents of Registers to Delay Counter (2/2)

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when TOM0n = 0 (master channel output mode).

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n 0	0: Sets master channel output mode.

Remark n: Channel number (n = 3, 5, 6, 7)
The TI07 pin can be used for P11/RxD0 pin by setting bit 1 (ISC1) in the input switch control register (ISC) to 1. For details, see **6.3.13 Input switch control register (ISC)**.

Figure 6-59. Operation Procedure When Delay Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (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 status. 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). INTTM0n output delay is set to timer data register 0n (TDR0n). Clears the TOE0n bit to 0 and stops operation of TO0n.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TS0n (TSH03) bit to 1. → The TS0n (TSH03) bit automatically returns to 0 because it is a trigger bit.	TE0n (TEH03) = 1, and the start trigger detection (the valid edge of the TI0n pin input is detected or the TS0n bit is set to 1) wait status is set.
	The counter starts counting down by the next start trigger detection. • Detects the TI0n pin input valid edge. → • Sets the TS0n bit to 1 by the software.	Value of the TDR0n register is loaded to the timer/counter register 0n (TCR0n).
During operation	Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used.	The counter (TCR0n) counts down. When the count value of TCR0n reaches to 0000H, the INTTM0n output is generated, and the count operation stops until the next start trigger detection (the valid edge of the TI0n pin input is detected or the TS0n bit is set to 1).
Operation stop	The TT0n (TTH03) bit is set to 1. → The TT0n (TTH03) bit automatically returns to 0 because it is a trigger bit.	TE0n (TEH03) = 0, and count operation stops. The TCR0n register holds count value and stops.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark n: Channel number (n = 3, 5, 6, 7)
The TI07 pin can be used for P11/RxD0 pin by setting bit 1 (ISC1) in the input switch control register (ISC) to 1. For details, see **6.3.13 Input switch control register (ISC)**.

6.9 Simultaneous Channel Operation Function of Timer Array Unit

6.9.1 Operation as one-shot pulse output function

By using two channels as a set, a one-shot pulse having any delay pulse width can be generated from the start trigger with software operation.

The delay time and pulse width can be calculated by the following expressions.

Delay time = {Set value of TDR0n (master) + 2} × Count clock period
Pulse width = {Set value of TDR0p (slave)} × Count clock period

The master channel operates in the one-count mode and counts the delays. Timer/counter register 0n (TCR0n) of the master channel starts operating upon start trigger detection and loads the value of timer data register 0n (TDR0n).

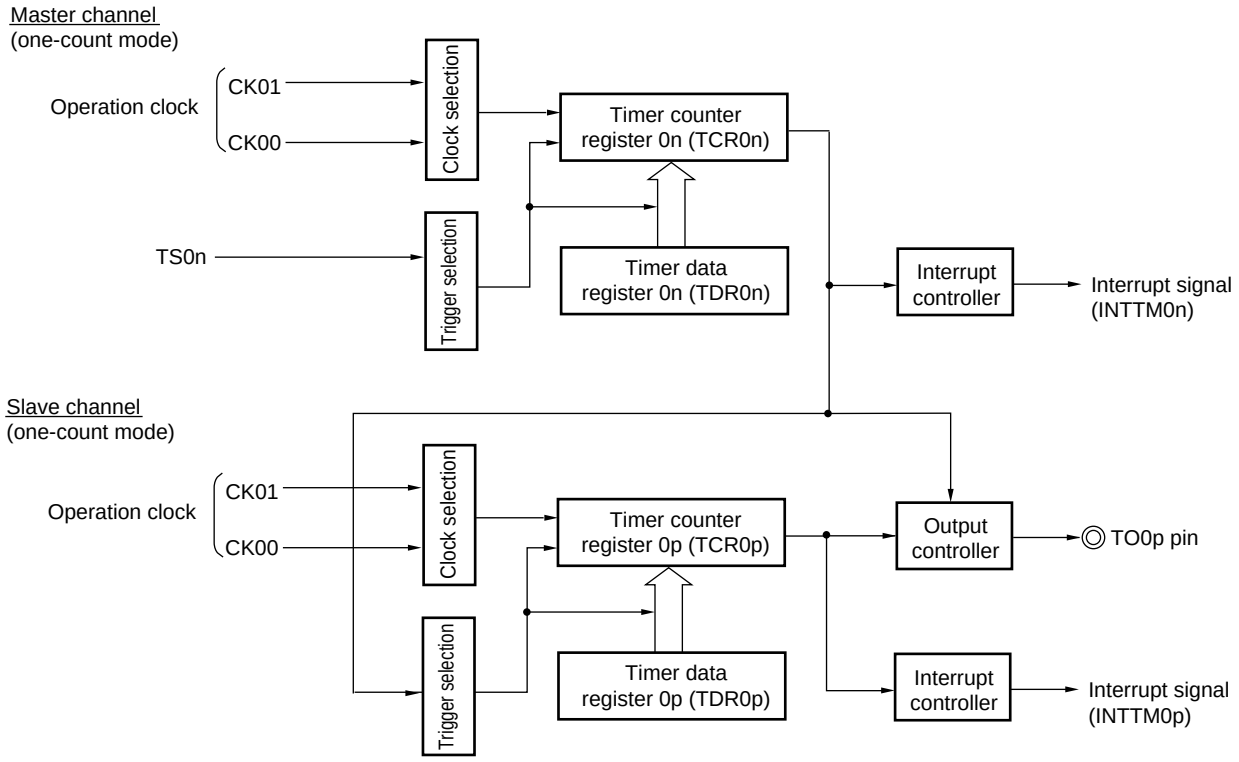
The TCR0n register counts down from the value of the TDR0n register it has loaded, in synchronization with the count clock. When TCR0n = 0000H, it outputs INTTM0n and stops counting until the next start trigger is detected.

The slave channel operates in the one-count mode and counts the pulse width. The TCR0p register of the slave channel starts operation using INTTM0n of the master channel as a start trigger, and loads the value of the TDR0p register. The TCR0p register counts down from the value of The TDR0p register it has loaded, in synchronization with the count value. When count value = 0000H, it outputs INTTM0p and stops counting until the next start trigger (INTTM0n of the master channel) is detected. The output level of TO0p becomes active one count clock after generation of INTTM0n from the master channel, and inactive when TCR0p = 0000H.

Caution The timing of loading of timer data register 0n (TDR0n) of the master channel is different from that of the TDR0p register of the slave channel. If the TDR0n and TDR0p registers are rewritten during operation, therefore, an illegal waveform is output. Rewrite the TDR0n register after INTTM0n is generated and the TDR0p register after INTTM0p is generated.

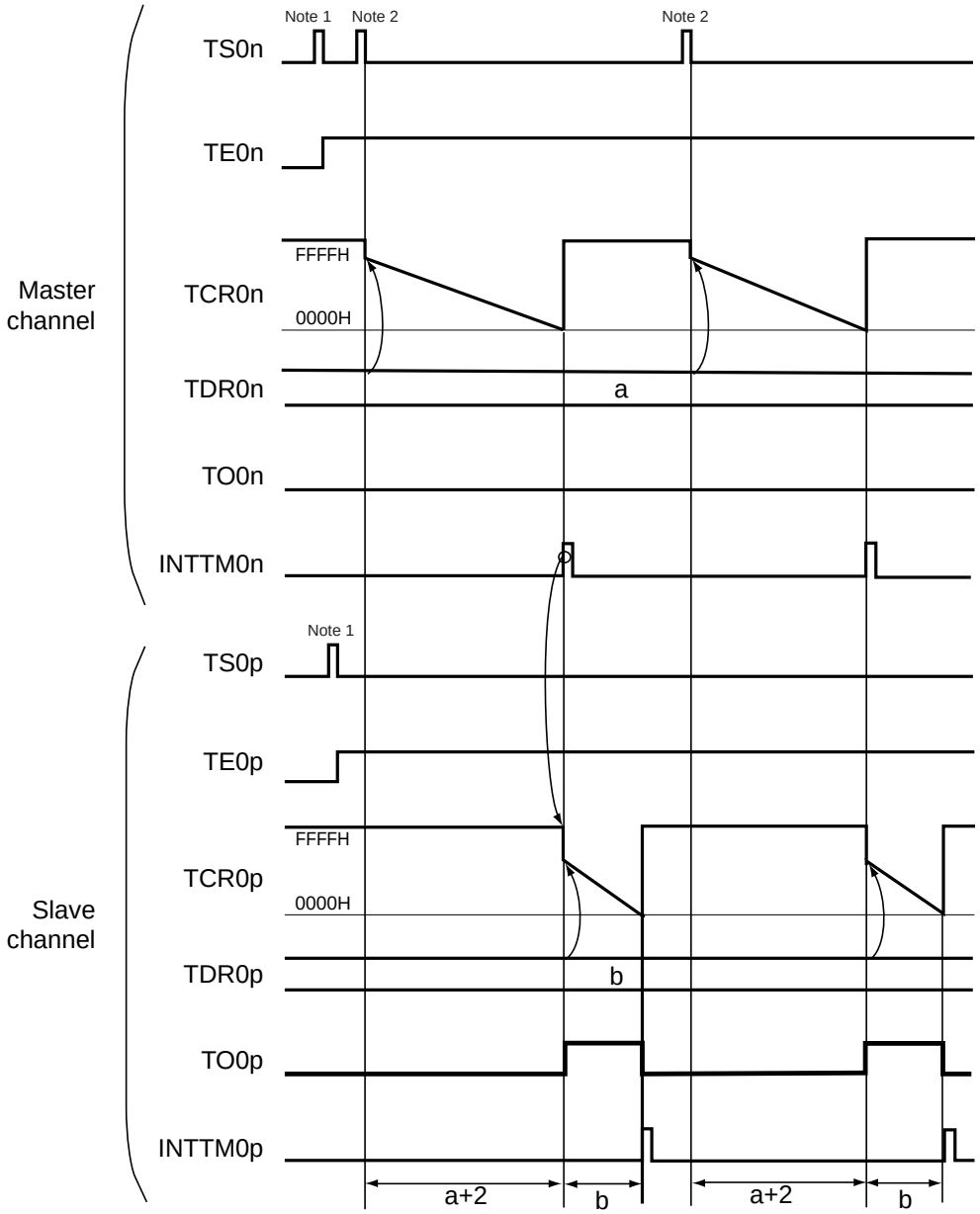
Remark n: Master channel number (n = 0, 2, 4)
p: Slave channel number (p = 3, 5, 6, where n < p ≤ 6)

Figure 6-60. Block Diagram of Operation as One-Shot Pulse Output Function



Remark n: Master channel number (n = 0, 2, 4)
p: Slave channel number (n < p ≤ 6)

Figure 6-61. Example of Basic Timing of Operation as One-Shot Pulse Output Function



Notes 1. If TS0n of the master channel and TS0p of the slave channel are set to “1” at the same time when TE0n and TE0p are both “0”, TE0n and TE0p are set to “1”.

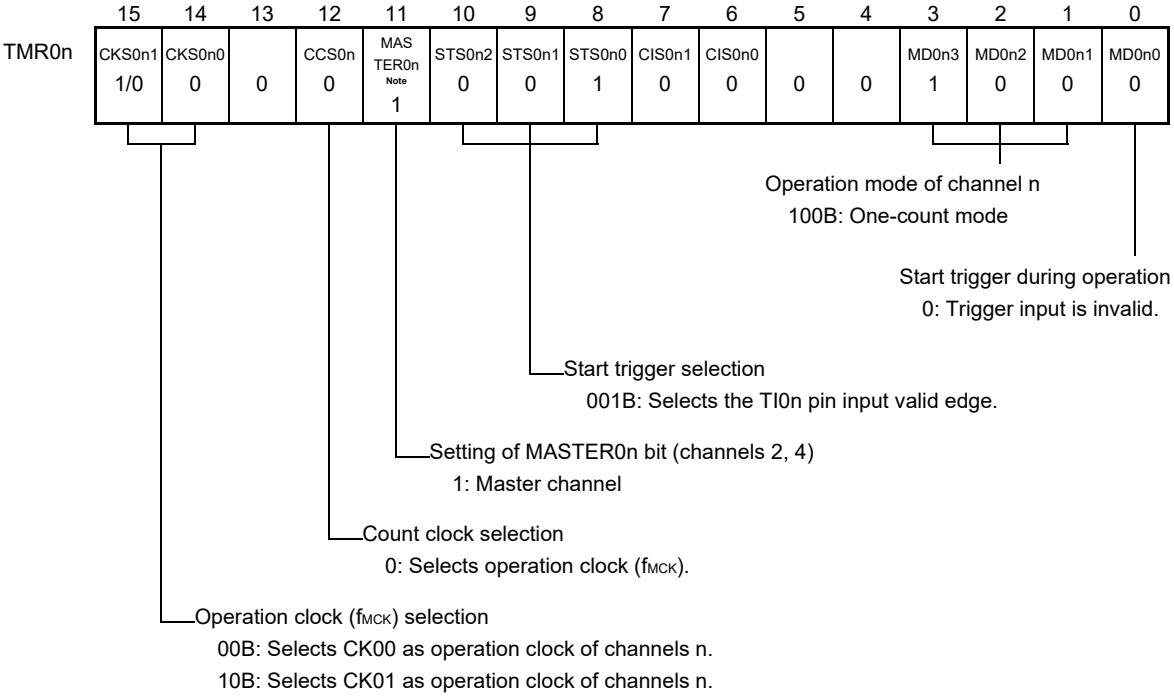
2. If TS0n of the master channel is set to “1” when TE0n and TE0p are both “1”, the one-shot pulse operation starts. Do not to set TS0p of the slave channel to “1” in this event.

Remarks 1. n: Master channel number (n = 0, 2, 4)
p: Slave channel number (p = 3, 5, 6, where n < p ≤ 6)

2. TS0n, TS0p: Bit n, p of timer channel start register 0 (TS0)
TE0n, TE0p: Bit n, p of timer channel enable status register 0 (TE0)
TCR0n, TCR0p: Timer/counter registers 0n, 0p (TCR0n, TCR0p)
TDR0n, TDR0p: Timer data registers 0n, 0p (TDR0n, TDR0p)
TO0n, TO0p: TO0n and TO0p pins output signal

Figure 6-62. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Master Channel)

(a) Timer mode register 0n (TMR0n)



(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n 0	0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n 0	0: Stops the TO0n output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when TOM0n = 0 (master channel output mode).

(e) Timer output mode register 0 (TOM0)

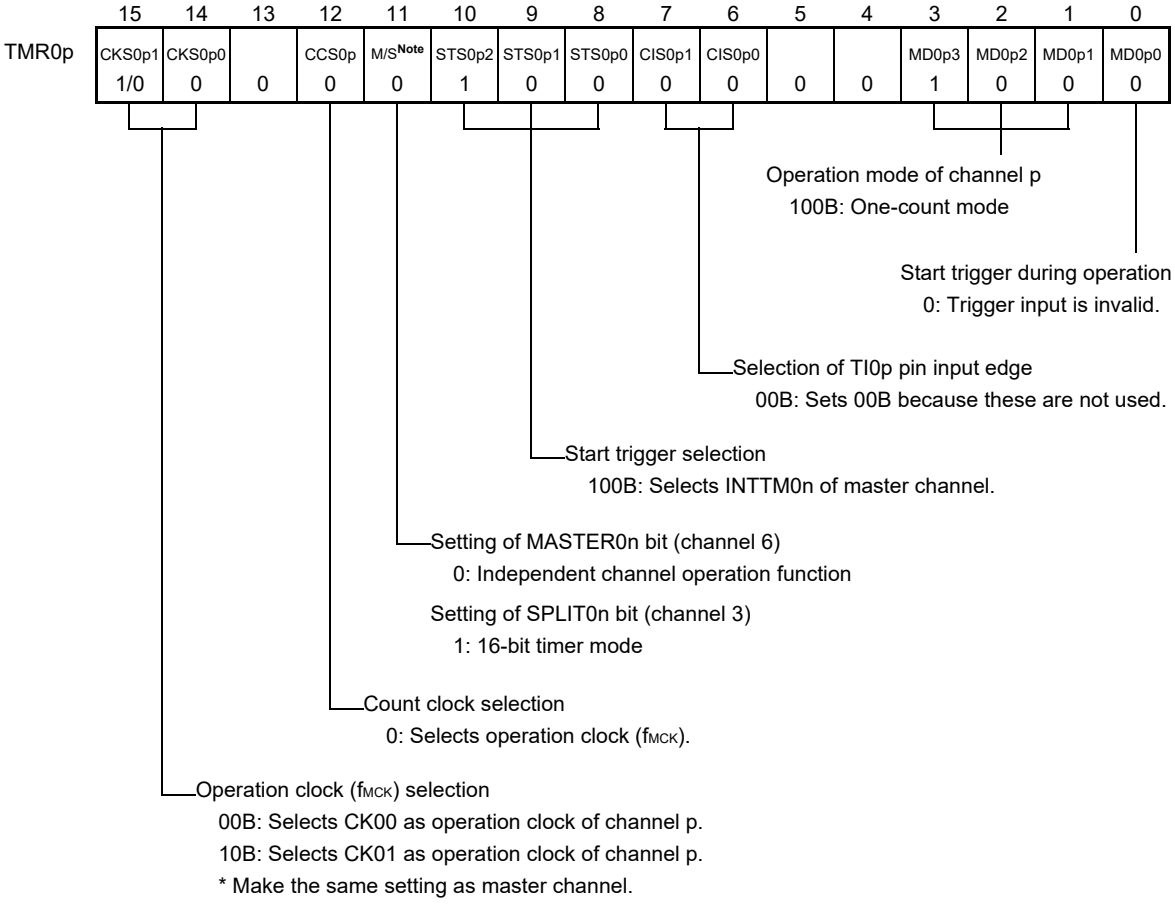
	Bit n	
TOM0	TOM0n 0	0: Sets master channel output mode.

Note TMR02, TMR04: MASTER0n = 1
TMR00: Fixed to 0

Remark n: Master channel number (n = 0, 2, 4)

Figure 6-63. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Slave Channel)

(a) Timer mode register 0p (TMR0p)



(b) Timer output register 0 (TO0)

	Bit p	
TO0	TO0p 1/0	0: Outputs 0 from TO0p. 1: Outputs 1 from TO0p.

(c) Timer output enable register 0 (TOE0)

	Bit p	
TOE0	TOE0p 1/0	0: Stops the TO0p output operation by counting operation. 1: Enables the TO0p output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit p	
TOL0	TOL0p 1/0	0: Positive logic output (active-high) 1: Inverted output (active-low)

(e) Timer output mode register 0 (TOM0)

	Bit p	
TOM0	TOM0p 1	1: Sets the slave channel output mode.

Note TMR02, TMR04, TMR06: MASTER0p bit
TMR01, TMR03: SPLIT0p bit
TMR05: Fixed to 0

Remark n: Master channel number (n = 0, 2, 4)
p: Slave channel number (p = 3, 5, 6, where n < p ≤ 6)

Figure 6-64. Operation Procedure of One-Shot Pulse Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable registers 0 (PER0) to 1. →	Power-on status. 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 and CK01.	
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, 0p (TMR0n, TMR0p) of two channels to be used (determines operation mode of channels). An output delay is set to timer data register 0n (TDR0n) of the master channel, and a pulse width is set to the TDR0p register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOM0p bit of timer output mode register 0 (TOM0) is set to 1 (slave channel output mode). Sets the TOL0p bit. Sets the TO0p bit and determines default level of the TO0p output. →	The TO0p pin goes into Hi-Z output state. The TO0p default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOE0p bit to 1 and enables operation of TO0p. →	TO0p does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO0p pin outputs the TO0p set level.

Remark n: Master channel number (n = 0, 2, 4)
p: Slave channel number (p = 3, 5, 6, where n < p ≤ 6)

Figure 6-64. Operation Procedure of One-Shot Pulse Output Function (2/2)

	Software Operation	Hardware Status
Operation start	Sets the TOE0p bit (slave) to 1 (only when operation is resumed). The TS0n (master) and TS0p (slave) bits of timer channel start register 0 (TS0) are set to 1 at the same time. → The TS0n and TS0p bits automatically return to 0 because they are trigger bits.	→ The TE0n and TE0p bits are set to 1 and the master channel enters the start trigger detection (the valid edge of the TI0n pin input is detected or the TS0n bit of the master channel is set to 1) wait status. Counter stops operating.
	Count operation of the master channel is started by start trigger detection of the master channel. • Detects the TI0n pin input valid edge • Sets the TS0n bit of the master channel to 1 by software [Note]. **Note** Do not set the TS0n bit of the slave channel to 1.	Master channel starts counting.
During operation	Set values of only the CIS0n1 and CIS0n0 bits of the TMR0n register can be changed. Set values of the TMR0p, TDR0n, TDR0p registers, TOM0n, TOM0p, TOL0n, and TOL0p bits cannot be changed. The TCR0n and TCR0p registers can always be read. The TSR0n and TSR0p registers are not used. Set values of the TO0 and TOE0 registers by slave channel can be changed.	Master channel loads the value of the TDR0n register to timer/counter register 0n (TCR0n) by the start trigger detection (the valid edge of the TI0n pin input is detected or the TS0n bit of the master channel is set to 1), and the counter starts counting down. When the count value reaches TCR0n = 0000H, the INTTM0n output is generated, and the counter stops until the next start trigger detection. The slave channel, triggered by INTTM0n of the master channel, loads the value of the TDR0p register to the TCR0p register, and the counter starts counting down. The output level of TO0p becomes active one count clock after generation of INTTM0n from the master channel. It becomes inactive when TCR0p = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TT0n (master) and TT0p (slave) bits are set to 1 at the same time. → The TT0n and TT0p bits automatically return to 0 because they are trigger bits.	→ TE0n, TE0p = 0, and count operation stops. The TCR0n and TCR0p registers hold count value and stop. The TO0p output is not initialized but holds current status.
	The TOE0p bit of slave channel is cleared to 0 and value is set to the TO0p bit. →	The TO0p pin outputs the TO0p set level.
TAU stop	To hold the TO0p pin output level Clears the TO0p bit to 0 after the value to be held is set to the port register. →	The TO0p pin output level is held by port function.
	When holding the TO0p pin output level is not necessary Switches the port mode register to input mode. →	The TO0p pin output level goes into Hi-Z output state.
	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0p bit is cleared to 0 and the TO0p pin is set to port mode.)

Operation is resumed.

Remark n: Master channel number (n = 0, 2, 4)
p: Slave channel number (p = 3, 5, 6, where n < p ≤ 6)

6.9.2 Operation as PWM function

Two channels can be used as a set to generate a pulse of any period and duty factor.

The period and duty factor of the output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDR0n (master) + 1} × Count clock period
Duty factor [%] = {Set value of TDR0p (slave)}/{Set value of TDR0n (master) + 1} × 100
0% output: Set value of TDR0p (slave) = 0000H
100% output: Set value of TDR0p (slave) ≥ {Set value of TDR0n (master) + 1}

Remark The duty factor exceeds 100% if the set value of TDR0p (slave) > (set value of TDR0n (master) + 1), it summarizes to 100% output.

The master channel operates in the interval timer mode. If the channel start trigger bit (TS0n) of timer channel start register 0 (TS0) is set to 1, an interrupt (INTTM0n) is output, the value set to timer data register 0n (TDR0n) is loaded to timer/counter register 0n (TCR0n), and the counter counts down in synchronization with the count clock. When the counter reaches 0000H, INTTM0n is output, the value of the TDR0n register is loaded again to the TCR0n register, and the counter counts down. This operation is repeated until the channel stop trigger bit (TT0n) of timer channel stop register 0 (TT0) 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 (TO0p) cycle.

The slave channel operates in one-count mode. By using INTTM0n from the master channel as a start trigger, the TCR0p register loads the value of the TDR0p register and the counter counts down to 0000H. When the counter reaches 0000H, it outputs INTTM0p and waits until the next start trigger (INTTM0n from the master channel) is generated.

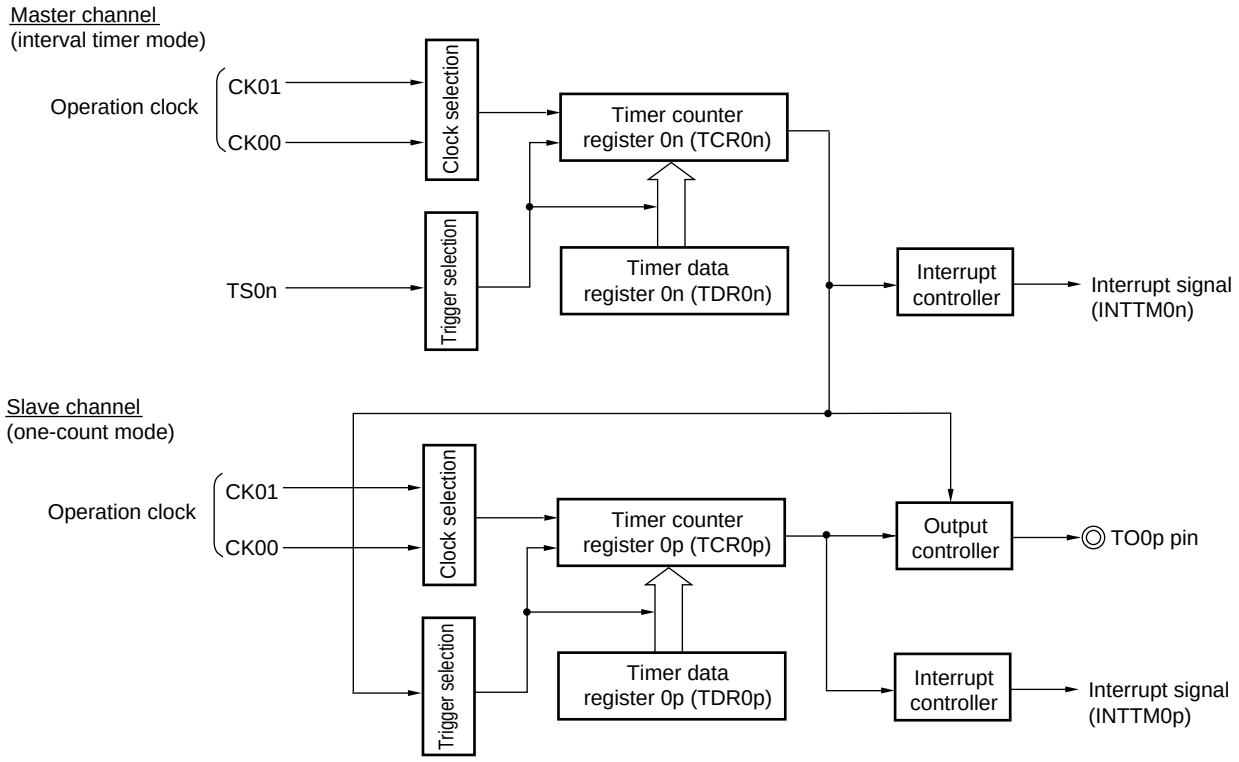
If two channels are used to output a PWM waveform, the period until the slave channel counts down to 0000H is the PWM output (TO0p) duty.

PWM output (TO0p) goes to the active level one clock after the master channel generates INTTM0n and goes to the inactive level when the TCR0p register of the slave channel becomes 0000H.

Caution To rewrite both timer data register 0n (TDR0n) of the master channel and the TDR0p register of the slave channel, a write access is necessary two times. The timing at which the values of the TDR0n and TDR0p registers are loaded to the TCR0n and TCR0p registers is upon occurrence of INTTM0n of the master channel. Thus, when rewriting is performed split before and after occurrence of INTTM0n of the master channel, the TO0p pin cannot output the expected waveform. To rewrite both the TDR0n register of the master and the TDR0p register of the slave, therefore, be sure to rewrite both the registers immediately after INTTM0n is generated from the master channel.

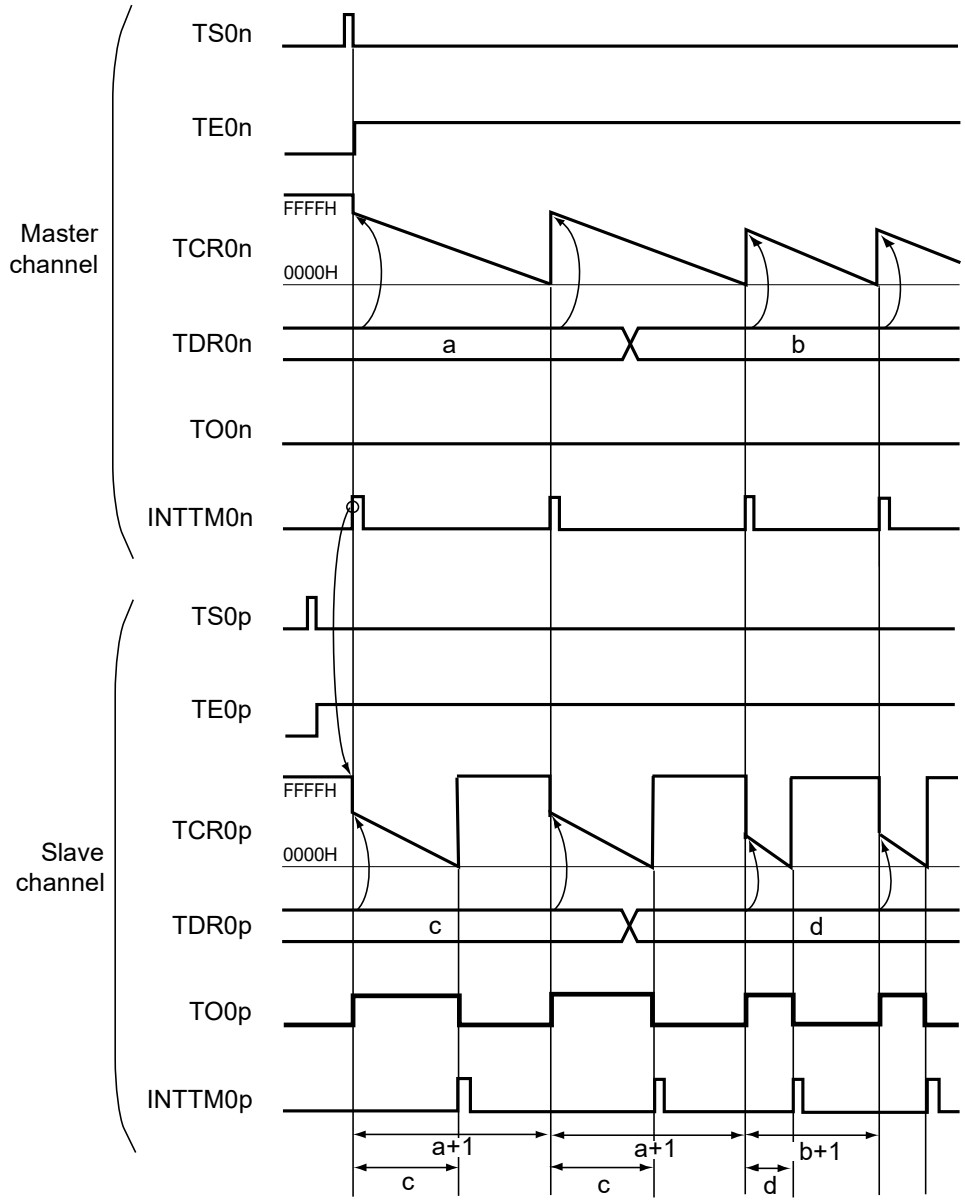
Remark n: Master channel number (n = 0, 2, 4)
p: Slave channel number (p = 3, 5, 6, where n < p ≤ 6)

Figure 6-65. Block Diagram of Operation as PWM Function



Remark n: Master channel number (n = 0, 2, 4)
p: Slave channel number (p = 3, 5, 6, where n < p ≤ 6)

Figure 6-66. Example of Basic Timing of Operation as PWM Function

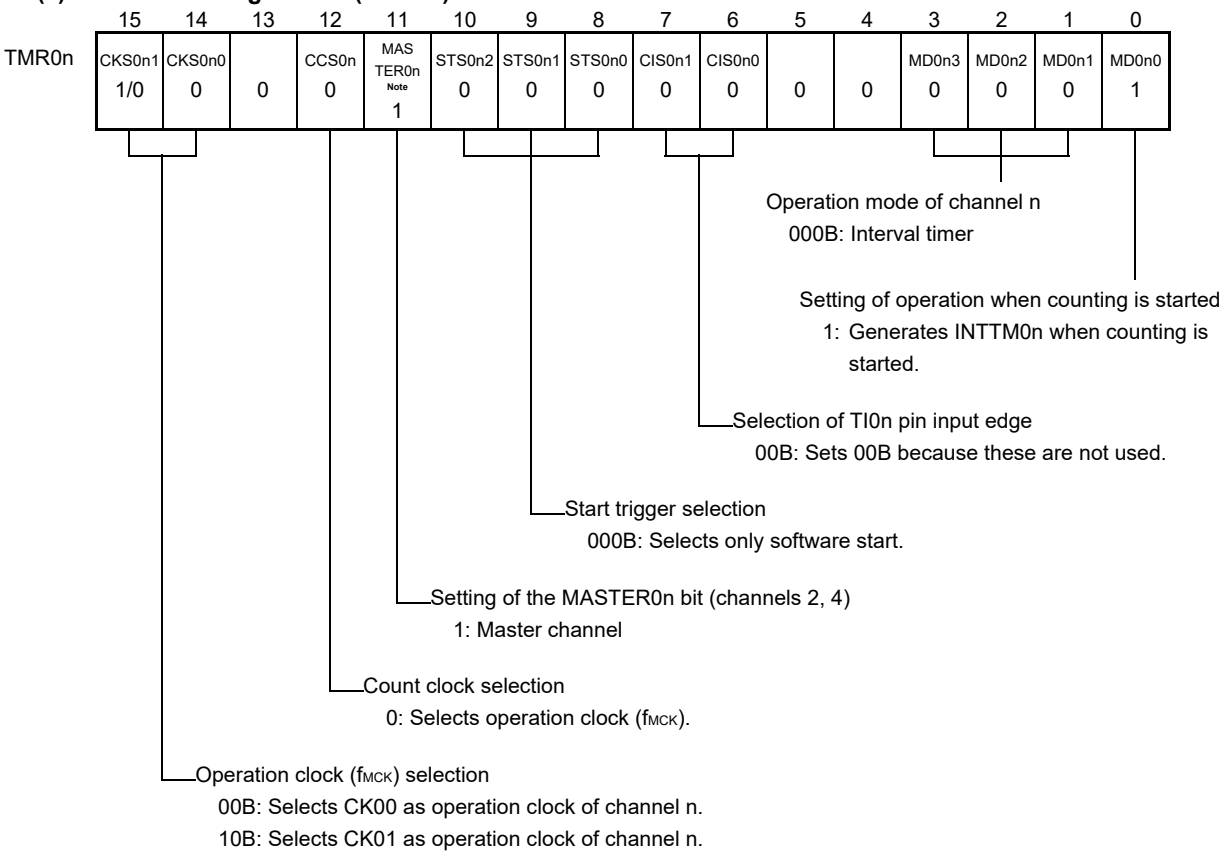


Remarks 1. n: Maste channel number (n = 0, 2, 4)
p: Slave channel number (p = 3, 5, 6, where n < p ≤ 6)

2. TS0n, TS0p: Bit n, p of timer channel start register 0 (TS0)
TE0n, TE0p: Bit n, p of timer channel enable status register 0 (TE0)
TCR0n, TCR0p: Timer/counter registers 0n, 0p (TCR0n, TCR0p)
TDR0n, TDR0p: Timer data registers 0n, 0p (TDR0n, TDR0p)
TO0n, TO0p: TO0n and TO0p pins output signal

Figure 6-67. Example of Set Contents of Registers When PWM Function (Master Channel) Is Used

(a) Timer mode register 0n (TMR0n)



(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n 0	0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n 0	0: Stops the TO0n output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when TOM0n = 0 (master channel output mode).

(e) Timer output mode register 0 (TOM0)

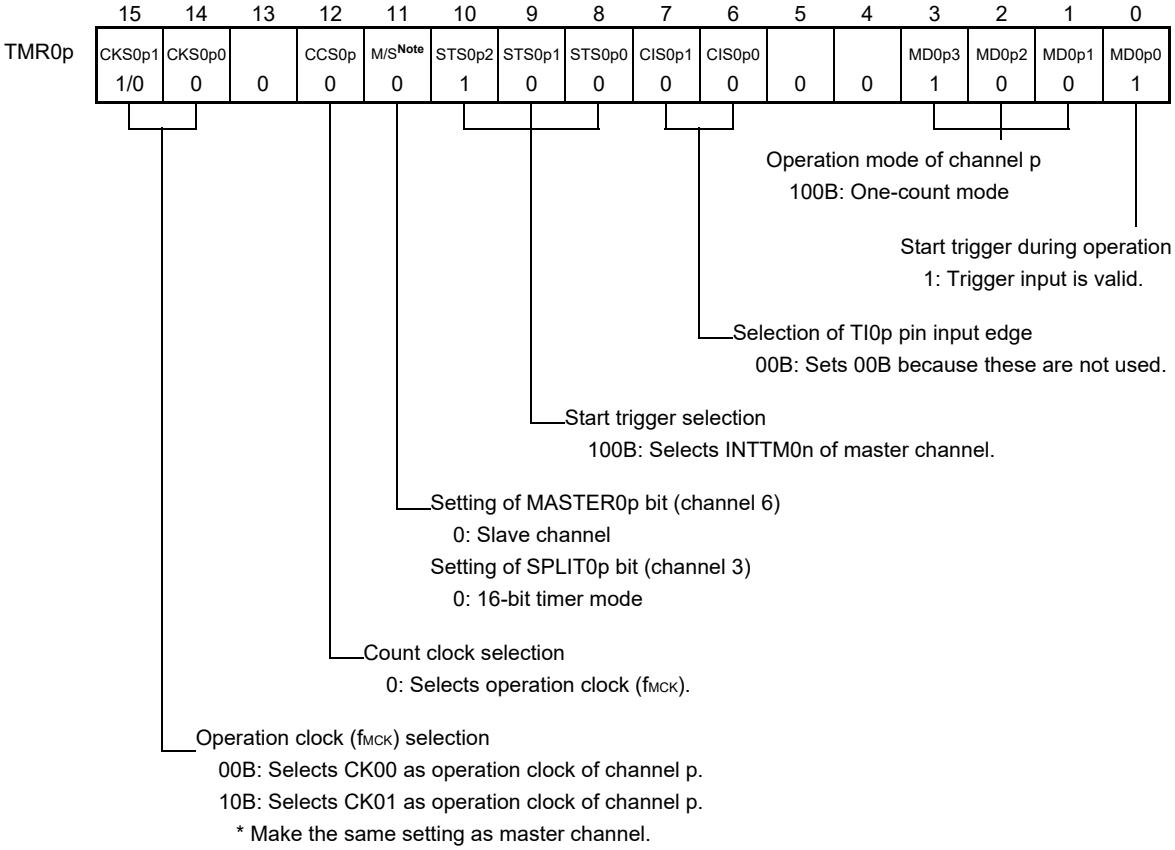
	Bit n	
TOM0	TOM0n 0	0: Sets master channel output mode.

Note TMR02, TMR04: MASTER0n = 1
TMR00: Fixed to 0

Remark n: Master channel number (n = 0, 2, 4)

Figure 6-68. Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used

(a) Timer mode register 0p (TMR0p)



(b) Timer output register 0 (TO0)

	Bit p	
TO0	TO0p 1/0	0: Outputs 0 from TO0p. 1: Outputs 1 from TO0p.

(c) Timer output enable register 0 (TOE0)

	Bit p	
TOE0	TOE0p 1/0	0: Stops the TO0p output operation by counting operation. 1: Enables the TO0p output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit p	
TOL0	TOL0p 1/0	0: Positive logic output (active-high) 1: Inverted output (active-low)

(e) Timer output mode register 0 (TOM0)

	Bit p	
TOM0	TOM0p 1	1: Sets the slave channel output mode.

Note TMR05: Fixed to 0
TMR03: SPLIT0p bit

Remark n: Master channel number (n = 0, 2, 4)
p: Slave channel number (p = 3, 5, 6, where n < p ≤ 6)

Figure 6-69. Operation Procedure When PWM Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (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 status. 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 and CK01.	
Channel default setting	Sets timer mode registers 0n, 0p (TMR0n, TMR0p) of two channels to be used (determines operation mode of channels). An interval (period) value is set to timer data register 0n (TDR0n) of the master channel, and a duty factor is set to the TDR0p register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOM0p bit of timer output mode register 0 (TOM0) is set to 1 (slave channel output mode). Sets the TOL0p bit. Sets the TO0p bit and determines default level of the TO0p output. →	The TO0p pin goes into Hi-Z output state. The TO0p default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOE0p bit to 1 and enables operation of TO0p. →	TO0p does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO0p pin outputs the TO0p set level.

Remark n: Master channel number (n = 0, 2, 4)
p: Slave channel number (p = 3, 5, 6, where n < p ≤ 6)

Figure 6-69. Operation Procedure When PWM Function Is Used (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets the TOE0p bit (slave) to 1 (only when operation is resumed). The TS0n (master) and TS0p (slave) bits of timer channel start register 0 (TS0) are set to 1 at the same time. The TS0n and TS0p bits automatically return to 0 because they are trigger bits.	TE0n = 1, TE0p = 1 → When the master channel starts counting, INTTM0n is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	Set values of the TMR0n and TMR0p registers, TOM0n, TOM0p, TOL0n, and TOL0p bits cannot be changed. Set values of the TDR0n and TDR0p registers can be changed after INTTM0n of the master channel is generated. The TCR0n and TCR0p registers can always be read. The TSR0n and TSR0p registers are not used. Set values of the TO0 and TOE0 registers can be changed.	The counter of the master channel loads the TDR0n register value to timer/counter register 0n (TCR0n), and counts down. When the count value reaches TCR0n = 0000H, INTTM0n output is generated. At the same time, the value of the TDR0n register is loaded to the TCR0n register, and the counter starts counting down again. At the slave channel, the value of the TDR0p register is loaded to the TCR0p register, triggered by INTTM0n of the master channel, and the counter starts counting down. The output level of TO0p becomes active one count clock after generation of the INTTM0n output from the master channel. It becomes inactive when TCR0p = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TT0n (master) and TT0p (slave) bits are set to 1 at the same time. The TT0n and TT0p bits automatically return to 0 because they are trigger bits.	→ TE0n, TE0p = 0, and count operation stops. The TCR0n and TCR0p registers hold count value and stop. The TO0p output is not initialized but holds current status.
	The TOE0p bit of slave channel is cleared to 0 and value is set to the TO0p bit.	→ The TO0p pin outputs the TO0p set level.
TAU stop	To hold the TO0p pin output level Clears the TO0p bit to 0 after the value to be held is set to the port register.	→ The TO0p pin output level is held by port function.
	When holding the TO0p pin output level is not necessary Switches the port mode register to input mode.	→ The TO0p pin output level goes into Hi-Z output state.
	The TAU0EN bit of the PER0 register is cleared to 0.	→ Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0p bit is cleared to 0 and the TO0p pin is set to port mode.)

Remark n: Master channel number (n = 0, 2, 4)
p: Slave channel number (p = 3, 5, 6, where n < p ≤ 6)

6.9.3 Operation as multiple PWM output function

By extending the PWM function and using multiple slave channels, many PWM waveforms with different duty values can be output.

For example, when using two slave channels, the period and duty factor of an output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDR0n (master) + 1} × Count clock period
Duty factor 1 [%] = {Set value of TDR0p (slave 1)}/{Set value of TDR0n (master) + 1} × 100
Duty factor 2 [%] = {Set value of TDR0q (slave 2)}/{Set value of TDR0n (master) + 1} × 100

Remark Although the duty factor exceeds 100% if the set value of TDR0p (slave 1) > {set value of TDR0n (master) + 1} or if the {set value of TDR0q (slave 2)} > {set value of TDR0n (master) + 1}, it is summarized into 100% output.

Timer/counter register 0n (TCR0n) of the master channel operates in the interval timer mode and counts the periods.

The TCR0p register of the slave channel 1 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TO0p pin. The TCR0p register loads the value of timer data register 0p (TDR0p), using INTTM0n of the master channel as a start trigger, and starts counting down. When TCR0p = 0000H, TCR0p outputs INTTM0p and stops counting until the next start trigger (INTTM0n of the master channel) has been input. The output level of TO0p becomes active one count clock after generation of INTTM0n from the master channel, and inactive when TCR0p = 0000H.

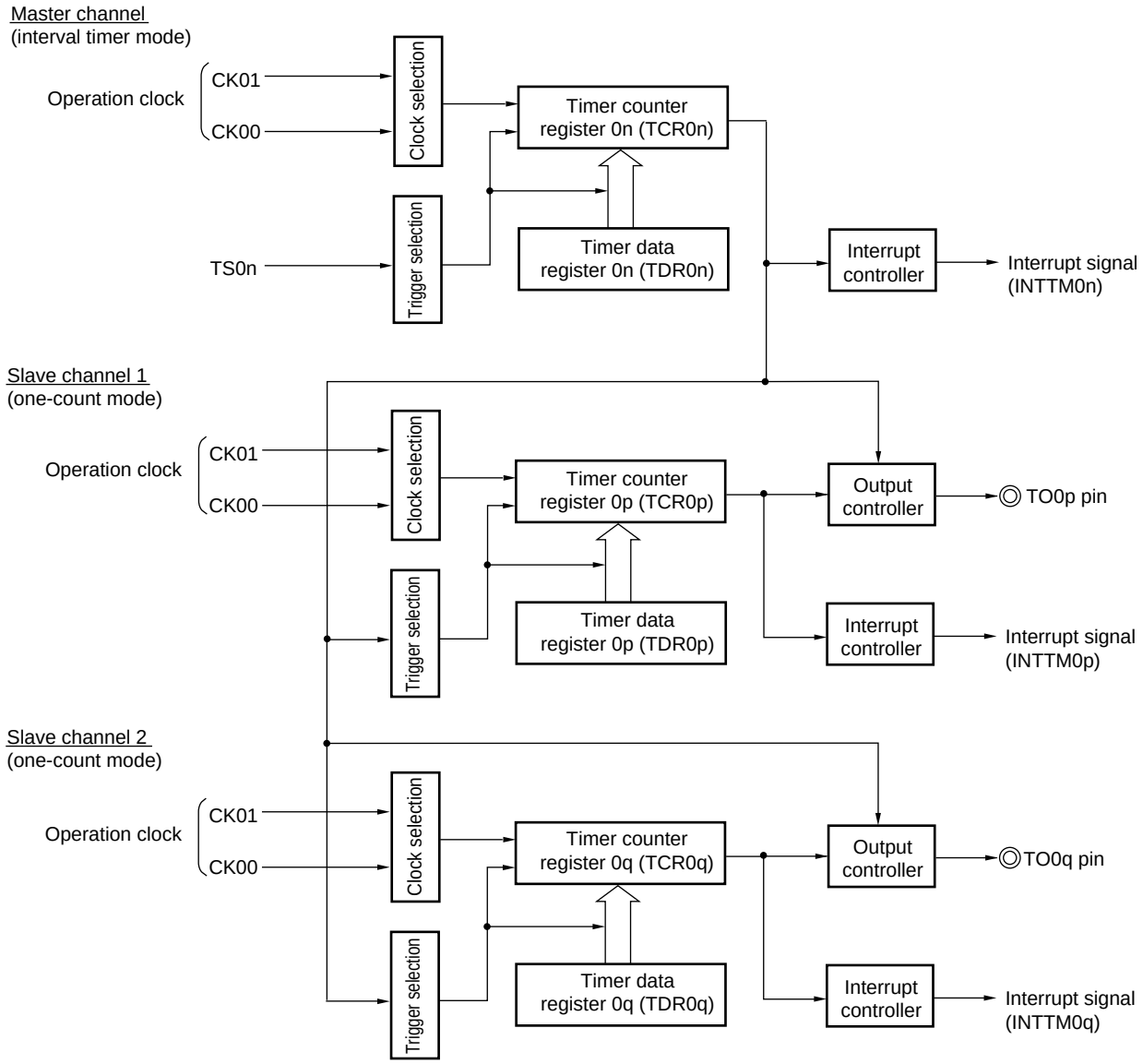
In the same way as the TCR0p register of the slave channel 1, the TCR0q register of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TO0q pin. The TCR0q register loads the value of the TDR0q register, using INTTM0n of the master channel as a start trigger, and starts counting down. When TCR0q = 0000H, the TCR0q register outputs INTTM0q and stops counting until the next start trigger (INTTM0n of the master channel) has been input. The output level of TO0q becomes active one count clock after generation of INTTM0n from the master channel, and inactive when TCR0q = 0000H.

When channel 0 is used as the master channel as above, up to seven types of PWM signals can be output at the same time.

Caution To rewrite both timer data register 0n (TDR0n) of the master channel and the TDR0p register of the slave channel 1, write access is necessary at least twice. Since the values of the TDR0n and TDR0p registers are loaded to the TCR0n and TCR0p registers after INTTM0n is generated from the master channel, if rewriting is performed separately before and after generation of INTTM0n from the master channel, the TO0p pin cannot output the expected waveform. To rewrite both the TDR0n register of the master and the TDR0p register of the slave, be sure to rewrite both the registers immediately after INTTM0n is generated from the master channel (This applies also to the TDR0q register of the slave channel 2).

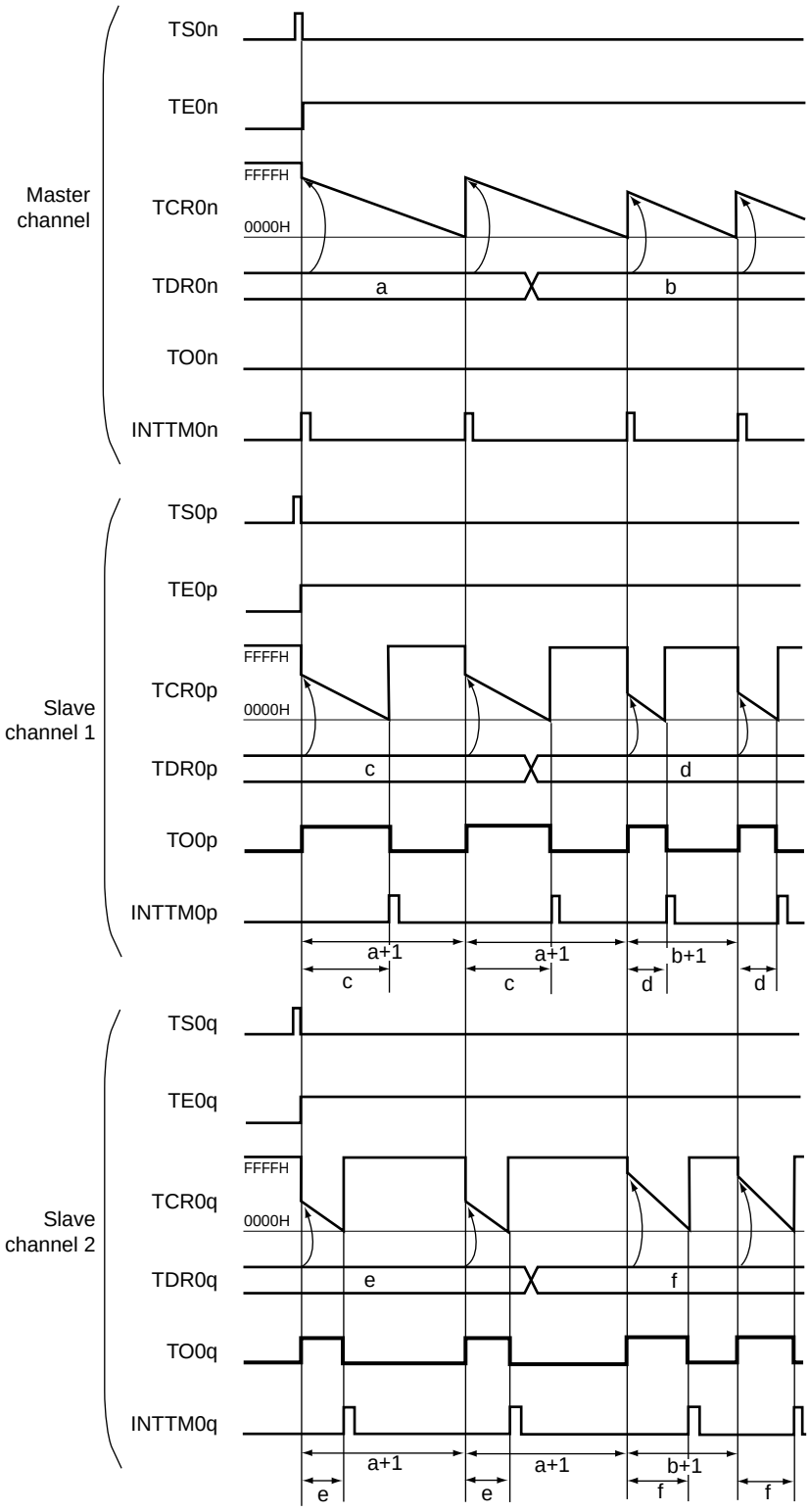
Remark n: Master channel number (n = 0, 2, 4)
p: Slave channel number
q: Slave channel number (p, q = 3, 5, 6, where n < p < q ≤ 6, and p and q are integers greater than n)

Figure 6-70. Block Diagram of Operation as Multiple PWM Output Function (Output Two Types of PWMs)



Remark n: Master channel number (n = 0, 2, 4)
p: Slave channel number
q: Slave channel number (p, q = 3, 5, 6, where $n < p < q \leq 6$, and p and q are integers greater than n)

Figure 6-71. Example of Basic Timing of Operation as Multiple PWM Output Function (Output Two Types of PWMs)



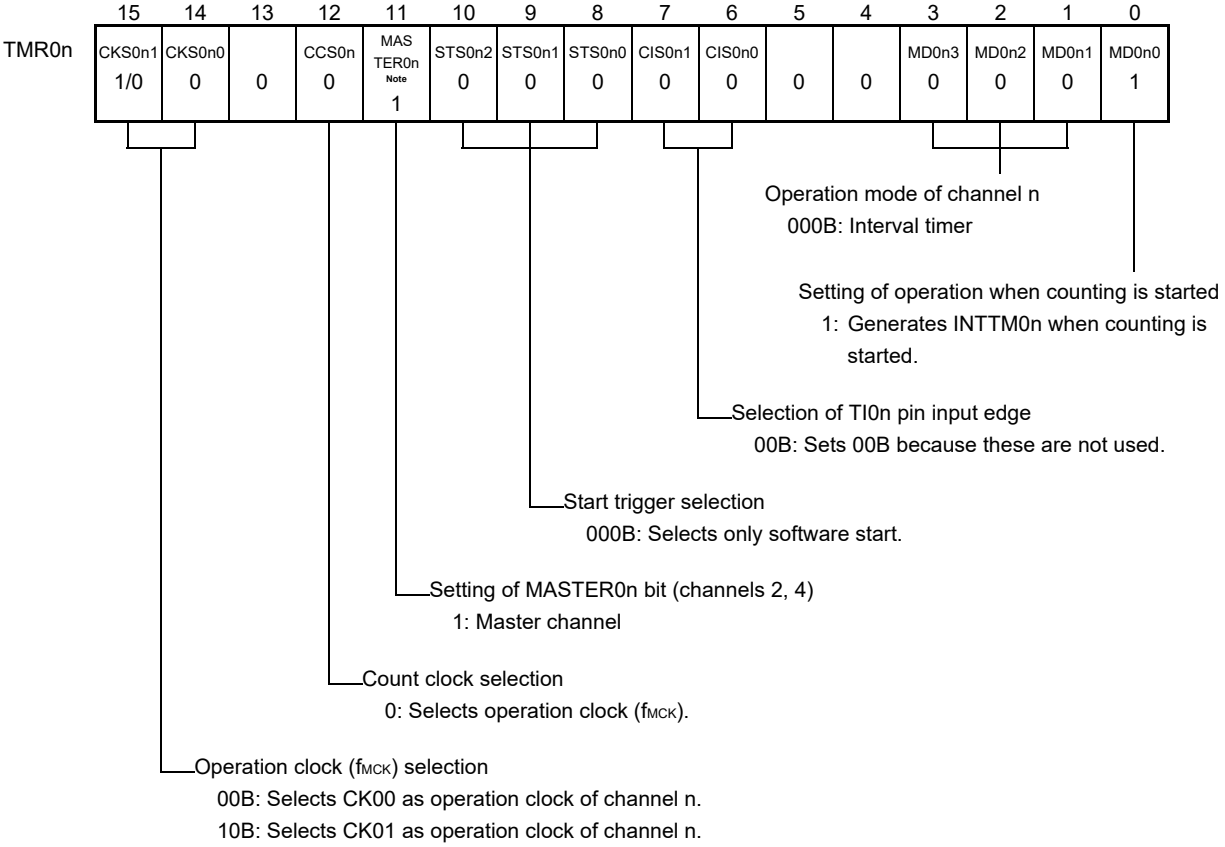
(**Remarks** are listed on the next page.)

Remarks 1. n: Maste channel number (n = 0, 2, 4)
p: Slave channel number
q: Slave channel number (p, q = 3, 5, 6, where n < p < q ≤ 6, and p and q are integers greater than n)

2. TS0n, TS0p, TS0q: Bit n, p, q of timer channel start register 0 (TS0)
TE0n, TE0p, TE0q: Bit n, p, q of timer channel enable status register 0 (TE0)
TCR0n, TCR0p, TCR0q: Timer/counter registers 0n, 0p, 0q (TCR0n, TCR0p, TCR0q)
TDR0n, TDR0p, TDR0q: Timer data registers 0n, 0p, 0q (TDR0n, TDR0p, TDR0q)
TO0n, TO0p, TO0q: TO0n, TO0p, and TO0q pins output signal

Figure 6-72. Example of Set Contents of Registers When Multiple PWM Output Function (Master Channel) Is Used

(a) Timer mode register 0n (TMR0n)



(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n 0	0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n 0	0: Stops the TO0n output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when TOM0n = 0 (master channel output mode).

(e) Timer output mode register 0 (TOM0)

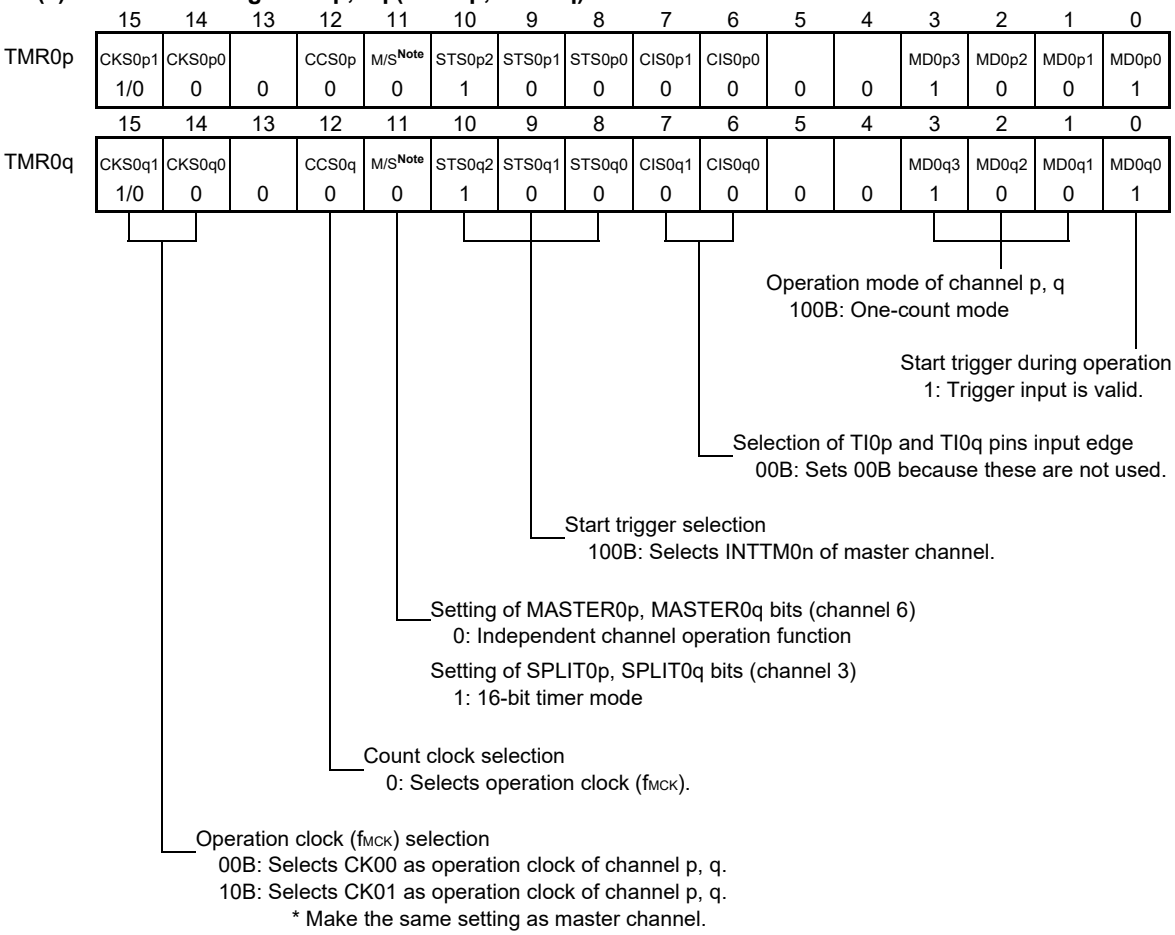
	Bit n	
TOM0	TOM0n 0	0: Sets master channel output mode.

Note TMR02, TMR04: MASTER0n = 1
TMR00: Fixed to 0

Remark n: Master channel number (n = 4)

Figure 6-73. Example of Set Contents of Registers When Multiple PWM Output Function (Slave Channel) Is Used (Output Two Types of PWMs) (1/2)

(a) Timer mode register 0p, 0q (TMR0p, TMR0q)



(b) Timer output register 0 (TO0)

	Bit q	Bit p	
TO0	TO0q 1/0	TO0p 1/0	0: Outputs 0 from TO0p or TO0q. 1: Outputs 1 from TO0p or TO0q.

(c) Timer output enable register 0 (TOE0)

	Bit q	Bit p	
TOE0	TOE0q 1/0	TOE0p 1/0	0: Stops the TO0p or TO0q output operation by counting operation. 1: Enables the TO0p or TO0q output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit q	Bit p	
TOL0	TOL0q 1/0	TOL0p 1/0	0: Positive logic output (active-high) 1: Inverted output (active-low)

Note TMR06: MASTER0p, MASTER0q bit
TMR03: SPLIT0p, SPLIT0q bit
TMR05: Fixed to 0

Remark n: Master channel number (n = 0, 2, 4)
p: Slave channel number
q: Slave channel number (p, q = 3, 5, 6, where $n < p < q \leq 6$, and p and q are integers greater than n)

Figure 6-73. Example of Set Contents of Registers
When Multiple PWM Output Function (Slave Channel) Is Used (Output Two Types of PWMs) (2/2)

(e) Timer output mode register 0 (TOM0)

	Bit q	Bit p	
TOM0	TOM0q	TOM0p	1: Sets the slave channel output mode.
	1	1	

Note TMR06: MASTER0p, MASTER0q bit
TMR03: SPLIT0p, SPLIT0q bit
TMR05: Fixed to 0

Remark p: Slave channel number
q: Slave channel number (p, q = 3, 5, 6, where $n < p < q \leq 6$, and p and q are integers greater than n)

Figure 6-74. Operation Procedure When Multiple PWM Output Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (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 status. 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 and CK01.	
Channel default setting	Sets timer mode registers 0n, 0p, 0q (TMR0n, TMR0p, TMR0q) of each channel to be used (determines operation mode of channels). An interval (period) value is set to timer data register 0n (TDR0n) of the master channel, and a duty factor is set to the TDR0p and TDR0q registers of the slave channels.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channels. The TOM0p and TOM0q bits of timer output mode register 0 (TOM0) are set to 1 (slave channel output mode). Sets the TOL0p and TOL0q bits. Sets the TO0p and TO0q bits and determines default level of the TO0p and TO0q outputs. →	The TO0p and TO0q pins go into Hi-Z output state. The TO0p and TO0q default setting levels are output when the port mode register is in output mode and the port register is 0.
	Sets the TOE0p and TOE0q bits to 1 and enables operation of TO0p and TO0q. →	TO0p and TO0q do not change because channels stop operating.
	Clears the port register and port mode register to 0. →	The TO0p and TO0q pins output the TO0p and TO0q set levels.

Remark n: Master channel number (n = 0, 2, 4)
p: Slave channel number
q: Slave channel number (p, q = 3, 5, 6, where $n < p < q \leq 6$, and p and q are integers greater than n)

Figure 6-74. Operation Procedure When Multiple PWM Output Function Is Used (2/2)

	Software Operation	Hardware Status
Operation start	(Sets the TOE0p and TOE0q (slave) bits to 1 only when resuming operation.) The TS0n bit (master), and TS0p and TS0q (slave) bits of timer channel start register 0 (TS0) are set to 1 at the same time. The TS0n, TS0p, and TS0q bits automatically return to 0 because they are trigger bits.	→ TE0n = 1, TE0p, TE0q = 1 When the master channel starts counting, INTTM0n is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	Set values of the TMR0n, TMR0p, TMR0q registers, TOM0n, TOM0p, TOM0q, TOL0n, TOL0p, and TOL0q bits cannot be changed. Set values of the TDR0n, TDR0p, and TDR0q registers can be changed after INTTM0n of the master channel is generated. The TCR0n, TCR0p, and TCR0q registers can always be read. The TSR0n, TSR0p, and TSR0q registers are not used. Set values of the TO0 and TOE0 registers can be changed.	The counter of the master channel loads the TDR0n register value to timer/counter register 0n (TCR0n) and counts down. When the count value reaches TCR0n = 0000H, INTTM0n output is generated. At the same time, the value of the TDR0n register is loaded to the TCR0n register, and the counter starts counting down again. At the slave channel 1, the values of the TDR0p register are transferred to the TCR0p register, triggered by INTTM0n of the master channel, and the counter starts counting down. The output levels of TO0p become active one count clock after generation of the INTTM0n output from the master channel. It becomes inactive when TCR0p = 0000H, and the counting operation is stopped. At the slave channel 2, the values of the TDR0q register are transferred to TCR0q register, triggered by INTTM0n of the master channel, and the counter starts counting down. The output levels of TO0q become active one count clock after generation of the INTTM0n output from the master channel. It becomes inactive when TCR0q = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TT0n bit (master), TT0p, and TT0q (slave) bits are set to 1 at the same time. The TT0n, TT0p, and TT0q bits automatically return to 0 because they are trigger bits.	→ TE0n, TE0p, TE0q = 0, and count operation stops. The TCR0n, TCR0p, and TCR0q registers hold count value and stop. The TO0p and TO0q output are not initialized but hold current status.
	The TOE0p and TOE0q bits of slave channels are cleared to 0 and value is set to the TO0p and TO0q bits.	→ The TO0p and TO0q pins output the TO0p and TO0q set levels.
TAU stop	To hold the TO0p and TO0q pin output levels Clears the TO0p and TO0q bits to 0 after the value to be held is set to the port register.	→ The TO0p and TO0q pin output levels are held by port function.
	When holding the TO0p and TO0q pin output levels are not necessary Switches the port mode register to input mode.	→ The TO0p and TO0q pin output levels go into Hi-Z output state.
	The TAU0EN bit of the PER0 register is cleared to 0.	→ Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0p and TO0q bits are cleared to 0 and the TO0p and TO0q pins are set to port mode.)

Operation is resumed.

Remark n: Master channel number (n = 0, 2, 4)
p: Slave channel number
q: Slave channel number (p, q = 3, 5, 6, where $n < p < q \leq 6$, and p and q are integers greater than n)

6.10 Cautions When Using Timer Array Unit

6.10.1 Cautions when using timer output

Depends on products, a pin is assigned a timer output and other alternate functions. In this case, outputs of the other alternate functions must be set in initial status.

For details, see **4.5 Register Settings When Using Alternate Function**.

CHAPTER 7 16-BIT TIMERS KB0, KB1, AND KB2

16-bit timers KB0, KB1 and KB2 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.

	20-pin	30-pin	38-pin
16-bit timer KB0	√	√	√
16-bit timer KB1	√	√	√
16-bit timer KB2	–	√	√

Cautions 1. Most of the following descriptions in this chapter use the 38-pin products as an example.

2. 16-bit timer KB2 is generated as an external pin for the 20 pins products.

7.1 Functions of 16-bit Timers KB0, KB1, and KB2

16-bit timers KB0, KB1, and KB2 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, or INTP21, PWM output can be stopped urgently.

16-bit timers KB0, KB1, and KB2 are provided with the following functions.

(1) PWM output

- 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 high level and low level, respectively.

(2) A/D conversion start timing signal output

The A/D conversion start timing signal can be output by using compare register TKBTGCRn. The 16-bit timer KBn and A/D conversion start timing can be synchronized with this function.

(3) Simultaneous Start / Stop Mode

By setting 16-bit timer KB0 as the master and 16-bit timers KB1 and KB2 as slaves, slave 16-bit timers KB1 and KB2 can be started/stopped at the same time synchronized with the count start/stop timing of timer KB0.

(4) Synchronous start/clear mode

By setting 16-bit timer KB0 as the master and 16-bit timers KB1 and KB2 as slaves, the counting cycles of the master and slave timers can be synchronized. In this mode, a complementary PWM output can be generated, for example.

(5) PWM output gating function (by interlocking with 16-bit timer KC0)

Up to 6 timer KBn outputs can be gate-controlled by using the output of 16-bit timer KC0 (the TKCO00 to TKCO05 outputs).

(6) Timer restart function (by interlocking with the comparator and INTP)

Timer output can be restarted directly (not via the CPU) when a trigger source occurs (comparator 1 to 3 output, INTP20, INTP21). By using this function, critical conduction mode PFC control can be implemented, for example.

(7) Forced output stop function 1 (by interlocking with the comparator)

Timer output can be fixed to Hi-Z, high, or low level directly (not via the CPU) and asynchronously with the operation clock f_{KBKC} of the 16-bit timer KBn and KC0 circuit when a trigger source occurs (comparator 0 to 5 output). The forced output stop status is canceled synchronously with the operation clock f_{KBKC} of the 16-bit timer KBn and KC0 circuits by setting the stop trigger of forced output stop function 1.

(8) Forced output stop function 2 (by interlocking with the comparator and INTP)

Timer output can be fixed to high or low level directly (not via the CPU) and asynchronously with the operation clock f_{KBKC} of the 16-bit timer KBn and KC0 circuit when a trigger source occurs (comparator 0 to 5 output, INTP20). 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.

(9) 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 that is 16 times the count clock can be output as the average resolution of 16 timer KBn cycles.

(10) PWM output smooth start function

It is possible to make a soft-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.

(11) 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.

(12) Interleave 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 with Critical Conduction Mode.

Remark Critical Conduction Mode is a PFC control method that activates a switching FET by detecting zero level of inductor current.

7.2 Configuration of 16-bit Timers KB0, KB1, and KB2

16-bit timers KB0, KB1, and KB2 include the following hardware.

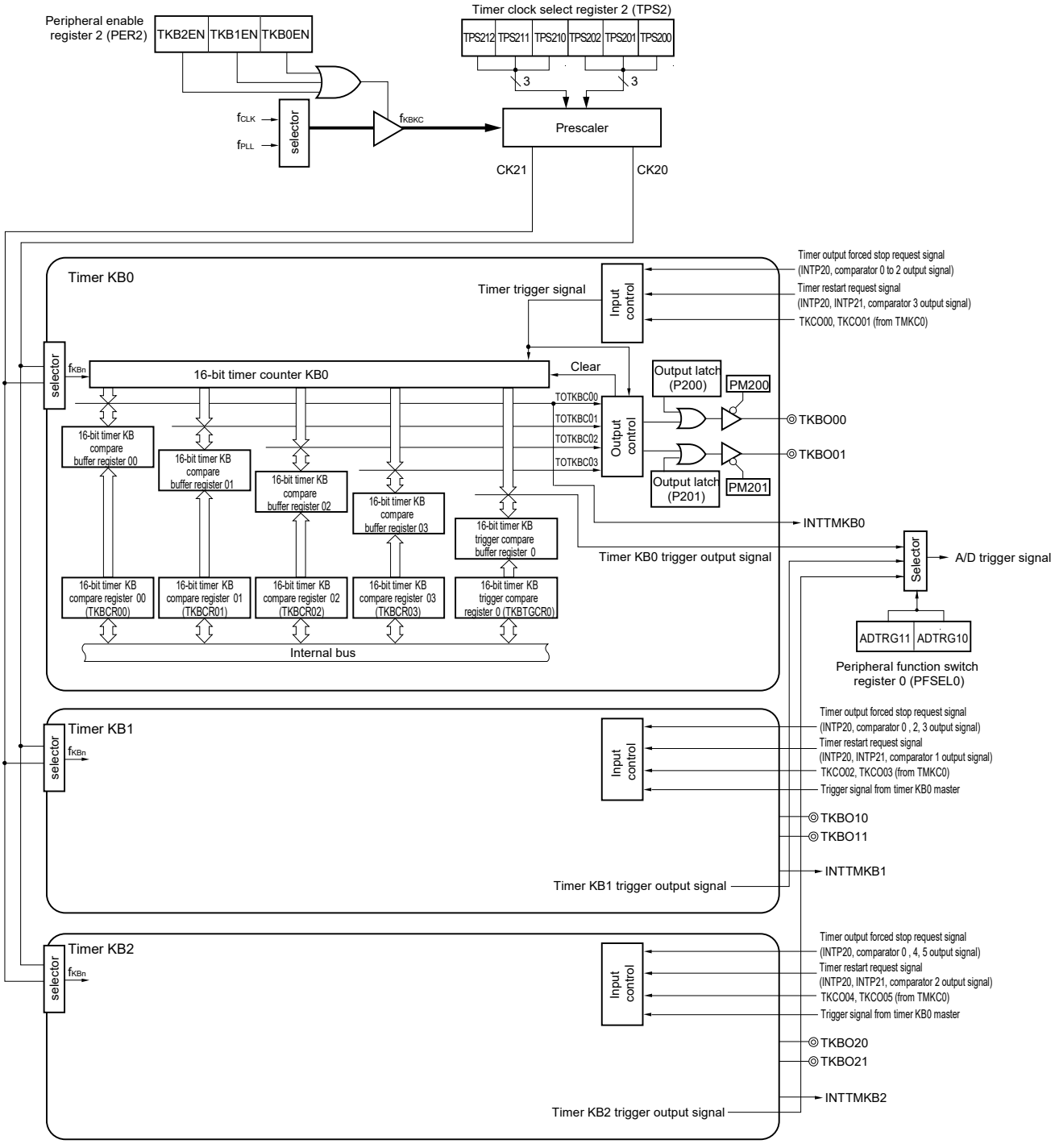
Table 7-1. Configuration of 16-bit Timers KB0, KB1, and KB2

Item	Configuration
Timer/counter	16-bit timer counter KBn (TKBCNTn)
Registers	16-bit timer KB compare registers n0 to n3 (TKBCRn0 to TKBCRn3) 16-bit timer KB trigger compare register n (TKBTGCRn)
Timer output	TKBOn0, TKBOn1
Control registers	Peripheral enable register 2 (PER2) Timer clock select register 2 (TPS2) 16-bit timer KB operation control register n0 (TKBCTLn0) 16-bit timer KB operation control register n1 (TKBCTLn1) 16-bit timer KB output control register n0 (TKBIOCn0) 16-bit timer KB output control register n1 (TKBIOCn1) 16-bit timer KB flag register n (TKBFLGn) 16-bit timer KB trigger register n (TKBTRGn) 16-bit timer KB flag clear trigger register n (TKBCLRn) 16-bit timer KB dithering count registers n0, n1 (TKBDNRn0, TKBDNRn1) 16-bit timer KB compare 1L & dithering count register n0 (TKBCRLDn0) 16-bit timer KB compare 3L & dithering count register n1 (TKBCRLDn1) 16-bit timer KB smooth start initial duty registers n0, n1 (TKBSIRn0, TKBSIRn1) 16-bit timer KB smooth start step width registers n0, n1 (TKBSSRn0, TKBSSRn1) 16-bit timer KB maximum frequency limit setting register n (TKBMFRn) Peripheral function switch register 0 (PFSEL0) Port mode register 20 (PM20) Port register 20 (P20)

Remark n = 0 to 2

Figure 7-1 shows a block diagram.

Figure 7-1. Block Diagram of 16-bit Timer KBn



Remarks 1. f_{KBKC}: Operation clock of whole 16-bit timer KBn and KC0 circuit

f_{KBn}: Count clock of 16-bit timer KBn

2. n = 0 to 2

7.2.1 16-bit timer KB compare registers n0 to n3 (TKBCRn0 to TKBCRn3)

TKBCRnm can be refreshed (writing the same value) and its value can be rewritten while the timer is counting (TKBCEn = 1). When the value of TKBCRnm is rewritten while the timer is operating, that value is latched, transferred to TKBCRnm at the following timing, and the value of TKBCRnm is changed.

- When starting count operation of counter (TKBCEn = 0)
- When a batch overwrite trigger (TKBRDTn = 1) or an external trigger (TKBTSEn = 1) occurs

This register can be read or written in 16-bit units.

Reset signal generation clears this register to 0000H.

Figure 7-2. Format of 16-bit Timer KB Compare Register nm (TKBCRnm)

Address: F0600H (TKBCR00), F0602H (TKBCR01), F0604H (TKBCR02), F0606H (TKBCR03),
F0640H (TKBCR10), F0642H (TKBCR11), F0644H (TKBCR12), F0646H (TKBCR13),
F0680H (TKBCR20), F0682H (TKBCR21), F0684H (TKBCR22), F0686H (TKBCR23)

After reset: 0000H R/W

	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

7.2.2 16-bit timer KB trigger compare register n (TKBTGCRn)

TKBTGCRn can be refreshed (writing the same value) and its value can be rewritten while the timer is counting (TKBCEn = 1). When the value of TKBTGCRn is rewritten while the timer is operating, that value is latched, transferred to TKBTGCRn at the following timing, and the value of TKBTGCRn is changed.

- When starting count operation of counter (TKBCEn = 0)
- When a batch overwrite trigger (TKBRDTn = 1) or an external trigger (TKBTSEn = 1) occurs

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.

Reset signal generation clears this register to 0000H.

Figure 7-3. Format of 16-bit Timer KB Trigger Compare Register n (TKBTGCRn)

Address: F0608H (TKBTGCR0), F0648H (TKBTGCR1), F0688H (TKBTGCR2) After reset: 0000H R/W

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

Remark n = 0 to 2

7.3 Registers Controlling 16-bit Timers KB0, KB1, and KB2

16-bit timers KB0, KB1, and KB2 are controlled by the following registers.

- Peripheral enable register 2 (PER2)
- Timer clock select register 2 (TPS2)
- 16-bit timer KB operation control registers n0, n1 (TKBCTLn0, TKBCTLn1)
- 16-bit timer KB output control registers n0, n1 (TKBIOCn0, TKBIOCn1)
- 16-bit timer KB flag register n (TKBFLGn)
- 16-bit timer KB trigger register n (TKBTRGn)
- 16-bit timer KB flag clear trigger register n (TKBCLRn)
- 16-bit timer KB dithering count registers n0, n1 (TKBDNRn0, TKBDNRn1)
- 16-bit timer KB compare 1L & dithering count register n0 (TKBCRLDn0)
- 16-bit timer KB compare 3L & dithering count register n1 (TKBCRLDn1)
- 16-bit timer KB smooth start initial duty registers n0, n1 (TKBSIRn0, TKBSIRn1)
- 16-bit timer KB smooth start step width registers n0, n1 (TKBSSRn0, TKBSSRn1)
- 16-bit timer KB maximum frequency limit setting register n (TKBMFRn)
- Peripheral function switch register 0 (PFSEL0)
- Port mode register 20 (PM20)
- Port register 20 (P20)

7.3.1 Peripheral enable register 2 (PER2)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When timers KB0, KB1, and KB2 are used, be sure to set bits 2 to 0 (TKB2EN to TKB0EN) of this register to 1.

The PER2 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-4. Format of Peripheral Enable Register 2 (PER2)

Address: F0509H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PER2	PGACMPEN	TKBPA2EN	TKBPA1EN	TKBPA0EN	TKC0EN	TKB2EN	TKB1EN	TKB0EN

TKBnEN	Control of timer KBn input clock
0	Stops supply of input clock. • SFR used by timer KBn cannot be written. • Timer KBn is in the reset status.
1	Supplies input clock. • SFR used by timer KBn can be read/written.

Caution When setting timer KBn, be sure to set the TKBnEN bit to 1 first. If TKBnEN = 0, writing to a control register of timer KBn is ignored, and all read values are default values (except for timer clock select register 2 (TPS2), peripheral function switch register 0 (PFSEL0), port mode register 20 (PM20), and port register 20 (P20)).

7.3.2 Timer clock select register 2 (TPS2)

The TPS2 register is a 16-bit register that is used to select two types of operation clocks (CK20, CK21) that are commonly supplied to timers KB0, KB1, KB2, and KC0 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 TPS200 to TPS202 bits can be rewritten (n = 0 to 2):

All timers for which CK20 is selected as the operation clock (TKBCKSn = 0, TKCCKS0 = 0) are stopped (TKBCEn = 0, TKCCE0 = 0).

If the TPS210 to TPS212 bits can be rewritten (n = 0 to 2):

All timers for which CK21 is selected as the operation clock (TKBCKSn = 1, TKCCKS0 = 1) are stopped (TKBCEn = 0, TKCCE0 = 0).

The TPS2 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-5. Format of Timer Clock Select Register 2 (TPS2)

Address: F05C5H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TPS2	0	TPS212	TPS211	TPS210	0	TPS202	TPS201	TPS200

TPS2k2	TPS2k1	TPS2k0	Selection of operation clock (CK2k)[Notes 1, 2] (k = 0, 1)				
				f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 20 MHz	f_{CLK} = 32 MHz	f_{PLL} = 64 MHz
0	0	0	f_{CLK}, f_{PLL}	2 MHz	5 MHz	20 MHz	32 MHz	64 MHz
0	0	1	$f_{CLK}/2$, $f_{PLL}/2$	1 MHz	2.5 MHz	10 MHz	16 MHz	32 MHz
0	1	0	$f_{CLK}/2^2$, $f_{PLL}/2^2$	500 kHz	1.25 MHz	5 MHz	8 MHz	16 MHz
0	1	1	$f_{CLK}/2^3$, $f_{PLL}/2^3$	250 kHz	625 kHz	2.5 MHz	4 MHz	8 MHz
1	0	0	$f_{CLK}/2^4$, $f_{PLL}/2^4$	125 kHz	312.5 kHz	1.25 MHz	2 MHz	4 MHz
1	0	1	$f_{CLK}/2^5$, $f_{PLL}/2^5$	62.5 kHz	156.2 kHz	625 kHz	1 MHz	2 MHz
1	1	0	Setting prohibited	–	–	–	–	–
1	1	1	Setting prohibited	–	–	–	–	–

Notes 1. When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timers KB0, KB1, KB2, and KC0 (TKBCEn = 0, TKCCE0 = 0).

2. When PLLON = 1 in the PLL control register (PLLCTL), f_{PLL} is supplied.

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

Remark f_{CLK}: CPU/peripheral hardware clock frequency

f_{PLL}: PLL output clock

7.3.3 16-bit timer KB operation control register n0 (TKBCTLn0)

TKBCTLn0 is a register that controls the count operation and sets the count clock of 16-bit timer.

TKBCTLn0 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 7-6. Format of 16-bit Timer KB Operation Control Register n0 (TKBCTLn0) (1/2)

Address: F0622H (TKBCTL00), F0662H (TKBCTL10), F06A2H (TKBCTL20) After reset: 0000H 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 of PWM output gate function by timer KC0 output
0	Does not use PWM output gate function.
1	Use PWM output gate function.

TKBSSEnp	Control of TKBOnp PWM output smooth start function
0	Does not use PWM output smooth start function.
1	Use PWM output smooth start function.

TKBDIEnp	Control of TKBOnp PWM output dithering function
0	Does not use PWM output dithering function.
1	Use 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.

TKBISRn1	TKBISRn0	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

TKBTSEn	Control of compare register batch overwrite function set by external trigger
0	Does not use compare register batch overwrite function set by external trigger.
1	Use compare register batch overwrite function set by external trigger.

Remarks 1. n = 0 to 2, p = 0, 1

2. T is the period of the last restart

Figure 7-6. Format of 16-bit Timer KB Operation Control Register n0 (TKBCTLn0) (2/2)

TKBSTSn1	TKBSTSn0	Selection of timer KBn count start trigger
0	0	Does not use trigger input.
0	1	External interrupt signal (INTP20)
1	0	External interrupt signal (INTP21)
1	1	When n = 0: Comparator 3 detection signal When n = 1: Comparator 1 detection signal When n = 2: Comparator 2 detection signal

Cautions 1. During timer operation, setting the other bits of the TKBCTLn0 register is prohibited. However, the TKBCTLn0 register can be refreshed (the same value is written).

2. Be sure to clear bits 15, 11, 6, and 3 to "0".

3. When using the PWM output gate function of TKBOnp, set corresponding bits 5 to 0 (TKCTOE05 to TKCTOE00) of the TKCIOC01 register to 1 (enable the TKCO05 to TKCO00 output).

4. For setting of INTP20/INTP21, see CHAPTER 14 COMPARATOR.

Remark n = 0 to 2, p = 0, 1

7.3.4 16-bit timer KB operation control register n1 (TKBCTLn1)

TKBCTLn1 is a register that controls the count operation and sets the count clock of 16-bit timer.

TKBCTLn1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-7. Format of 16-bit Timer KB Operation Control Register n1 (TKBCTLn1)

Address: F0629H (TKBCTL01), F0669H (TKBCTL11), F06A9H (TKBCTL21) After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
TKBCTL01	TKBCE0	0	0	TKBCKS0	TKBSCM0	0	TKBMD01	TKBMD00

Symbol	<7>	6	5	4	3	2	1	0
TKBCTLm1	TKBCEm	0	0	TKBCKSm	0	0	TKBMDm1	TKBMDm0

(m = 1, 2)

TKBCEn	Timer KBn operation control
0	Stops timer operation (counter is set to FFFF).
1	Enables timer count operation.

TKBCKS0	Timer KB0 clock selection
0	CK20 clock selected by TPS202 to TPS200 bits
1	CK21 clock selected by TPS212 to TPS210 bits

TKBSCMn	Timer KBn start operation control
0	Operates using clock selected by TKBCKSn bit
1	The count start timing per se is to start when the CK20 and CK21 clock are matched. After the operation is started, the clock selected by the TKBCKSn bit is used for operation. **Caution By setting simultaneous start mode to the slave with TKBSCMn bit, start timing of the slave and master can be matched.**

TKBMDn1	TKBMDn0	Timer KBn operation mode selection
0	0	Standalone mode (uses master)
0	1	Simultaneous start / stop mode (uses slave)
1	0	Synchronous start / clear mode (uses slave)
1	1	Interleave PFC output mode

Cautions 1. During timer operation, setting the other bits of the TKBCTLn1 register is prohibited. However, the TKBCTLn1 register can be refreshed (the same value is written).

2. In TKBCTL01, be sure to clear bits 6, 5, and 2 to "0".

3. In TKBCTLm1, be sure to clear bits 6, 5, 3, and 2 to "0".

Remark n = 0 to 2, m = 1, 2

7.3.5 16-bit timer KB output control register n0 (TKBIOCn0)

TKBIOCn0 is a register that setting the default level/active level in 16-bit timer KBn output (TKBOnp).

TKBIOCn0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-8. Format of 16-bit Timer KB Output Control Register n0 (TKBIOCn0)

Address: F0626H (TKBIOC00), F0666H (TKBIOC10), F06A6H (TKBIOC20) After reset: 00H 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

Cautions 1. During timer operation, setting the other bits of the TKBIOCn0 register is prohibited. However, the TKBIOCn0 register can be refreshed (the same value is written).

2. Be sure to clear bits 7 to 4 to "0".

3. Actual TKBOnp pin output is set not only by TKBOnp output but by the port mode registers (PMxx) and port registers (Pxx) for the shared ports.

Remark n = 0 to 2, p = 0, 1

7.3.6 16-bit timer KB output control register n1 (TKBIOCn1)

TKBIOCn1 is a register that controls disable/enable timer control in 16-bit timer KBn output (TKBOnp).

TKBIOCn1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-9. Format of 16-bit Timer KB Output Control Register n1 (TKBIOCn1)

Address: F0628H (TKBIOC01), F0668H (TKBIOC11), F06A8H (TKBIOC21) After reset: 00H 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

Cautions 1. The TKBIOCn1 register can be overwritten while the timer is operating.

2. Be sure to clear bits 7 to 2 to "0".

3. Actual TKBOnp pin output is set not only by TKBOnp output but by the port mode registers (PMxx) and port registers (Pxx) for the shared ports.

Remark n = 0 to 2, p = 0, 1

7.3.7 16-bit timer KB flag register n (TKBFLGn)

TKBFLGn is a register with status flags for 16-bit timer KBn.

TKBFLGn can be read by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-10. Format of 16-bit Timer KB Flag Register n (TKBFLGn)

Address: F0613H (TKBFLG0), F0653H (TKBFLG1), F0693H (TKBFLG2) After reset: 00H 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 (TKBRDTnp = 1 occurred during PWM output smooth start execute (TKBSSFnp = 0))

TKBIRFn	Undetected INTP21 trigger error flag for interleave 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 TKBIRSn1 and TKBIRSn0)

TKBIEFn	Multiplex detection INTP21 trigger error flag for interleave 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 TKBCLMFn
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 TKBRDTn.

Remarks 1. n = 0 to 2, p = 0, 1

2. T is the period of the last restart

7.3.8 16-bit timer KB trigger register n (TKBTRGn)

TKBTRGn is a trigger register used for batch overwriting of the compare register for 16-bit timer KBn.

TKBTRGn can be written by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-11. Format of 16-bit Timer KB Trigger Register n (TKBTRGn)

Address: F0612H (TKBTRG0), F0652H (TKBTRG1), F0692H (TKBTRG2) After reset: 00H 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

7.3.9 16-bit timer KB flag clear trigger register n (TKBCLRn)

TKBCLRn is a register used to clear flags in the 16-bit timer KB flag register n (TKBFLGn).

TKBCLRn can be written by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-12. Format of 16-bit Timer KB Flag Clear Trigger Register n (TKBCLRn)

Address: F0627H (TKBCLR0), F0667H (TKBCLR1), F06A7H (TKBCLR2) After reset: 00H 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 interleave PFC mode
0	Invalid setting
1	Clear the TKBIRFn flag to “0”.

TKBCLIEn	Trigger for clearing multiplex detection INTP21 trigger error flag for interleave 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

7.3.10 16-bit timer KB dithering count registers n0, n1 (TKBDNRn0, TKBDNRn1)

TKBDNRnp is a register that 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 7-2 shows the relation between the TKBDNRnp setting and the active period for N repetitions of one count clock extended.

TKBDNRnp can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-13. Format of 16-bit Timer KB Dithering Count Register np (TKBDNRnp)

Address: F060EH (TKBDNR00), F064EH (TKBDNR10), F068EH (TKBDNR20) After reset: 00H R/W
F0610H (TKBDNR01), F0650H (TKBDNR11), F0690H (TKBDNR21)

Symbol	7	6	5	4	3	2	1	0
TKBDNRnp					0	0	0	0

Caution Be sure to clear bits 3 to 0 to “0”.

Remark n = 0 to 2, p = 0, 1

Table 7-2. 16-bit Timer KB Dithering Count Register np (TKBDNRnp) Setting

Period / Repetitions (N)	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	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	

Remarks 1. □ cell period: Inactive output via settings in the TKBCRn1 abd TKBCRn3 registers
■ cell period: Inactive output via “settings + 1” in the TKBCRn1 and TKBCRn3 registers

2. n = 0 to 2, p = 0, 1

7.3.11 16-bit timer KB compare 1L & dithering count register n0 (TKBCRLDn0)

TKBCRLDn0 is a register that stores the “lower 8 bits of TKBCRn1 register” values in its higher 8 bits and the “TKBDNRn0 register” values in its lower 8 bits.

TKBCRLDn0 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 7-14. Format of 16-bit Timer KB Compare 1L & Dithering Count Register n0 (TKBCRLDn0)

Address: F0614H (TKBCRLD00), F0654H (TKBCRLD10), F0694H (TKBCRLD20) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKBCRLDn0													0	0	0	0

Lower 8 bits of TKBCRn1 register — TKBDNRn0 register

Caution Be sure to clear bits 3 to 0 to “0”.

Remark n = 0 to 2

7.3.12 16-bit timer KB compare 3L & dithering count register n1 (TKBCRLDn1)

TKBCRLDn1 is a register that stores the “lower 8 bits of TKBCRn3 register” values in its higher 8 bits and the “TKBDNRn1 register” values in its lower 8 bits.

TKBCRLDn1 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 7-15. Format of 16-bit Timer KB Compare 3L & Dithering Count Register n1 (TKBCRLDn1)

Address: F0616H (TKBCRLD01), F0656H (TKBCRLD11), F0696H (TKBCRLD21) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKBCRLDn1													0	0	0	0

Lower 8 bits of TKBCRn3 register — TKBDNRn1 register

Caution Be sure to clear bits 3 to 0 to “0”.

Remark n = 0 to 2

7.3.13 16-bit timer KB smooth start initial duty registers n0, n1 (TKBSIRn0, TKBSIRn1)

TKBSIRnp is a register that sets the default duty for the PWM output smooth start function for TKBOnp output.

TKBSIRnp can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 7-16. Format of 16-bit Timer KB Smooth Start Initial Duty Register np (TKBSIRnp)

Address: F060AH (TKBSIR00), F064AH (TKBSIR10), F068AH (TKBSIR20) After reset: 0000H R/W
F060CH (TKBSIR01), F064CH (TKBSIR11), F068CH (TKBSIR21)

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

7.3.14 16-bit timer KB smooth start step width registers n0, n1 (TKBSSRn0, TKBSSRn1)

TKBSSRnp is a register that is used by the PWM output smooth start function for TKBOnp output.

When the value of this register is N (N = 0000B to 1111B), output of a PWM with the active output period is continued for N + 1 times by setting TKBSIRnp. 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 TKBCRn1 and TKBCRn3 have the same duty, the PWM output smooth start function is cleared and normal PWM output is set.

TKBSSRnp can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-17. Format of 16-bit Timer KB Smooth Start Step Width Register np (TKBSSRnp)

Address: F060FH (TKBSSR00), F064FH (TKBSSR10), F068FH (TKBSSR20) After reset: 00H R/W
F0611H (TKBSSR01), F0651H (TKBSSR11), F0691H (TKBSSR21)

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

7.3.15 16-bit timer KB maximum frequency limit setting register n (TKBMFRn)

TKBMFRn is a register that sets the minimum period for the timer restart of external trigger.

When the counter (TKBCNTn) value is smaller than this TKBMFRn value, if trigger input is detected, the trigger is held pending, and the counter (TKBCNTn) is cleared (restart) after counting to the value set to TKBMFRn.

TKBMFRn can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 7-18. Format of 16-bit Timer KB Maximum Frequency Limit Setting Register n (TKBMFRn)

Address: F0624H (TKBMFR0), F0664H (TKBMFR1), F06A4H (TKBMFR2) After reset: 0000H 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

7.3.16 Peripheral function switch register 0 (PFSEL0)

PFSEL0 selects function setting I/O in peripheral function and 16-bit timers KB0, KB1, and KB2.

Bits 0 and 1 use external interrupts INTP20 and INTP21 either for PWM control of 16-bit timers KB0, KB1, and KB2 or for clearing stop mode.

Bits 2 and 3 select the timer trigger for A/D conversion.

Bit 4 selects whether or not to use the INTP20 noise filter. When INTP20 is used by forced output stop function 2, select to not use the noise filter in order to speed up the reaction time between trigger input and stopping output.

PFSEL0 can be written by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Remark When INTP20 is used with forced output stop function 2, select none for the noise filter to shorten the response interval from input of a trigger to termination of output.

Figure 7-19. Format of Peripheral Function Switch Register 0 (PFSEL0)

Address: F05C6H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	3	2	<1>	<0>
PFSEL0	0	CMP2STEN	CMP0STEN	PNFEN	ADTRG11	ADTRG10	TMRSTEN1	TMRSTEN0

CMP2STEN	CMP0STEN	Comparator interrupt selection
See **CHAPTER 14 COMPARATOR**.		

PNFEN	Use/Do not use external interrupt INTP20 noise filter
0	Use noise filter
1	Do not use noise filter

ADTRG11	ADTRG10	Timer trigger selection for A/D conversion
0	0	Timer KB0 trigger source
0	1	Timer KB1 trigger source
1	0	Timer KB2 trigger source
1	1	Setting prohibited

TMRSTEN1	Switch of external interrupt INTP21[Note]
0	External interrupt function is selected (stop mode release enabled, timer restart disabled)
1	Timer restart function is selected (stop mode release disabled, timer restart enabled).

TMRSTEN0	Switch of external interrupt INTP20[Note]
0	External interrupt function is selected (stop mode release enabled, timer restart disabled)
1	Timer restart function/forced output stop function 2 is selected (stop mode release disabled, timer restart enabled).

Note When INTP20 or INTP21 is used as a trigger of the timer KB forced output stop function 2 or timer restart function, see **14.5 Caution for Using Timer KB Simultaneous Operation Function**.

Remark See **Figure 14-1 Block Diagram of Comparator**.

7.3.17 Port mode register 20 (PM20)

This register specifies input or output mode for port 20 in 1-bit units.

When using the P200/TKBO00/INTP22, P201/TKBO01, P202/TKBO10/(INTP21), P203/TKBO11/TKCO02/(INTP20), P204/TKBO20/TKCO03, and P205/TKBO21/TKCO04/DALITxD4 pins for timer output, set PM200 to PM205 and the output latches of P200 to P205 to 0.

PM20 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 7-20. Format of Port Mode Register 20 (PM20) (38-Pin Products)

Address: F0510H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM20	1	PM206	PM205	PM204	PM203	PM202	PM201	PM200

s	P20n pin I/O mode selection (n = 0 to 6)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bit 7 of the PM20 register to "1".
For 20-pin products, bits 4 to 6 of the PM20 register must be set to output mode (specified by setting the port registers and port mode registers to 0) by software after reset is cancelled.

Remark The figure shown above presents the format of port mode register 20 of the 38-pin products. For the format of the port mode register of other products, see **Table 4-3 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx Registers and Bits Mounted on Each Product**.

7.4 Operation of 16-bit Timers KB0, KB1 and KB2

Operation specifications of 16-bit timers KB0, KB1 and KB2 described below.

- Counter basic operation (See **7.4.1**)
- Default level and active level (See **7.4.2**)
- Stop/restart operation (See **7.4.3**)
- Batch overwrite (See **7.4.4**)

There are 6 different operation modes for 16-bit timers KB0, KB1 and KB2

- Standalone mode (period controlled by TKBCRn0) (See **7.4.5**)
- Standalone mode (period controlled by external trigger input) (See **7.4.6**)
- Simultaneous start/stop mode (period controlled by TKBCRn0) (See **7.4.7**)
- Simultaneous start/stop mode (period controlled by external trigger input) (See **7.4.7**)
- Synchronous start/clear mode (period controlled by master) (See **7.4.8**)
- Interleave PFC output mode (See **7.4.9**)

Figure 7-21. Timer KB Operation Setting Example (Operation Start Flow)

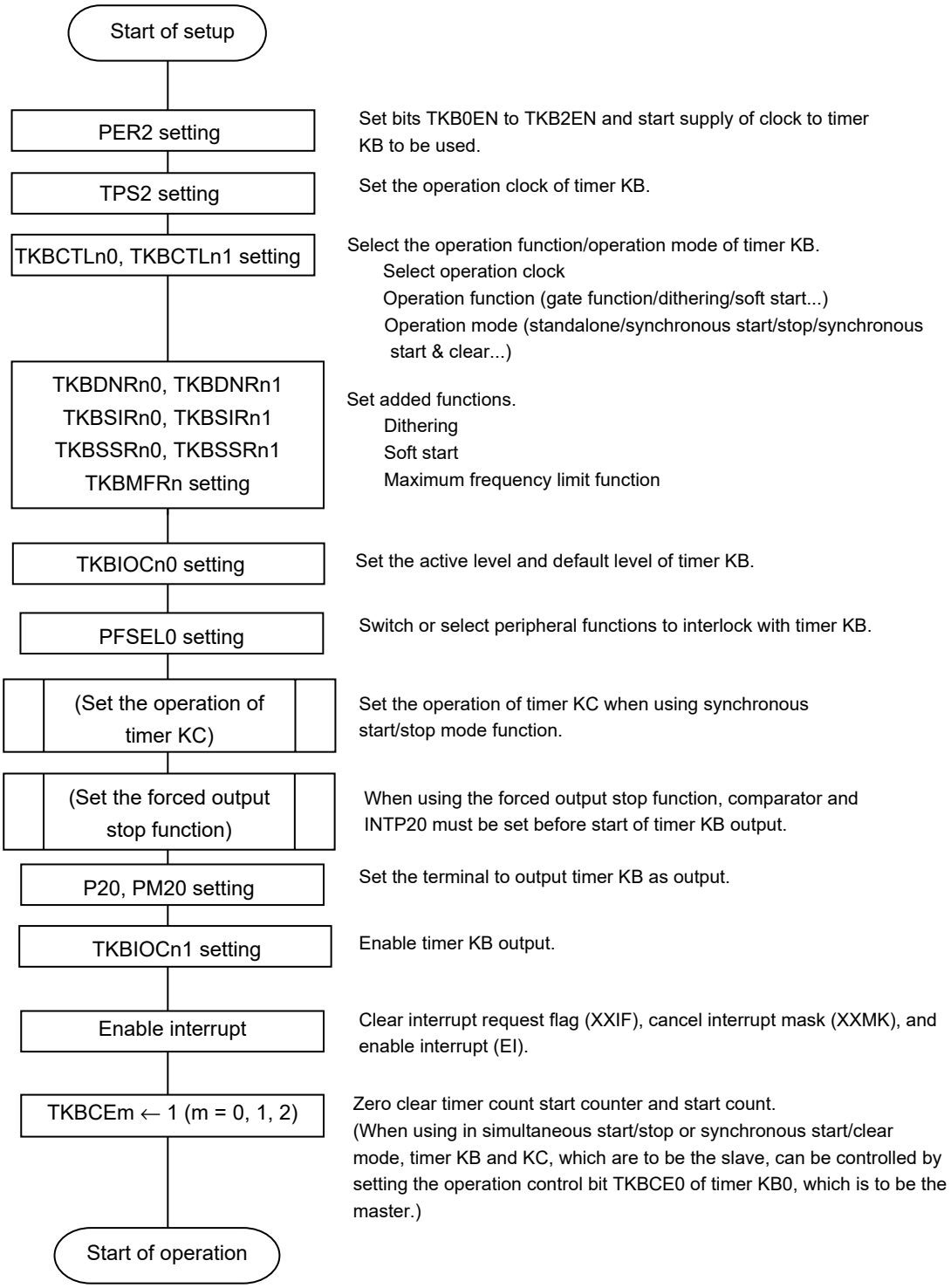
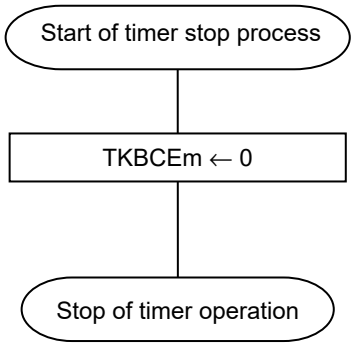
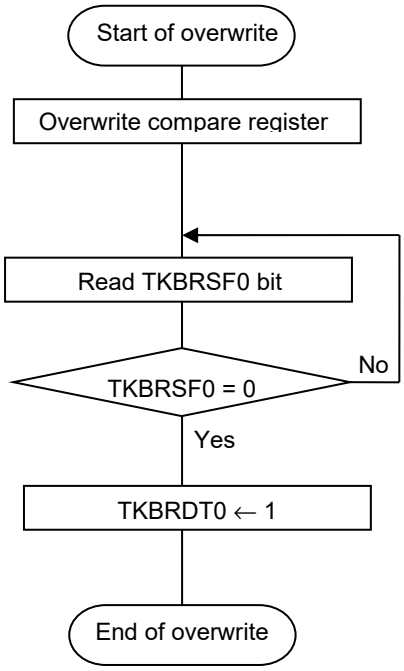


Figure 7-22. Timer KB Operation Setting Example (Operation Stop Flow)



Timer count stops and timer KB output becomes fixed output at the set default level.
(When using in synchronous start & stop mode, clearing of timer operation enable bit TKBCE0 of the master also stops operation of the slave timer.)

Figure 7-23. Timer KB Operation Setting Example (Compare Register Batch Overwrite Flow)



TKBCNTn, TKBCRn0 to TKBCRn3, TKBTGCRn (n = 0, 1, 2) Overwrite TKBCNTn, TKBCRn0 to TKBCRn3, TKBTGCRn (n = 0, 1, 2) TKBDNRn0, TKBDNRn1, TKBCRLDn0, and TKBCRLDn1 register.

Read TKBRSF0 of TKBFLG0 register.
Accept batch overwrite request when TKBRSF0 = 0.

Compare register batch overwrite request

After accepting overwrite request, TKBRSF0 bit remains “1” until setting is actually changed.

Remark The batch overwrite function is used to change the timer counter operation setting while timer KB is operating. The set value is reflected to the operation from the next restart.

7.4.1 Counter basic operation

(1) Count start operation

In any mode, the 16-bit counter of timer KB starts its counting from initial value of FFFFH. It increments the counter from FFFFH to 0000H, 0001H, 0002H, 0003H and so on.

(2) Clear operation

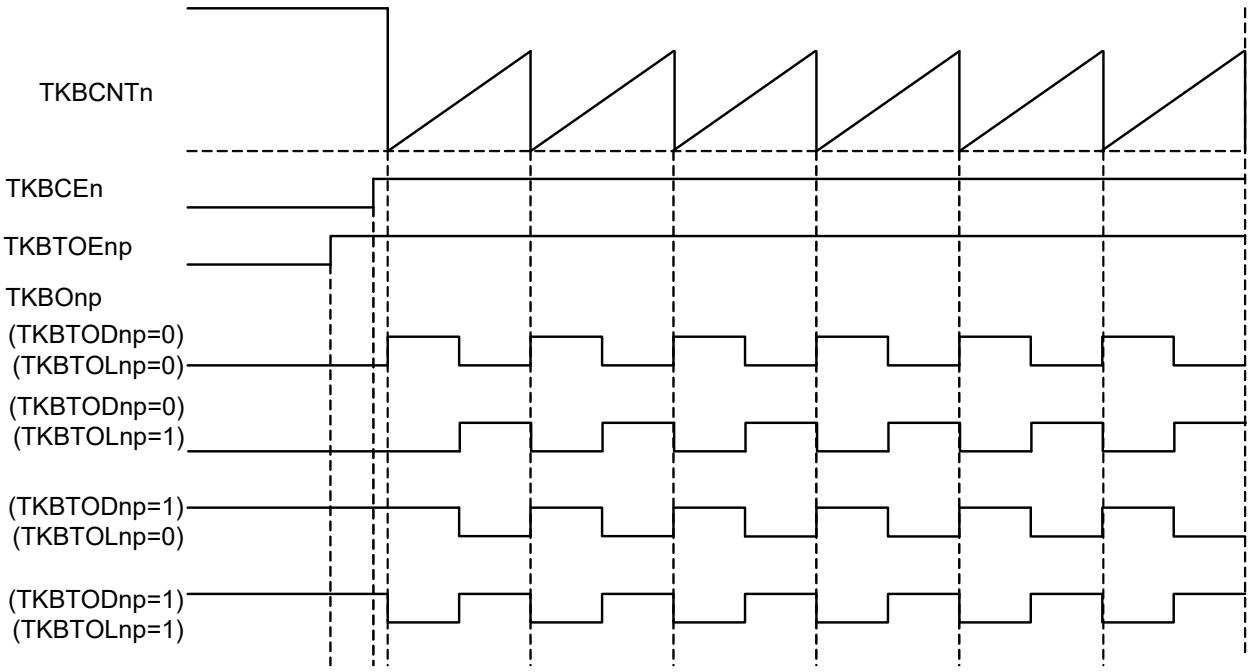
The 16-bit counter is reset to 0000H when the 16-bit counter value matches with the value defined in TKBCRn0 or an external trigger is in effect if the period is determined by external triggers. INTTMKBn interruption occurs when the counter is cleared at the time when it matches with the value defined in TKBCRn0, but it does not occur when the counter is cleared by an external trigger.

7.4.2 Default level and active level

(1) Basic operation

Default level and active level settings are available for timer KB output by 16-bit timer KB output control register n0 (TKBIOCn0).

Figure 7-24. Figure of Timing of Default and Active Level (Basic Operation)



When TKBTOEnp is switched from “0” to “1”, PWM waveform is output according to the generation of TKBOnp set condition/reset condition and TKBTOLnp setting.

When TKBTOEnp is switched from “1” to “0”, default level is output for TKBOnp according to TKBTODnp setting.

(2) TKBTOEnp switched from "0" to "1"

When TKBTOEnp is changed from 0 to 1 before the value of counter TKBCNTn matches with the value of compare register 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 TKBTOEnp is changed from 0 to 1 after the value of counter TKBCNTn matches with the value of compare register TKBCRnp, the timer output remains its initial setting level until the next timing of match occurs.

Figure 7-25. Figure of Timing of Default and Active Level
(TKBTOEnp = 0 Switched to 1 Before Matching Counter and Compare Register (TKBCRn1 to TKBCRn3))

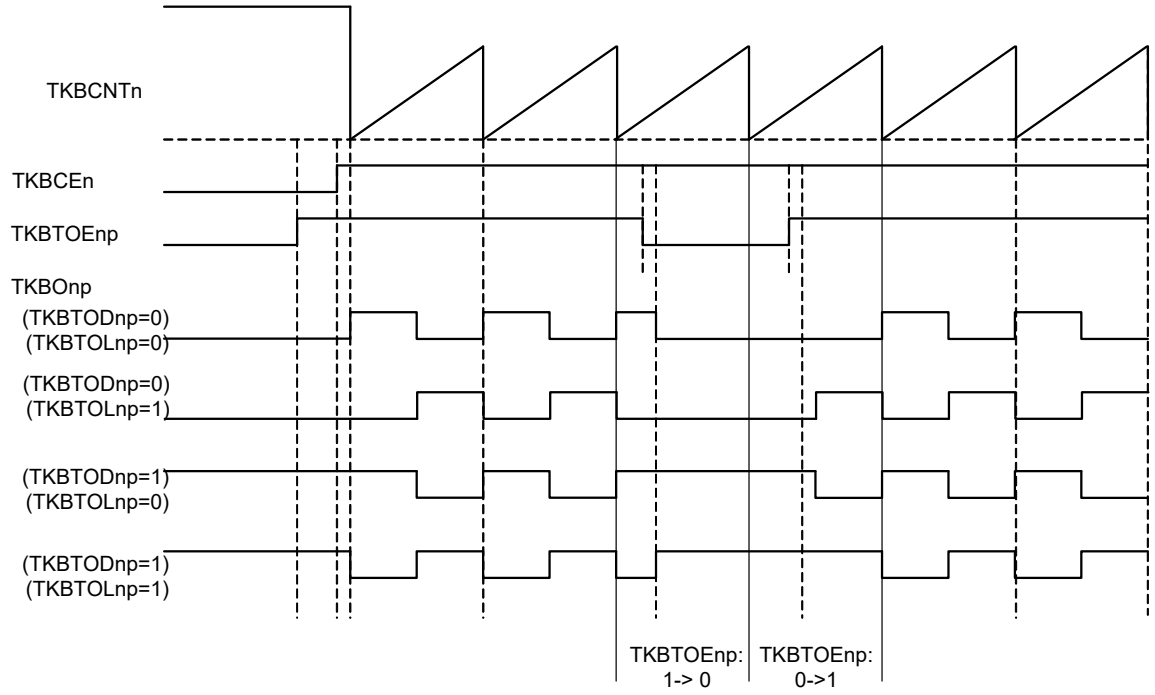
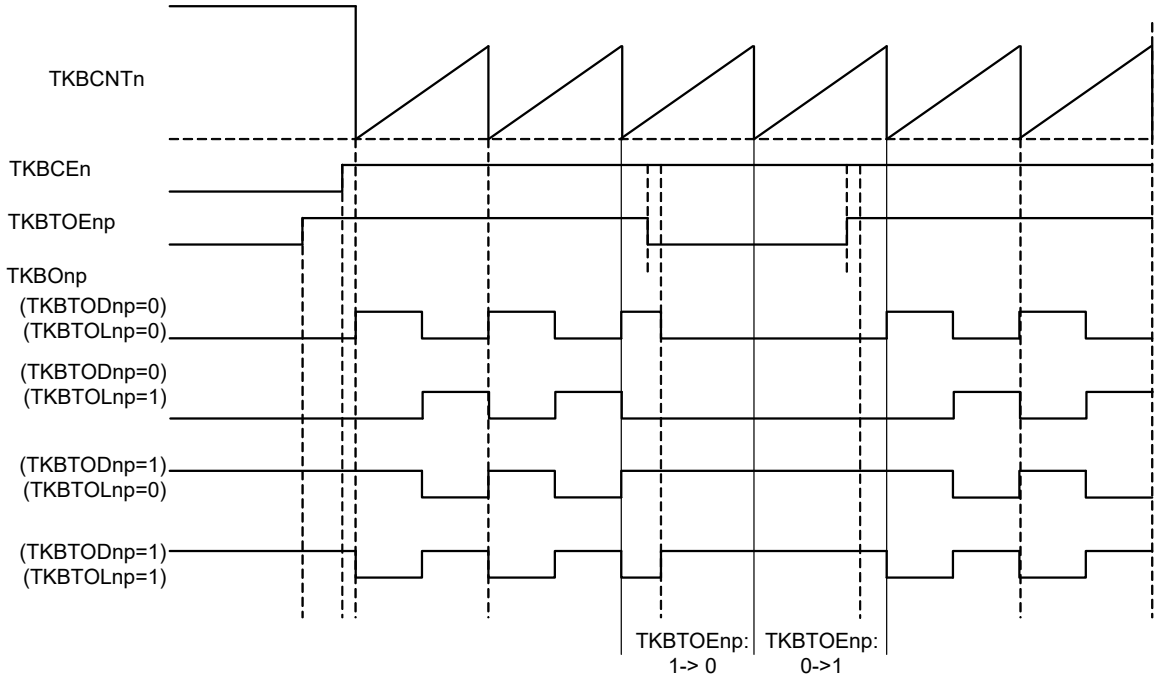


Figure 7-26. Figure of Timing of Default and Active Level
(TKBTOEnp = 0 Switched to 1 After Matching Counter and Compare Register (TKBCRn1 to TKBCRn3))

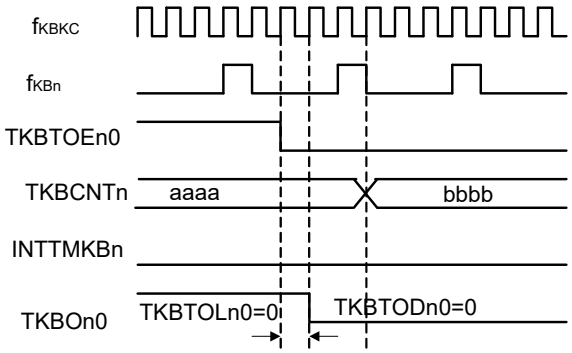


(3) TKBTOEnp switched from “1” to “0”

(a) Basic timing

TKBOnp is default level set by TKBTODnp after 1 f_{KBKC} when TKBTOEnp is switched from “1” to “0”.

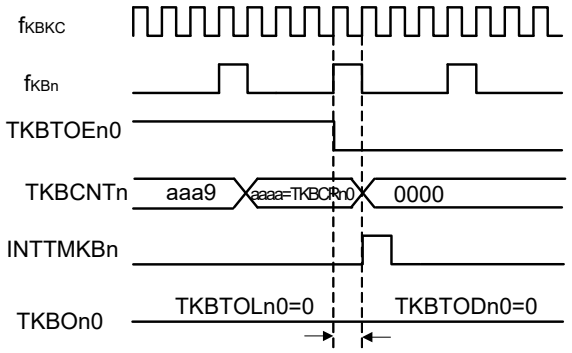
Figure 7-27. Figure of Timing of Default and Active Level (TKBTOEn0 Switched from “1” to “0”)



(b) When the setting due to the matched value of TKBCRn0 and the event that TKBTOEnp is cleared occur at the same instant:

When TKBOnp set timing (Low to High) is simultaneous with the matching between TKBCNTn and TKBCRnm, the change of TKBTOEnp is given priority to become default level set for TKBTODnp.

Figure 7-28. Figure of Timing for Default and Active Level (TKBOn0 Set Timing (Low to High) Is Simultaneous with the Matching Between TKBCNTn and TKBCRnm)

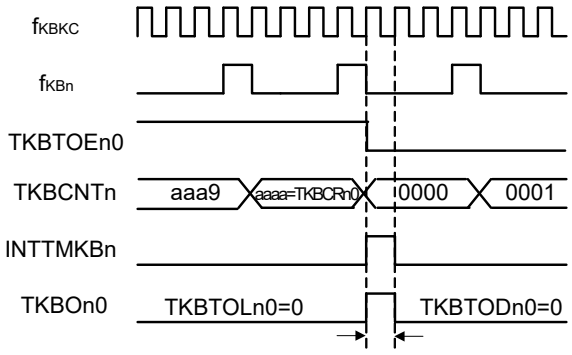


(c) When the operation of TKBTOEnp is simultaneous with generation of timer count clock

TKBOnp is set by the matching of TKBCNTn = TKBCRnm in case when the operation of TKBTOEnp is simultaneous with generation of timer count clock.

After 1 f_{KBn}, TKBOnp is default level which is set with TKBTODnp.

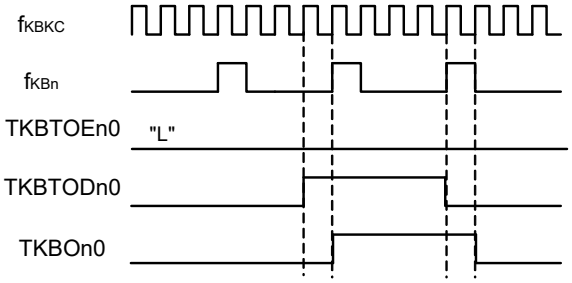
Figure 7-29. Figure of Timing of Default and Active Level
(Operation of TKBTOEn0 Is Simultaneous with generation of Timer Count Clock.)



(4) Change TKBTODnp at TKBTOEnp = 0

When TKBTODnp being changed at TKBTOEnp = 0, after 1 f_{KBKC}, TKBOnp is default level which is set with TKBTODnp.

Figure 7-30. Figure of Timing of Default and Active Level
(Change TKBTODn0 at TKBTOEn0 = 0)



7.4.3 Stop/restart operation

Stop and start of operation of 16-bit timer KB is available by controlling TKBCEn.

16-bit timer KB is reset and stop operation by changing TKBCEn from “1” to “0”.

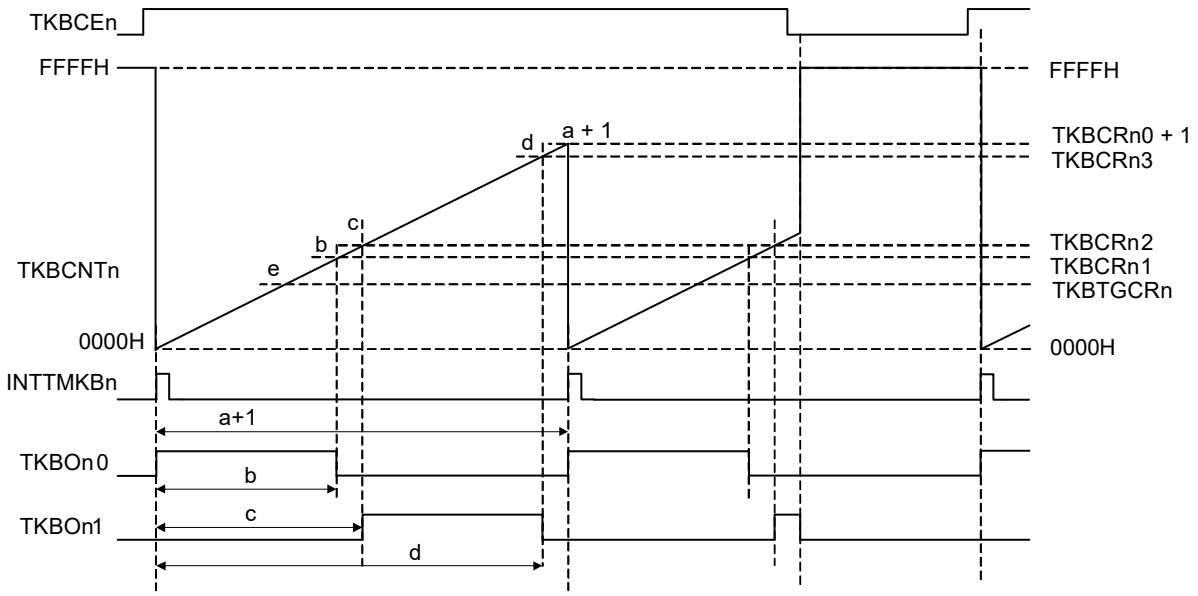
Counter TKBCNTn is reset to FFFFH and stop operation then.

TKBOnp output outputs default level set by TKBTODnp.

16-bit timer KB starts operation by changing TKBCEn from “0” to “1”.

Counter TKBCNTn maintains FFFFH when TKBCEn = 0 and start up counting operation by changing TKBCEn from “0” to “1”.

Figure 7-31. Figure of Timing of Stop Operation (TKBTOLnp = 0, TKBTODnp = 0)

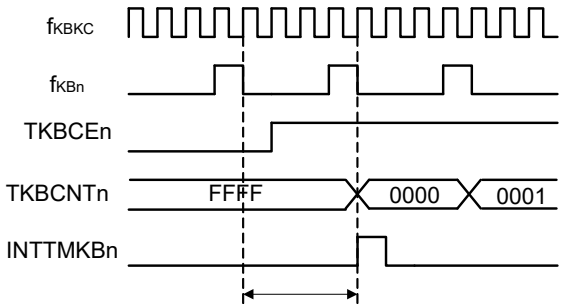


(1) Count operation start timing

When TKBCEn 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 7-32. Figure of Timing of Start Operation (TKBCEn Switched from “0” to “1”)

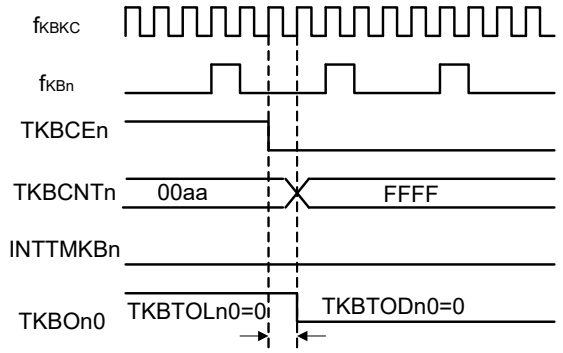


(2) Count operation stop timing

When TKBCEn is switched from “1” to “0” counting operation is stopped after the progress of minimum 1 f_{KBKC}.

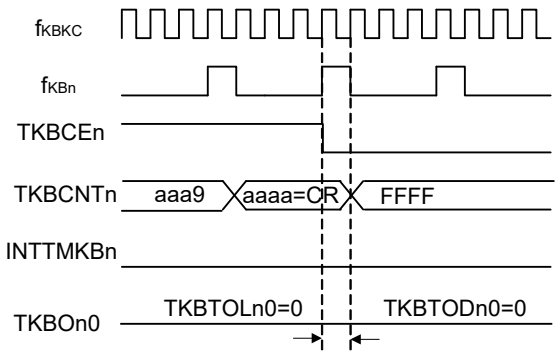
TKBCNTn is reset to FFFFH and TKBOnp is default level set by TKBTODnp.

Figure 7-33. Figure of Timing of Stop Operation (TKBCEn Switched from “1” to “0”)



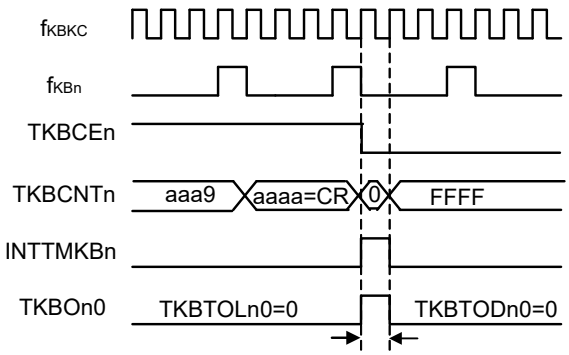
When TKBCEn is switched from “1” to “0” counting operation is stopped after the progress of minimum 1 f_{KBKC}.
Before the generation of 1 f_{KBn}, INTTMKBn is not output even matching of TKBCNTn = TKBCRn0 being generated.

Figure 7-34. 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 TKBCEn is simultaneous with the generation of 1 f_{KBKC}.
TKBCNTn is reset to FFFFH after the progress of 1 f_{KBKC} and TKBOnp is default level set by TKBTODnp.

Figure 7-35. Figure of Timing of Stop Operation
(Operation of TKBCEn Is Simultaneous with the Generation of Timer Count Clock.)

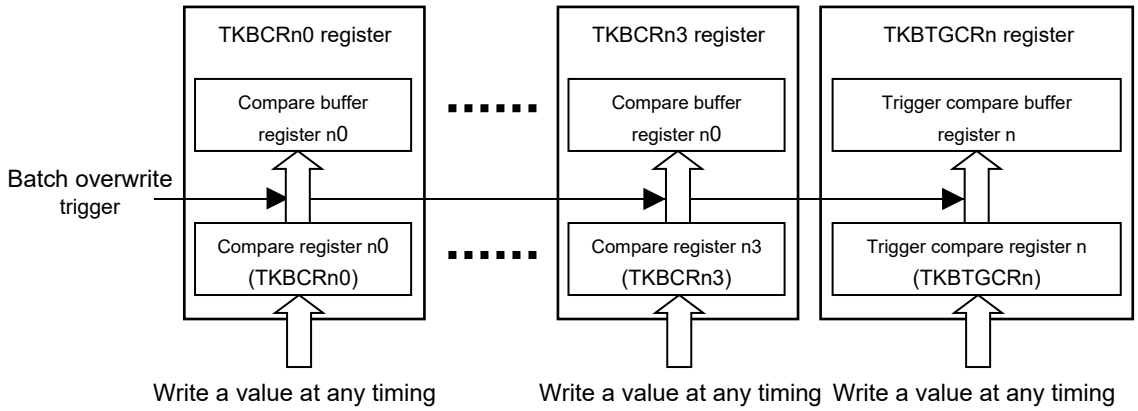


7.4.4 Batch overwrite operation

TMKB compare register np (TKBCRnp) for timer KB has, as shown in Figure 7-36, two stages.

Therefore, its value does not become effective immediately even if any value is set to TKBCRnp by a program. The value set to TKBCRnp 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 7-36. Compare Register Batch Overwrite Function



Remark As shown above, TMKB compare registers np (TKBCRnp) have two stage structure and they are treated as a single register except when values are written to them.

(1) Timing ob batch overwrite

There are three cases when the compare registers are written all together. Among these, (c) can be controlled by configuration of the register.

(a) When starting count operation of timer KB

(b) Count value of the 16-bit counter and the value that is set to TMKB compare register n0 (TKBCRn0) matches.

(c) An external trigger occurs, while batch overwrite with an external trigger is permitted.

7.4.5 Standalone mode (period controlled by TKBCRn0)

(1) Outline of functions

In standalone operation mode, the period is defined by setting value of TKBCRn0, then TKBOn0 is generated by TKBCRn0 and TKBCRn1, and then TKBOn1 is generated by TKBCRn2 and TKBCRn3.

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

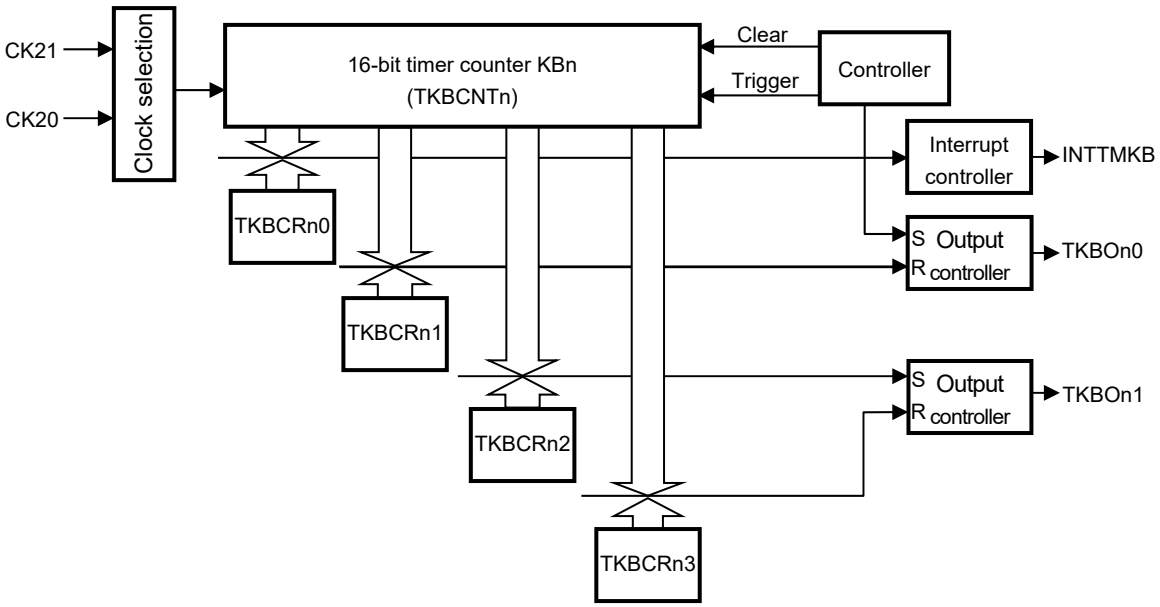
0% Output: TKBCRn3 setting = TKBCRn2 setting

100% Output: TKBCRn2 setting = 0000H, TKBCRn3 setting ≥ TKBCRn0 setting + 1

Caution It should always be: TKBCRn2 setting ≤ TKBCRn3 setting.

Figure 7-37 shows the configuration of standalone mode.

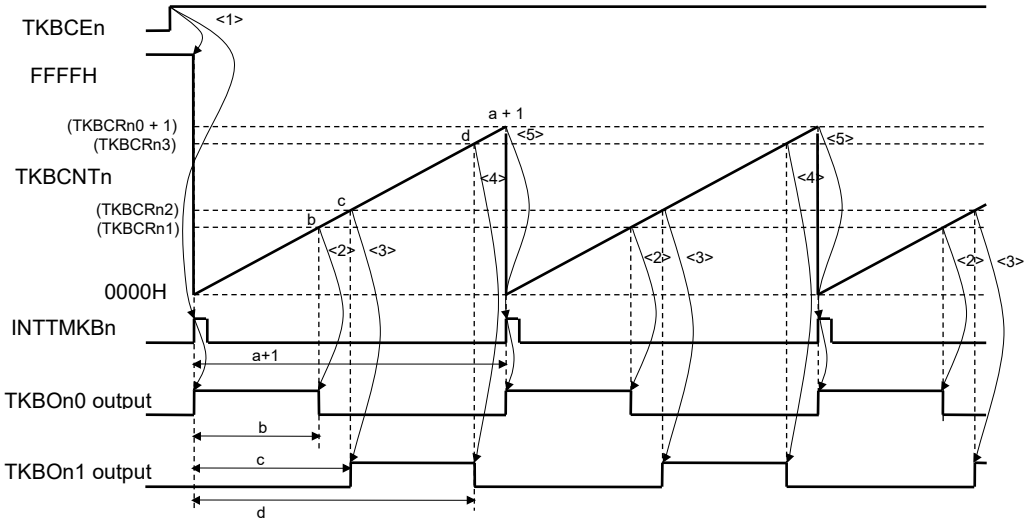
Figure 7-37. Configuration of Standalone Mode (Period Controlled by TKBCRn0)



(2) Outline of operation

Figure 7-38 shows the timing sample for standalone mode.

Figure 7-38. Timing Sample for Standalone Mode (Period Controlled by TKBCRn0)
(at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



This section describes an example about the standalone operation (periodic control by TKBCR0). The following descriptions are linked with <1> to <5> in Figure 7-38.

<1> When TKBCEn 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 TKB0TOD0 bit of TKB0IOC0 register to its active value (high level in this example) specified with TKB0TOL0 bit (TKBOn1 output hold its initial value specified with TKB0TOD1 bit).

<2> When TKBCNTn is counted up and its value matches with the value specified in TMKB compare register n1 (TKBCRn1), TKBOn0 output becomes inactive level.

<3> When TKBCNTn is counted up and its value matches with the value specified in TMKB compare register n2 (TKBCRn2), TKBOn1 output becomes active level.

<4> When TKBCNTn is counted up and its value matches with the value specified in TMKB compare register n3 (TKBCRn3), TKBOn1 output becomes inactive level.

<5> When TKBCNTn is counted up and its value matches with the value specified in TMKB compare register n0 (TKBCRn0), INTTMKBn output is generated at the next count clock and TKBOn0 output becomes active level. TKBCNTn starts its upward counting from 0000H.

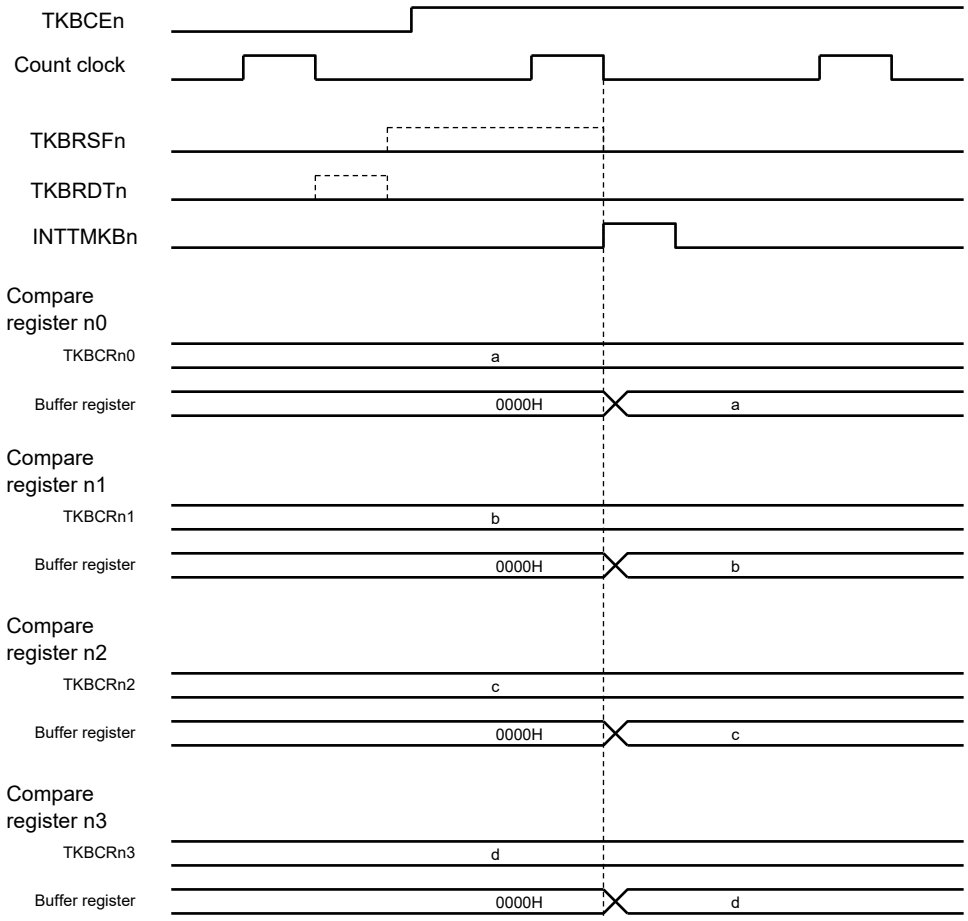
<6> Repeats <2> through <5>.

(3) Operation of batch overwrite (at starting the counting operation)

Compare register of the timers KB0, KB1 and KB2 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 TKBCEn bit.

Batch overwrite is generated without writing “1” on TKBRDTn bit only in case of counting operation start timing (see **Figure 7-39**).

Figure 7-39. Batch Overwrite Function: Figure of Buffer Updating Timing at Counting Operation Start



Remark When TKBCEn = 0, TKBRSFn is set to “1” at writing “1” to TKBRDTn.
TKBRSFn is cleared to “0” at counting operation start timing (counter start trigger generated).

(4) Batch overwrite function: Update buffer during counting operation

Compare register of the timers KB0, KB1 and KB2 has a function which updates internal buffer register simultaneously at the next counter clear (TKBCNTn and TKBCRn0 matched), identifying the writing “1” to TKBRDTn bit as batch overwriting trigger. TKBRSFn is provided as flag to indicate from writing of “1” to TKBRDTn bit until the completion of batch overwrite (see **Figure 7-40**).

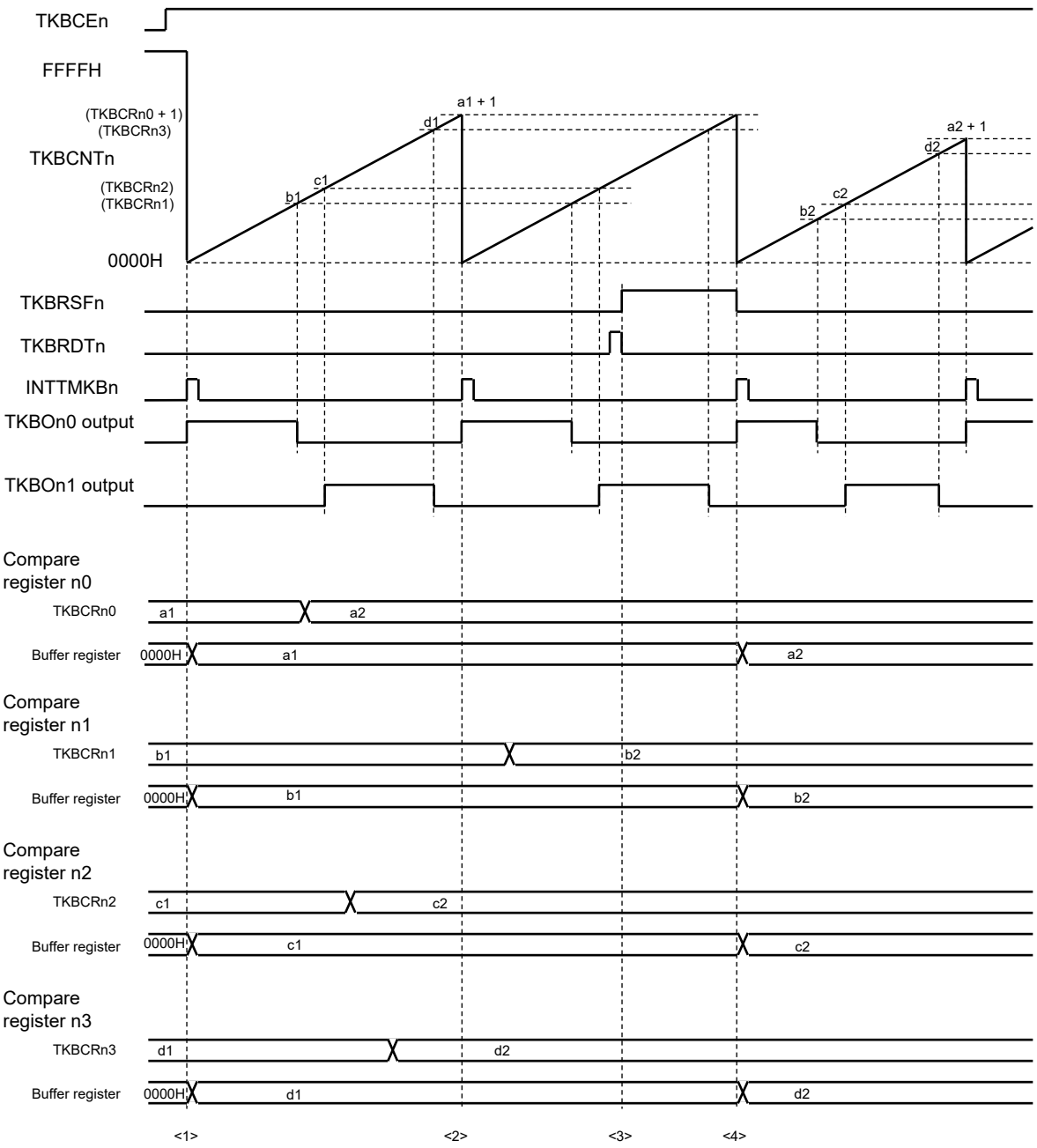
<1> Compare register setting is transferred to buffer register at the timing when TKBCEn is set from “0” to “1” and TKBCNTn starts counting operation.

<2> Overwriting is not generated if writing of “1” to TKBRDTn is not implemented even counter clear is generated after TKBCRn0 to TKBCRn3 and TKBTGCRn registers are overwritten.

<3> Batch overwrite pending flag (TKBRSFn) is “1” by writing “1” to TKBRDTn.

<4> Compare register setting is transferred to buffer register by counter clear generated at TKBRSFn = 1. TKBRSFn is “0” simultaneously.

Figure 7-40. Batch Overwrite Function: Figure of the Timing of Buffer Updating During Counting Operation



(5) Sample of register setting details at standalone mode (period controlled by TKBCRn0)

bit No.	15	14	13	12	11	10	9	8
TKBCTLn0	–	TKBGTEn1	TKBSSEn1	TKBDIEn1	–	TKBGTEn0	TKBSSEn0	TKBDIEn0
Setting	0	1/0	1/0	1/0	0	1/0	1/0	1/0

bit No.	7	6	5	4	3	2	1	0
TKBCTLn0	TKBMFEn	–	TKBIRSn1	TKBIRSn0	–	TKBTSEn	TKBSTSn1	TKBSTSn0
Setting	0	0	0	0	0	0	0	0

bit No.	7	6	5	4	3	2	1	0
TKBCTLn1	TKBCEn	–	–	TKBCKSn	TKBSCMn	–	TKBMDn1	TKBMDn0
Setting	1	0	0	1/0	0	0	0	0

bit No.	7	6	5	4	3	2	1	0
TKBIOCn0	–	–	–	–	TKBTOLn1	TKBTOLn0	TKBTODn1	TKBTODn0
Setting	0	0	0	0	1/0	1/0	1/0	1/0

bit No.	7	6	5	4	3	2	1	0
TKBIOCn1	–	–	–	–	–	–	TKBTOEn1	TKBTOEn0
Setting	0	0	0	0	0	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

□: Setting is fixed for this mode ■: Setting is not needed (default setting)

7.4.6 Standalone mode (period controlled by external trigger input)

(1) Outline of functions

By standalone mode, period can be controlled not only by TKBCRn0 but also by external trigger input.

Input signals selected by TKBSTSn1 and TKBSTSn0 bits of 16-bit timer KB operation control register are used to detect external trigger input (Timer restart function). By using this function, critical conduction mode PFC control can be implemented, for example.

When the external trigger input is detected, counter TKBCNTn is cleared with 0000H and TKBOn0/TKBOn1 output is respectively set to active level and inactive level. When setting value of TKBCRn0 and the counter (TKBCNTn) match is generated before external trigger input detection, 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 TKBCRn0, see **7.4.5 Standalone mode (period controlled by TKBCRn0)**.

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 ≤ KBCRn3 setting.

Figure 7-41 shows the configuration of standalone mode (period controlled by external trigger input).

Figure 7-41. Configuration of Standalone Mode (Period Controlled by External Trigger Input)

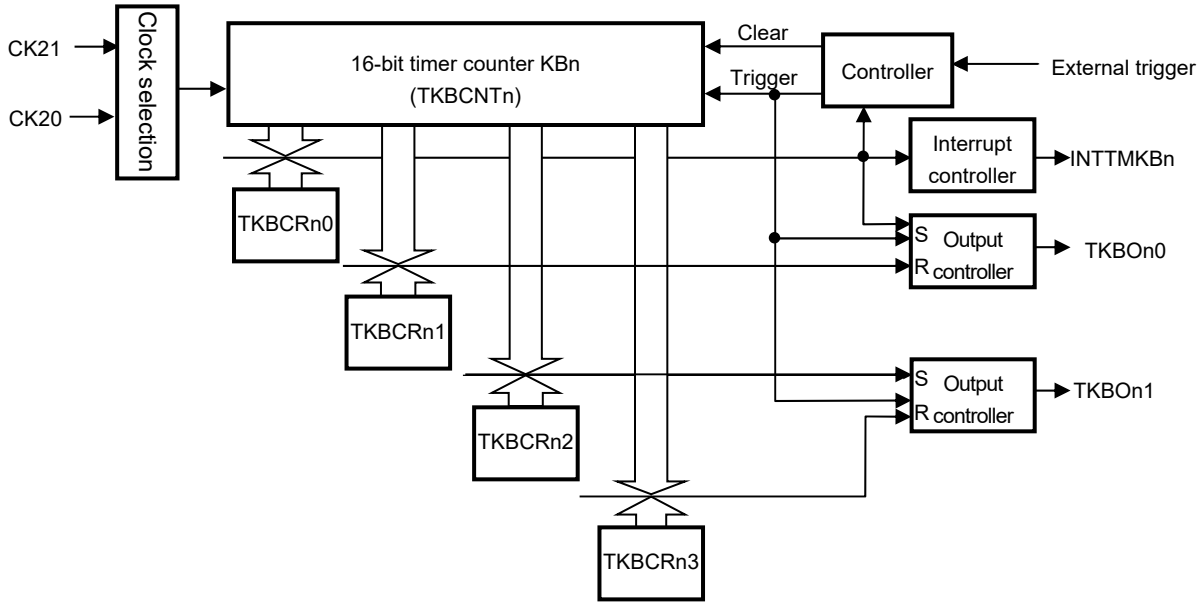


Table 7-3. External Trigger Assignment List of Standalone Operation (Period Controlled by External Trigger Input)

	Timer KB0	Timer KB1	Timer KB2
Comparator 0	–	–	–
Comparator 1	–	√	–
Comparator 2	–	–	√
Comparator 3	√	–	–
Comparator 4	–	–	–
Comparator 5	–	–	–
INTP20	√	√	√
INTP21	√	√	√

(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 TKBRDTn bit and through setting TKBTSEn bit of TKBCTLn0 register in “1”.

Same as in counter clear, batch overwriting is implemented as well in case when TKBCRn0 and counter (TKBCNTn) being matched before the detection of external trigger input after writing “1” to TKBRDTn bit.

Factor of external trigger input is selected at TKBSTSn1 and TKBSTSn0 of TKBCTLn0 register. Figure 7-42 shows an example of the timing of the batch overwrite operation with TKBTSEn bit set to “1”.

<1> Compare register setting is transferred to buffer register at the timing when TKBCEn is set from “0” to “1” and TKBCNTn starts counting operation.

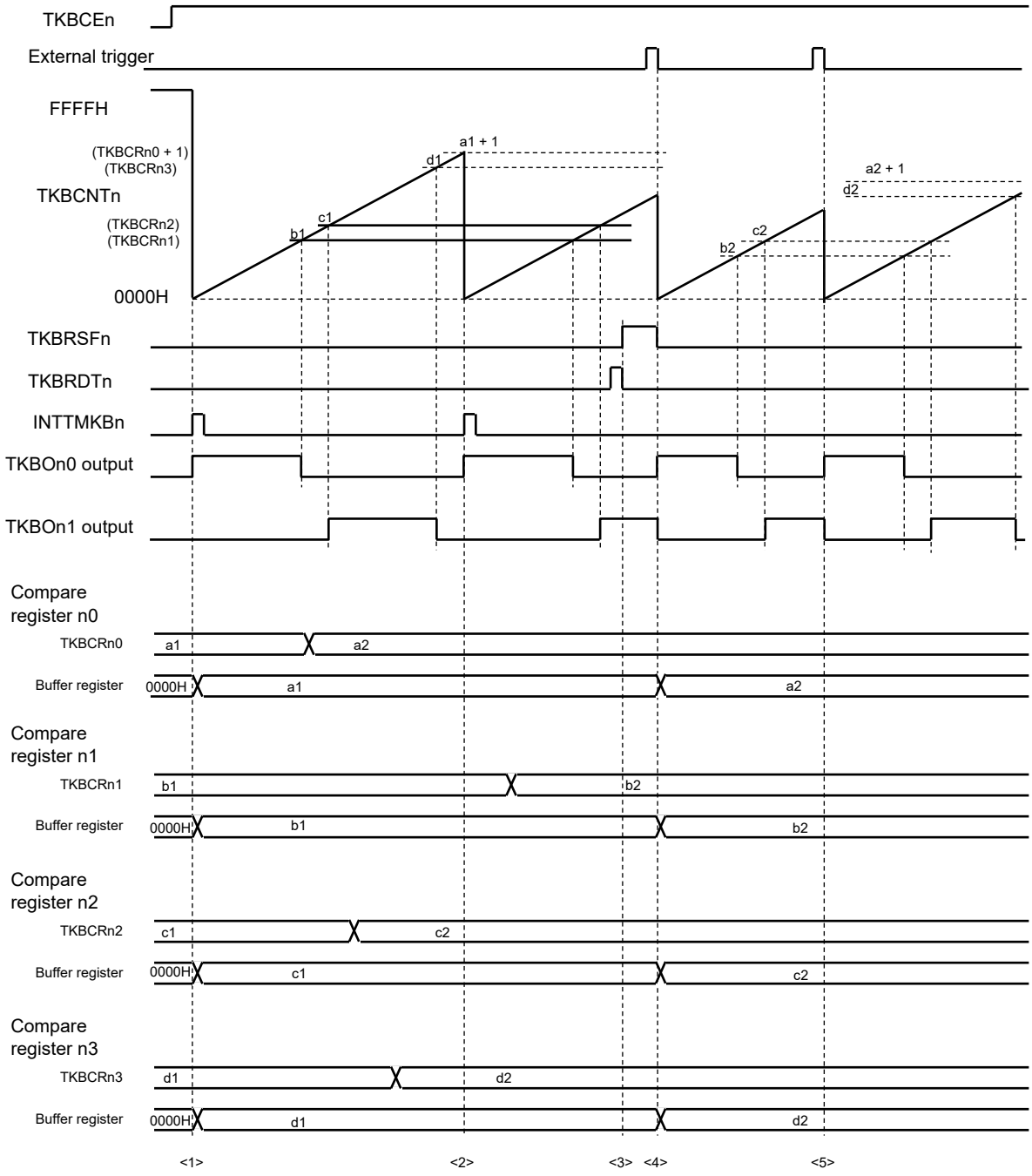
<2> Overwriting is not generated if writing of “1” to TKBRDTn is not implemented even counter clear is generated after TKBCRn0 to TKBCRn3 and TKBTGCRn registers are overwritten.

<3> Batch overwrite pending flag (TKBRSFn) is “1” by writing “1” to TKBRDTn.

<4> When a counter clear is generated by an external trigger input while TKBTSEn bit is set to “1” and TKBRSFn bit is “1”, the setting value in the compare register is transferred to the buffer register. At the same time, TKBRSFn bit 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 TKBRDTn bit.

Figure 7-42. 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)

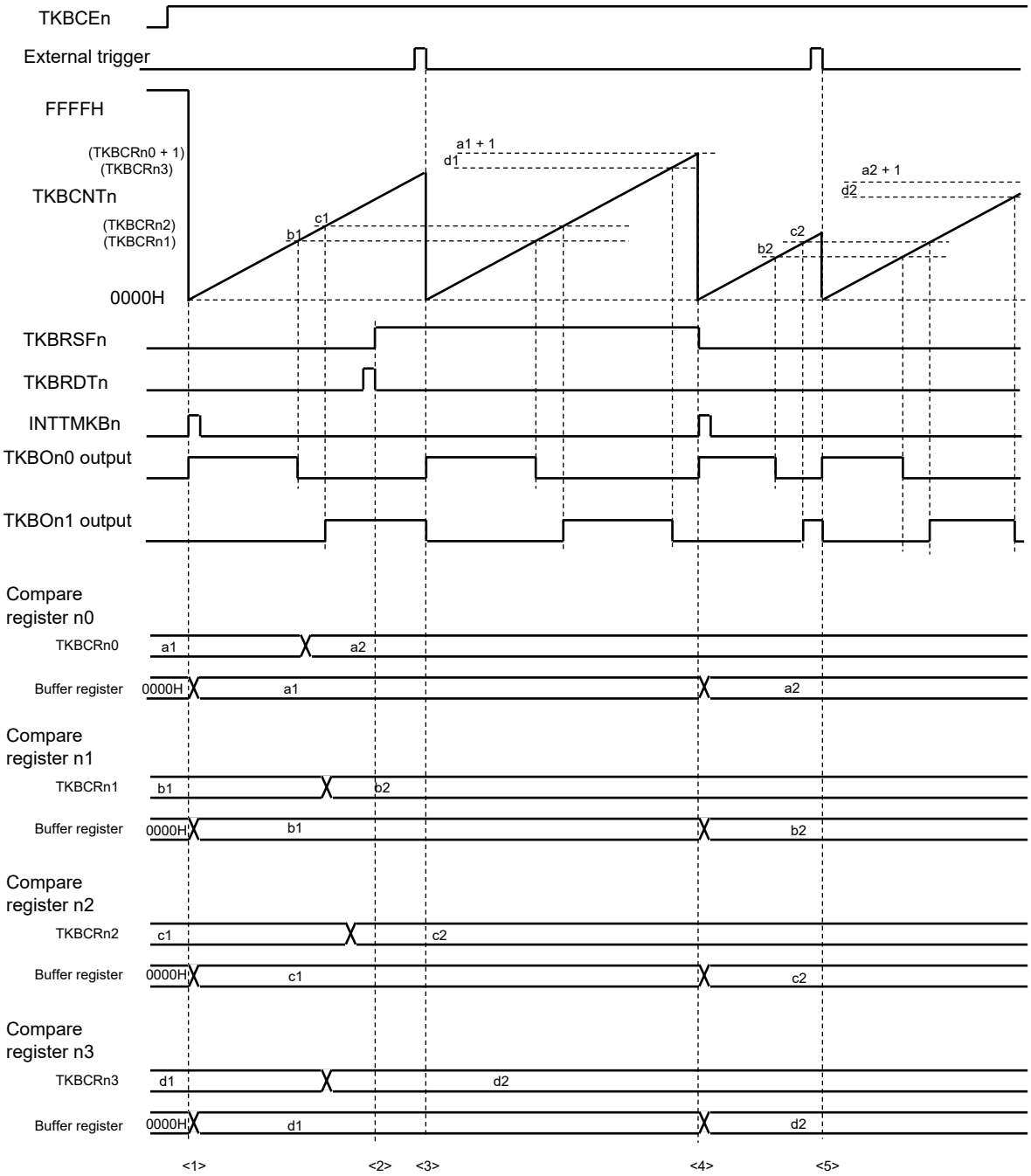


(3) Batch overwrite function (period controlled by external trigger input, buffer updating during counting operation (TKBTSEn bit clear to 0))

This is an example of the case where TKBTSEn bit in TKBCTLn0 register is set to “0” during standalone operation under the periodic control by external trigger input. In this case, the counter is cleared when a value “1” is written in TKBRDTn bit and the external trigger input is detected while batch overwrite suspension flag (TKBRSFn bit) is “1”.
An external trigger factor is selected with TKBSTSn1 and TKBSTSn0 bit in TKBCTLn0 register. Figure 7-43 shows an example of the batch overwrite operation timing when TKBTSEn bit is set to “0”.

<1> Compare register setting is transferred to buffer register at the timing when TKBCEn is set from “0” to “1” and TKBCNTn starts counting operation.
<2> After rewritten TKBCRn0 to TKBCRn3 and TKBTGCRn register, batch overwrite pending flag (TKBRSFn) is “1” by writing “1” to TKBRDTn.
<3> Even if the counter clear event is generated by the external trigger input, the batch overwrite does not occur unless TKBTSEn bit is “1”.
<4> When the counter clear event (TKBCNTn matches with TKBCRn0) occurs under the status of TKBRSFn bit is “1”, the value set to the compare register is transferred to the buffer register. At the same time, TKBRSFn bit becomes “0”.
<5> Even if the counter clear event is generated by the external trigger input, the batch overwrite does not occur unless TKBTSEn and TKBRSFn bits are both “1”.

Figure 7-43. 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 setting details at standalone mode (period controlled by external trigger input)

	15	14	13	12	11	10	9	8
TKBCTLn0	–	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	0	0	0	1/0	1/0	1/0

	7	6	5	4	3	2	1	0
TKBCTLn1	TKBCEn	–	–	TKBCKSn	TKBSCMn	–	TKBMDn1	TKBMDn0
	1/0	0	0	1/0	0	0	0	0

	7	6	5	4	3	2	1	0
TKBIOCn0	–	–	–	–	TKBTOLn1	TKBTOLn0	TKBTODn1	TKBTODn0
	0	0	0	0	1/0	1/0	1/0	1/0

	7	6	5	4	3	2	1	0
TKBIOCn1	–	–	–	–	–	–	TKBTOEn1	TKBTOEn0
	0	0	0	0	0	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
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)

7.4.7 Simultaneous start/stop mode

(1) Outline of functions

Slave timer KBm can be start/stop simultaneously by synchronization with count start/stop of master timer KB when master/slave is configured using multiple KBn timers.

Select “Standalone Mode (TKBMD01, TKBMD00 = 0, 0)” for master and “Simultaneous Start/Stop Mode (TKBMDm1, TKBMDm0 = 0, 1)” 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.

Cautions 1. Master is timer KB only.

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	O
Selecting different clocks for master and slave	When master selected clock is faster than slave selected clock	1	O
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 **7.4.5 Standalone mode (period controlled by TKBCRn0)** and **7.4.6 Standalone mode (period controlled by external trigger input)**.

(2) Operation mode combination available for simultaneous start/stop mode

Shows the operation mode available for simultaneous start/stop mode.

Master:

Operation Mode	TKBMDn1, TKBMDn0	TKBSTSn1, TKBSTSn0	Setting Available
Standalone mode (period controlled by TKBCRn0)	00B	00B	O
Standalone mode (period controlled by external trigger input)	00B	01B/10B/11B	O
Simultaneous start/stop mode (period controlled by TKBCRn0)	01B	00B	×
Simultaneous start/stop mode (period controlled by external trigger input)	01B	01B/10B/11B	×
Synchronous start/clear mode (period controlled by master)	10B	–	×
Interleave PFC output mode	11B	–	×

Slave:

Operation Mode	TKBMDn1, TKBMDn0	TKBSTSn1, TKBSTSn0	Setting Available
Standalone mode (period controlled by TKBCRn0)	00B	00B	×
Standalone mode (period controlled by external trigger input)	00B	01B/10B/11B	×
Simultaneous start/stop mode (period controlled by TKBCRn0)	01B	00B	O
Simultaneous start/stop mode (period controlled by external trigger input)	01B	01B/10B/11B	O
Synchronous start/clear mode (period controlled by master)	10B	–	×
Interleave PFC output mode	11B	–	×

(3) Simultaneous start/stop mode

Master: Sample of register setting details at standalone mode (period controlled by TKBCRn0)

	15	14	13	12	11	10	9	8
TKBCTLn0	–	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

	7	6	5	4	3	2	1	0
TKBCTLn1	TKBCEn	–	–	TKBCKSn	TKBSCMn	–	TKBMDn1	TKBMDn0
	1	0	0	1/0	1/0	0	0	0

	7	6	5	4	3	2	1	0
TKBIOCn0	–	–	–	–	TKBTOLn1	TKBTOLn0	TKBTODn1	TKBTODn0
	0	0	0	0	1/0	1/0	1/0	1/0

	7	6	5	4	3	2	1	0
TKBIOCn1	–	–	–	–	–	–	TKBTOEn1	TKBTOEn0
	0	0	0	0	0	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

□ : Setting is fixed for this mode ■ : Setting is not needed (default setting)

(4) Simultaneous start/stop mode

Slave: Sample of register setting details at standalone mode (period controlled by TKBCRn0)

	15	14	13	12	11	10	9	8
TKBCTLn0	–	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

	7	6	5	4	3	2	1	0
TKBCTLn1	TKBCEn	–	–	TKBCKSn	TKBSCMn	–	TKBMDn1	TKBMDn0
	1	0	0	1/0	0	0	0	1

	7	6	5	4	3	2	1	0
TKBIOCn0	–	–	–	–	TKBTOLn1	TKBTOLn0	TKBTODn1	TKBTODn0
	0	0	0	0	1/0	1/0	1/0	1/0

	7	6	5	4	3	2	1	0
TKBIOCn1	–	–	–	–	–	–	TKBTOEn1	TKBTOEn0
	0	0	0	0	0	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

□ : Setting is fixed for this mode ■ : Setting is not needed (default setting)

(5) Simultaneous start/stop mode

Master: Sample of register setting details at standalone mode (period controlled by external trigger input)

	15	14	13	12	11	10	9	8
TKBCTLn0	–	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	0	0	0	1/0	1/0	1/0

	7	6	5	4	3	2	1	0
TKBCTLn1	TKBCEn	–	–	TKBCKSn	TKBSCMn	–	TKBMDn1	TKBMDn0
	1	0	0	1/0	1/0	0	0	0

	7	6	5	4	3	2	1	0
TKBIOCn0	–	–	–	–	TKBTOLn1	TKBTOLn0	TKBTODn1	TKBTODn0
	0	0	0	0	1/0	1/0	1/0	1/0

	7	6	5	4	3	2	1	0
TKBIOCn1	–	–	–	–	–	–	TKBTOEn1	TKBTOEn0
	0	0	0	0	0	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
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)

(6) Simultaneous start/stop mode

Slave: Sample of register setting details at standalone mode (period controlled by external trigger input)

	15	14	13	12	11	10	9	8
TKBCTLn0	–	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	0	0	0	1/0	1/0	1/0

	7	6	5	4	3	2	1	0
TKBCTLn1	TKBCEn	–	–	TKBCKSn	TKBSCMn	–	TKBMDn1	TKBMDn0
	1	0	0	1/0	0	0	0	1

	7	6	5	4	3	2	1	0
TKBIOCn0	–	–	–	–	TKBTOLn1	TKBTOLn0	TKBTODn1	TKBTODn0
	0	0	0	0	1/0	1/0	1/0	1/0

	7	6	5	4	3	2	1	0
TKBIOCn1	–	–	–	–	–	–	TKBTOEn1	TKBTOEn0
	0	0	0	0	0	0	1/0	1/0

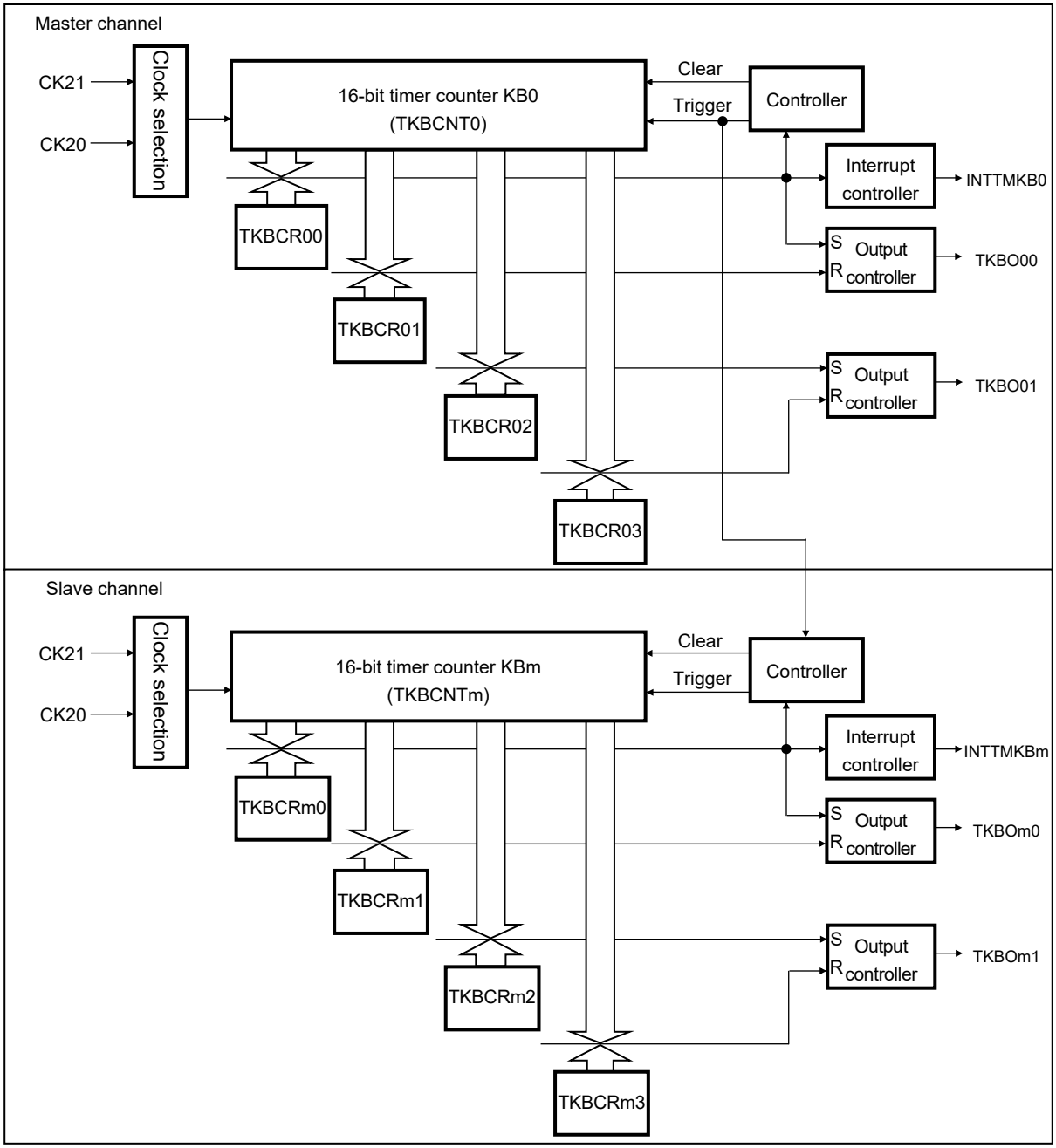
TKBCRn0	0000H to FFFFH
TKBCRn1	0000H to FFFFH
TKBCRn2	0000H to FFFFH
TKBCRn3	0000H to FFFFH
TKBTGCRn	0000H to FFFFH
TKBSIRn0	0000H
TKBSIRn1	0000H
TKBSSRn0	00H
TKBSSRn1	00H
TKBDNRn0	00H
TKBDNRn1	00H
TKBMFRn	0000H to FFFFH

□ : Setting is fixed for this mode ▒ : Setting is not needed (default setting)

(7) Configuration of simultaneous start/stop mode (period controlled by TKBCR00)

Figure 7-44 shows configuration of simultaneous start/stop mode.

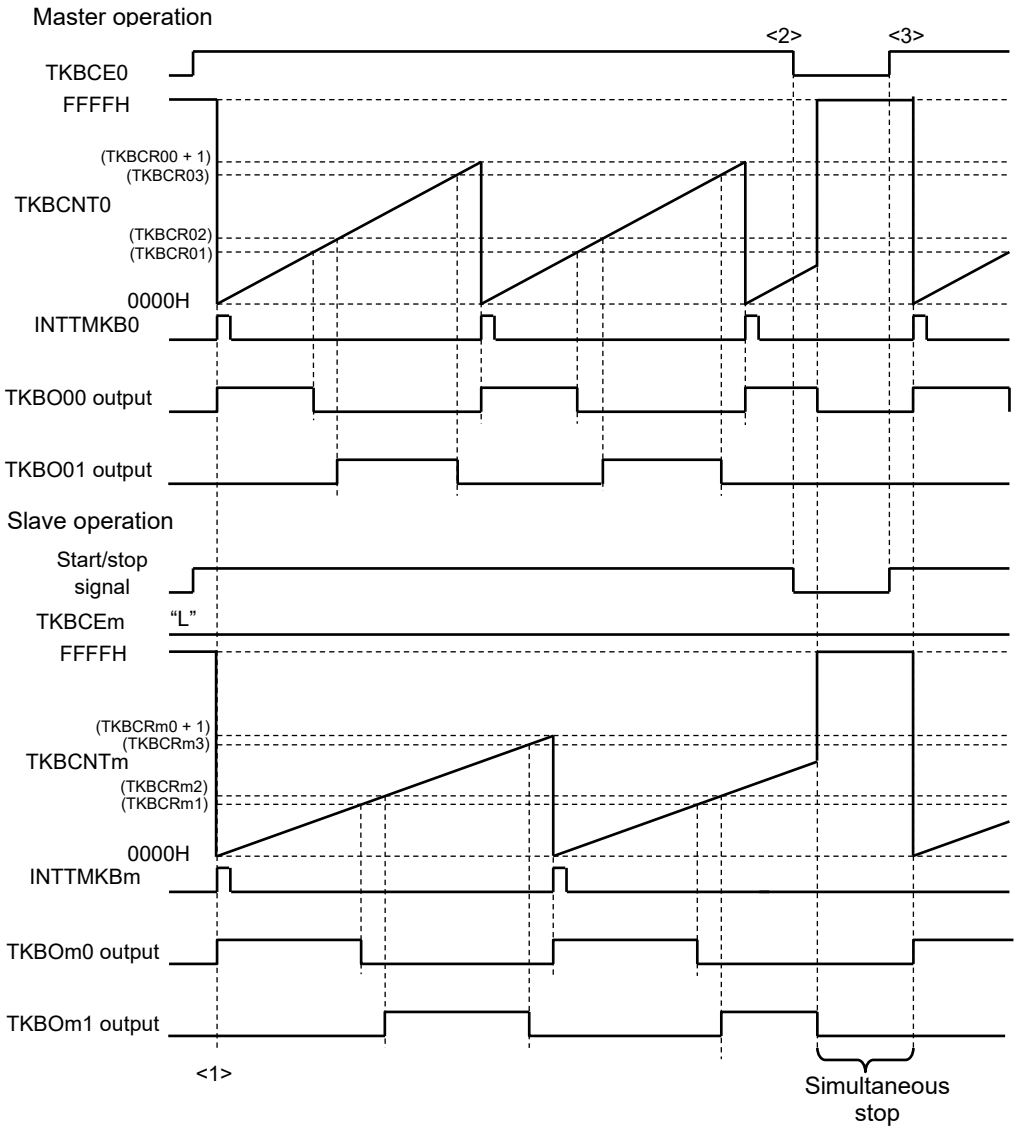
Figure 7-44. Configuration of Simultaneous Start & Stop Mode (Period Controlled by TKBCR00)



(8) Outline of operation

Figure 7-45 shows timing sample for simultaneous start & stop mode.

Figure 7-45. Timing Sample for Simultaneous Start/Stop Mode (Period Controlled by TKBCR00)
(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. The following descriptions are linked with <1> to <3> in Figure 7-45.

<1> When the master TKBCE0 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 7-38 Timing Sample for Standalone Mode (Period Controlled by TKBCRn0) (at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0)).**

<2> If TKBCE0 is set to 0, synching with the input clock of timer KB, TKBCNT0 of the master and TKBCNTm 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 the master's TKBCE0 is set as 1.

<3> If the master's TKBCE0 is set as 1, the same sequence of operation starting from <1> is repeated.

7.4.8 Synchronous start/clear mode

Slave timer KBm can be started and cleared simultaneously by synchronization with timings of start/stop of counting by KB0 Master timer, counter clearing and batch overwriting when master/slave is configured using multiple KBn timers.

Operate by “Standalone Mode (TKBMD01, TKBMD00 = 00)” for master and “Synchronous Start/Clear Mode (TKBMDm1, TKBMDm0 = 10)” for slave in such case.

Select the same division clock for Master/Slave in TKBCKS0 and TKBCKSm bits.

See **7.4.5 Standalone mode (period controlled by TKBCRn0)** for the calculation of master TKBO00/TKBO01 output.

Batch overwriting is controlled by writing “1” to master TKBRDT0.

Verifying of master TKBRSF0 is needed to read TKBRSFn flag.

Slave TKBCNTm is cleared at the same timing for Master TKBCNT0 clearing.

Batch overwriting for Slave compare register is executed at the same timing for master batch overwriting.

The role of slave TKBCRm0 is shifted to register which sets TKBOm0 active timing as slave operates according to the period generated by master TKBCR00.

INTTMKBm is generated when matching with TKBCNTm and TKBCRm0 is detected. Although INTTMKBm for the timing to start counting operation is not output.

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 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

(1) Operation mode combination available for synchronous start/clear mode

Shows the operation mode available for synchronous start/clear mode.

Master:

Operation Mode	TKBMD01, TKBMD00	TKBSTS01, TKBSTS00	Setting Available
Standalone mode (period controlled by TKBCR00)	00B	00B	O
Standalone mode (period controlled by external trigger input)	00B	01B/10B/11B	×
Simultaneous start/stop mode (period controlled by TKBCR00)	01B	00B	×
Simultaneous start/stop mode (period controlled by external trigger input)	01B	01B/10B/11B	×
Synchronous start/clear mode (period controlled by master)	10B	–	×
Interleave PFC output mode	11B	–	×

Slave:

Operation Mode	TKBMDm1, TKBMDm0	TKBSTSm1, TKBSTSm0	Setting Available
Standalone mode (period controlled by TKBCRm0)	00B	00B	×
Standalone mode (period controlled by external trigger input)	00B	01B/10B/11B	×
Simultaneous start/stop mode (period controlled by TKBCRm0)	01B	00B	×
Simultaneous start/stop mode (period controlled by external trigger input)	01B	01B/10B/11B	×
Synchronous start/clear mode (period controlled by master)	10B	–	O
Interleave PFC output mode	11B	–	×

(2) Synchronous start/clear mode: List of register setting by master

bit No.	15	14	13	12	11	10	9	8
TKBCTL00	–	TKBGTE01	TKBSSE01	TKBDIE01	–	TKBGTE00	TKBSSE00	TKBDIE00
Setting	0	1/0	1/0	1/0	0	1/0	1/0	1/0

bit No.	7	6	5	4	3	2	1	0
TKBCTL00	TKBMFE0	–	TKBIRS01	TKBIRS00	–	TKBTSE0	TKBSTS01	TKBSTS00
Setting	0	0	0	0	0	0	0	0

bit No.	7	6	5	4	3	2	1	0
TKBCTL01	TKBCE0	–	–	TKBCKS0	TKBSCM0	–	TKBMD01	TKBMD00
Setting	1	0	0	1/0	0	0	0	0

bit No.	7	6	5	4	3	2	1	0
TKBIOC00	–	–	–	–	TKBTOL01	TKBTOL00	TKBTOD01	TKBTOD00
Setting	0	0	0	0	1/0	1/0	1/0	1/0

bit No.	7	6	5	4	3	2	1	0
TKBIOC01	–	–	–	–	–	–	TKBTOE01	TKBTOE00
Setting	0	0	0	0	0	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

□ : Setting is fixed for this mode ■ : Setting is not needed (default setting)

(3) Simultaneous start/clear mode: List of register setting by slave

bit No.	15	14	13	12	11	10	9	8
TKBCTLm0	–	TKBGTEm1	TKBSSEm1	TKBDIEm1	–	TKBGTEm0	TKBSSEm0	TKBDIEm0
Setting	0	1/0	1/0	1/0	0	1/0	1/0	1/0

bit No.	7	6	5	4	3	2	1	0
TKBCTLm0	TKBMFEm	–	TKBIRSm1	TKBIRSm0	–	TKBTSEm	TKBSTSm1	TKBSTSm0
Setting	0	0	0	0	0	0	0	0

bit No.	7	6	5	4	3	2	1	0
TKBCTLm1	TKBCEm	–	–	TKBCKSm	TKBSCMm	–	TKBMDm1	TKBMDm0
Setting	1	0	0	1/0	0	0	1	0

bit No.	7	6	5	4	3	2	1	0
TKBIOCm0	–	–	–	–	TKBTOLm1	TKBTOLm0	TKBTODm1	TKBTODm0
Setting	0	0	0	0	1/0	1/0	1/0	1/0

bit No.	7	6	5	4	3	2	1	0
TKBIOCm1	-	–	–	–	–	–	TKBTOEm1	TKBTOEm0
Setting	0	0	0	0	0	0	1/0	1/0

TKBCRm0	0000H to FFFFH
TKBCRm1	0000H to FFFFH
TKBCRm2	0000H to FFFFH
TKBCRm3	0000H to FFFFH
TKBTGCRm	0000H to FFFFH
TKBSIRm0	0000H to FFFFH
TKBSIRm1	0000H to FFFFH
TKBSSRm0	00H to 0FH
TKBSSRm1	00H to 0FH
TKBDNRm0	00H to F0H
TKBDNRm1	00H to F0H
TKBMFRm	0000H

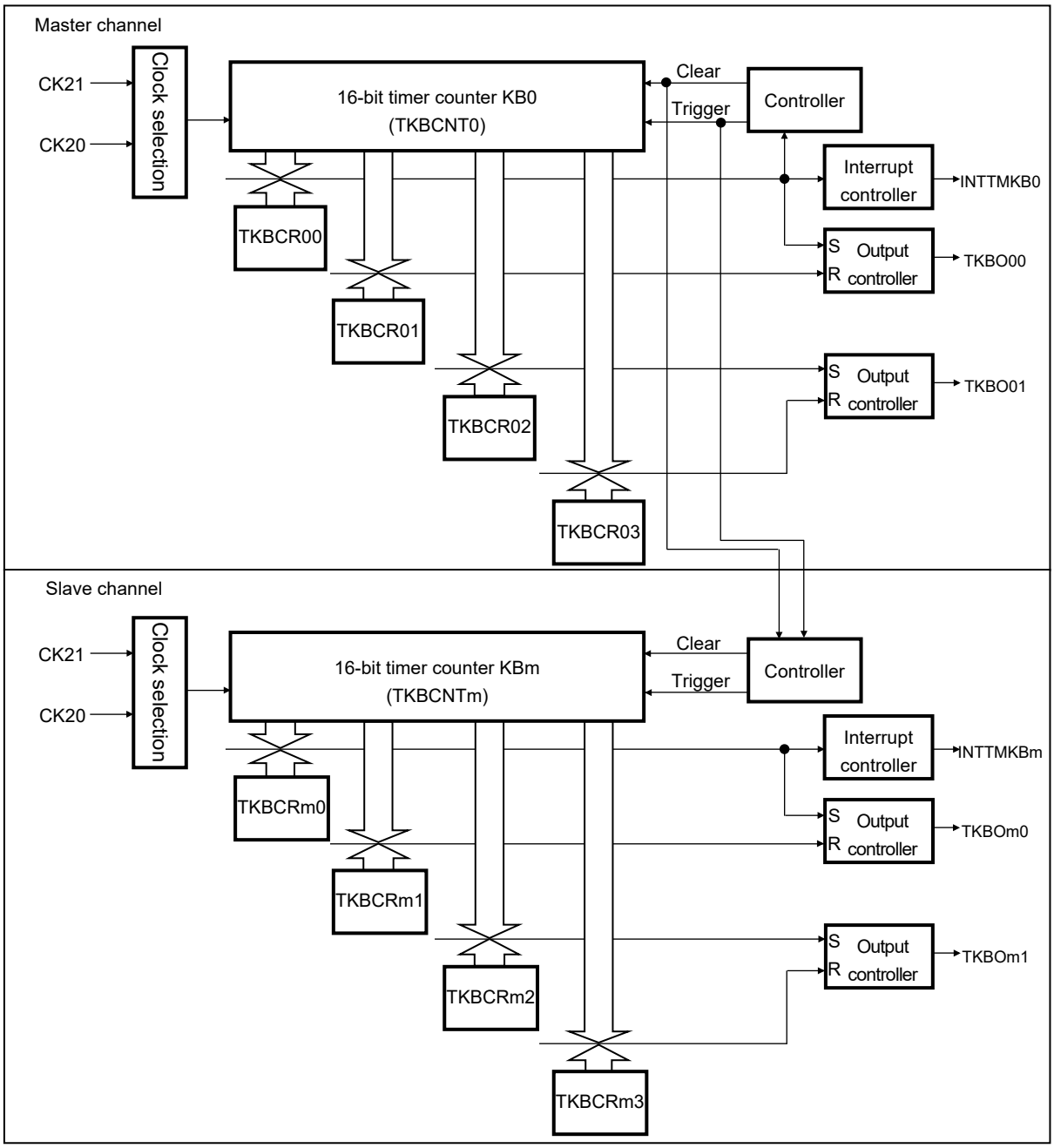
□ : Setting is fixed for this mode ■ : Setting is not needed (default setting)

Remark m = 1, 2

(4) Configuration of synchronous start/clear mode (period controlled by master)

Figure 7-46 shows configuration of synchronous start/clear mode.

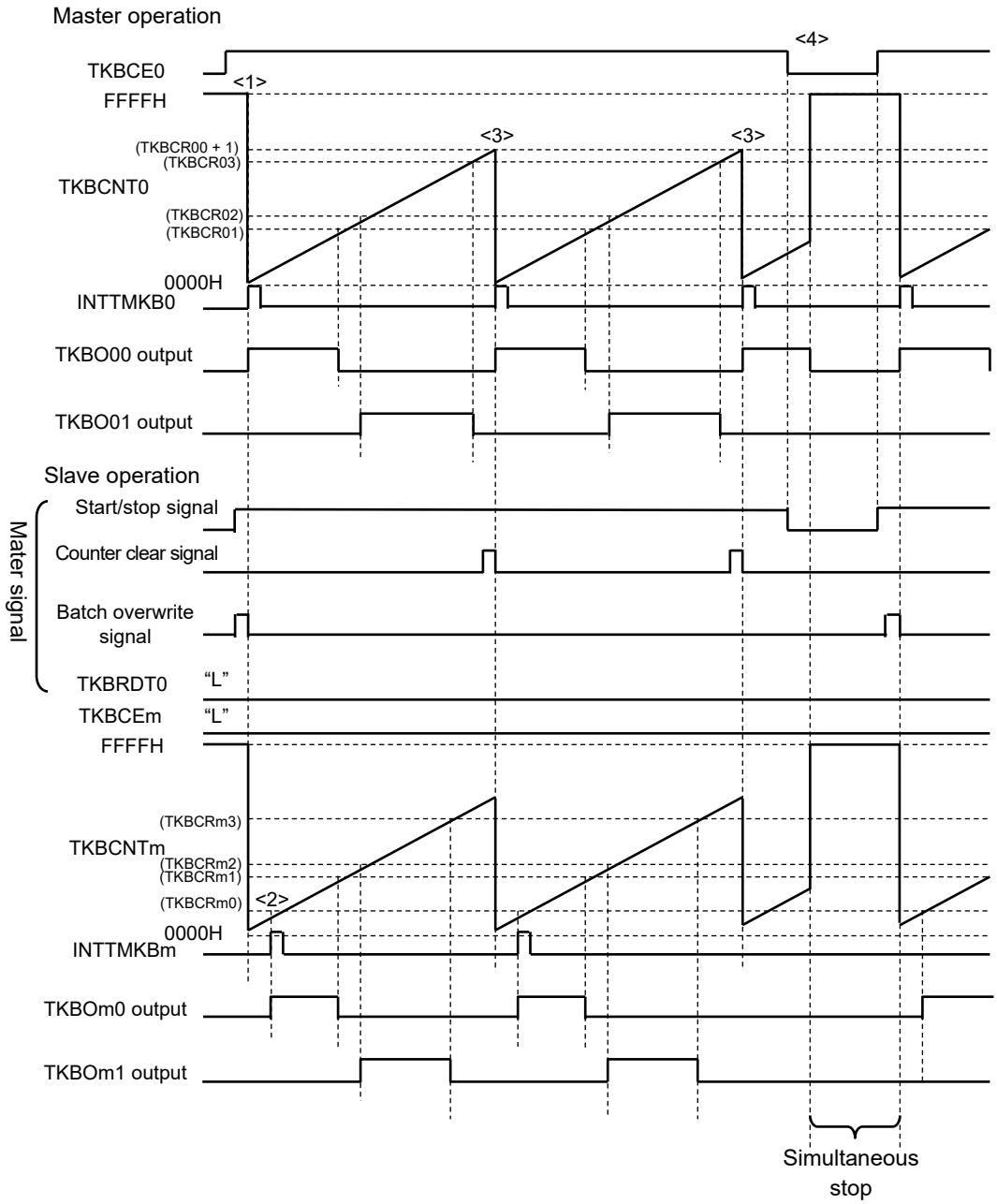
Figure 7-46. Configuration of Synchronous Start/Clear Mode (Period Controlled by Master)



(5) Outline of operation

Figure 7-47 shows timing sample for Synchronous start/clear mode.

Figure 7-47. Timing Sample for Synchronous 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 synchronous start/clear mode. The following descriptions are linked with <1> to <4> in Figure 7-47.

<1> When the master TKBCE0 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).

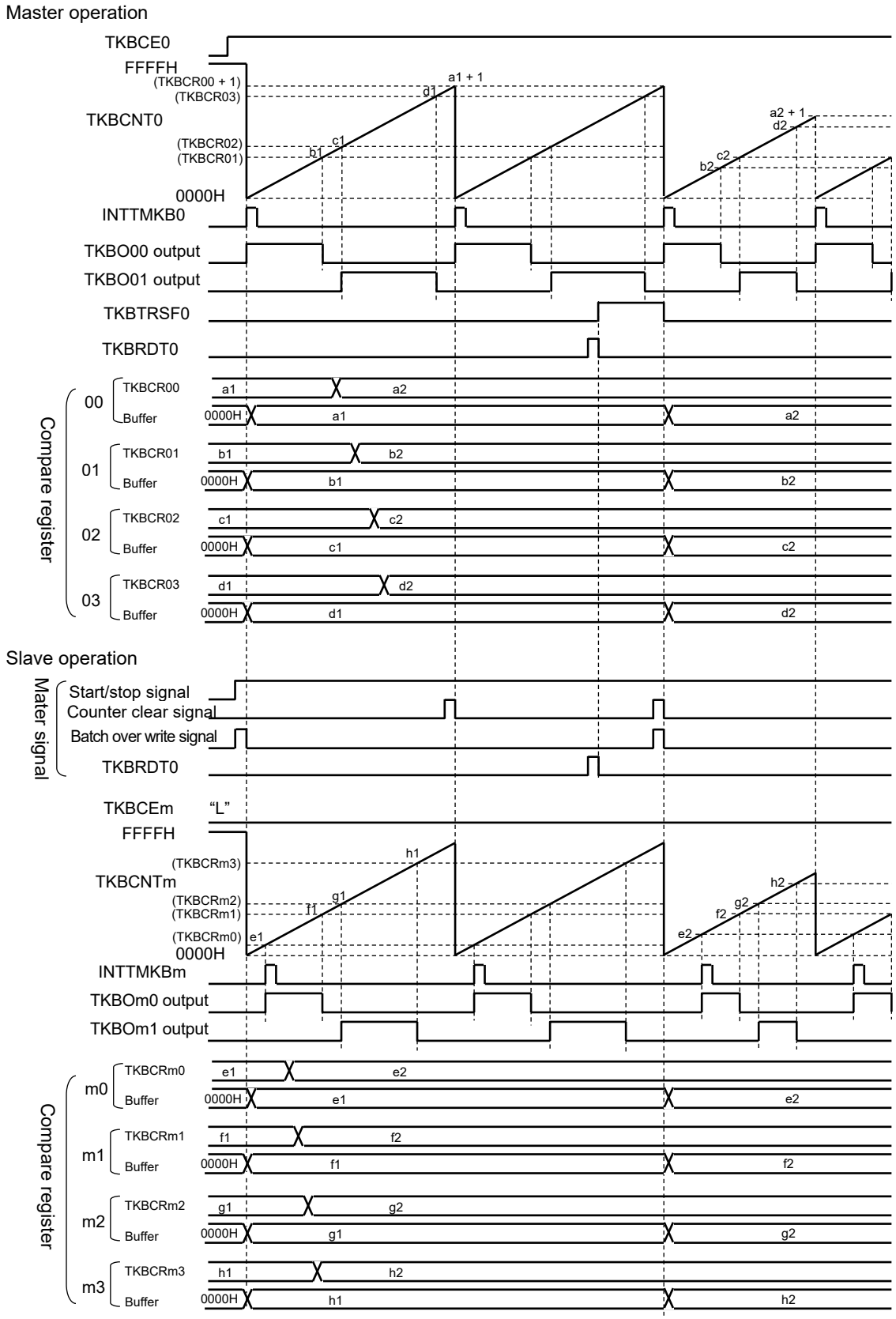
<2> When count value of TKBCNTm matches with the value specified in TMKB compare register n2 (TKBCRn2), TKBOn1 output of slave becomes active level. For further detailed operation, see **Figure 7-38 Timing Sample for Standalone Mode (Period Controlled by TKBCRn0) (at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0)).**

<3> When count value of TKBCNT0 matches with the value specified in TMKB compare register 00 (TKBCR00), clear signal of master is output. At the same time, 16-bit timer counter (TKBCCNT0, TKBCNTm) for master and slave is cleared.

<4> If TKBCE0 is set to 0, synching with the input clock of timer KB, TKBCNT0 of the master and TKBCNTm 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 the master's TKBCE0 is set as 1.

Figure 7-48 shows the operation timing sample when batch overwriting in synchronous 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 7-48. Timing Sample for Synchronous Start/Clear Mode (Period Controlled by Master) (at Batch Overwrite)



7.4.9 Interleave PFC (Power Factor Correction) output mode

This is the mode that can generate a signal as interleave 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.

Interleaved PFC control requires two inputs for zero current detection and two PWM outputs for switching. TMKB implements 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 standalone mode (periodic control by external input trigger). For more detail, see **7.4.6 Standalone mode (periodic control by external trigger input)**.

Timer KBn restart period is set by TKBCRn0 for cases in which external input INTP20 not being detected.

Active width for TKBOn0 output is set by TKBCRn1.

Active width for TKBOn1 output is set by TKBCRn3.

Therefore, TKBCRn2 is not used for this function.

Remark Interleave PFC (Power Factor Correction) 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

Notes 1. This is the timer KBn restart period in case when external interrupt input INTP20 not being detected.

2. Except when condition No. 7 holds.

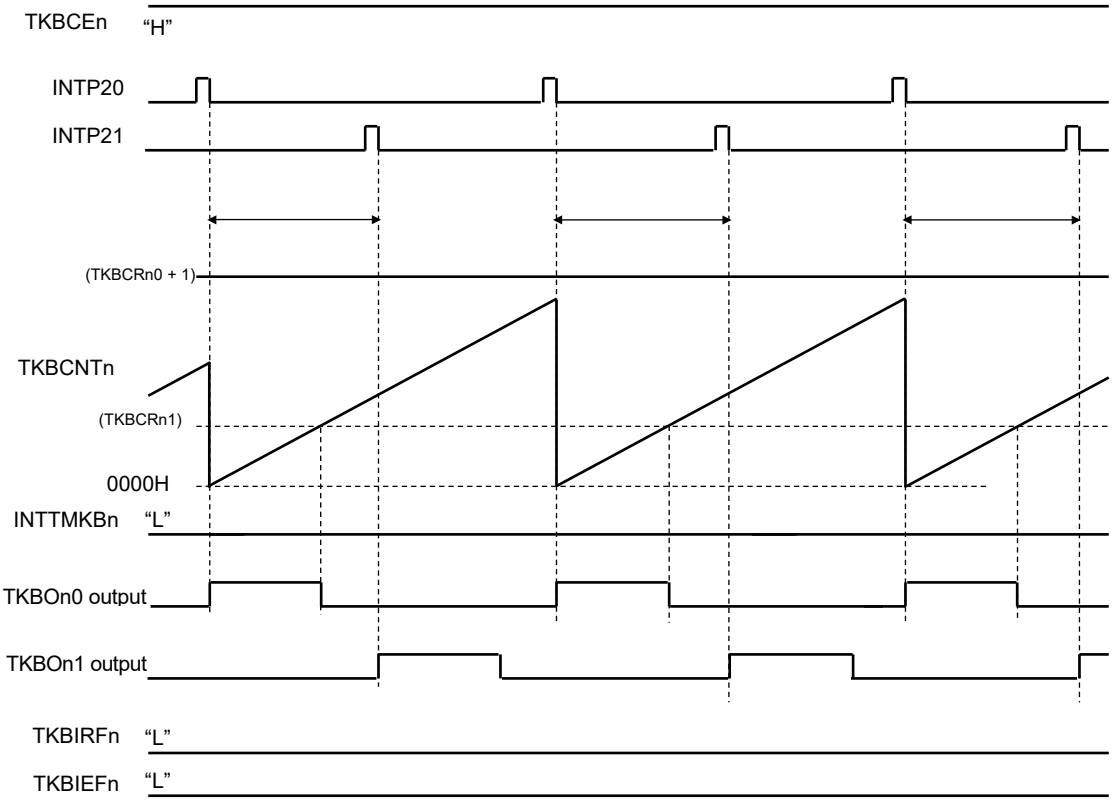
3. When condition No. 1 holds, this is determined by the setting of TKBCRn0.

Figure 7-49 shows overview of the basic operation of the interleave PFC mode. In the basic operation of the interleave PFC mode, TKBCNTn 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 TKBCNTn matches with the setting value of TKBCRn0 register, then the above operation is repeated.

Figure 7-49. Operation Outline of Basic Operation for Interleave PFC Mode
(at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



(1) Output condition of TKBOn1 at interleave PFC

There are output conditions for TKBOn1 output which are controlled according to the table below.

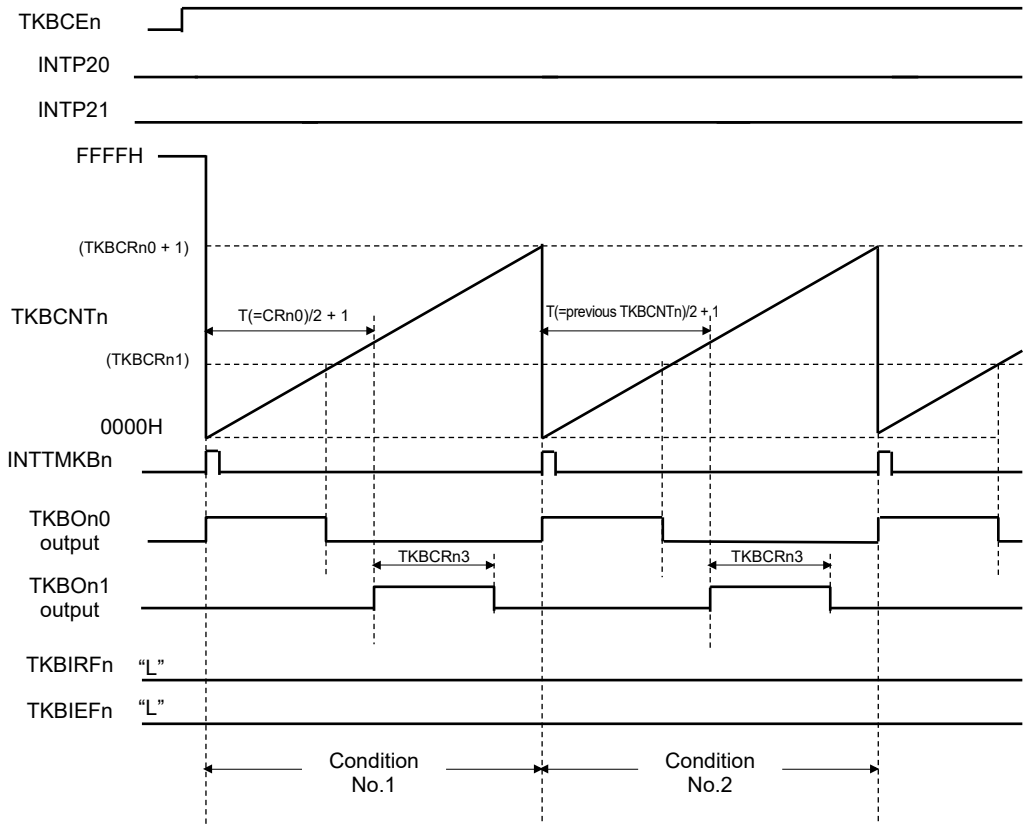
Condition No.	Judgment Status 1 (INTP20 Input)	Judgment Status 2 (Matching with CR0/INTP21 Input)	Judgment Status 3 (Period Width)	Output Status
1	First period (Generate a wave form setting T for CR0)	–	–	Output by T/2
2	INTP20 input not detected	matching of CNTn and CRn0 (Ignore INTP21 input detection)	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)*1	–	–	Output by T/2
6	INTP20 input detected (from the second time)*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)*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)*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

*1 INTP20 input detected (for the first time) means that the previous period wasn't cleared for INTP20 input being detected.

*2 INTP20 input detected (from the second time) means that the previous period being cleared for INTP20 input being detected.

Please see the following figures of wave form corresponding to each “Condition No.”

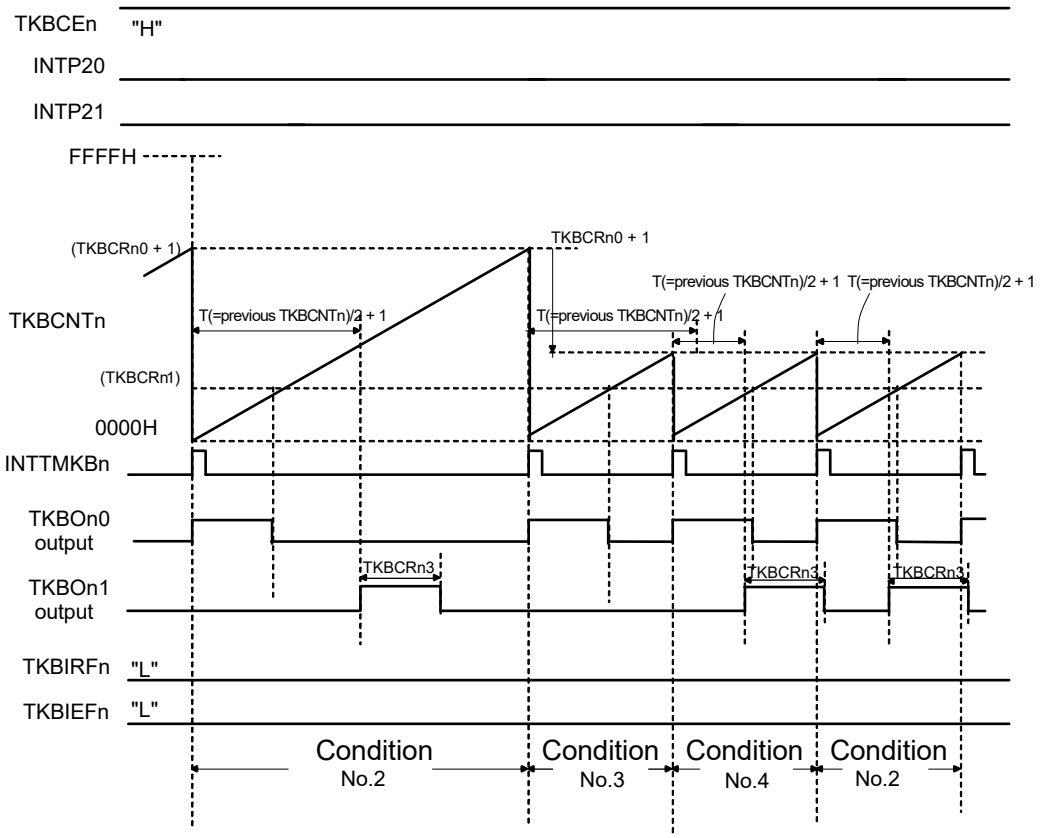
Figure 7-50. Figure of Timing of Interleave PFC Mode (Operation for Conditions No. 1 and No. 2)
(at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



Condition No.1 Only for the first period after TKBCEn = 1 setting, TKBOn1 with setting width of TKBCRn3 is output setting "T" as TKBCRn0.

Condition No.2 In the second period, TKBOn1 with setting width of TKBCRn3 is output at T/2 of previous period.

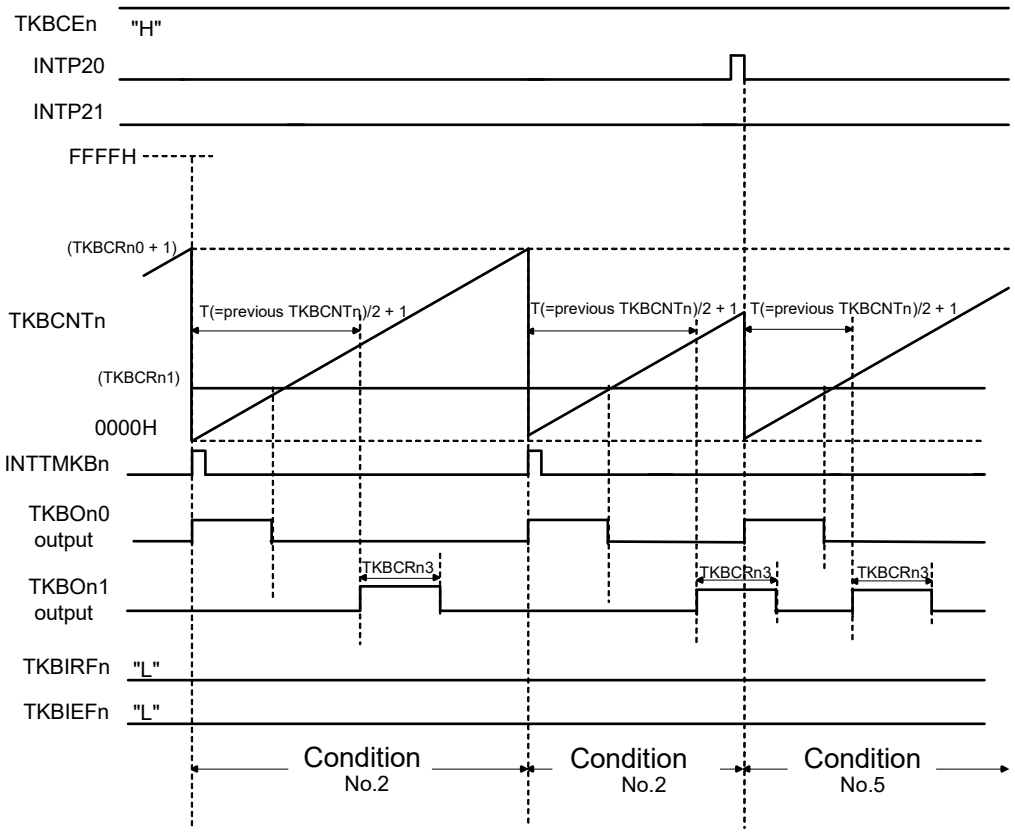
Figure 7-51. Figure of Timing of Interleave PFC Mode (Below T/2s No. 3 and No. 4)
(at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



Condition No.3 TKBOn1 keeps the status and T/2 of the previous period not ensured.

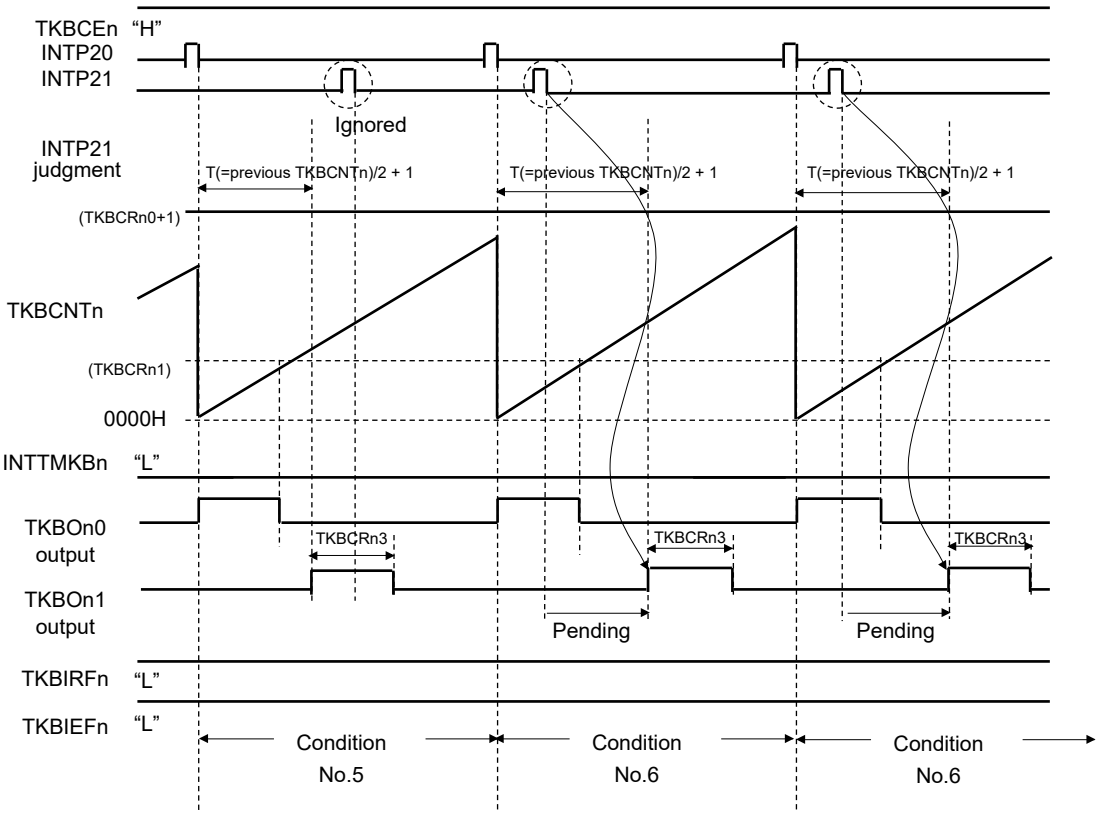
Condition No.4 TKBOn1 with setting width of TKBCRn3 is output at T/2 of previous period.

Figure 7-52. Figure of Timing of Interleave 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))



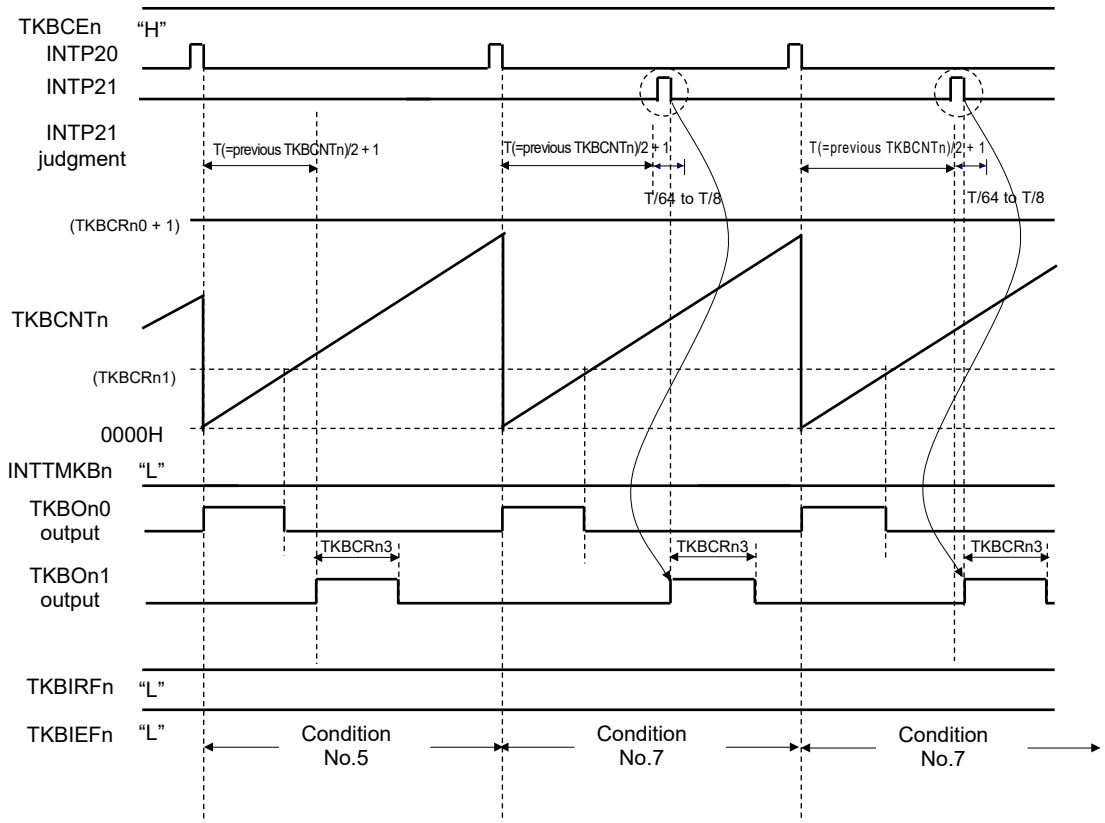
Condition No.5 INTP20 which was first detected after setting TKBCEn = 1 outputs TKBOn1 with setting width of TKBCRn3.

Figure 7-53. Figure of Timing of Interleave PFC Mode (Operation for Conditions No. 6)
(at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



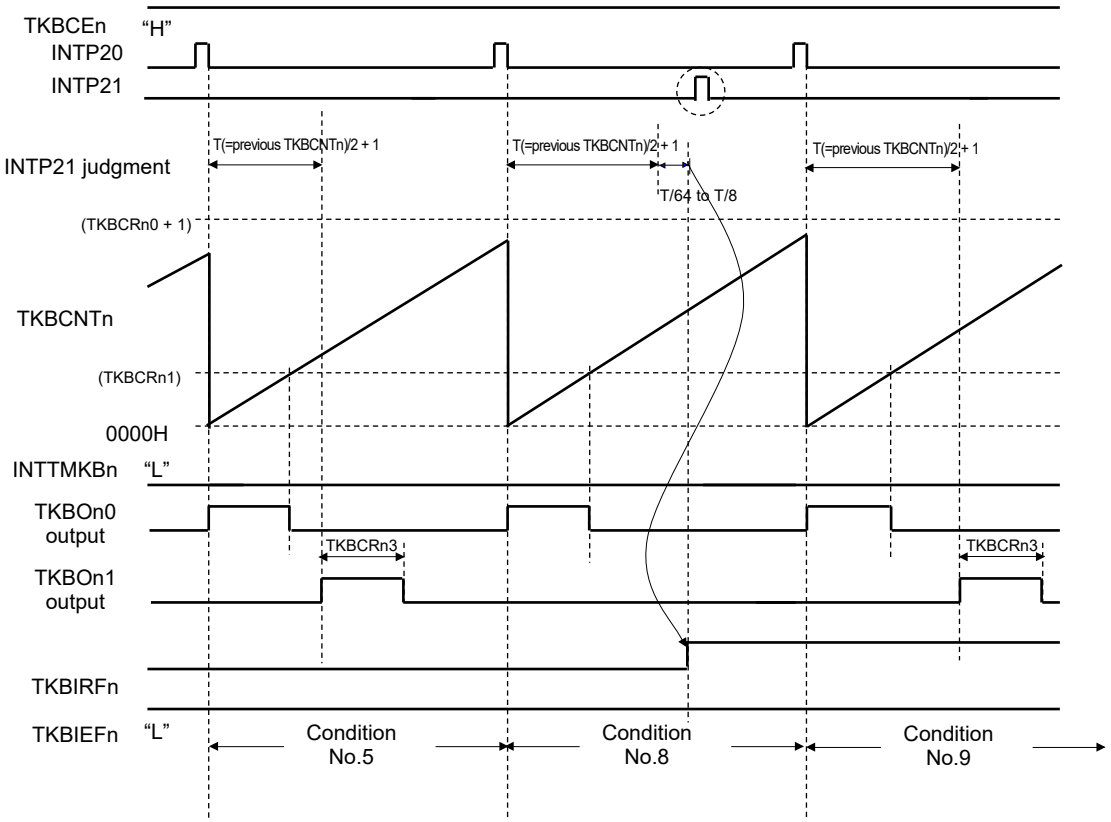
Condition No.6 TKBOn1 with setting width of TKBCRn3 is output at T/2 of previous period as INTP21 input is below T/2 of the previous period.

Figure 7-54. Figure of Timing of Interleave PFC Output Mode (Operation for Conditions 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 TKBIRSn1, TKBIRSn0), TKBOn1 is output by setting width of TKBCRn3.

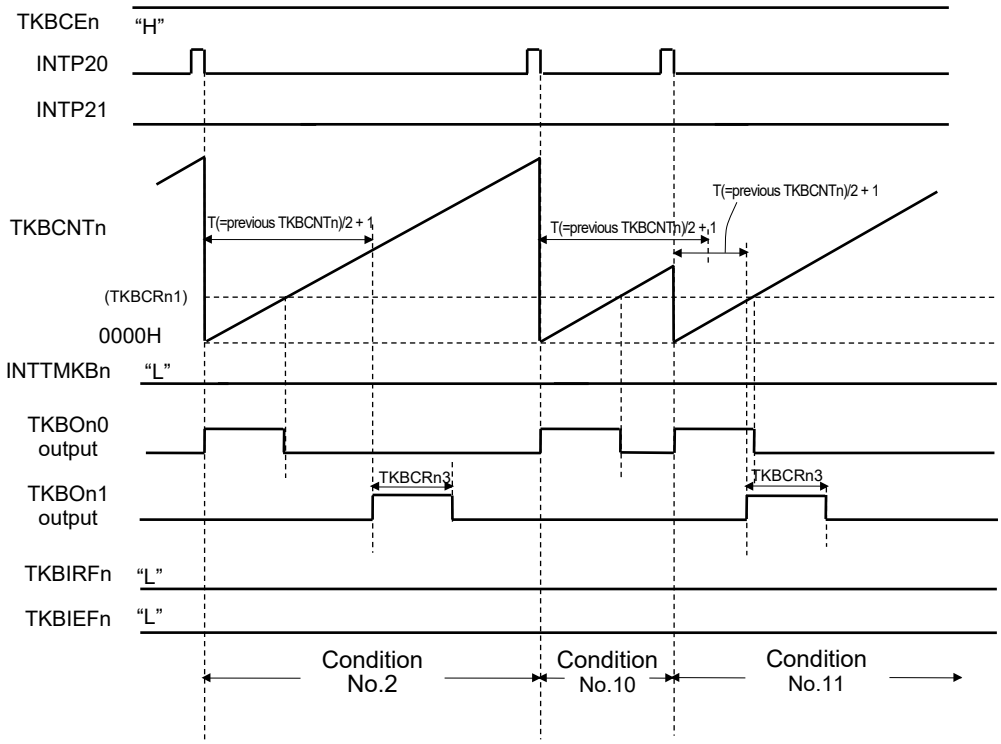
Figure 7-55. Figure of Timing of Interleave PFC Output Mode (Operation for Conditions No. 8 to 9)
(at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



Condition No.8 INTP21 wasn't detected within T/2 + T/m (m stands for 8/16/32/64; set at TKBIRSn1 & TKBIRSn0) of the previous period and TKBOn1 maintains the status. Then TKBIRFn is set by "1".

Condition No.9 TKBOn1 with setting width of TKBCRn3 is output at T/2 of previous period.

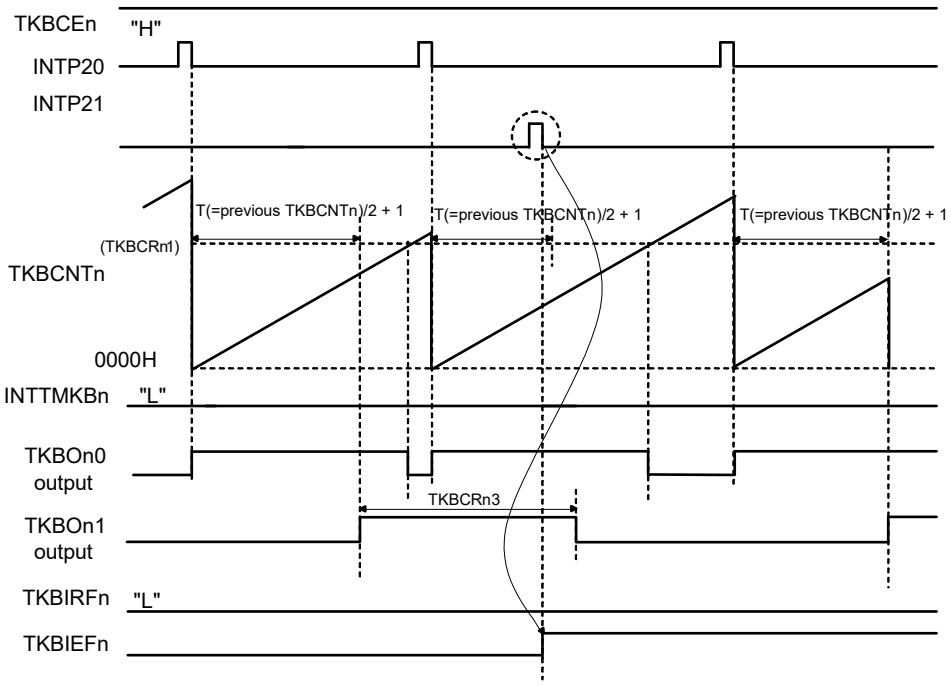
Figure 7-56. Figure of Timing of Interleave PFC Output Mode (Operation for Conditions No. 10 and No. 11) (at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



Condition No.10 TKBOn1 keeps the status and T/2 of the previous period not ensured.

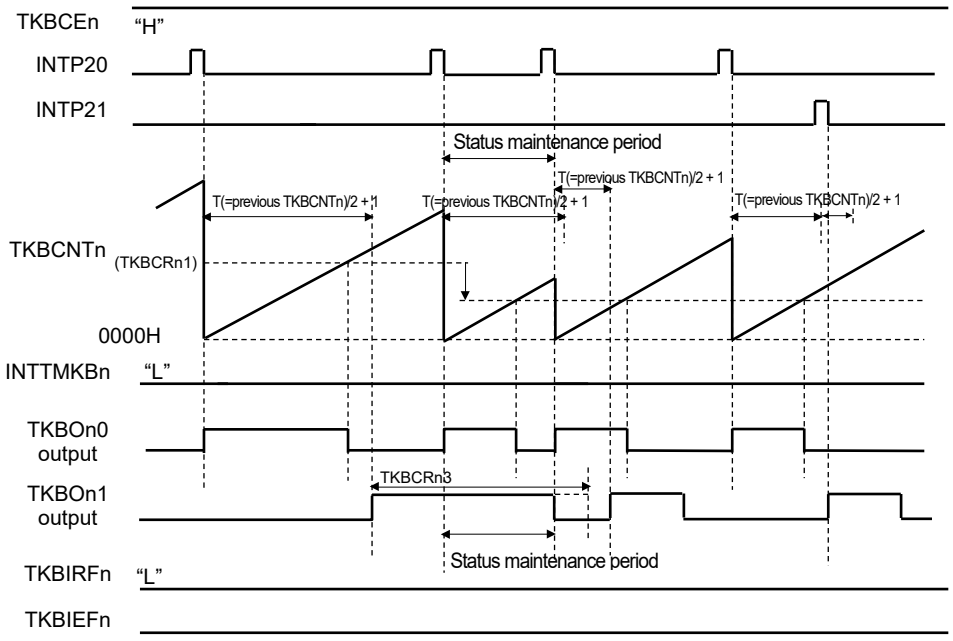
Condition No.11 TKBOn1 with setting width of TKBCRn3 is output at T/2 of previous period.

Figure 7-57. Figure of Timing of Interleave PFC Output 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 TKBIEFn is set by “1”.

Figure 7-58. Figure of Timing of Interleave PFC Output 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.

(2) List of register setting at interleave PFC output mode

bit No.	15	14	13	12	11	10	9	8
TKBCTLn0	–	TKBGTEn1	TKBSSEn1	TKBDIEn1	–	TKBGTEn0	TKBSSEn0	TKBDIEn0
Setting	0	0	0	0	0	0	0	0

bit No.	7	6	5	4	3	2	1	0
TKBCTLn0	TKBMFEn	–	TKBIRSn1	TKBIRSn0	–	TKBTSEn	TKBSTSn1	TKBSTSn0
Setting	1/0	0	1/0	1/0	0	1	0	0

bit No.	7	6	5	4	3	2	1	0
TKBCTLn1	TKBCEn	–	–	TKBCKSn	TKBSCMn	–	TKBMDn1	TKBMDn0
Setting	1	0	0	1/0	0	0	1	1

bit No.	7	6	5	4	3	2	1	0
TKBIOCn0	–	–	–	–	TKBTOLn1	TKBTOLn0	TKBTODn1	TKBTODn0
Setting	0	0	0	0	1/0	1/0	1/0	1/0

bit No.	7	6	5	4	3	2	1	0
TKBIOCn1	–	–	–	–	–	–	TKBTOEn1	TKBTOEn0
Setting	0	0	0	0	0	0	1/0	1/0

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)

7.5 Option Function of 16-bit Timers KB0, KB1 and KB2

Option function can be added to timers KB0, KB1 and KB2.

The following table shows available option for each operation modes for timer KB0, KB1 and KB2.

Operation Mode		Standalone Mode		Simultaneous Start/ Stop Mode		Synchronous Start/Clear Mode	Interleave PFC Output Mode
Period Controlling Method for Operation 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 Gate Function	○	×	○	×	○	×
	Maximum Frequency Limit Function	×	○	×	○	×	○

Remark For details of the operation specifications, see **7.4.2 Default level and active level** and **7.4.3 Stop/start operation**.

7.5.1 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 KBn and A/D conversion start timing can be synchronized.

Timer KBn trigger output signal is output by detecting the match between TKBCNTn and TKBTGCRn which makes trigger output available at any timing corresponding to set period of TKBCRnm. Output width of timer KBn trigger output signal 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 = TKBTGCRn setting × Count clock period

Caution Timer KBn trigger output signal is not output when TKBCRn0 < TKBTGCRn.

Figure 7-59. A/D Conversion Start Timing Signal Output Function for Standalone Mode (Period Controlled by TKB0CR0)

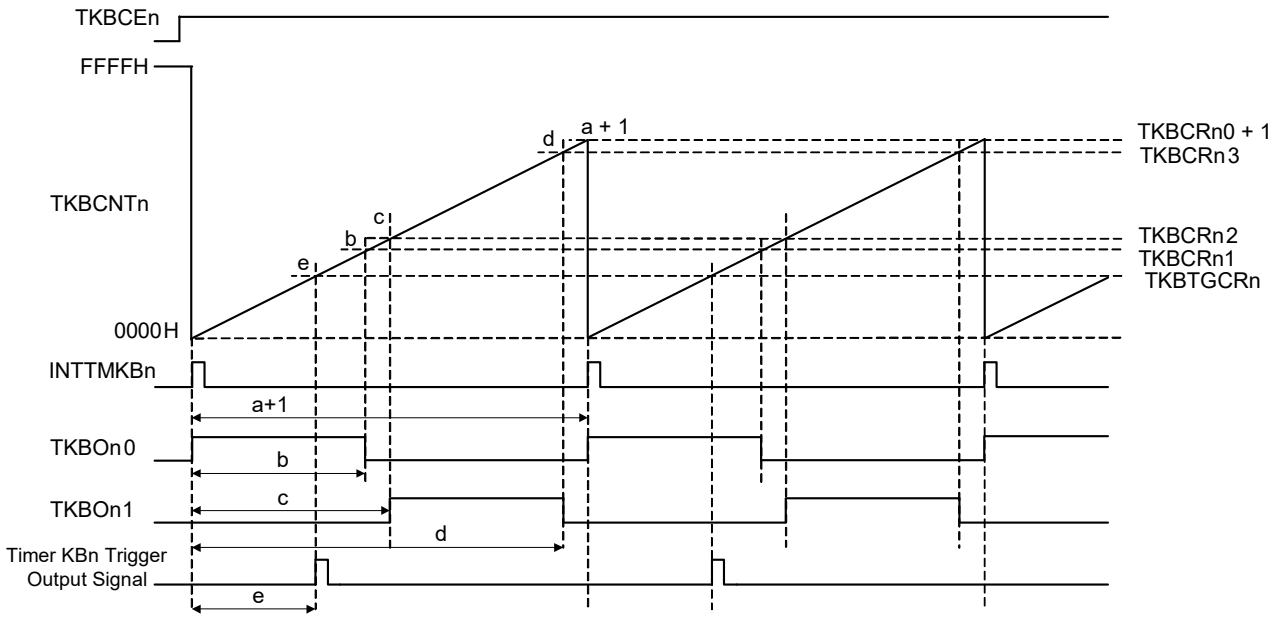
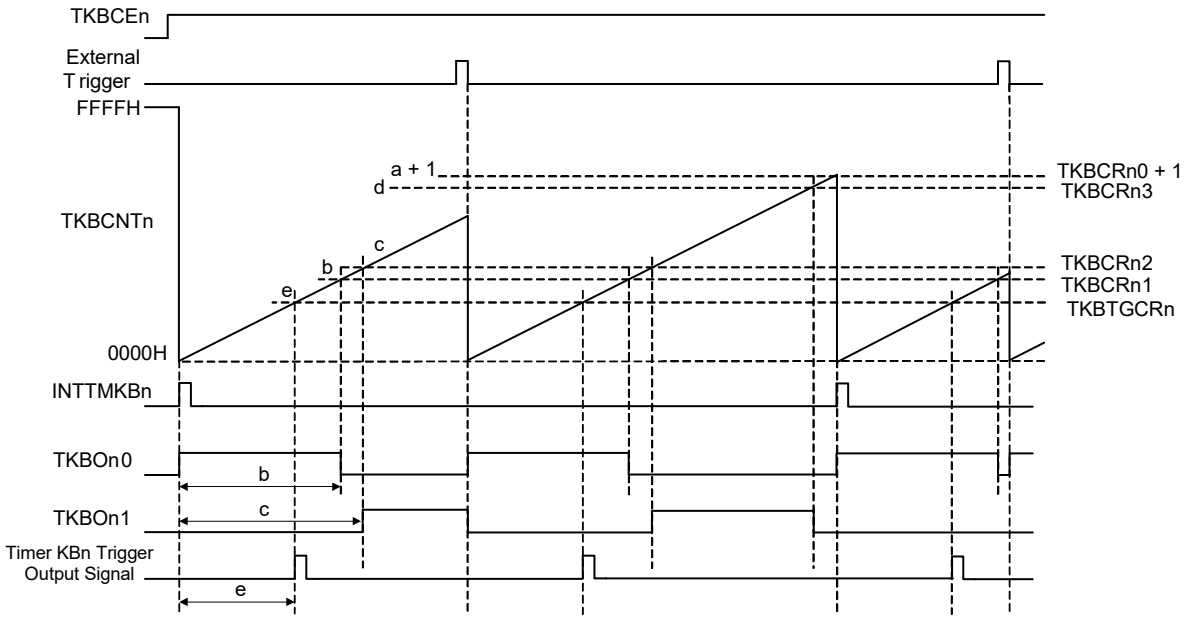


Figure 7-60. A/D Conversion Start Timing Signal Output Function for Standalone Mode (Period Controlled by External Trigger Input)



7.5.2 PWM output dithering function

16-bit timer KB is available for high resolution PWM output using 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 TKBDNRnp.

The relationship between TKBDNRnp and the period extending active period for 1 count clock is as follows:

Figure 7-61. Figure of Relationship Between TKBDNRnp and the Period which Extends Active Period for 1 Count Clock

Period / Repetitions (N)	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	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	

Remarks 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

2. n = 0 to 2, p = 0, 1

Figure 7-62. Figure of Waveform at Dithering Operation

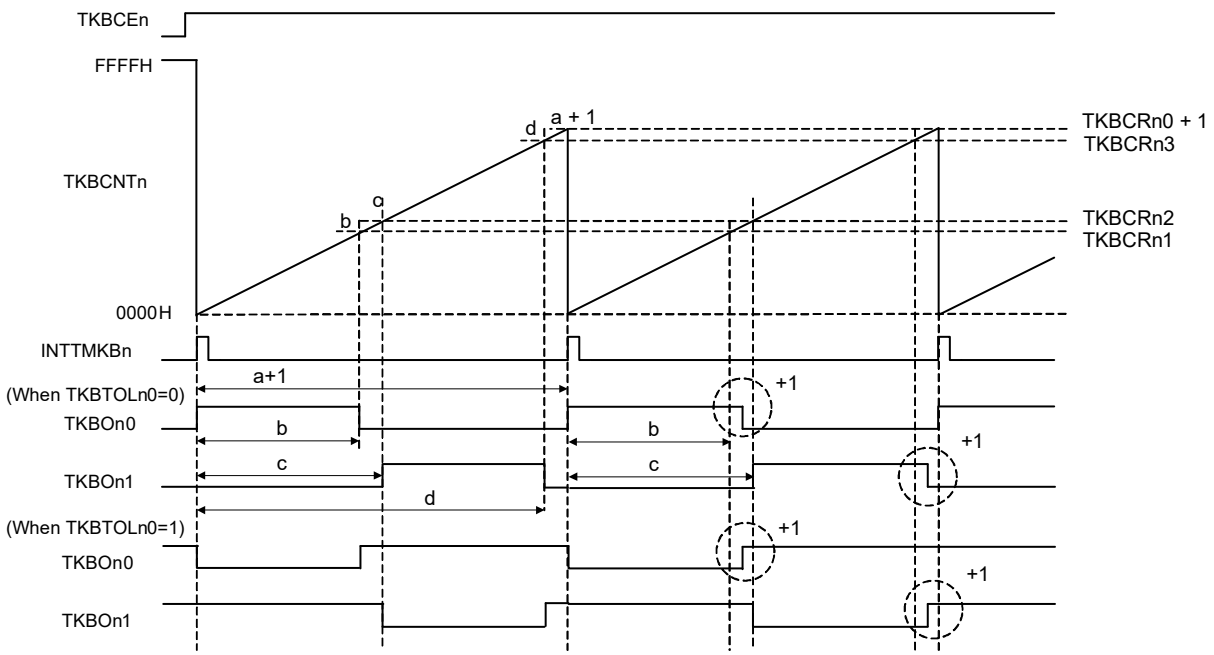


Figure 7-63. Figure of Waveform at Dithering Operation
(When TKBCRn1 = TKBCRn0 (100% Nearest Neighbor), TKBCRn2 = TKBCRn3(0% Nearest Neighbor)

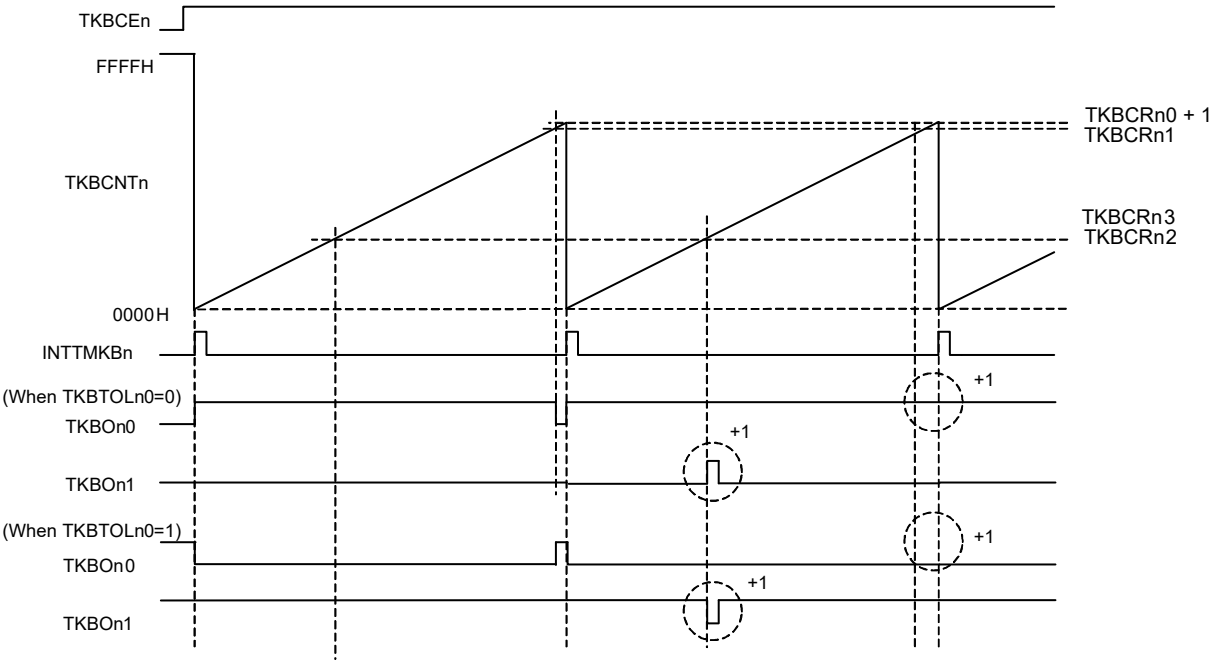
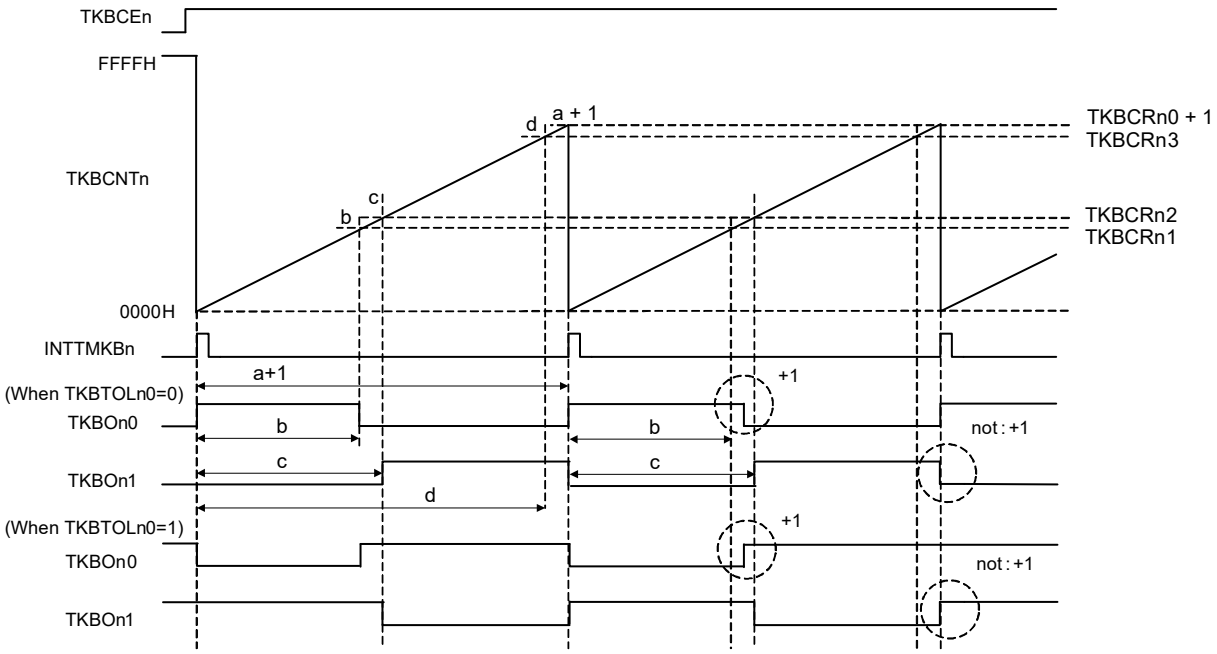


Figure 7-64. Figure of Waveform at Dithering Operation
(When TKBCRn3 = TKBCRn0+1)



(1) Available for operation mode

This shows enable or disable status under each mode that is specified by TKBCTLn0 register (TKBSTSn1 and TKBSTSn0 bit) and TKBCTLn1 register (TKBMDn1 and TKBMDn0 bit).

Operation Mode	TKBMDn1, TKBMDn0	TKBSTSn1, TKBSTSn0	Setting Available
Standalone mode (period controlled by TKBCRn0)	00	00	O
Standalone mode (period controlled by external trigger input)	00	01/10/11	×
Simultaneous start/stop mode (period controlled by TKBCRn0)	01	00	O
Simultaneous start/stop mode (period controlled by external trigger input)	01	01/10/11	×
Synchronous start/clear mode (period controlled by master)	10	-	O
Interleave PFC output mode	11	-	×

PWM output dithering function is available when external trigger input isn't used and the period being controlled by TKBCRn0.

TKBDNRn0/TKBDNRn1 control PWM output dithering function of relative TKBOn0/TKBOn1.

Cautions 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.

2. [Access by TKBCRLDn0/TKBCRLDn1 register]

TKBCRLDn0 is a 16-bit register mapping lower 8 bit TKBCRn1 and TKBDNRn0. TKBCRLDn1 is a 16-bit register mapping lower 8 bit TKBCRn3 and TKBDNRn1.

Value of TKBDNRn0/TKBDNRn1 is changed even in case that they have accessed TKBCRLDn0/TKBCRLDn1 register.

Value of TKBCRn1/TKBCRn3 is changed even in case that they have accessed TKBCRLDn0/TKBCRLDn1 register.

Only the lower 8 bit of TKBCRn1/TKBCRn3 is changed when it is accessed to TKBCRLDn0/TKBCRLDn1 register.

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).

7.5.3 PWM output smooth start function

Timer KB0, KB1 and KB2 own PMW output smooth start function corresponding to rush current control and over-voltage prevention. PWM output smooth start function begins at timer start timing. PWM output smooth start function is initiated by the timer start timing. 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 cancelled when the same active period defined by TKBCRn1 and TKBCRn3 is reached.

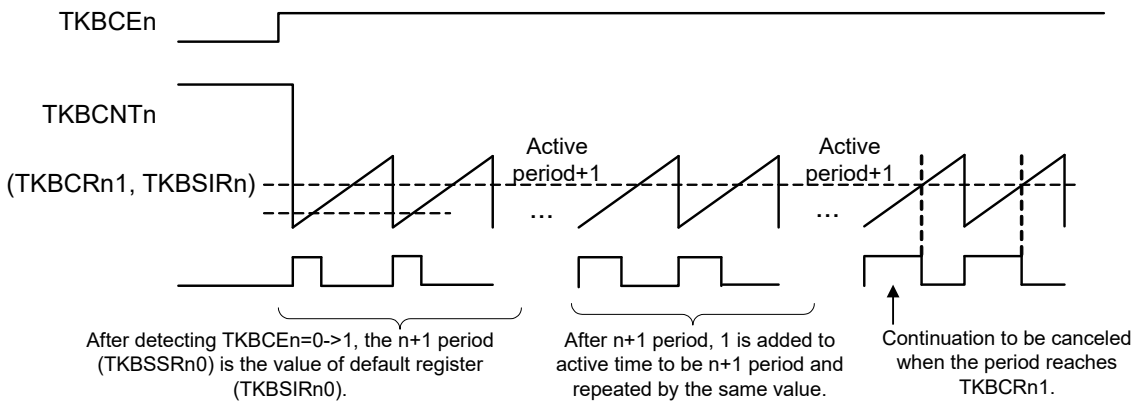
16-bit timer KB smooth start default duty register should be set according to following condition;

0000H ≤ TKBSIRn0 < TKBCRn1 ≤ TKBCRn0 + 1
TKBCRn2 ≤ TKBSIRn1 < TKBCRn3 ≤ TKBCRn0 + 1

It should be set according to following condition when synchronous start/clear mode is applied;

TKBCRn0 ≤ TKBSIRn0 < TKBCRn1 ≤ TKBCR00 + 1 of Master

Figure 7-65. PWM Output Smooth Start Function



(1) Operation mode available for PMW output smooth start function

Operation Mode	TKBMDn1, TKBMDn0	TKBSTSn1, TKBSTSn0	Setting Available
Standalone mode (period controlled by TKBCRn0)	00B	00B	O
Standalone mode (period controlled by external trigger input)	00B	01B/10B/11B	×
Simultaneous start/stop mode (period controlled by TKBCRn0)	01B	00B	O
Simultaneous start/stop mode (period controlled by external trigger input)	01B	01B/10B/11B	×
Synchronous start/clear mode (period controlled by master)	10B	–	O
Interleave PFC output mode	11B	–	×

(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.

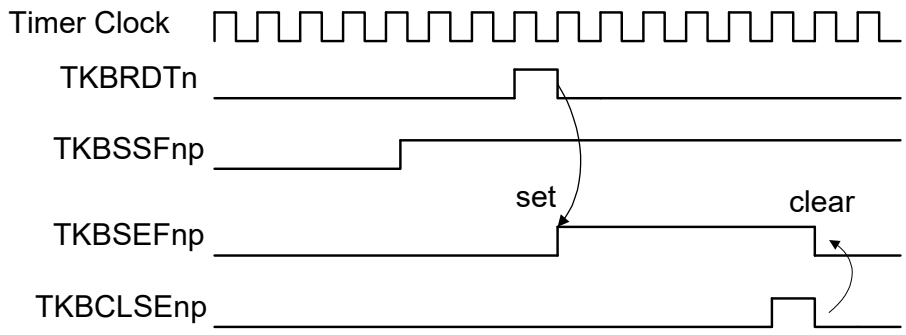
TKBSIRn0/TKBSIRn1/TKBSSRn0/TKBSSRn1 own buffer and batch overwriting is available via writing “1” to TKBRDTn bit. In TKBSIRn0/TKBSIRn1, the buffer value at starting PWM output smooth start function is duty default, and in TKBSSRn0/TKBSSRn1, it is comparison value of internal 4-bit counter.

The internal 4-bit counter is incremented upward using the period of TKBCNTn 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 TKBRDTn is set as “1” during the period of PWM output soft-start (TKBSSFn0 = 1 and TKBSSFn1 = 1), batch overwrite is masked and TKBSEFnp flag is set. In order to perform batch overwrite, clear TKBSEFnp and confirm TKBSSFnp becomes “0”, then set “1” to TKBRDTn.

Figure 7-66. 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

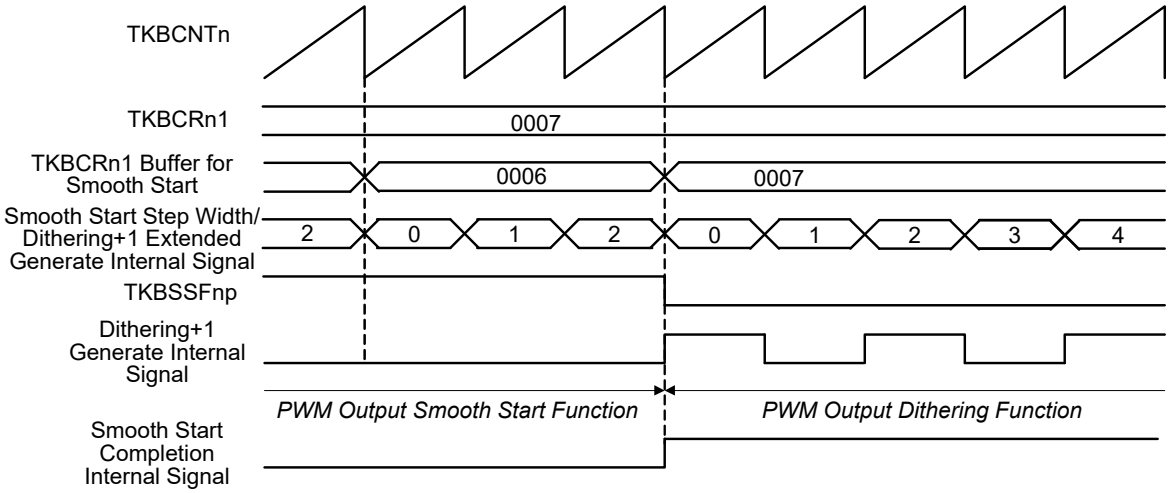
PWM output dithering function is invalid during the execution of PWM output smooth start function (TKBSSFnp = 1).

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 7-67 shows When TKBCRn1 is 0007H, TKBDNRnp is 70H and TKBSSRnp is 02H. At the timing that TKBCRn1 = 0007H and the value of TKBCRn1 buffer for internal soft-start matches, TKBSSFnp is cleared then dithering function begins.

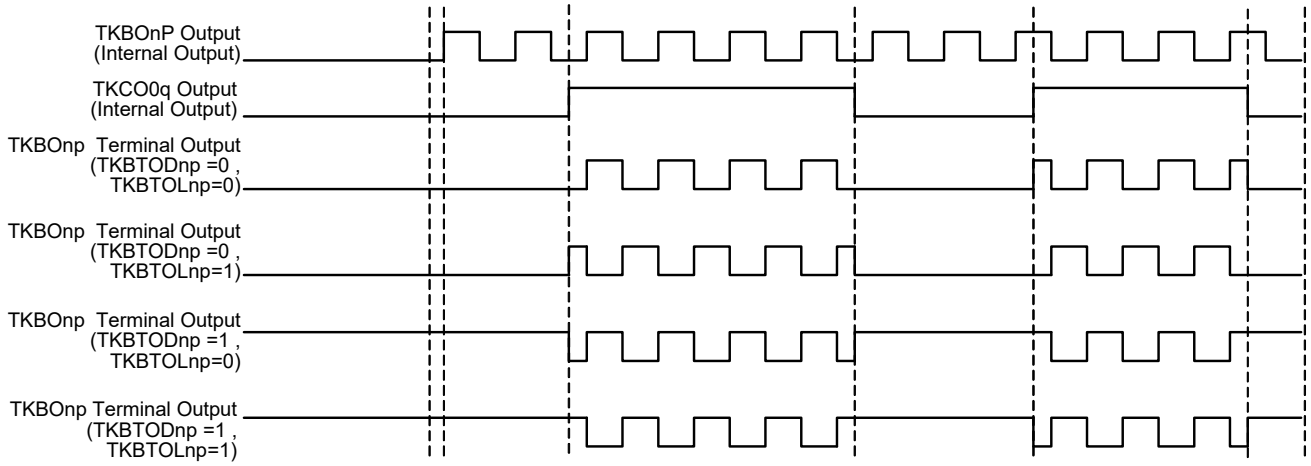
Figure 7-67. Completion of PWM Output Smooth Start Function and Operation of TKBSSFnp



7.5.4 PWM output gate function (without combining with PWM output smooth start function)

With this function, during high-level period of 16-bit timer KC0 output (TKCO00 to TKCO05), PWM pulse is output from the output terminal of TKBOnp 16-bit timer KB0 to KB2. During low-level period of 16-bit timer KC0 output (TKCO00 to TKCO05), default level is output from the output terminal of TKBOnp 16-bit timer KB0 to KB2.

Figure 7-68. PWM Output Gate Function



Corresponding relationship between TKCO0q gating output and TKBOnp output to be gated is 1:1; please refer the following.

TKCO00 : TKBO00
TKCO01 : TKBO01
TKCO02 : TKBO10
TKCO03 : TKBO11
TKCO04 : TKBO20
TKCO05 : TKBO21

(1) Operation mode available for PMW output gate function

Output gate function can be used under the following operational modes.

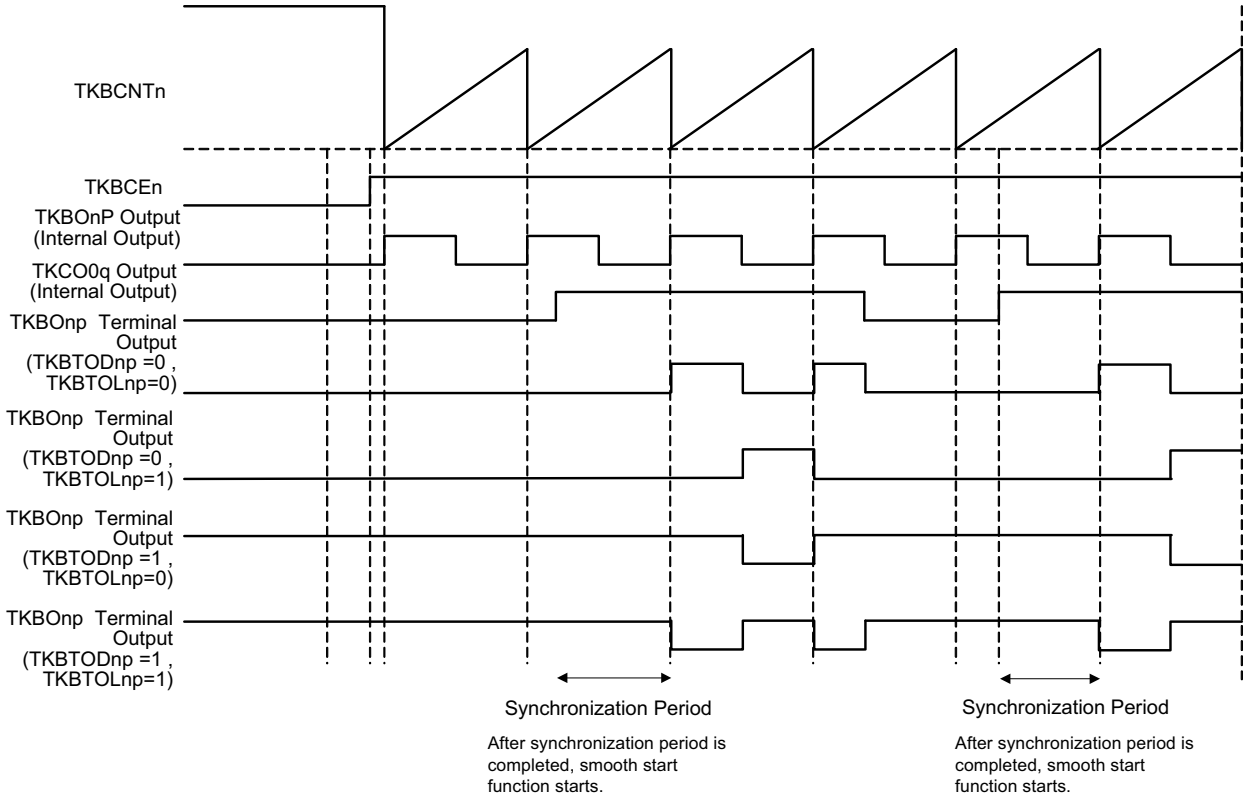
Operation Mode	TKBMDn1, TKBMDn0	TKBSTSn1, TKBSTSn0	Setting Available
Standalone mode (period controlled by TKBCRn0)	00B	00B	O
Standalone mode (period controlled by external trigger input)	00B	01B/10B/11B	×
Simultaneous start/stop mode (period controlled by TKBCRn0)	01B	00B	O
Simultaneous start/stop mode (period controlled by external trigger input)	01B	01B/10B/11B	×
Synchronous start/clear mode (period controlled by master)	10B	–	O
Interleave PFC output mode	11B	–	×

7.5.5 PWM output gate function (combining with PWM output smooth start function)

The functions PWM output gate and PWM output smooth start can be combined.

When soft-start is also used at the same time, PWM pulse is generated from TKBOnp output pin of 16-bit timers KB0 to KB2 synching with the period of 16-bit timers KB0 to KB2 after detecting rising edge of timer KC0 output (TKCO00 to TKCO05) of 16-bit timer KC. Through the detection of falling edge of timer KC0 output (TKCO00 to TKCO05) of 16-bit timer KC, default level (TKBTODnp) is output from the output terminal of TKBOnp 16-bit timer KB0 to KB2.

Figure 7-69. Sample of TKBOnp Output Synchronization Waveform at Starting TKCO0q Output of PWM Output Gate Function (Combining with PWM Output Smooth Start Function)



(1) Operation mode and additional function available for PMW output gate function

Output gate function can be used under the following operational modes.

Operation Mode	TKBMDn1, TKBMDn0	TKBSTSn1, TKBSTSn0	Setting Available
Standalone mode (period controlled by TKBCRn0)	00B	00B	O
Standalone mode (period controlled by external trigger input)	00B	01B/10B/11B	×
Simultaneous start/stop mode (period controlled by TKBCRn0)	01B	00B	O
Simultaneous start/stop mode (period controlled by external trigger input)	01B	01B/10B/11B	×
Synchronous start/clear mode (period controlled by master)	10B	–	O
Interleave PFC output mode	11B	–	×

Please see **7.5.3 PWM output smooth start function** for the details of PWM output smooth start function.

7.5.6 Maximum frequency limit function

Timers KB0, KB1 and KB2 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.

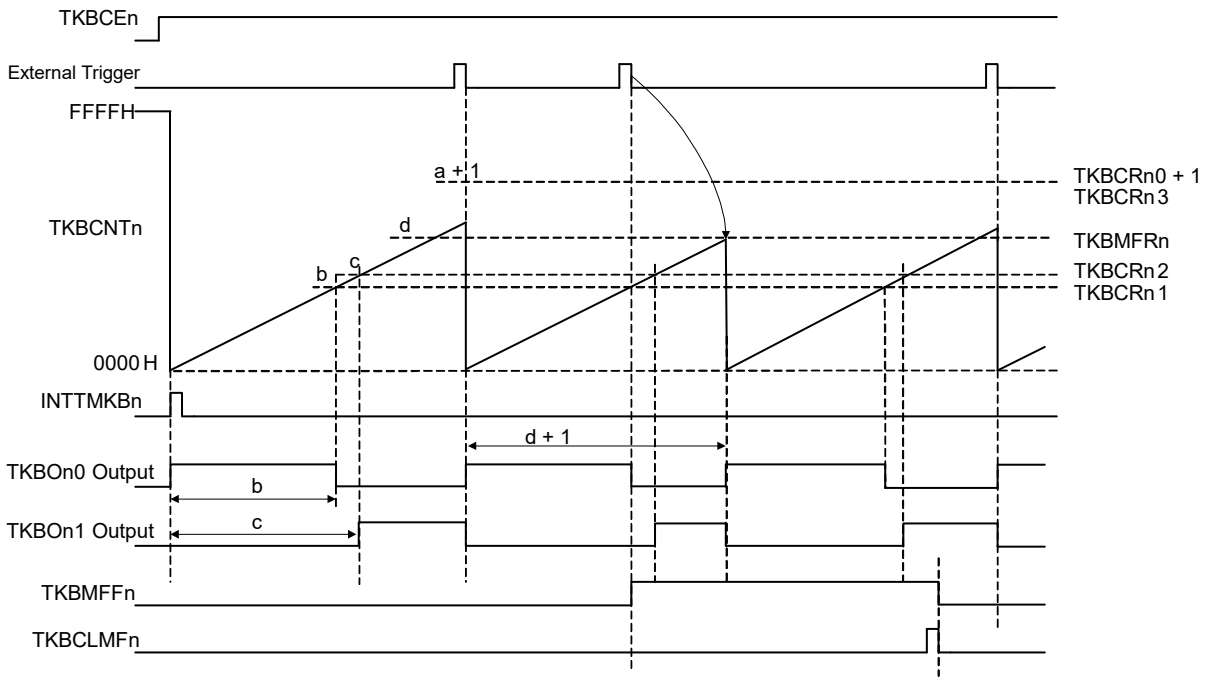
(1) Formula for maximum frequency limit (= 1/Minimum period)

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 counter value is smaller than TKBMFRn at the timing for external trigger input detection, “1” is set for TKBMFFn flag. TKBMFFn flag is cleared to “0” by writing “1” to TKBCLMFn bit.

Figure 7-70. Maximum Frequency Limit Function



Remark Period controlled by external trigger input.

(2) Operation Mode Available for Maximum Frequency Limit Function

Operation Mode	TKBMDn1, TKBMDn0	TKBSTSn1, TKBSTSn0	Setting Available
Standalone mode (period controlled by TKBCRn0)	00B	00B	×
Standalone mode (period controlled by external trigger input)	00B	01B/10B/11B	O
Simultaneous start/stop mode (period controlled by TKBCRn0)	01B	00B	×
Simultaneous start/stop mode (period controlled by external trigger input)	01B	01B/10B/11B	O
Synchronous start/clear mode (period controlled by master)	10B	–	×
Interleave PFC output mode	11B	–	O

Remark Available when the period is controlled by external trigger input.

7.6 Forced Output Stop Function

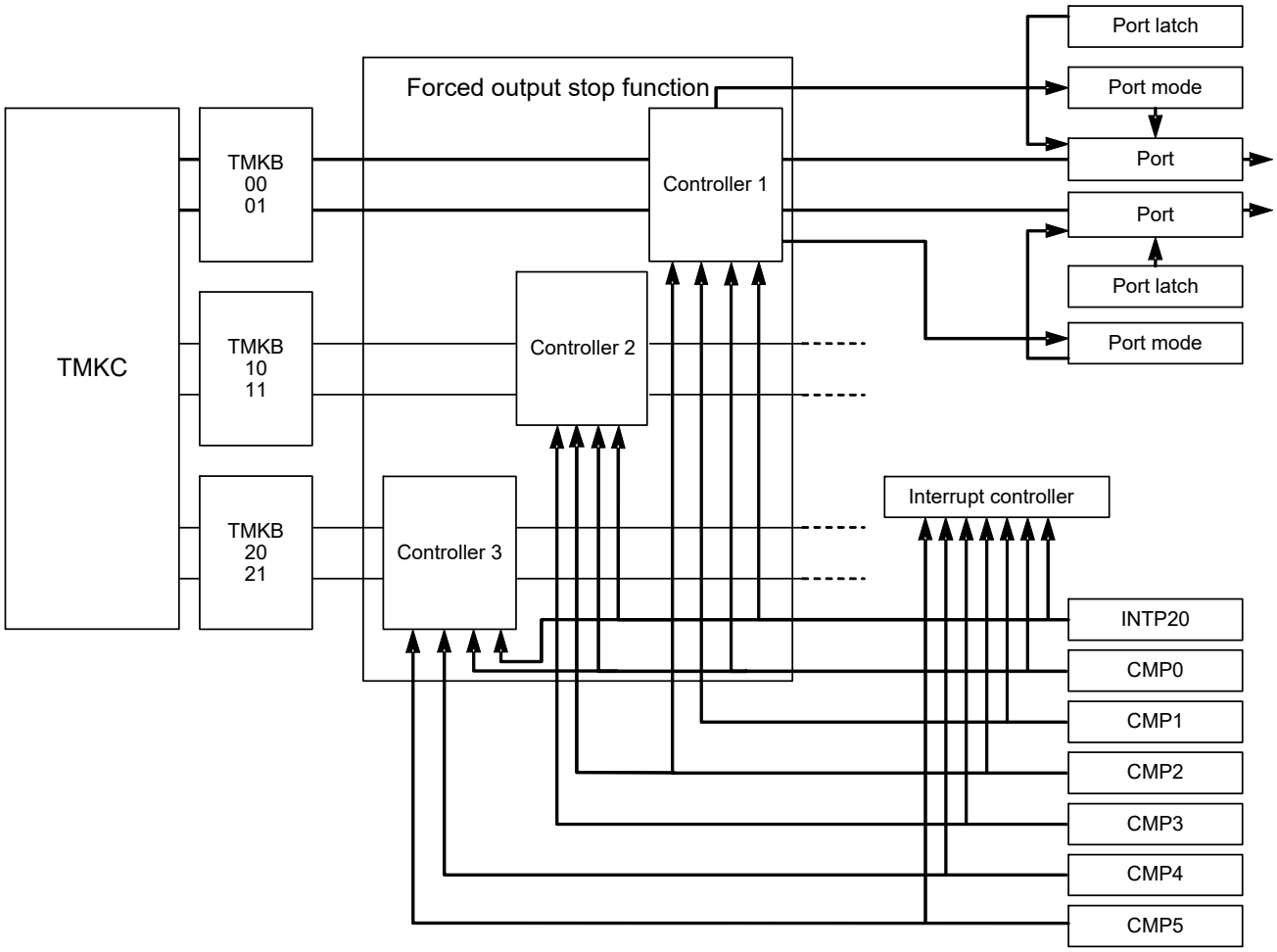
Forced output stop function is a function to protect power supply, etc.

If any abnormal situation that occurs in a power circuit configured outside of a micro-computer leads to over-voltage of over-current, making voltage or current sense signal into INTP20/comparator can protect the circuit by maintaining the timer output high impedance or fixed output state without being intermediated by a CPU's program control.

With this function, abnormality is identified only when input signal edge have been detected. Fixed level without edge is not recognized as abnormality.

The following figure shows the system structure of forced output stop function.

Figure 7-71. System Structure of Forced Output Stop Function



7.6.1 Forced output stop function 1 and 2

There are two ways of controls in forced output stop function. Forced output stop function 1 can select fixed level output or high-impedance output, and forced output stop function 2 can only set fixed level output. Then the difference of the control method is shown.

(1) Selectable Output Levels for Forced Output Stop Function 1 & 2

Selectable Output Levels	Forced Output Stop	
	Function 1	Function 2
High-impedance output	○	×
Low-level fixed output	○	○
High-level fixed output	○	○

(2) Start/cancel of Forced Output Stop Function 1 & 2

Function/Operation Details (Start Forced Output Stop)	Forced Output Stop	
	Function 1	Function 2
Start forced output stop following the detection of the rising edge of comparator output.	○	○
Start forced output stop following the rising or falling edge detection by external interruption input (INTP20).	×	○
Start forced output stop by software bit (TKBPAHTSnp) setting.	○	×

Function/Operation Details (Cancel Forced Output Stop)	Forced Output Stop	
	Function 1	Function 2
Cancel forced output stop by software bit (TKBPAHTTnp) setting.	○	×
Cancel forced output stop by synchronization with TMKB period after the software bit (TKBPAHTTnp) setting.	○	×
Forced output stop cancelled at the next counter period after the beginning of forced output stop.	×	○
Forced output stop cancelled at the next counter period after the detection of falling edge by trigger signal of forced output stop.	×	○

Remark n = 0 to 2, p = 0, 1

(3) Trigger Signal Selectable and the Conditions for Trigger Bit Available for Forced Output Stop Function 1 & 2

Selectable Trigger Signals	Forced Output Stop	
	Function 1	Function 2
Comparator 0 to 5	○	○
External interrupt input (INTP20)	×	○

Trigger Bits Available	Forced Output Stop	
	Function 1	Function 2
TKBPAHTSnp (Trigger bit which starts forced output stop of TKBOnp output)	○	×
TKBPAHTTnp (Trigger bit which cancels forced output stop of TKBOnp output)	○	×

Remark n = 0 to 2, p = 0, 1

Table 7-4. External Trigger Assignment List of Forced Output Stop Function 1

	TKBO00	TKBO01	TKBO10	TKBO11	TKBO20	TKBO21
Comparator 0	√	√	√	√	√	√
Comparator 1	√	√	–		–	–
Comparator 2	√	√	√	√	–	–
Comparator 3	–	–	√	√	–	–
Comparator 4	–	–	–	–	√	√
Comparator 5	–	–	–	–	√	√
INTP20	–	–	–	–	–	–
INTP21	–	–	–	–	–	–

Table 7-5. External Trigger Assignment List of Forced Output Stop Function 2

	TKBO00	TKBO01	TKBO10	TKBO11	TKBO20	TKBO21
Comparator 0	√	√	√	√	√	√
Comparator 1	√	√	–	–	–	–
Comparator 2	√	√	√	√	–	–
Comparator 3	–	–	√	√	–	–
Comparator 4	–	–	–	–	√	√
Comparator 5	–	–	–	–	√	√
INTP20	√	√	√	√	√	√
INTP21	–	–	–	–	–	–

Caution For setting of INTP20/INTP21, see CHAPTER 14 COMPARATOR.

7.6.2 Configuration of forced output stop function

Forced output stop function includes the following hardware.

Table 7-6. Configuration of Forced Output Stop Function

Item	Configuration
Control registers	Peripheral enable register 2 (PER2) Forced output stop function control register n0 (TKBPACTLn0) Forced output stop function control register n1 (TKBPACTLn1) Forced output stop function control register n2 (TKBPACTLn2) Forced output stop function flag register n (TKBPAFLGn) Forced output stop function 1 start trigger register n (TKBPAHFSn) Forced output stop function cancel 1 trigger register n (TKBPAHFTn)

7.6.3 Registers controlling forced output stop function

Forced output stop function is controlled by the following registers.

- Peripheral enable register 2 (PER2)
- Forced output stop function control register n0 (TKBPACTLn0)
- Forced output stop function control register n1 (TKBPACTLn1)
- Forced output stop function control register n2 (TKBPACTLn2)
- Forced output stop function flag register n (TKBPAFLGn)
- Forced output stop function start 1 trigger register n (TKBPAHFSn)
- Forced output stop function cancel 1 trigger register n (TKBPAHFTn)

7.6.3.1 Peripheral enable register 2 (PER2)

This register is used to supply or stop the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the forced output stop function is used, be sure to set bits 6 to 4 (TKBPA2EN to TKBPA0EN) of this register to 1.

The PER2 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-72. Format of Peripheral Enable Register 2 (PER2)

Address: F0509H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PER2	PGACMPEN	TKBPA2EN	TKBPA1EN	TKBPA0EN	TKC0EN	TKB2EN	TKB1EN	TKB0EN

TKBPAnEN	Control of input clock to forced output stop function of timer KBn
0	Stops supply of input clock. • SFR used by forced output stop function of timer KBn cannot be written. • Forced output stop function of timer KBn is in the initial status.
1	Supplies input clock. • SFR used by forced output stop function of timer KBn can be read/written.

Caution When setting the forced output stop function of timer KBn, be sure to set the TKBPAnEN bit to 1 first. If TKBPAnEN = 0, writing to a control register of forced output stop function of timer KBn is ignored, and all read values are default values.

Remark n = 0 to 2

7.6.3.2 Forced output stop function control registers n0, n1 (TKBPACTLn0, TKBPACTLn1)

TKBPACTLnp 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.

TKBPACTLnp can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 7-73. Format of Forced Output Stop Function Control Register 0p (TKBPACTL0p) (1/2)

Address: F0630H (TKBPACTL00), F0632H (TKBPACTL01) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
TKBPACTL0p	TKBPAFXS0p3	TKBPAFXS0p2	TKBPAFXS0p1	TKBPAFXS0p0	0	0	0	TKBPAFCM0p

7	6	5	4	3	2	1	0
0	TKBPAHZS0p2	TKBPAHZS0p1	TKBPAHZS0p0	TKBPAHCM0p1	TKBPAHCM0p0	TKBPAMD0p1	TKBPAMD0p0

TKBPAFXS0p3	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

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 3

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

TKBPAFCM0p	Operation 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 4
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 4

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 3

Figure 7-73. Format of Forced Output Stop Function Control Register 0p (TKBPACTL0p) (2/2)

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

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 active 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 inactive 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 4
1	1	Forced output stop function 1 starts with trigger input, and when the trigger signal is in its active 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 inactive period. Note 4

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

Notes 1. When INTP20 is used as the forced output stop function 2, see **14.5 Caution for Using Timer KB Simultaneous Operation Function**.

2. When CMP0 or CMP2 is used as the timer KB forced output stop function, set CMPnSTEN = 1. See **14.5 Caution for Using Timer KB Simultaneous Operation Function**.

3. When CMP1 is used as the timer KB forced output stop function, see **14.5 Caution for Using Timer KB Simultaneous Operation Function**.

4. When timer KB is stopped (TKBCEn = 0) without waiting for the next counter period, the forced output stop function is kept on until timer KB is restarted (TKBCEn = 1).

Cautions 1. During timer operation, setting the other bits of the TKBPACTL0p register is prohibited. However, the TKBPACTL0p register can be refreshed (the same value is written).

2. Be sure to clear bits 11 to 9 and 7 to “0”.

Remark n = 0 to 2, p = 0, 1

Figure 7-74. Format of Forced Output Stop Function Control Register 1p (TKBPACTL1p) (1/2)

Address: F0670H (TKBPACTL10), F0672H (TKBPACTL11) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
TKBPACTL1p	TKBPAFXS1p3	TKBPAFXS1p2	TKBPAFXS1p1	TKBPAFXS1p0	0	0	0	TKBPAFCM1p

7	6	5	4	3	2	1	0
0	TKBPAHZS1p2	TKBPAHZS1p1	TKBPAHZS1p0	TKBPAHCM1p1	TKBPAHCM1p0	TKBPAMD1p1	TKBPAMD1p0

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 3

TKBPAFXS1p0	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 3

TKBPAFCM1p	Operation 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 4
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 4

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 3

TKBPAHZS1p0	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 3

Figure 7-74. Format of Forced Output Stop Function Control Register 1p (TKBPACTL1p) (2/2)

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 active 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 inactive 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 4
1	1	Forced output stop function 1 starts with trigger input, and when the trigger signal is in its active 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 inactive period. Note 4

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

Notes 1. When INTP20 is used as the forced output stop function 2, see **14.5 Caution for Using Timer KB Simultaneous Operation Function**.

2. When CMP3 is used as the timer KB forced output stop function, see **14.5 Caution for Using Timer KB Simultaneous Operation Function**.

3. When CMP0 or CMP2 is used as the timer KB forced output stop function, set CMPnSTEN = 1. For details, see **14.5 Caution for Using Timer KB Simultaneous Operation Function**.

4. When timer KB is stopped (TKBCEn = 0) without waiting for the next counter period, the forced output stop function is kept on until timer KB is restarted (TKBCEn = 1).

Cautions 1. During timer operation, setting the other bits of the TKBPACTL1p register is prohibited. However, the TKBPACTL1p register can be refreshed (the same value is written).

2. Be sure to clear bits 11 to 9 and 7 to “0”.

Remark n = 0 to 2, p = 0, 1

Figure 7-75. Format of Forced Output Stop Function Control Register 2p (TKBPACTL2p) (1/2)

Address: F06B0H (TKBPACTL20) , F06B2H (TKBPACTL21) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
TKBPACTL2p	TKBPAFXS2p3	TKBPAFXS2p2	TKBPAFXS2p1	TKBPAFXS2p0	0	0	0	TKBPAFCM2p

7	6	5	4	3	2	1	0
0	TKBPAHZS2p2	TKBPAHZS2p1	TKBPAHZS2p0	TKBPAHCM2p1	TKBPAHCM2p0	TKBPAMD2p1	TKBPAMD2p0

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 5 can not be used as a trigger.
1	Comparator 5 can be used as a trigger.Note 2

TKBPAFXS2p1	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

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 3

TKBPAFCM2p	Operation 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 4
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 4

TKBPAHZS2p2	Comparator trigger selection for forced output stop function 1
0	Comparator 5 can not be used as a trigger.
1	Comparator 5 can be used as a trigger.Note 2

TKBPAHZS2p1	Comparator trigger selection for forced output stop function 1
0	Comparator 4 can not be used as a trigger.
1	Comparator 4 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 3

Figure 7-75. Format of Forced Output Stop Function Control Register 2p (TKBPACTL2p) (2/2)

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 active 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 inactive 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 4
1	1	Forced output stop function 1 starts with trigger input, and when the trigger signal is in its active 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 inactive period.Note 4

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

Notes 1. When INTP20 is used as the forced output stop function 2, see **14.5 Caution for Using Timer KB Simultaneous Operation Function**.

2. When CMP4 or CMP5 is used as the timer KB forced output stop function, see **14.5 Caution for Using Timer KB Simultaneous Operation Function**.

3. When CMP0 is used as the timer KB forced output stop function, set CMP0STEN = 1. For details, see **14.5 Caution for Using Timer KB Simultaneous Operation Function**.

4. When timer KB is stopped (TKBCEn = 0) without waiting for the next counter period, the forced output stop function is kept on until timer KB is restarted (TKBCEn = 1).

Cautions 1. During timer operation, setting the other bits of the TKBPACTL2p register is prohibited. However, the TKBPACTL2p register can be refreshed (the same value is written).

2. Be sure to clear bits 11 to 9 and 7 to “0”.

Remark n = 0 to 2, p = 0, 1

7.6.3.3 Forced output stop function control register n2 (TKBPACTLn2)

TKBPACTLn2 is a register that enables or disables the forced output stop function of the TKBOnp pin.

TKBPACTLn2 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-76. Format of Forced Output Stop Function Control Register n2 (TKBPACTLn2)

Address: F0637H (TKBPACTL02), F0677H (TKBPACTL02), F06B7H (TKBPACTL02)

After reset: 00H 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

Cautions 1. The TKBPACTLn2 register can be overwritten while the timer is operating.

2. Be sure to clear bits 7 to 2 to “0”.

Remark n = 0 to 2, p = 0, 1

7.6.3.4 Forced output stop function flag register (TKBPAFLGn)

TKBPAFLGn is a register with status flags for forced output stop function of the TKBOnp pin.

TKBPAFLGn can be read by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-77. Format of Forced Output Stop Function Flag Register (TKBPAFLGn)

Address: F0636H (TKBPAFLG0), F0676H (TKBPAFLG1), F06B6H (TKBPAFLG2) After reset: 00H 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 7.7.2 Software cancel operation for forced output stop function 1.

Remark n = 0 to 2, p = 0, 1

7.6.3.5 Forced output stop function 1 start trigger register n (TKBPAHFSn)

TKBPAHFSn is the start trigger register used by forced output stop function 1 of the TKBOnp pin.

TKBPAHFSn can be written by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-78. Format of Forced Output Stop Function 1 Start Trigger Register n (TKBPAHFSn)

Address: F0634H (TKBPAHFS0), F0674H (TKBPAHFS1), F06B4H (TKBPAHFS2)

After reset: 00H 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

Cautions 1. The TKBPAHFSn register can be overwritten while the timer is operating.

2. Be sure to clear bits 7 to 2 to "0".

3. When TKBPAHFSn register is read, 0 is read.

7.6.3.6 Forced output stop function cancel 1 trigger register n (TKBPAHFTn)

TKBPAHFTn is the cancel trigger register used by forced output stop function 1 of the TKBOnp pin.

TKBPAHFTn can be written by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-79. Format of Forced Output Stop Function Cancel 1 Trigger Register n (TKBPAHFTn)

Address: F0635H (TKBPAHFT0), F0675H (TKBPAHFT1), F06B5H (TKBPAHFT2)

After reset: 00H 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

Cautions 1. The TKBPAHFTn register can be overwritten while the timer is operating.

2. Be sure to clear bits 7 to 2 to "0".

3. When TKBPAHFSn register is read, 0 is read.

4. The timing to cancel the forced output stop function 1 depends on the setting. For details, see 7.7.2 Software cancel operation for forced output stop function 1.

Remark n = 0 to 2, p = 0, 1

7.7 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 operation clock f_{KBKC} of the 16-bit timer KBn and KC0 circuit when a trigger source occurs (comparator 0 to 5 output). The forced output stop status is canceled synchronously with the operation clock f_{KBKC} of the 16-bit timer KBn and KC0 circuits by setting the stop trigger of forced output stop function 1.

7.7.1 Summary for forced output stop function 1

In this function, comparator output signal and software trigger is used as trigger signal for forced output stop function 1.

The output level selectable at forced output stop is controlled byTKBPAMDnp0, TKBPAMDnp1 bit of TKBPACTLnp register.

The following table shows the relationship of forced output stop function 1 of output p terminal (TKBOnp) for timer KBn.

Table 7-7. The 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	Low level fixed output
1	1	High-level fixed output

The selection for comparator output being used is controlled by TKBPAHZSnp2 to TKBPAHZSnp0 bit for forced output stop function control register np (TKBPACTLnp).

The following table shows trigger selection for forced output stop function 1 of output p terminal (TKBOnp) for timer KBn.

Table 7-8. The Trigger Selection for Forced Output Stop Function 1 of the TKBOnp Pin

Bit	Selectable Trigger Signals		
	Timer KB0	Timer KB1	Timer KB2
TKBPAHZSnp0	Comparator 0		
TKBPAHZSnp1	Comparator 1	Comparator 2	Comparator 4
TKBPAHZSnp2	Comparator 2	Comparator 3	Comparator 5

Remark n = 0 to 2, p = 0, 1

7.7.2 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 7-9. 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 high-impedance/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 7-10. 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 active 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 inactive 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 active 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 inactive period. Note

Note When timer KB is stopped (TKBCEn = 0) without waiting for the next counter period, the forced output stop function is kept on until timer KB is restarted (TKBCEn = 1).

Remark n = 0 to 2, p = 0, 1

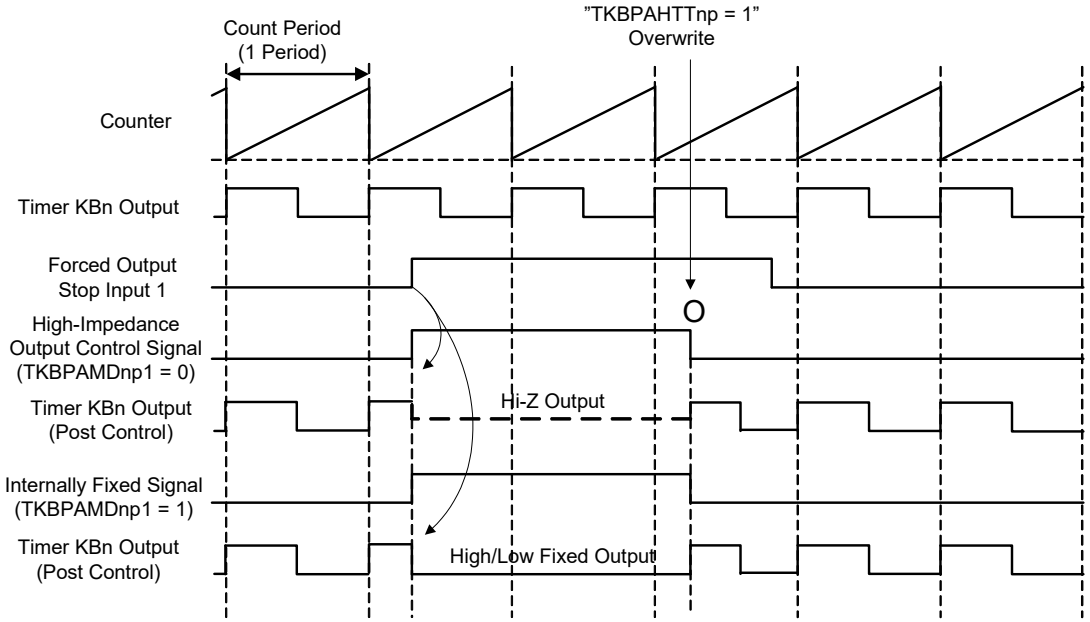
7.7.3 Basic operation of forced output stop function 1

This shows the operations of forced output function 1 with different setting of TKBPAHCMnp1 and TKBPAHCMnp0 registers.

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 TKBPAHZSnp0 to TKBPAHZSnp2 bits of forced output stop function control register np (TKBPACTLnp) and TKBPAHTSnp bit of the trigger register n (TKBPAHFSn) that initiates forced output stop function 1.

(1) Forced output stop function 1 at TKBPAHCMnp1, TKBPAHCMnp0 = 0, 0

Figure 7-80. Forced Output Stop Function 1 at TKBPAHCMnp1, TKBPAHCMnp0 = 0, 0



(a) TKBPAMDnp1 = 0 (Hi-z output)

High-impedance 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 high-impedance output control signal is the period for forced output stop 1 (high-impedance output).

(b) TKBPAMDnp1 = 1 (fixed output)

Fixed output is realized in low-level/high-level according to TKBPAMDnp0 setting at the detection of rising edge of forced output stop input 1.

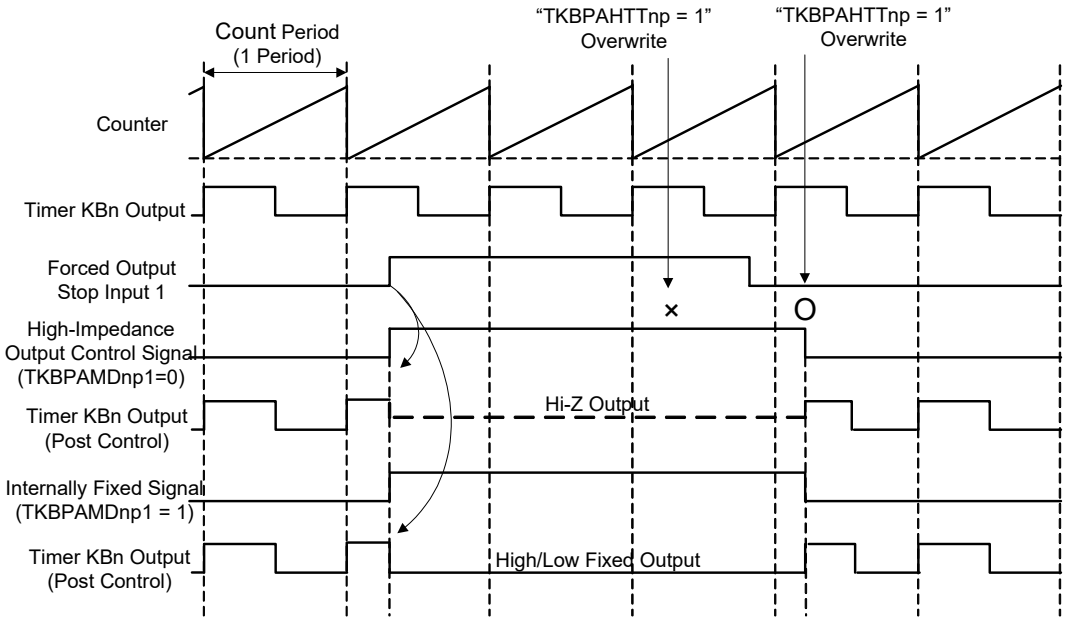
Regardless of input level of forced output stop input 1, output level fixing is cancelled and returned to timer output via writing of “1” to cancel trigger (TKBPAHTTnp bit).

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 TKBPAHCMnp1, TKBPAHCMnp0 = 0, 1

Figure 7-81. Forced Output Stop Function 1 at TKBPAHCMnp1, TKBPAHCMnp0 = 0, 1



(a) TKBPAMDnp1 = 0 (Hi-z output)

High-impedance output is realized via the detection of rising edge of forced output stop input 1.

During the active level (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 inactive level (low-level), it returns to timer output via writing of "1" to cancel trigger (TKBPAHTTnp bit).

High-level period of high-impedance output control signal is the period for forced output stop 1 (high-impedance output).

(b) TKBPAMDnp1 = 1 (fixed output)

Fixed output is realized in low-level/high-level according to TKBPAMDnp0 setting at the detection of rising edge of forced output stop input 1.

During the active level (high-level) period of forced output stop input 1, writing "1" to cancel trigger (TKBPAHTTnp bit) is invalid.

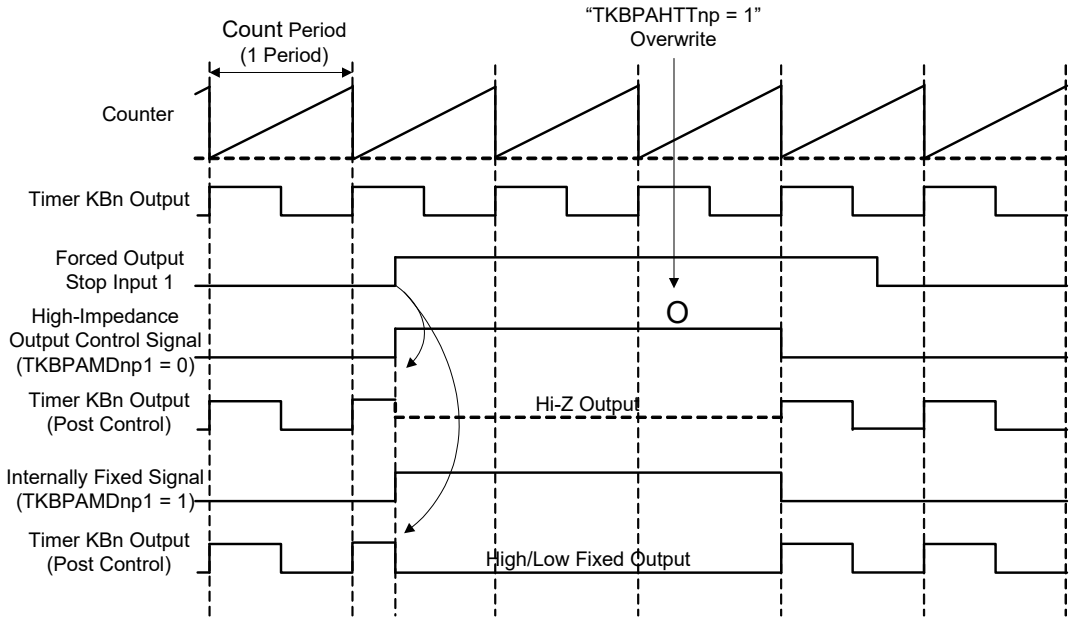
After the forced output stop input 1 turned into inactive level (low-level), output level fixing is cancelled and returned to timer output via writing of "1" to cancel trigger (TKBPAHTTnp bit).

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 TKBPAHCMnp1, TKBPAHCMnp0 = 1, 0

Figure 7-82. Forced Output Stop Function 1 at TKBPAHCMnp1, TKBPAHCMnp0 = 1, 0



(a) TKBPAMDnp1 = 0 (Hi-z output)

High-impedance 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 high-impedance output control signal is the period for forced output stop 1 (high-impedance output).

(b) TKBPAMDnp1 = 1 (fixed output)

Fixed output is realized in low-level/high-level according to TKBPAMDnp0 setting at the detection of rising edge of forced output stop input 1.

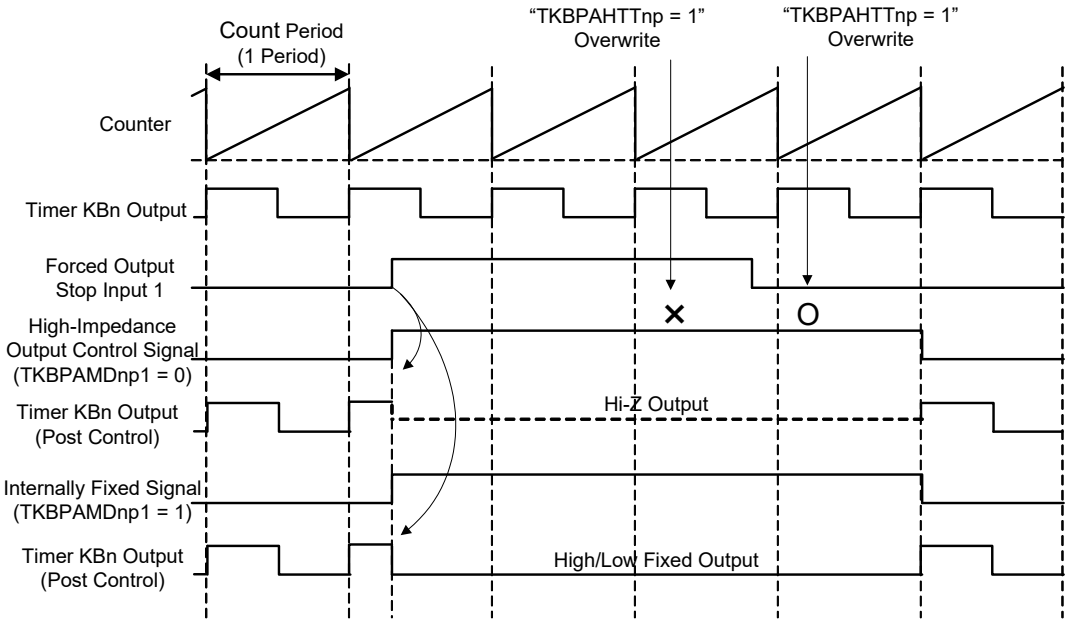
Regardless of input level of forced output stop input 1, output level fixing is cancelled and returned to timer output after writing of "1" to cancel trigger (TKBPAHTTnp bit) in the next counter period.

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 TKBPAHCMnp1, TKBPAHCMnp0 = 1, 1

Figure 7-83. Forced Output Stop Function 1 at TKBPAHCMnp1, TKBPAHCMnp0 = 1, 1



(a) TKBPAMDnp1 = 0 (Hi-z output)

High-impedance output is realized via the detection of rising edge of forced output stop input 1.

During the active level (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 inactive level (low-level) of forced output stop input 1, in the next period.

High-level period of high-impedance output control signal is the period for forced output stop 1 (high-impedance output).

(b) TKBPAMDnp1 = 1 (fixed output)

Fixed output is realized in low-level/high-level according to TKBPAMDnp0 setting at the detection of rising edge of forced output stop input 1.

During the active level (high-level) period of forced output stop input 1, writing "1" to cancel trigger (TKBPAHTTnp bit) is invalid.

During the inactive level (low-level) period of forced output stop input 1, output level fixing is cancelled 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

7.8 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 operation clock f_{KBKC} of the 16-bit timer KBn and KC0 circuit when a trigger source occurs (comparator 0 to 5 output, INTP20). 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.

7.8.1 Summary for forced output stop function 2

In this function, comparator output signal and external interrupt (INTP20) are used as trigger signals for forced output stop function 2.

The output level selectable at forced output stop is controlled byTKBPAMDnp0, TKBPAMDnp1 bit of TKBPACTLnp register.

The following table shows the relationship of forced output stop function 2 of output p terminal (TKBOnp) for timer KBn.

Table 7-11. The 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	Low level fixed output
0	1	High-level fixed output
1	0	Low level fixed output
1	1	High-level fixed output

The selection for comparator output being used is controlled by TKBPAFXSnp3 to TKBPAFXSnp0 bit for forced output stop function control register np (TKBPACTLnp).

The following table shows trigger selection for forced output stop function 2 of output p terminal (TKBOnp) for timer KBn.

Table 7-12. The Trigger Selection for Forced Output Stop Function 2 of the TKBOnp Pin

Bit	Selectable Trigger Signals		
	Timer KB0	Timer KB1	Timer KB2
TKBPAFXSnp0	Comparator 0		
TKBPAFXSnp1	Comparator 1	Comparator 2	Comparator 4
TKBPAFXSnp2	Comparator 2	Comparator 3	Comparator 5
TKBPAFXSnp3	INTP20		

Remark n = 0 to 2, p = 0, 1

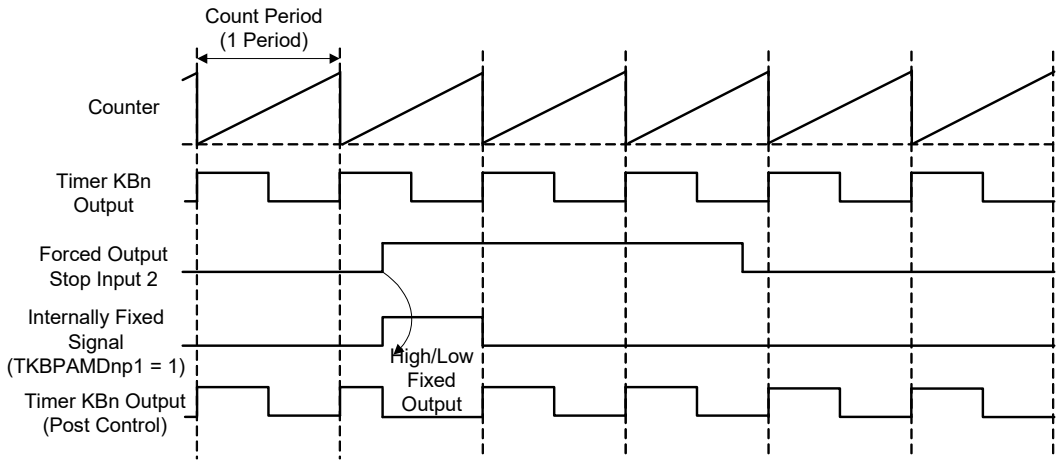
7.8.2 Basic operation of forced output stop function 2

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 TKBPAFXSnp0 to TKBPAFXSnp3 bits of forced output stop function control register np.

(1) Forced output stop function 2 at TKBPAFCMnp = 0

Figure 7-84. 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.

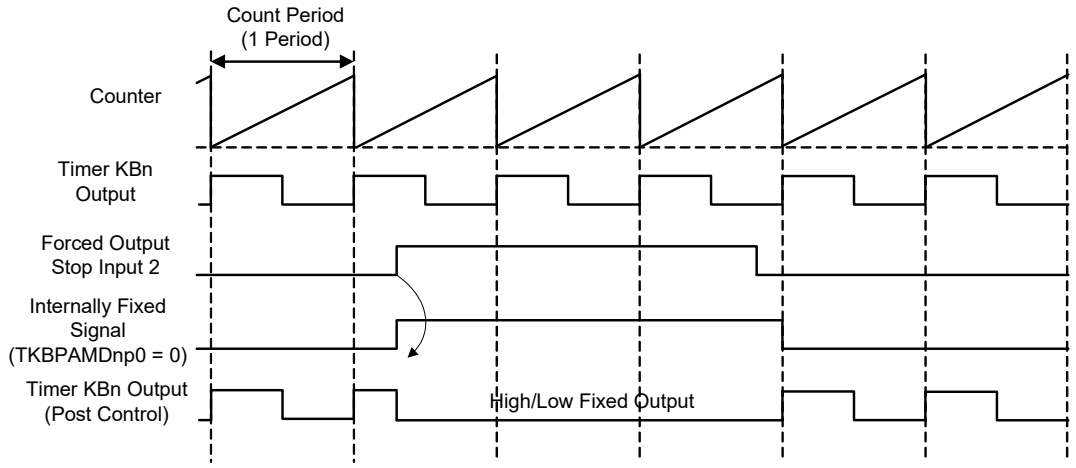
Regardless of the input level of the forced output stop input 2, the fixing of output level is canceled at the next counter cycle and returned to timer output.

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 TKBPAFCMnp0 = 1

Figure 7-85. Forced Output Stop Function 2 at TKBPAFCMnp0 = 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.

During the active level (high-level) period of forced output stop input 1, writing “1” to cancel trigger (TKBPAHTTnp bit) is invalid.

After the forced output stop input 2 becomes reverse edge, 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

CHAPTER 8 16-BIT TIMER KC0

8.1 Functions of 16-bit Timer KC0

16-bit timer KC0 is mounted onto all RL78/I1A microcontroller products.

16-bit timer KC0 is a timer with six outputs, enabling the generation of up to six PWM outputs.

In addition, gate control can be implemented for up to six PWM outputs by interlocking with timers KB0, KB1, and KB2.

16-bit timer KC0 is provided with the following functions.

(1) PWM output

- A variable pulse with any duty can be output for a period while the timer is operating.
- The default timer output level (high or low level) can be set.

(2) PWM output gating function (by interlocking with 16-bit timers KB0, KB1, and KB2)

Maximum of six lines of 16-bit timers, KB0, KB1, KB2 output (TKBO00, TKBO01, TKBO10, TKBO11, TKBO20, TKBO21 output) can be controlled via a gate respectively with any duty by timer KC output.

(3) Simultaneous start & stop mode

By assigning the 16-bit timer KB as master and the 16-bit timer KC as slave, the 16-bit timer KC at slave can be started or stopped in synchronizing with the count start/stop timing of timer KB.

8.2 Configuration of 16-bit Timer KC0

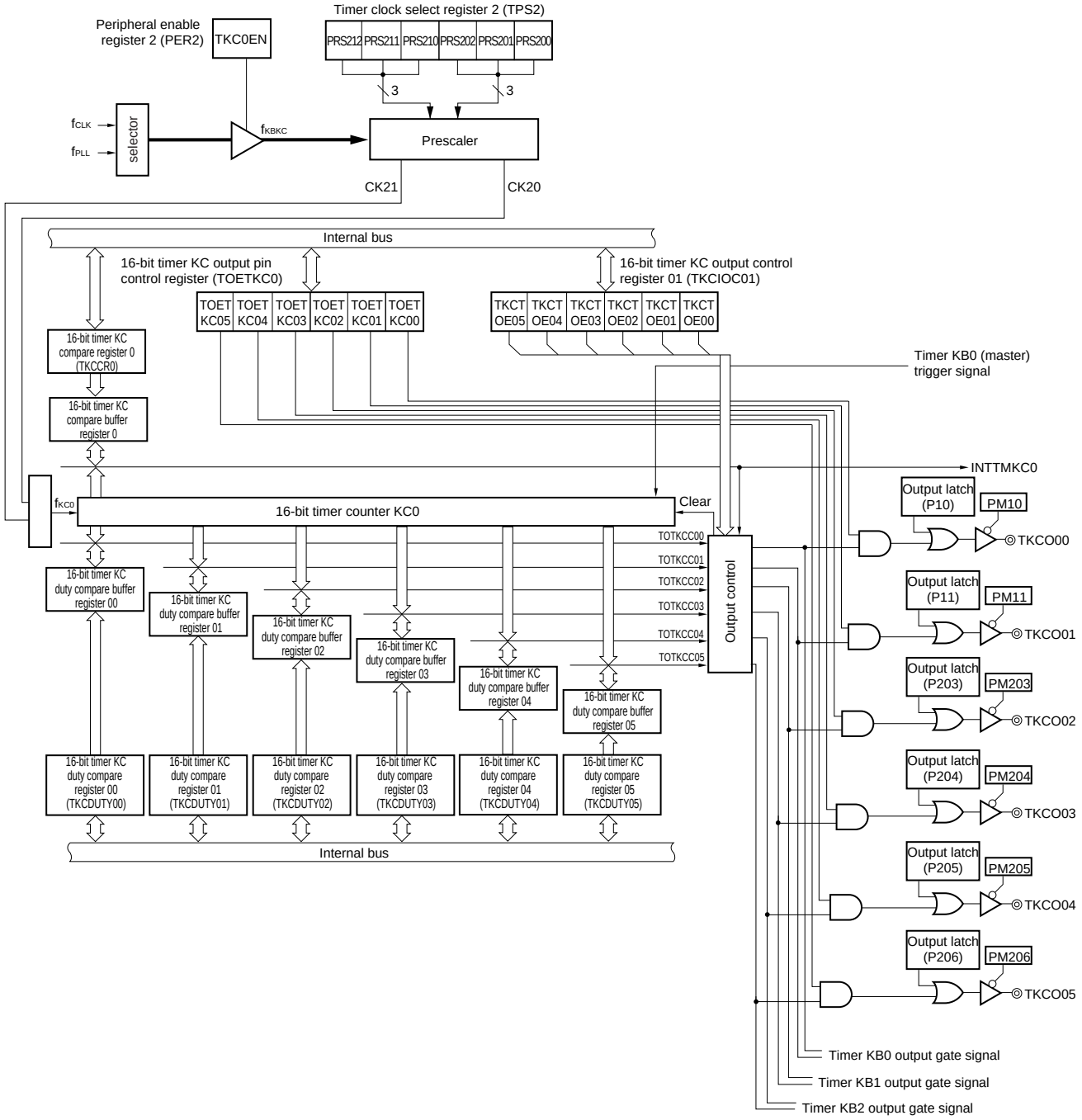
16-bit timer KC0 includes the following hardware.

Table 8-1. Configuration of 16-bit Timer KC0

Item	Configuration
Timer/counter	16-bit timer counter KC0 (TKCCNT0)
Registers	16-bit timer KC compare register 0 (TKCCR0) 16-bit timer KC duty compare registers 0 to 5 (TKCDUTY00 to TKCDUTY05)
Timer output	TKCO00 to TKCO05
Control registers	16-bit timer KC operation control register 0 (TKCCTL0) 16-bit timer KC output control register 00 (TKCIOC00) 16-bit timer KC output control register 01 (TKCIOC01) 16-bit timer KC output pin control register (TOETKC0) 16-bit timer KC output flag register 0 (TKCTOF0) 16-bit timer KC flag register 0 (TKCFLG0) 16-bit timer KC trigger register 0 (TKCTRG0) Port mode registers 1, 20 (PM1, PM20) Port registers 1, 20 (P1, P20)

Figure 8-1 shows a block diagram.

Figure 8-1. Block Diagram of 16-bit Timer KC0



Remarks 1. f_{KBKC}: Operation clock of whole 16-bit timer KBn and KC0 circuit

f_{KC0}: Count clock of 16-bit timer KC0

2. n = 0 to 2

8.2.1 16-bit timer KC compare register 0 (TKCCR0)

TKCCR0 is setting to period of timer KC.

TKCCR0 can be refreshed (writing the same value) and its value can be rewritten while the timer is counting (TKCCE0 = 1). When the value of TKCCR0 is rewritten while the timer is operating, that value is latched, transferred to TKCCR0 at the following timing, and the value of TKCCR0 is changed.

- When starting count operation of counter (TKCCE0 = 0)
- When a batch overwrite trigger (TKCRDT0 = 1) or an external trigger (TKCTSE0 = 1) occurs

This register can be read or written in 16-bit units.

Reset signal generation clears this register to 0000H.

Figure 8-2. Format of 16-bit Timer KC Compare Register 0 (TKCCR0)

Address: F06D0H After reset: 0000H R/W

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKCCR0																

8.2.2 16-bit timer KC duty compare registers 00 to 05 (TKCDUTY00 to TKCDUTY05)

TKCDUTY0m is setting to active period of timer KC output.

TKCDUTY0m can be refreshed (writing the same value) and its value can be rewritten while the timer is counting (TKCCE0 = 1). When the value of TKCDUTY0m is rewritten while the timer is operating, that value is latched, transferred to TKCDUTY0m at the following timing, and the value of TKCDUTY0m is changed.

- When starting count operation of counter (TKCCE0 = 0)
- When a batch overwrite trigger (TKCRDT0 = 1) or an external trigger (TKCTSE0 = 1) occurs

This register can be read or written in 16-bit units.

Reset signal generation clears this register to 0000H.

Figure 8-3. Format of 16-bit Timer KC Duty Compare Register 0m (TKCDUTY0m)

Address: F06D2H (TKCDUTY00), F06D4H (TKCDUTY01), F06D6H (TKCDUTY02), F06D8H (TKCDUTY03), F06DAH (TKCDUTY04), F06DCH (TKCDUTY05) After reset: 0000H R/W

After reset: 0000H R/W

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKCDUTY0m																

Remark m = 0 to 5

8.3 Registers Controlling 16-bit Timer KC0

16-bit timer KC0 is controlled by the following registers.

- Peripheral enable register 2 (PER2)
- Timer clock select register 2 (TPS2)
- 16-bit timer KC operation control register 0 (TKCCTL0)
- 16-bit timer KC output control register 00 (TKCIOC00)
- 16-bit timer KC output control register 01 (TKCIOC01)
- 16-bit timer KC output pin control register (TOETKC0)
- 16-bit timer KC output flag register 0 (TKCTOF0)
- 16-bit timer KC flag register 0 (TKCFLG0)
- 16-bit timer KC trigger register 0 (TKCTRG0)
- Port mode registers 1, 20 (PM1, PM20)
- Port registers 1, 20 (P1, P20)

8.3.1 Peripheral enable register 2 (PER2)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When timer KC0 is used, be sure to set bit 3 (TKC0EN) of this register to 1.

The PER2 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8-4. Format of Peripheral Enable Register 2 (PER2)

Address: F0509H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PER2	PGACMPEN	TKBPA2EN	TKBPA1EN	TKBPA0EN	TKC0EN	TKB2EN	TKB1EN	TKB0EN

TKC0EN	Control of timer KC0 input clock
0	Stops supply of input clock. • SFR used by timer KC0 cannot be written. • Timer KC0 is in the initial status.
1	Supplies input clock. • SFR used by timer KC0 can be read/written.

Caution When setting timer KC0, be sure to set the TKC0EN bit to 1 first. If TKC0EN = 0, writing to a control register of timer KC0 is ignored, and all read values are default values (except for timer clock select register 2 (TPS2), port mode registers 1, 20 (PM1, PM20), and port registers 1, 20 (P1, P20)).

8.3.2 Timer clock select register 2 (TPS2)

The TPS2 register is a 16-bit register that is used to select two types of operation clocks (CK20, CK21) that are commonly supplied to timers KB0, KB1, KB2, and KC0 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 PRS200 to PRS202 bits can be rewritten (n = 0 to 2):

All timers for which CK20 is selected as the operation clock (TKBCKSn = 0, TKCCKS0 = 0) are stopped (TKBCEn = 0, TKCCE0 = 0).

If the PRS210 to PRS212 bits can be rewritten (n = 0 to 2):

All timers for which CK21 is selected as the operation clock (TKBCKSn = 1, TKCCKS0 = 1) are stopped (TKBCEn = 0, TKCCE0 = 0).

The TPS2 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8-5. Format of Timer Clock Select Register 2 (TPS2)

Address: F05C5H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TPS2	0	TPS212	TPS211	TPS210	0	TPS202	TPS201	TPS200

TPS2k2	TPS2k1	TPS2k0	Selection of operation clock (CK2k)[Notes 1, 2] (k = 0, 1)					
				f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 20 MHz	f_{CLK} = 32 MHz	f_{PLL} = 64 MHz
0	0	0	f_{CLK}, f_{PLL}	2 MHz	5 MHz	20 MHz	32 MHz	64 MHz
0	0	1	$f_{CLK}/2$, $f_{PLL}/2$	1 MHz	2.5 MHz	10 MHz	16 MHz	32 MHz
0	1	0	$f_{CLK}/2^2$, $f_{PLL}/2^2$	500 kHz	1.25 MHz	5 MHz	8 MHz	16 MHz
0	1	1	$f_{CLK}/2^3$, $f_{PLL}/2^3$	250 kHz	625 kHz	2.5 MHz	4 MHz	8 MHz
1	0	0	$f_{CLK}/2^4$, $f_{PLL}/2^4$	125 kHz	312.5 kHz	1.25 MHz	2 MHz	4 MHz
1	0	1	$f_{CLK}/2^5$, $f_{PLL}/2^5$	62.5 kHz	156.2 kHz	625 kHz	1 MHz	2 MHz
1	1	0	Setting prohibited	–	–	–	–	–
1	1	1	Setting prohibited	–	–	–	–	–

Notes 1. When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timers KB0, KB1, KB2, and KC0 (TKBCEn = 0, TKCCE0 = 0).

2. When PLLON = 1 in the PLL control register (PLLCTL), f_{PLL} is supplied.

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

Remark f_{CLK}: CPU/peripheral hardware clock frequency

f_{PLL}: PLL output clock

8.3.3 16-bit timer KC operation control register 0 (TKCCTL0)

TKCCTL0 is a register that controls the count operation and sets the count clock of 16-bit timer.

TKCCTL0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8-6. Format of 16-bit Timer KC Operation Control Register 0 (TKCCTL0)

Address: 06E5H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TKCCTL0	TKCCE0	0	0	TKCCKS0	0	0	0	TKCMD0

TKCCE0	Timer KC0 operation control
0	Stops timer operation (counter is cleared to FFFFH).
1	Enables timer count operation.

TKCCKS0	Timer KC0 clock selection
0	CK20 clock selected by TPS202 to TPS200 bits
1	CK21 clock selected by TPS212 to TPS210 bits

TKCMD0	Timer KC0 operation mode selection
0	Standalone mode
1	Synchronous start/stop mode (uses slave, with timer KB0 as master)

Cautions 1. During timer operation, setting the other bits of the TKCCTL0 register is prohibited. However, the TKCCTL0 register can be refreshed (the same value is written).

2. Be sure to clear bits 6, 5, and 3 to 1 to “0”.

8.3.4 16-bit timer KC output control register 00 (TKCIOC00)

TKCIOC00 is a register that setting default/active level of TKCO0m in 16-bit timer KC0 output.

TKCIOC00 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8-7. Format of 16-bit Timer KC Output Control Register 00 (TKCIOC00)

Address: F06E2H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
TKCIOC00	0	0	TKCTOL05	TKCTOL04	TKCTOL03	TKCTOL02	TKCTOL01	TKCTOL00
	7	6	5	4	3	2	1	0
	0	0	TKCTOD05	TKCTOD04	TKCTOD03	TKCTOD02	TKCTOD01	TKCTOD00

TKCTOL0m	Active level setting of timer output TKCO0m
0	Active high
1	Active low

TKCTOD0m	Default level setting of timer output TKCO0m
0	Low level (normal output)
1	High level (inverted output)

Cautions
1. **During timer operation, setting the other bits of the TKCIOC00 register is prohibited. However, the TKCIOC00 register can be refreshed (the same value is written).**
2. **Be sure to clear bits 15, 14, 7, and 6 to "0".**
3. **Actual TKCO0m pin output is set not only by TKCO0m output but by the port mode registers (PMxx) and port registers (Pxx) for the shared ports.**

Remark m = 0 to 5

8.3.5 16-bit timer KC output control register 01 (TKCIOC01)

TKCIOC01 is the register that sets the functionality of making 16-bit timer KC0 output as PWM output gate function of 16-bit timer KB and TKCO0p output.

When it is used as the PWM output gate function, the setting for 16-bit timer KB is required.

Also, if the output is actually used as TKCO0p output, settings for TOETKC0 register as well as port register and port-mode register are required.

TKCIOC01 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8-8. Format of 16-bit Timer KC Output Control Register 01 (TKCIOC01)

Address: F06E4H After reset: 00H R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
TKCIOC01	0	0	TKCTOE05	TKCTOE04	TKCTOE03	TKCTOE02	TKCTOE01	TKCTOE00

TKCTOE0m	PWM output gating function of timer KB used and timer output TKCO0m output enable/disable
0	Using PWM output gating function of timer KB and disables timer output (Fixes to low-level output when TKCTOD0m = 0, and fixes to high-level output when TKCTOD0m = 1.)
1	Using PWM output gating function of timer KB and enables timer output (PWM output)

Cautions 1. The TKCIOC01 register can be overwritten while the timer is operating.

2. Be sure to clear bits 7 and 6 to "0".

3. Actual TKCO0m pin output is set not only by TKCO0m output but by the port mode registers (PMxx) and port registers (Pxx) for the shared ports.

Remark m = 0 to 5

8.3.6 16-bit timer KC output pin control register (TOETKC0)

It is the register that controls output enable/disable toward pins for the timer output generated from 16-bit timer KC.

The output control of TKCOn is possible regardless of the setting of the timer output gate function which is coupled with 16-bit timer KB.

TOETKC0 can be read by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8-9. Format of 16-bit timer KC output pin control register (TOETKC0)

Address: F05C8H After reset: 00H R

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
TOETKC0	0	0	TOETKC05	TOETKC04	TOETKC03	TOETKC02	TOETKC01	TOETKC00

TOETKC0m	Pin of timer output TKCO0m output enable/disable
0	Disables pin output of TKCO0m.
1	Enables pin output of TKCO0m.

Remark m = 0 to 5

8.3.7 16-bit timer KC output flag register 0 (TKCTOF0)

TKCTOF0 is the monitor flag register for TKCO0m among the 16-bit timer KC0 outputs.

TKCTOF0 can be read by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8-10. Format of 16-bit Timer KC Output Flag Register 0 (TKCTOF0)

Address: F06E6H After reset: 00H R

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
TKCTOF0	0	0	TKCTOF05	TKCTOF04	TKCTOF03	TKCTOF02	TKCTOF01	TKCTOF00

TKCTOF0m	Monitor flag of timer output TKCO0m
0	Low-level status
1	High-level status

Caution Actual TKCO0m pin output is set not only by TKCO0m output but by the port mode registers (PMxx) and port registers (Pxx) for the shared ports.

Remark m = 0 to 5

8.3.8 16-bit timer KC flag register 0 (TKCFLG0)

TKCFLG0 is a register with status flags for 16-bit timer KC0.

TKCFLG0 can be read by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8-11. Format of 16-bit Timer KC Flag Register 0 (TKCFLG0)

Address: F06DFH After reset: 00H R

Symbol	7	6	5	4	3	2	1	<0>
TKCFLG0	0	0	0	0	0	0	0	TKCRSF0

TKCRSF0	Pending status flag for batch overwrite trigger
0	Batch overwrite enabled status or completion of batch overwrite caused by to batch overwrite trigger
1	Batch overwrite pending (wait for completion) status, or “01H” is written to batch write trigger register (TKCTRG0)

8.3.9 16-bit timer KC trigger register 0 (TKCTRG0)

TKCTRG0 is a trigger register used for batch overwriting of the compare register for 16-bit timer KC0.

TKCTRG0 can be written by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8-12. Format of 16-bit Timer KC Trigger Register 0 (TKCTRG0)

Address: F06DEH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
TKCTRG0	0	0	0	0	0	0	0	TKCRDT0

TKCRDT0	Trigger for batch overwrite request of compare register
0	Invalid setting
1	Batch overwrite request of compare register (set the TKCRSF0 flag to “1”)

Remark When TKCTRG0register is read, 0 is read.

8.3.10 Port mode registers 1, 20 (PM1, PM20)

These registers specify input or output mode for port 1 or 20 in 1-bit units.

When using the P10/TKCO00/INTP20/SO00/TxD0/SCLA0/(DALITxD4), P11/TKCO01/INTP21/SI00/RxD0/SDAA0/(DALIRxD4)/(TI07)/(TxRx4), P12/(TKCO03)/SCK00, P203/TKCO02/TKBO11/(INTP20), P204/TKCO03/TKBO20, P205/TKCO04/TKBO21/DALITxD4, and P206/TKCO05/DALIRxD4/TxRx4/INTP23 pins for timer output, set PM10 to PM12 and PM203 to PM206 and the output latches of P10 to P12 and P203 to P206 to 0.

PM1 and PM20 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 8-13. Format of Port Mode Registers 1, 20 (PM1, PM20) (38-pin Products)

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	1	1	1	1	1	PM12	PM11	PM10

PM1y	P1y pin I/O mode selection (y = 0 to 2)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Address: F0510H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM20	1	PM206	PM205	PM204	PM203	PM202	PM201	PM200

PM20x	P20x pin I/O mode selection (x = 0 to 6)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bits 3 to 7 of the PM1 register, and bit 7 of the PM20 register to "1".
For 30- and 20-pin products, the following bits must be set to output mode (specified by setting the port registers and port mode registers to 0) by software after reset is cancelled.
30-pin products: Bit 2 of the PM1 register
20-pin products: Bit 2 of the PM1 register, and bits 4 to 6 of the PM20 register

Remark The figure shown above presents the format of port mode registers 1 and 20 of the 38-pin products. For the format of the port mode register of other products, see **Table 4-3 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx Registers and Bits Mounted on Each Product**.

8.4 Operation of 16-bit Timer KC0

Figure 8-14. Timer KC Operation Setting Example (Operation Start Flow)

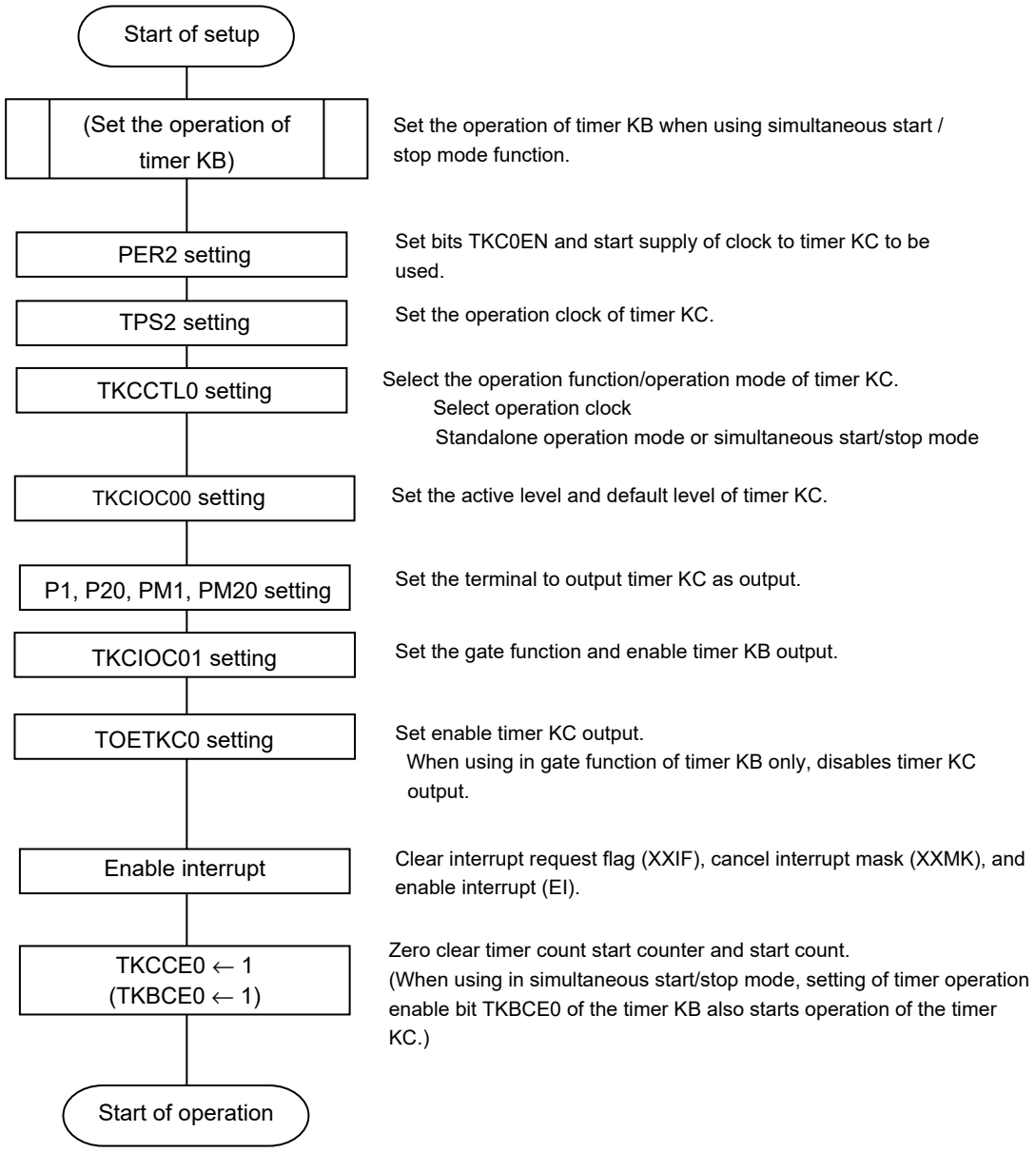
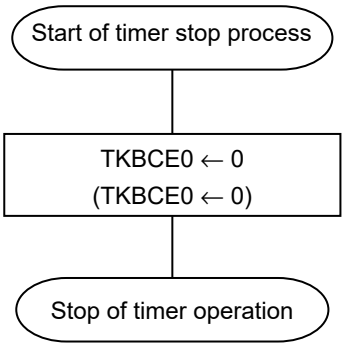
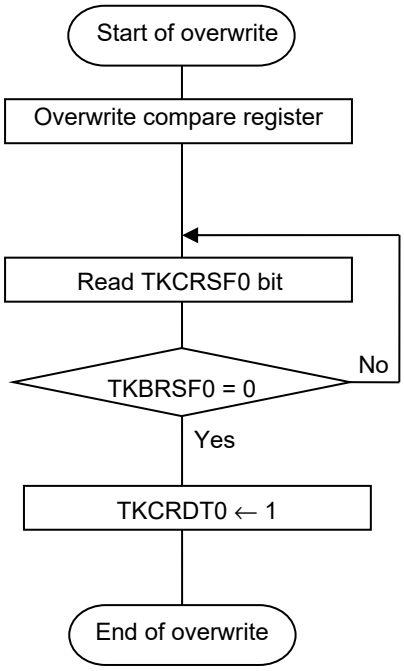


Figure 8-15. Timer KC Operation Setting Example (Operation Stop Flow)



Timer count stops and timer KC output becomes fixed output at the set default level.
(When using in simultaneous start/stop mode, clearing of timer operation enable bit TKBCE0 of the timer KB also stops operation of the timer KC.)

Figure 8-16. Timer KC Operation Setting Example (Compare Register Batch Overwrite Flow)



TKBCNTn, TKBCRn0-TKBCRn3, TKBTGCRn (n = 0, 1, 2)
Overwrite TKBDNRn0, TKBDNRn1, TKBCRLDn0, and TKBCRLDn1 register.

Read TKCRSF0 of TKCFLG0 register.
Accept batch overwrite request when TKCRSF0 = 0.

Compare register batch overwrite request

After accepting overwrite request, TKCRSF0 bit remains “1” until setting is actually changed.

Remark The batch overwrite function is used to change the timer counter operation setting while timer KC is operating. The set value is reflected to the operation from the next restart.

8.4.1 PWM output function

16-bit timer KC0 can output six PWM waveforms with same period and different duty values can be output.

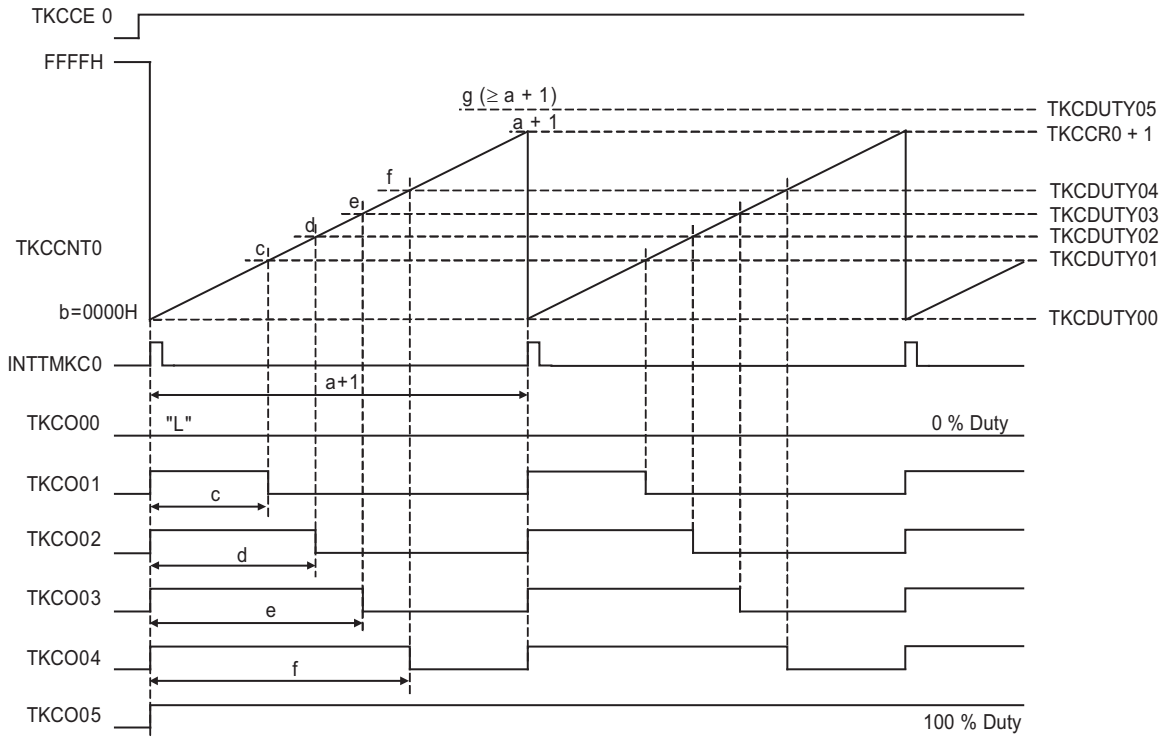
Duty can be set individually within range of 0% to 100% and the TKCO0m period and Duty can be calculated using the following formula.

Pulse period = (TKCCR0 setting + 1) × Counter clock period
Duty [%] = (TKCDUTY0m setting/(TKCCR0 setting + 1)) × 100

0% output: TKCDUTY0m setting = 0000H
100% output: TKCDUTY0m setting ≤ TKCCR0 setting + 1

Remark m = 0 to 5

Figure 8-17. Basic Timing Sample (at TKCTOL0m = 0, TKCTOD0m = 0) for PWM Output Function

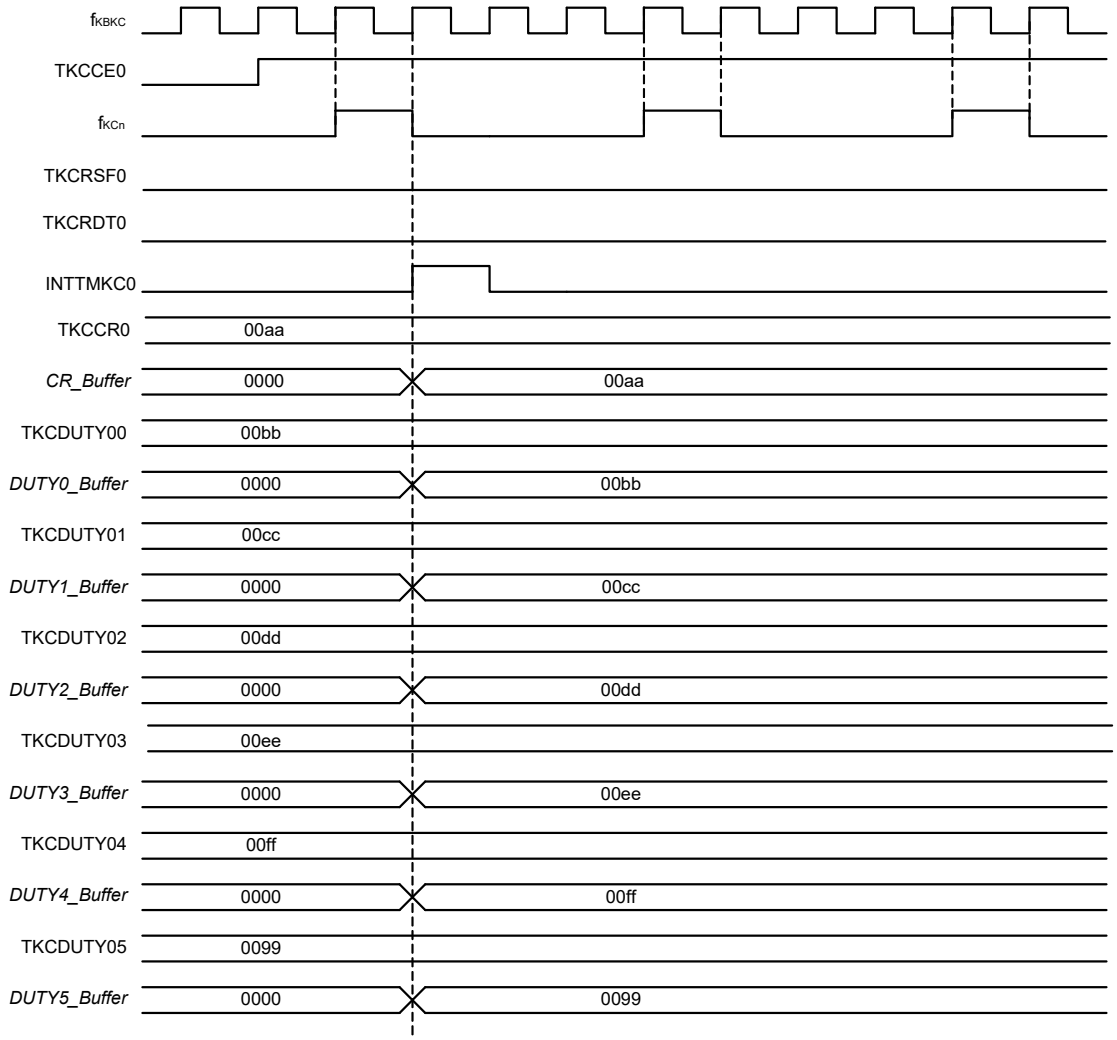


Remark Above figure in TKCO00 (0%) to TKCO05 (100%)

(1) Batch overwrite function: Update buffer at starting the counting operation

Compare register of the timer KC0 have function which update internal buffer register simultaneously at the starting of counter operation caused by count clock which is generated after overwriting “1” to TKCCE0 bit. Batch overwrite is generated without writing “1” on TKCRDT0 bit only in case of counting operation start timing.

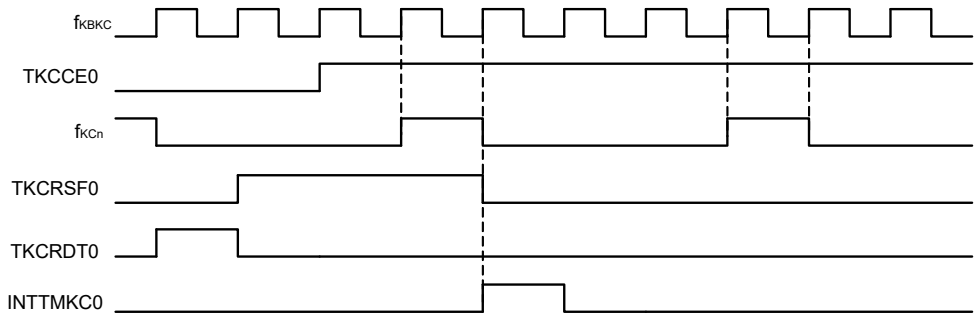
Figure 8-18. Batch Overwrite Function: Figure of Buffer Updating Timing at Counting Operation Start (TKCTOL0m = 0, TKCTOD0m = 0)



When TKCCE0 = 0, TKCRSF0 will be set to “1” at writing “1” to TKCRDT0.

TKCRSF0 will be cleared to “0” at counting operation start timing (counter start trigger generated).

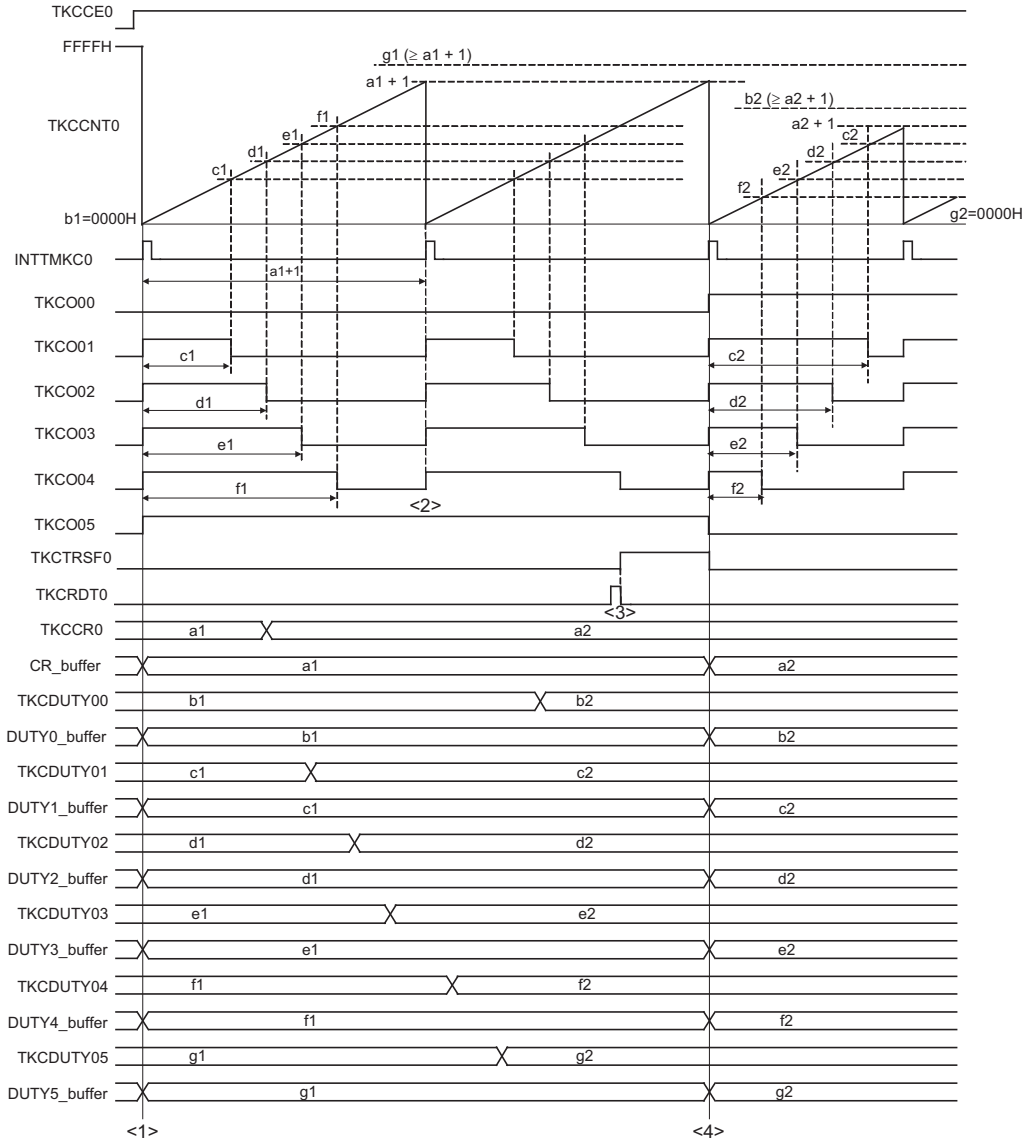
Figure 8-19. Batch Overwrite Function: Figure of Buffer Updating Timing at Counting Operation Start



(2) Batch overwrite function: Update buffer during counting operation

Compare register of the timer KC0 has a function which update internal buffer register simultaneously at the next counter clear (TKCCNT0 and TKCCR0 matched), identifying the writing “1” to TKCRDT0 bit as batch overwriting trigger. TKCRSF0 is provided as flag to indicate from writing of “1” to TKCRDT0 bit until the completion of batch overwrite. For details, see **Figure 8-20**.

Figure 8-20. Batch Overwrite Function: Figure of the Timing of Buffer Updating During Counting Operation (TKCTOL0m = 0, TKCTOD0m = 0)



<1>: Compare register setting will be transferred to buffer register at the timing when TKCCE0 is set from “0” to “1” and TKCCNT0 starts counting operation.

<2>: Overwriting will be not generated if writing of “1” to TKCRDT0 is not implemented even counter clear is generated after TKCCR0 and TKCDUTY05 to TKCDUTY00 registers are overwritten.

<3>: Batch overwrite pending flag (TKCRSF0) will be “1” by writing “1” to TKCRDT0.

<4>: Compare register setting will be transferred to buffer register by counter clear generated at TKCRSF0 = 1. TKCRSF0 will be “0” simultaneously.

8.4.2 Stop/restart operation

Stop and restart of operation of 16-bit timer KC0 will be available by controlling TKCCE0.

(1) 16-bit timer KC0 will be reset and stop operation by changing TKCCE0 from “1” to “0”.
Counter TKCCNT0 will be reset to FFFFH and stop operation then.
TKCO0m output will output default level set by TKCTOD0m.

(2) 16-bit timer KC0 will start operation by changing TKCCE0 from “0” to “1”.
Counter TKCCNT0 will maintain FFFFH when TKCCE0 = 0 and start up counting operation by changing TKCCE0 from “0” to “1”.
TKCO0m output will follow the settings of TKCTOE0m/TKCTOL0m/TKCTOD0m to output waveform.

Figure 8-21. Figure of Timing of Stop/Restart Operation (TKCTOL0m = 0, TKBTOD0m = 0)

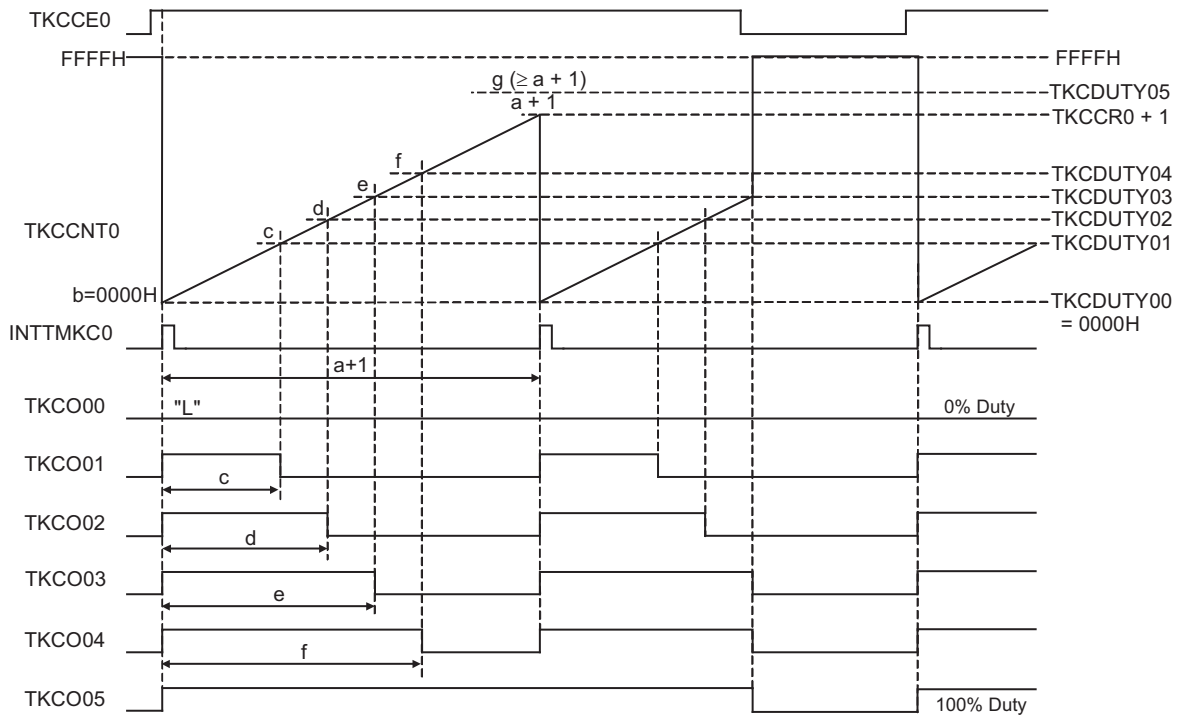
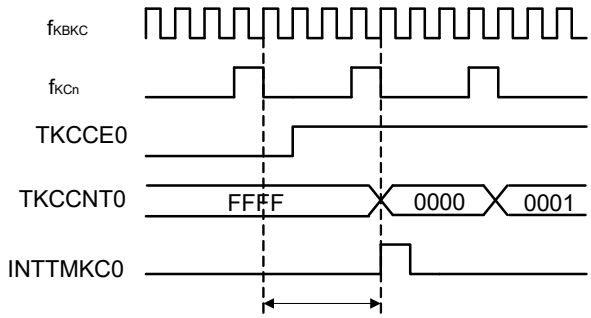
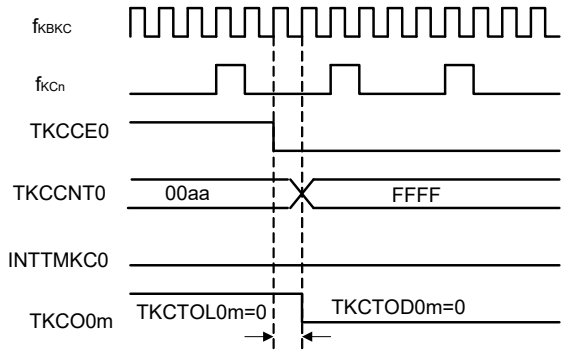


Figure 8-22. Count Operation Start Timing

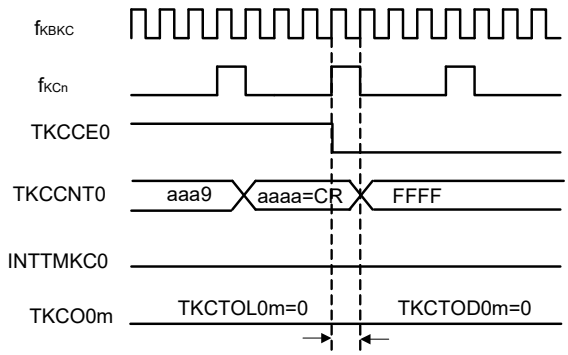


When TKCCE0 is switched from “0” to “1”, counting operation will start after the progress of the minimum 1 f_{KBKC} to the maximum 1 f_{KCn}.
INTTMKC0 is output at counting operation start timing.

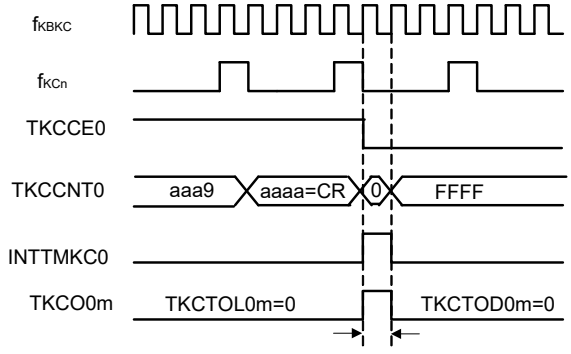
Figure 8-23. Count Operation Stop Timing



When TKCCE0 is switched from “1” to “0”, counting operation will be stopped after the progress of minimum 1 f_{KBKC}.
TKCCNT0 will be reset to FFFFH and TKCO0m will be default level set by TKCTOD0m.



When TKCCE0 is switched from “1” to “0”, counting operation will be stopped after the progress of minimum 1 f_{KBKC}.
Before the generation of 1 f_{KCn}, INTTMKC0 will not be output even matching of TKCCNT0 = TKCCR0 being generated.



TKCO0m will be set and INTTMKC0 being output via the matching of TKCCNT0 = TKCCR0 generated in case when the operation of TKCCE0 is simultaneous with the generation of 1 f_{KCn}.

TKCCNT0 will be reset to FFFFH after the progress of 1 f_{KBKC} and TKCO0m will be default level set by TKCTOD0m.

8.4.3 Default level and active level

Default level and active level settings are available for timer KC output by 16-bit timer KC output control register 00 (TKCIOC00).

When TKCTOE0m is switched from “1” to “0”, default level is output for TKCO0m according to TKCTOD0m setting.

When TKCTOE0m is switched from “0” to “1”, PWM waveform is output according to the generation of TKCO0m set condition/reset condition and TKCTOL0m setting.

Figure 8-24. TKCTOE0m = 0 Switched to 1 Before PWM Reset Timing (Matching Duty)

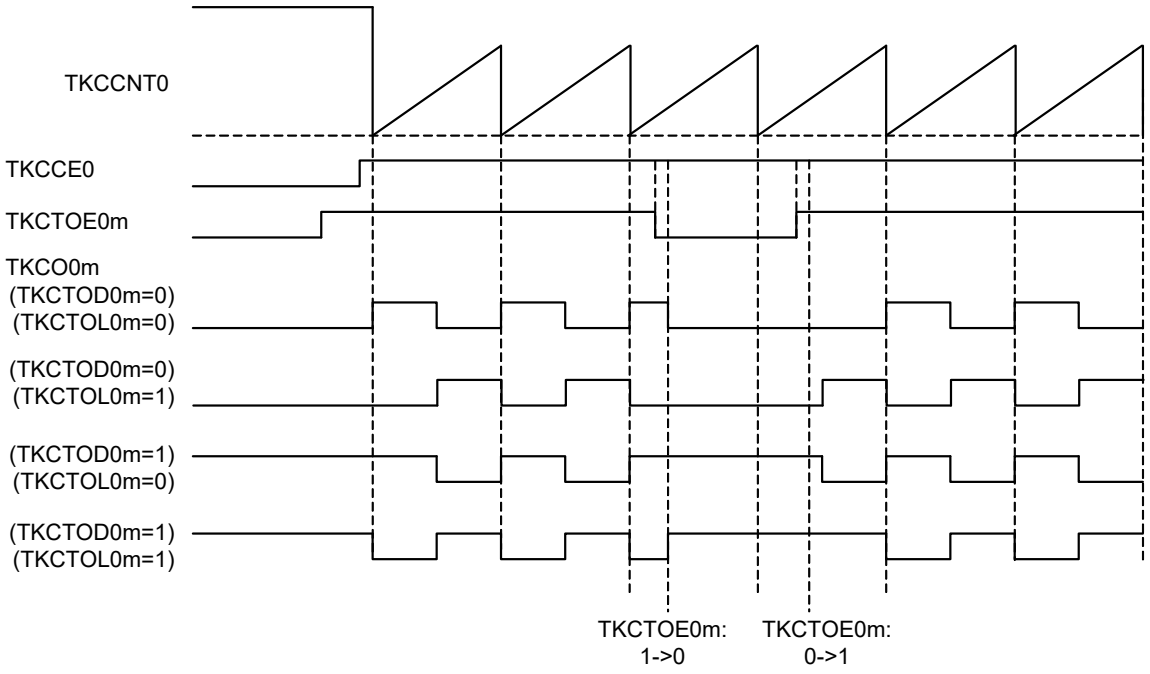


Figure 8-25. TKCTOE0m = 0 Switched to 1 After PWM Reset Timing (Matching Duty)

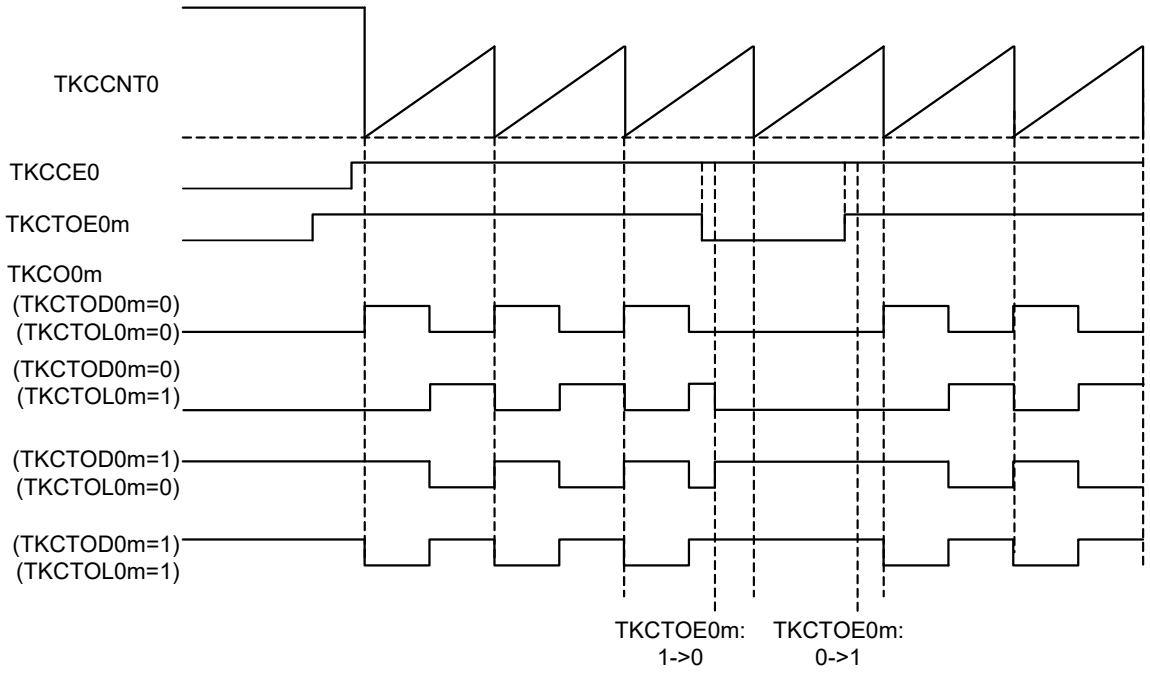
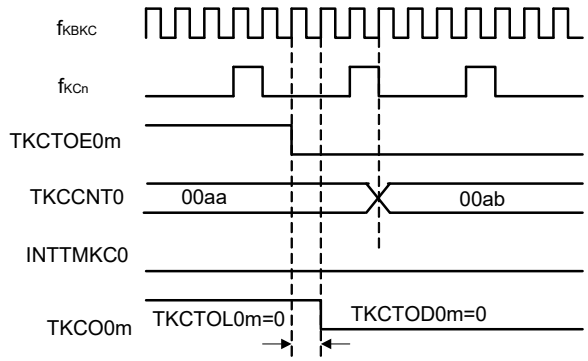
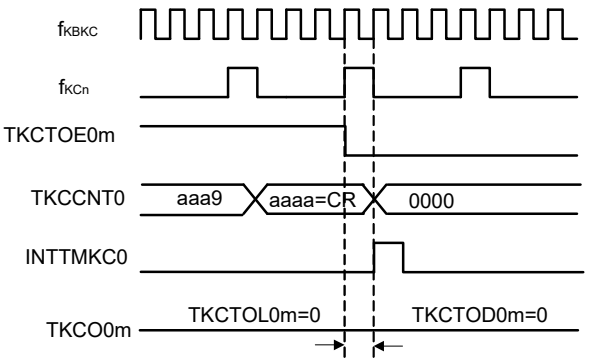


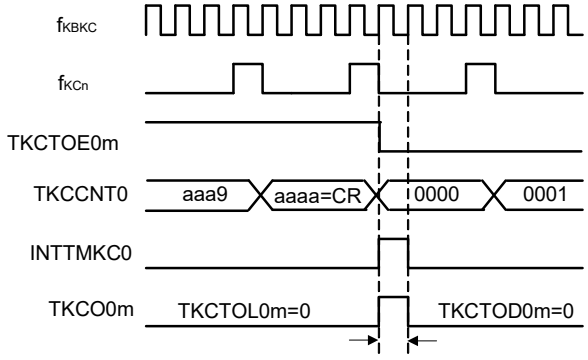
Figure 8-26. TKCTOE0m Switched from “1” to “0”



TKCO0m will be default level set by TKCTOD0m after 1 f_{KBKC} time when TKCTOE0m is switched from “1” to “0”.



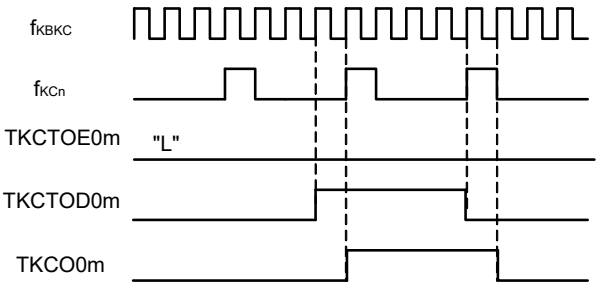
When TKCO0m set timing (Low level to High level) is simultaneous with the matching between TKCCNT0 and TKCCR0, the change of TKCTOE0m will be given priority to become default level set for TKCTOD0m.



TKCO0m will be set by the matching of TKCCNT0 = TKCCR0 in case when the operation of TKCTOE0m is simultaneous with generation of 1 f_{KCn}.

After 1 f_{KBKC} time, TKCO0m will be default level which is set with TKCTOD0m.

Figure 8-27. Change TKCTOD0m at TKCTOE0m = 0



TKCO0m will be default level set by TKCTOD0m after 1 f_{KBKC} time when TKCTOE0m is switched from “1” to “0”.

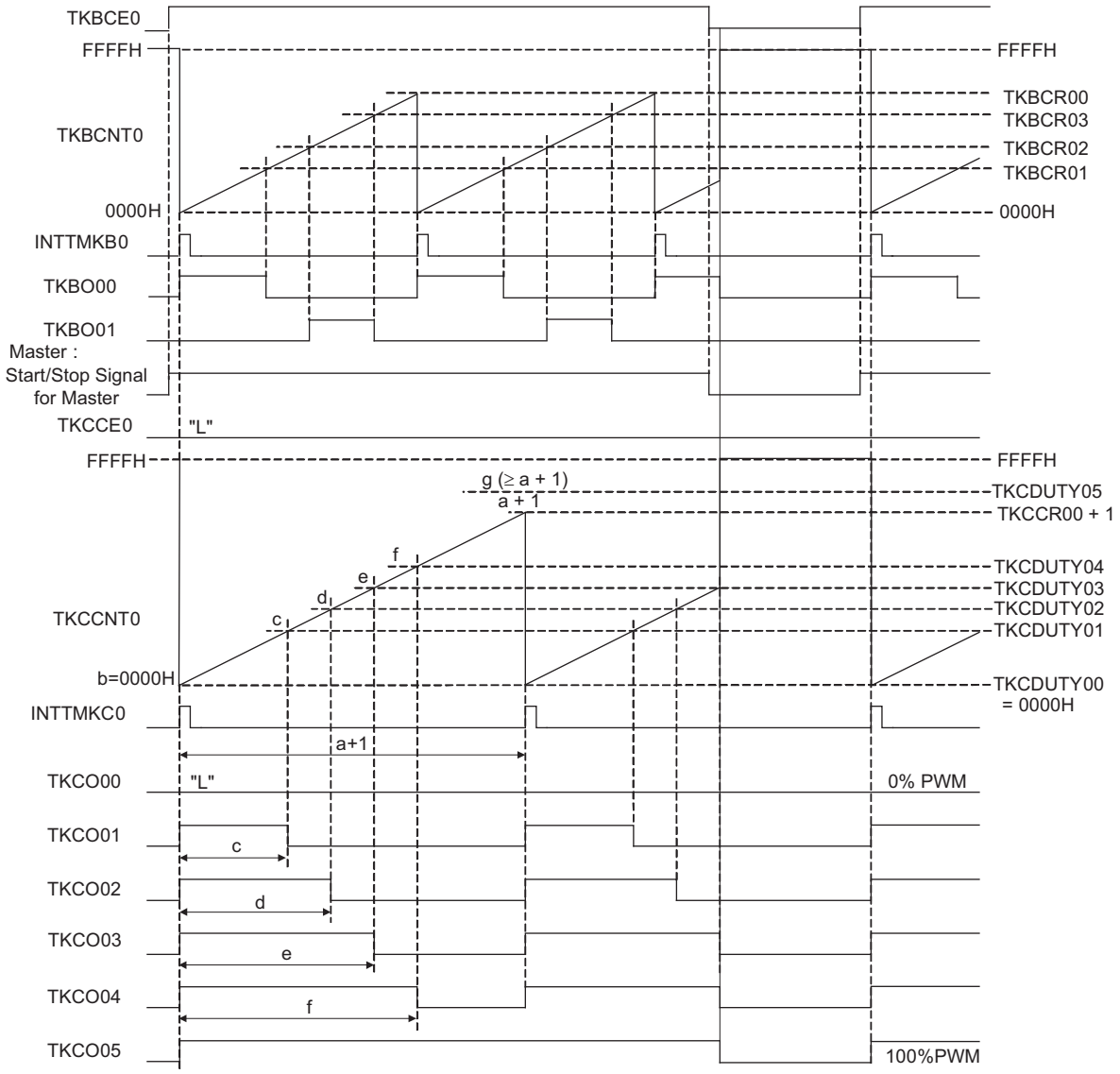
8.4.4 Simultaneous start & stop mode

Slave timer KC0 can be start/stop simultaneously by synchronization with count start/stop of master timer KB0 when master/slave is configured using 16-bit timer KB0/KC0.

Select “Standalone Operation Mode (TKBMDn1, TKBMDn0 = 0, 0)” for master and “Simultaneous Start & Stop Mode (TKCMD0 = 1)” for slave in this case.

For the formula to calculate TKCO05 to TKCO00 outputs see **8.4.1 PWM output function**.

Figure 8-28. Basic Timing Sample (at TKCTOL0m = 0, TKCTOD0m = 0) for Simultaneous Start & Stop Mode



CHAPTER 9 REAL-TIME CLOCK

9.1 Functions of Real-time Clock

The real-time clock has the following features.

- Having counters of year, month, week, day, hour, minute, and second, and can count up to 99 years.
- Constant-period interrupt function (period: 0.5 seconds, 1 second, 1 minute, 1 hour, 1 day, 1 month)
- Alarm interrupt function (alarm: week, hour, minute) (38-pin products only).
- Pin output function of 1 Hz (38-pin products only).

The real-time clock interrupt signal (INTRTC) can be utilized for wakeup from STOP mode and triggering an A/D converter's SNOOZE mode.

Cautions 1. The count of year, month, week, day, hour, minutes and second can only be performed when a subsystem clock (f_{SUB} = 32.768 kHz) is selected as the operation clock of the real-time clock. When the low-speed on-chip oscillator clock (f_{IL} = 15 kHz) is selected, only the constant-period interrupt function is available. The 20-pin products have the constant-period interrupt function only, because these products have no subsystem clock.
However, the constant-period interrupt interval when f_{IL} is selected will be calculated with the constant-period (the value selected with RTCC0 register) × f_{SUB}/f_{IL}.
2. Most of the following descriptions in this chapter use the 38-pin as an example.

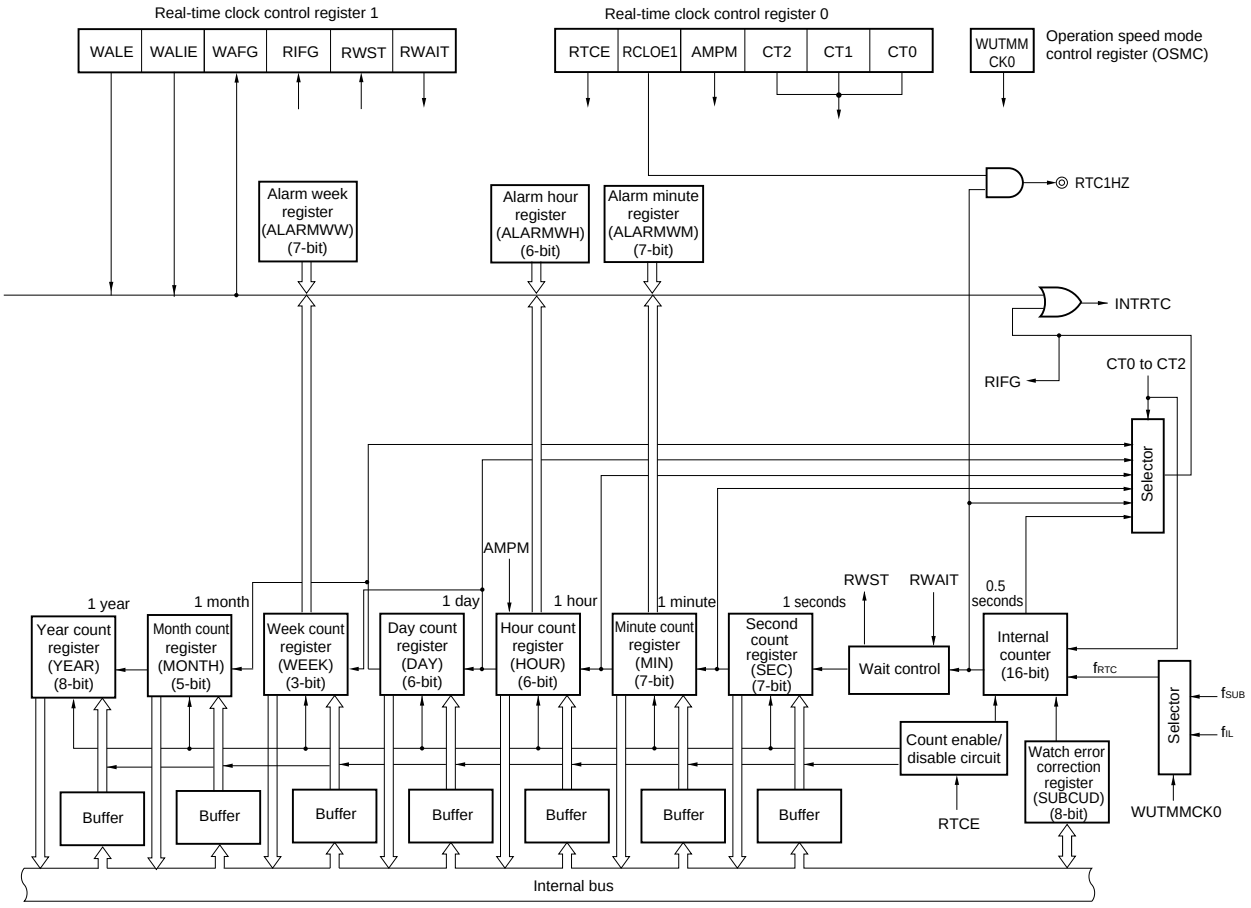
9.2 Configuration of Real-time Clock

The real-time clock includes the following hardware.

Table 9-1. Configuration of Real-time Clock

Item	Configuration
Counter	Internal counter (16-bit)
Control registers	Peripheral enable register 0 (PER0)
	Subsystem clock supply mode control register (OSMC)
	Real-time clock control register 0 (RTCC0)
	Real-time clock control register 1 (RTCC1)
	Second count register (SEC)
	Minute count register (MIN)
	Hour count register (HOUR)
	Day count register (DAY)
	Week count register (WEEK)
	Month count register (MONTH)
	Year count register (YEAR)
	Watch error correction register (SUBCUD)
	Alarm minute register (ALARMWM)
	Alarm hour register (ALARMWH)
	Alarm week register (ALARMWW)

Figure 9-1. Block Diagram of Real-time Clock



Caution The count of year, month, week, day, hour, minutes and second can only be performed when a subsystem clock (f_{SUB} = 32.768 kHz) is selected as the operation clock of the real-time clock. When the low-speed on-chip oscillator clock (f_{IL} = 15 kHz) is selected, only the constant-period interrupt function is available. The 20-pin products have the constant-period interrupt function only, because these products have no subsystem clock.

However, the constant-period interrupt interval when f_{IL} is selected will be calculated with the constant-period (the value selected with RTCC0 register) × f_{SUB}/f_{IL}.

9.3 Registers Controlling Real-time Clock

The real-time clock is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Subsystem clock supply mode control register (OSMC)
- Real-time clock control register 0 (RTCC0)
- Real-time clock control register 1 (RTCC1)
- Second count register (SEC)
- Minute count register (MIN)
- Hour count register (HOUR)
- Day count register (DAY)
- Week count register (WEEK)
- Month count register (MONTH)
- Year count register (YEAR)
- Watch error correction register (SUBCUD)
- Alarm minute register (ALARMWM)
- Alarm hour register (ALARMWH)
- Alarm week register (ALARMWW)
- Port mode register 3 (PM3)
- Port register 3 (P3)

9.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the real-time clock is used, be sure to set bit 7 (RTCEN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Figure 9-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	3	<2>	1	<0>
PER0	RTCEN	0	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

RTCEN	Control of real-time clock (RTC) and 12-bit interval timer input clock supply
0	Stops input clock supply. • SFR used by the real-time clock (RTC) and 12-bit interval timer cannot be written. • The real-time clock (RTC) and 12-bit interval timer are in the reset status.
1	Enables input clock supply. • SFR used by the real-time clock (RTC) and 12-bit interval timer can be read and written.

Cautions 1. When using the real-time clock, first set the RTCEN bit to 1 and then set the following registers, while oscillation of the count clock (f_{RTC}) is stable. If RTCEN = 0, writing to the control registers of the real-time clock is ignored, and, even if the registers are read, only the default values are read (except for the subsystem clock supply mode control register (OSMC), port mode register 3 (PM3), port register 3 (P3)).

- **Real-time clock control register 0 (RTCC0)**
- **Real-time clock control register 1 (RTCC1)**
- **Second count register (SEC)**
- **Minute count register (MIN)**
- **Hour count register (HOUR)**
- **Day count register (DAY)**
- **Week count register (WEEK)**
- **Month count register (MONTH)**
- **Year count register (YEAR)**
- **Watch error correction register (SUBCUD)**
- **Alarm minute register (ALARMWM)**
- **Alarm hour register (ALARMWH)**
- **Alarm week register (ALARMWW)**

2. The subsystem clock supply to peripheral functions other than the real-time clock and 12-bit interval timer can be stopped in STOP mode or HALT mode when the subsystem clock is used, by setting the RTCLPC bit of the subsystem clock supply mode control register (OSMC) to "1".

3. Be sure to clear bits 1, 3, and 6 to "0".

9.3.2 Subsystem clock supply mode control register (OSMC)

The WUTMMCK0 bit can be used to select the real-time clock count clock (f_{RTC}).

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 **CHAPTER 5 CLOCK GENERATOR**.

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

Reset signal generation clears this register to 00H.

Figure 9-3. Format of Subsystem Clock Supply Mode Control Register (OSMC)

Address: F00F3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
OSMC	RTCLPC	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	Selection of operation clock (f_{RTC}) for real-time clock and 12-bit interval timer.
0	Subsystem clock (f_{SUB})
1	Low-speed on-chip oscillator clock (f_{IL})

Cautions 1. The count of year, month, week, day, hour, minutes and second can only be performed when a subsystem clock (f_{SUB} = 32.768 kHz) is selected as the operation clock of the real-time clock. When the low-speed on-chip oscillator clock (f_{IL} = 15 kHz) is selected, only the constant-period interrupt function is available. The 20- to 32-pin products have the constant-period interrupt function only, because these products have no subsystem clock.

However, the constant-period interrupt interval when f_{IL} is selected will be calculated with the constant-period (the value selected with RTCC0 register) × f_{SUB}/f_{IL}.

2. Most of the following descriptions in this chapter use the 38-pin as an example.

9.3.3 Real-time clock control register 0 (RTCC0)

The RTCC0 register is an 8-bit register that is used to start or stop the real-time clock operation, control the RTC1HZ pin, and set a 12- or 24-hour system and the constant-period interrupt function.

The RTCC0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9-4. Format of Real-time Clock Control Register 0 (RTCC0)

Address: FFF9DH After reset: 00H R/W

Symbol	<7>	6	<5>	4	3	2	1	0
RTCC0	RTCE	0	RCLOE1	0	AMPM	CT2	CT1	CT0

RTCE	Real-time clock operation control
0	Stops counter operation.
1	Starts counter operation.

RCLOE1	RTC1HZ pin output control
0	Disables output of the RTC1HZ pin (1 Hz).
1	Enables output of the RTC1HZ pin (1 Hz).

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 real-time 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 9-2 shows the displayed time digits that are displayed.	

CT2	CT1	CT0	Constant-period interrupt (INTRTC) selection	
			When selecting subsystem clock (f_{SUB})	When selecting low-speed on-chip oscillator clock (f_{IL})
0	0	0	Does not use constant-period interrupt function.	
0	0	1	Once per 0.5 s (synchronized with second count up)	Once per 1.1 s
0	1	0	Once per 1 s (same time as second count up)	Once per 2.2 s
0	1	1	Once per 1 m (second 00 of every minute)	Once per 2.2 m
1	0	0	Once per 1 hour (minute 00 and second 00 of every hour)	Once per 2.2 hour
1	0	1	Once per 1 day (hour 00, minute 00, and second 00 of every day)	Once per 2.2 day
1	1	×	Once per 1 month (Day 1, hour 00 a.m., minute 00, and second 00 of every month)	Once per 2.2 month
When changing the values of the CT2 to CT0 bits while the counter operates (RTCE = 1), rewrite the values of the CT2 to CT0 bits after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, after rewriting the values of the CT2 to CT0 bits, enable interrupt servicing after clearing the RIFG and RTCIF flags.				

Cautions 1. Do not change the value of the RTCLOE1 bit when RTCE = 1.

2. The RTC1HZ pin does not output a 1-Hz signal if the RCLOE1 bit is set to 1 while the RTCE bit is 0.

Remark ×: don't care

9.3.4 Real-time clock control register 1 (RTCC1)

The RTCC1 register is an 8-bit register that is used to control the alarm interrupt function and the wait time of the counter.

The RTCC1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9-5. Format of Real-time Clock Control Register 1 (RTCC1) (1/2)

Address: FFF9EH After reset: 00H 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.
When setting a value to the WALE bit while the counter operates (RTCE = 1) and WALIE = 1, rewrite the WALE bit after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG and RTCIF flags after rewriting the WALE bit. When setting each alarm register (WALIE flag of real-time clock control register 1 (RTCC1), the alarm minute register (ALARMWM), the alarm hour register (ALARMWH), and the alarm week register (ALARMWW)), set match operation to be invalid (“0”) for the WALE bit.	

WALIE	Control of alarm interrupt (INTRTC) function operation
0	Does not generate interrupt on matching of alarm.
1	Generates interrupt on matching of alarm.

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 valid only when WALE = 1 and is set to “1” one cycle of f_{RTC} after matching of the alarm is detected. This flag is cleared when “0” is written to it. Writing “1” to it is invalid.	

RIFG	Constant-period interrupt status flag
0	Constant-period interrupt is not generated.
1	Constant-period interrupt is generated.
This flag indicates the status of generation of the constant-period interrupt. When the constant-period 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 9-5. Format of Real-time Clock Control Register 1 (RTCC1) (2/2)

RWST	Wait status flag of real-time clock
0	Counter is operating.
1	Mode to read or write counter value
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 real-time clock
0	Sets counter operation.
1	Stops SEC to YEAR counters. Mode to read or write counter value
This bit controls the operation of the counter. Be sure to write "1" to it to read or write the counter value. As the internal counter (16-bit) is continuing to run, complete reading or writing within one second and turn back to 0.[Note 1, 2] When reading or writing to the counter is required while generation of the alarm interrupt is enabled, first set the CT2 to CT0 bits to 010B (generating the constant-period interrupt once per 1 second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next constant-period interrupt. When RWAIT = 1, it takes up to 1 clock of f_{RTC} until the counter value can be read or written (RWST = 1). When the internal counter (16-bit) overflowed while RWAIT = 1, it keeps the event of overflow until RWAIT = 0, then counts up. However, when it wrote a value to second count register, it will not keep the overflow event.	

Notes 1. If RWAIT is set to 1 within 1 f_{RTC} clock time after setting RTCE to 1, two operation clocks time (f_{RTC}) might be taken to set RWST bit to 1.

2. If RWAIT is set to 1 within 1 f_{RTC} clock time after return from standby (HALT mode, STOP mode or SNOOZE mode), two operation clocks time (f_{RTC}) might be taken to set RWST bit to 1.

Caution If writing is performed to the RTCC1 register with a 1-bit manipulation instruction, the RIFG flag and WAFG flag may be cleared. Therefore, to perform writing to the RTCC1 register, be sure to use an 8-bit manipulation instruction. To prevent the RIFG flag and WAFG flag from being cleared during writing, disable writing by setting 1 to the corresponding bit. If the RIFG flag and WAFG flag are not used and the value may be changed, the RTCC1 register may be written by using a 1-bit manipulation instruction.

Remarks 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.

2. The internal counter (16 bits) is cleared when the second count register (SEC) is written.

9.3.5 Second count register (SEC)

The SEC register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of seconds.

It counts up when the internal counter (16-bit) overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 cycles of f_{RTC} 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.

Reset signal generation clears this register to 00H.

Figure 9-6. Format of Second Count Register (SEC)

Address: FFF92H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

Remark The internal counter (16 bits) is cleared when the second count register (SEC) is written.

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in 9.4.3 Reading/writing real-time clock.

9.3.6 Minute count register (MIN)

The MIN register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of minutes.

It counts up when the second counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 cycles of f_{RTC} 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.

Reset signal generation clears this register to 00H.

Figure 9-7. Format of Minute Count Register (MIN)

Address: FFF93H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in 9.4.3 Reading/writing real-time clock.

9.3.7 Hour count register (HOUR)

The HOUR register 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.

It counts up when the minute counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 cycles of f_{RTC} 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 real-time 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.

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit (bit 3 of the RTCC0 register) is set to 1 after reset.

Figure 9-8. Format of Hour Count Register (HOUR)

Address: FFF94H After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

Cautions 1. Bit 5 (HOUR20) of the HOUR register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

2. When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in 9.4.3 Reading/writing real-time clock.

Table 9-2 shows the relationship between the setting value of the AMPM bit, the hour count register (HOUR) value, and time.

Table 9-2. Displayed Time Digits

24-Hour Display (AMPM = 1)		12-Hour Display (AMPM = 1)	
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.

9.3.8 Day count register (DAY)

The DAY register is an 8-bit register that takes a value of 1 to 31 (decimal) and indicates the count value of days.

It counts up when the hour counter overflows.

This counter counts as follows.

- 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 2 cycles of f_{RTC} 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.

Reset signal generation clears this register to 01H.

Figure 9-9. Format of Day Count Register (DAY)

Address: FFF96H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in 9.4.3 Reading/writing real-time clock.

9.3.9 Week count register (WEEK)

The WEEK register is an 8-bit register that takes a value of 0 to 6 (decimal) and indicates the count value of weekdays.

It counts up in synchronization with the day counter.

When data is written to this register, it is written to a buffer and then to the counter up to 2 cycles of f_{RTC} 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.

Reset signal generation clears this register to 00H.

Figure 9-10. Format of Week Count Register (WEEK)

Address: FFF95H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

Cautions 1. The value corresponding to the month count register (MONTH) or the day count register (DAY) is not stored in the week count register (WEEK) automatically. After reset release, set the week count register as follow.

Day	WEEK
Sunday	00H
Monday	01H
Tuesday	02H
Wednesday	03H
Thursday	04H
Friday	05H
Saturday	06H

2. When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in 9.4.3 Reading/writing real-time clock.

9.3.10 Month count register (MONTH)

The MONTH register is an 8-bit register that takes a value of 1 to 12 (decimal) and indicates the count value of months.

It counts up when the day counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 cycles of f_{RTC} 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.

Reset signal generation clears this register to 01H.

Figure 9-11. Format of Month Count Register (MONTH)

Address: FFF97H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in 9.4.3 Reading/writing real-time clock.

9.3.11 Year count register (YEAR)

The YEAR register is an 8-bit register that takes a value of 0 to 99 (decimal) and indicates the count value of years.

It counts up when 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 2 cycles of f_{RTC} 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.

Reset signal generation clears this register to 00H.

Figure 9-12. Format of Year Count Register (YEAR)

Address: FFF98H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in 9.4.3 Reading/writing real-time clock.

9.3.12 Watch error correction register (SUBCUD)

This register is used to correct the watch with high accuracy when it is slow or fast by changing the value that overflows from the internal counter (16-bit) to the second count register (SEC) (reference value: 7FFFH).

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

Reset signal generation clears this register to 00H.

Figure 9-13. Format of Watch Error Correction Register (SUBCUD)

Address: FFF99H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SUBCUD	DEV	F6	F5	F4	F3	F2	F1	F0

DEV	Setting of watch error correction timing
0	Corrects watch error when the second digits are at 00, 20, or 40 (every 20 seconds).
1	Corrects watch 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 watch 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 watch error is not corrected. * is 0 or 1. /F5 to /F0 are the inverted values of the corresponding bits (000011 when 111100). 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 watch 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

Remark If a correctable range is –63.1 ppm or lower and 63.1 ppm or higher, set 0 to DEV.

9.3.13 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.

Reset signal generation clears this register to 00H.

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 9-14. Format of Alarm Minute Register (ALARMWM)

Address: FFF9AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

9.3.14 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.

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit (bit 3 of the RTCC0 register) is set to 1 after reset.

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 9-15. Format of Alarm Hour Register (ALARMWH)

Address: FFF9BH After reset: 12H 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)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

9.3.15 Alarm week register (ALARMWW)

This register is used to set date of alarm.

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

Reset signal generation clears this register to 00H.

Figure 9-16. Format of Alarm Week Register (ALARMWW)

Address: FFF9CH After reset: 00H 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							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

9.3.16 Port mode register 3 (PM3)

The PM3 register can be set by a 1-bit or 8-bit manipulation instruction.

Reset signal generation sets this register to FFH.

When using the port 3 as the RTC1HZ pin for output of 1 Hz, set the PM30 bit to “0”.

Figure 9-17. Format of Port Mode Register 3 (PM3)

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	1	PM31	PM30

9.3.17 Port register 3 (P3)

The P3 register can be set by a 1-bit or 8-bit manipulation instruction.

Reset signal generation sets this register to 00H.

When using the port 3 as 1-Hz output to the RTC1Hz pin, set the P30 bit to “0”.

Figure 9-18. Format of Port Register 3 (P3)

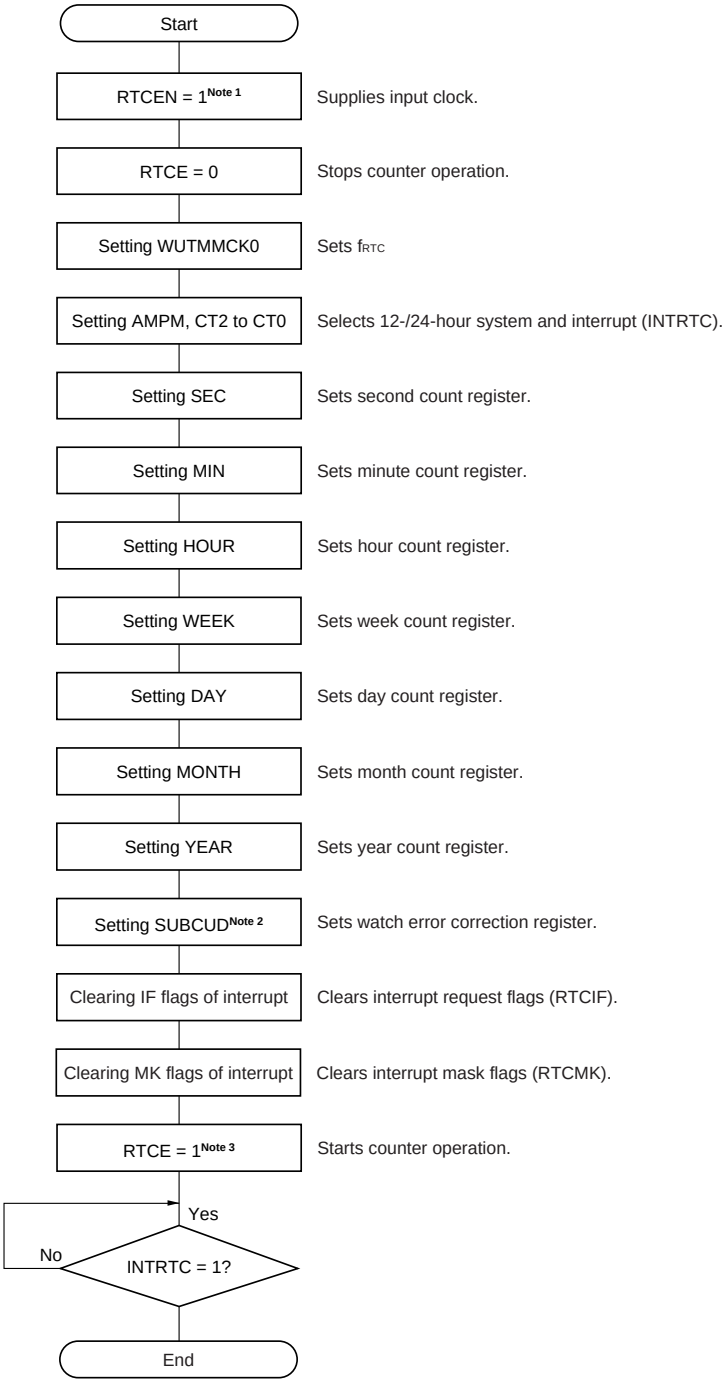
Address: FFF03H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
P3	0	0	0	0	0	0	P31	P30

9.4 Real-time Clock Operation

9.4.1 Starting operation of real-time clock

Figure 9-19. Procedure for Starting Operation of Real-time Clock



Notes 1. First set the RTCEN bit to 1, while oscillation of the count clock (f_{RTC}) is stable.

2. Set up the SUBCUD register only if the watch error must be corrected. For details about how to calculate the correction value, see **9.4.6 Example of watch error correction of real-time clock**.

3. Confirm the procedure described in **9.4.2 Shifting to HALT/STOP mode after starting operation** when shifting to HALT/STOP mode without waiting for INTRTC = 1 after RTCE = 1.

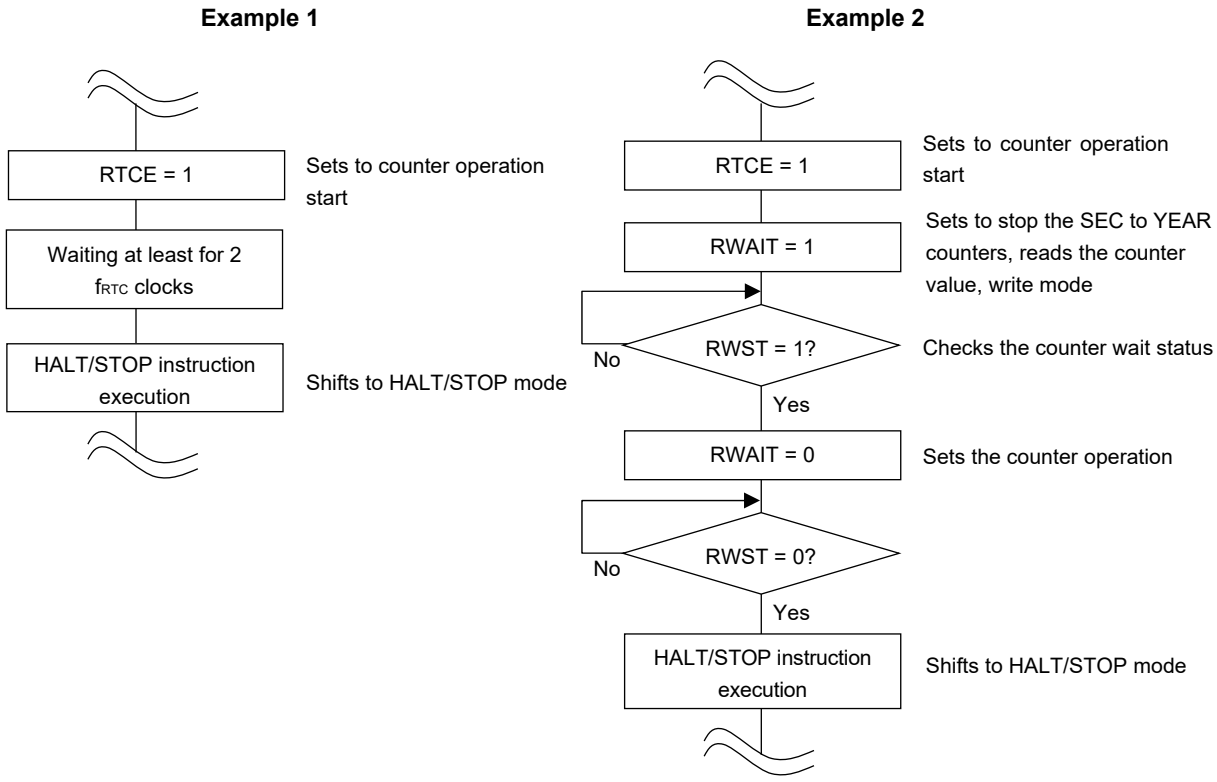
9.4.2 Shifting to HALT/STOP mode after starting operation

Perform one of the following processing when shifting to HALT/STOP mode immediately after setting the RTCE bit to 1.

However, after setting the RTCE bit to 1, this processing is not required when shifting to HALT/STOP mode after the INTRTC interrupt has occurred.

- Shifting to HALT/STOP mode when at least two cycles of the count clock (f_{RTC}) have elapsed after setting the RTCE bit to 1 (see **Figure 9-20**, **Example 1**).
- Checking by polling the RWST bit to become 1, after setting the RTCE bit to 1 and then setting the RWAIT bit to 1. Afterward, setting the RWAIT bit to 0 and shifting to HALT/STOP mode after checking again by polling that the RWST bit has become 0 (see **Figure 9-20**, **Example 2**).

Figure 9-20. Procedure for Shifting to HALT/STOP Mode After Setting RTCE Bit to 1

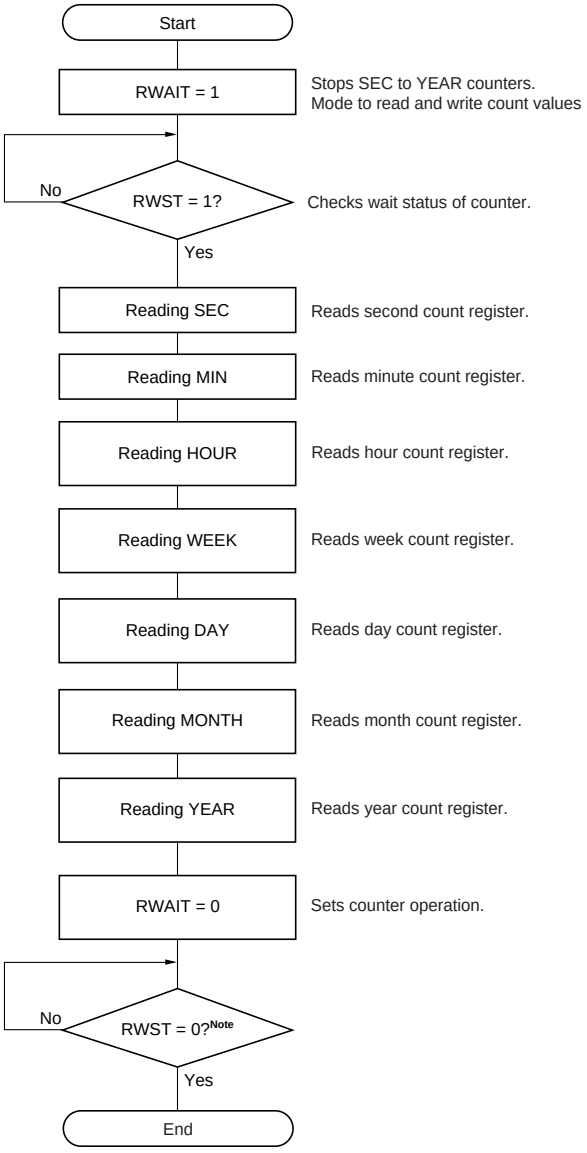


9.4.3 Reading/writing real-time clock

Read or write the counter after setting 1 to RWAIT first.

Set RWAIT to 0 after completion of reading or writing the counter.

Figure 9-21. Procedure for Reading Real-time Clock



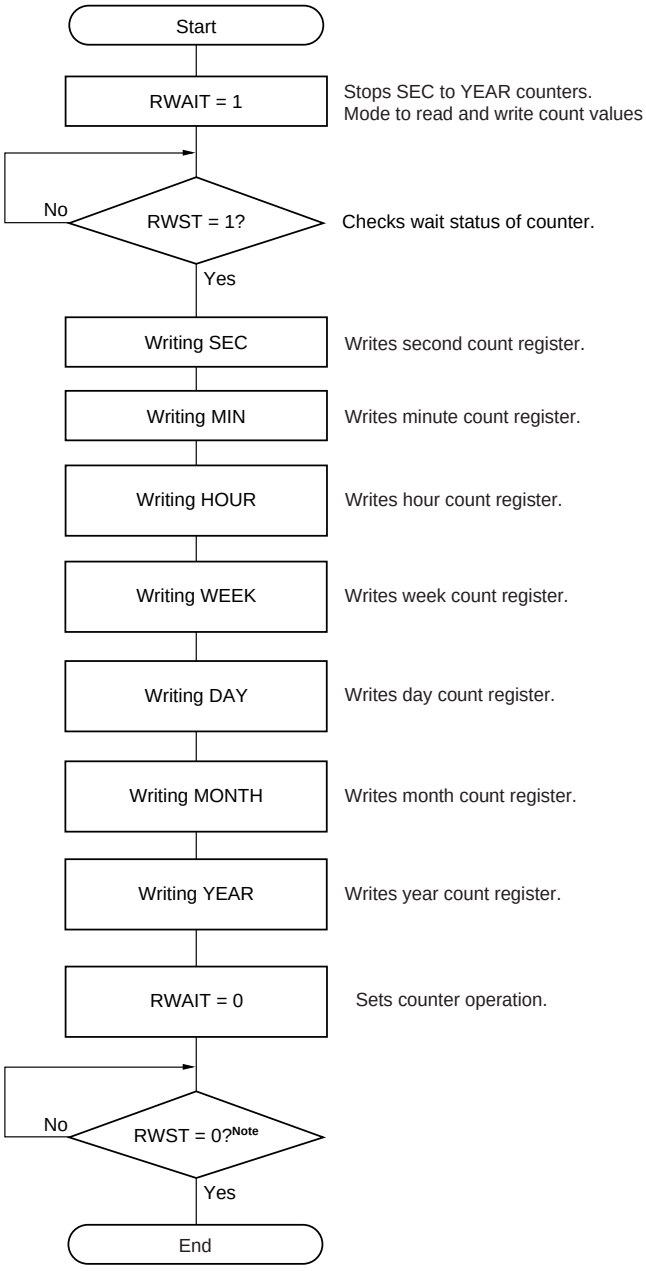
Note Be sure to confirm that RWST = 0 before setting STOP mode.

Caution Complete the series of process of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second. When reading to the counter is required while generation of the alarm interrupt is enabled, first set the CT2 to CT0 bits to 010B (generating the constant-period interrupt once per 1 second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next constant-period interrupt.

Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), 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 read and only some registers may be read.

Figure 9-22. Procedure for Writing Real-time Clock



Note Be sure to confirm that RWST = 0 before setting STOP mode.

Cautions 1. Complete the series of operations of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second. When writing to the counter is required while generation of the alarm interrupt is enabled, first set the CT2 to CT0 bits to 010B (generating the constant-period interrupt once per 1 second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next constant-period interrupt.

2. When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR register while the counter operates (RTCE = 1), rewrite the values of the MIN register after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags after rewriting the MIN register.

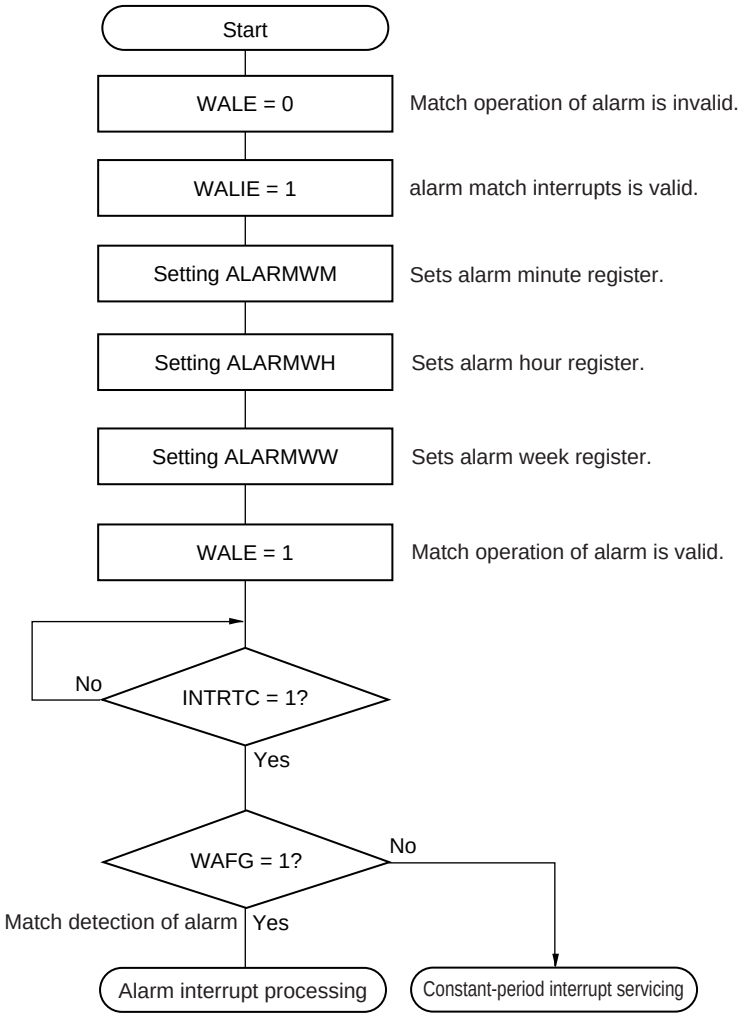
Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be written in any sequence.

All the registers do not have to be set and only some registers may be written.

9.4.4 Setting alarm of real-time clock

Set time of alarm after setting 0 to WALE (alarm operation invalid.) first.

Figure 9-23. Alarm processing Procedure

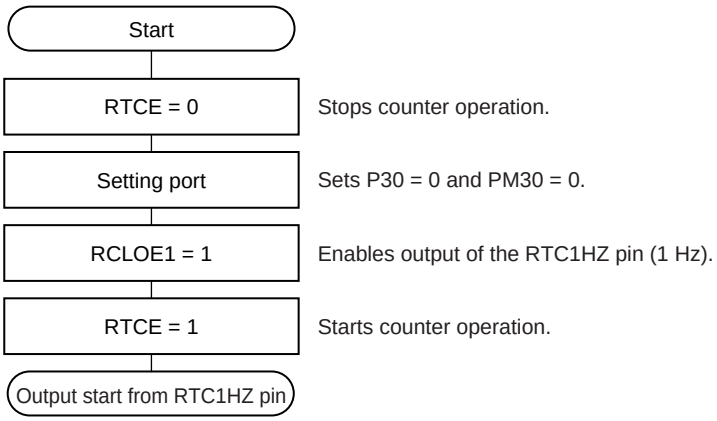


Remarks 1. The alarm week register (ALARMWW), alarm hour register (ALARMWH), and alarm week register (ALARMWW) may be written in any sequence.

2. 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.

9.4.5 1 Hz output of real-time clock

Figure 9-24. 1 Hz Output Setting Procedure



Cautions 1. First set the RTCEN bit to 1, while oscillation of the count clock (f_{SUB}) is stable.

2. Pin output function of 1 Hz is not available in the 20- and 30-pin products.

9.4.6 Example of watch error correction of real-time clock

The watch can be corrected with high accuracy when it is slow or fast, by setting a value to the watch error correction register.

Example of calculating the correction value

The correction value used when correcting the count value of the internal counter (16-bit) 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 watch error correction value calculated by using bits 6 to 0 of the watch 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, *), watch error correction is not performed. "*" is 0 or 1. /F5 to /F0 are bit-inverted values (000011 when 111100).

Remarks 1. The correction value is 2, 4, 6, 8, … 120, 122, 124 or –2, –4, –6, –8, … –120, –122, –124.

2. The oscillation frequency is the count clock (f_{RTC}).
It can be calculated from the output frequency of the RTC1HZ pin × 32768 when the watch error correction register is set to its initial value (00H).

3. The target frequency is the frequency resulting after correction performed by using the watch 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]

The oscillation frequency[Note] of each product is measured by outputting about 1 Hz from the RTC1HZ pin when the watch error correction register (SUBCUD) is set to its initial value (00H).

Note See **9.4.5 1 Hz output of real-time clock** for the setting procedure of the RTC1Hz output.

[Calculating the correction value]

(When the output frequency from the PCLBUZ0 pin is 32772.3 Hz)

Assume the target frequency to be 32768 Hz (32772.3 Hz–131.2 ppm) and DEV to be 0, because the correctable range of –131.2 ppm is –63.1 ppm or lower.

The expression for calculating the correction value when DEV is 0 is applied.

Correction value = Number of correction counts in 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 (when slowing), assume F6 to be 0.

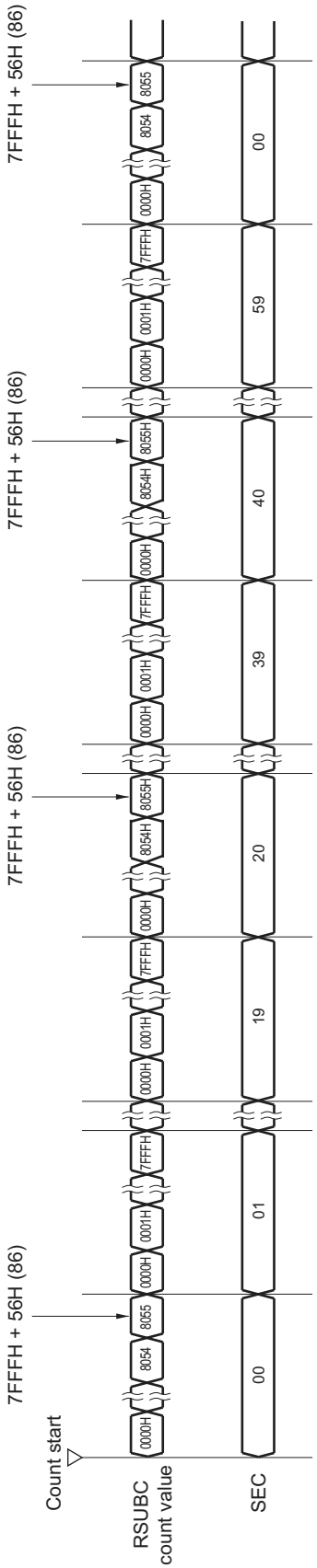
Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

{ (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: 0101100) results in 32768 Hz (0 ppm).

Figure 9-25 shows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (0, 0, 1, 0, 1, 1, 0, 0).

Figure 9-25. Correction Operation when (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]

The oscillation frequency[Note] of each product is measured by outputting about 1 Hz from the RTC1HZ pin when the watch error correction register (SUBCUD) is set to its initial value (00H).

Note See **9.4.5 1 Hz output of real-time clock** for the setting procedure of outputting about 1 Hz from the RTC1HZ pin.

[Calculating the correction value]

(When the output frequency from the RTCCL pin is 0.9999817 Hz)

Oscillation frequency = $32768 \times 0.9999817 \approx 32767.4$ Hz

Assume the target frequency to be 32768 Hz (32767.4 Hz + 18.3 ppm) and DEV to be 1.

The expression for calculating the correction value when DEV is 1 is applied.

Correction value = Number of correction counts in 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 (when quickening), assume F6 to be 1.

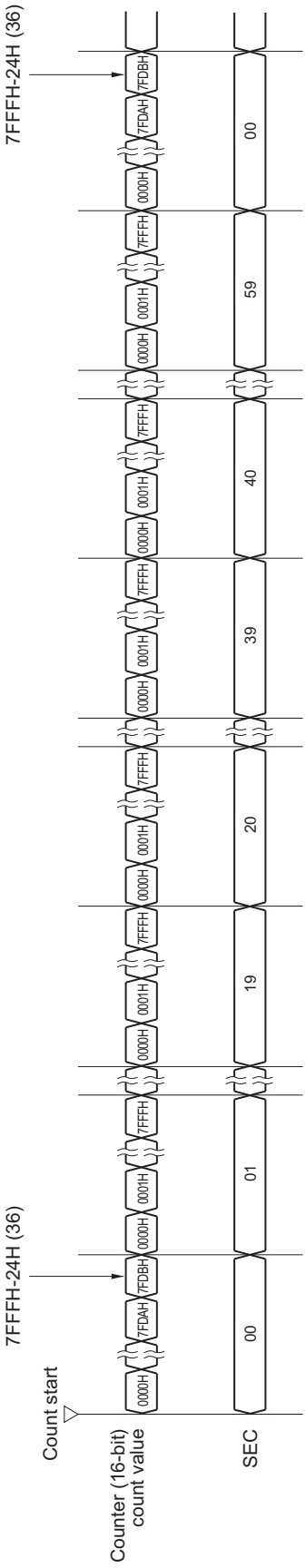
Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

$-\{(/F5, /F4, /F3, /F2, /F1, /F0) - 1\} \times 2$	= −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: 1101110) results in 32768 Hz (0 ppm).

Figure 9-26 shows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (1, 1, 1, 0, 1, 1, 1, 0).

Figure 9-26. Correction Operation When (DEV, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 0, 1, 1, 1, 0)



CHAPTER 10 12-BIT INTERVAL TIMER

10.1 Functions of 12-bit Interval Timer

An interrupt (INTIT) is generated at any previously specified time interval. It can be utilized for wakeup from STOP mode and triggering an A/D converter's SNOOZE mode.

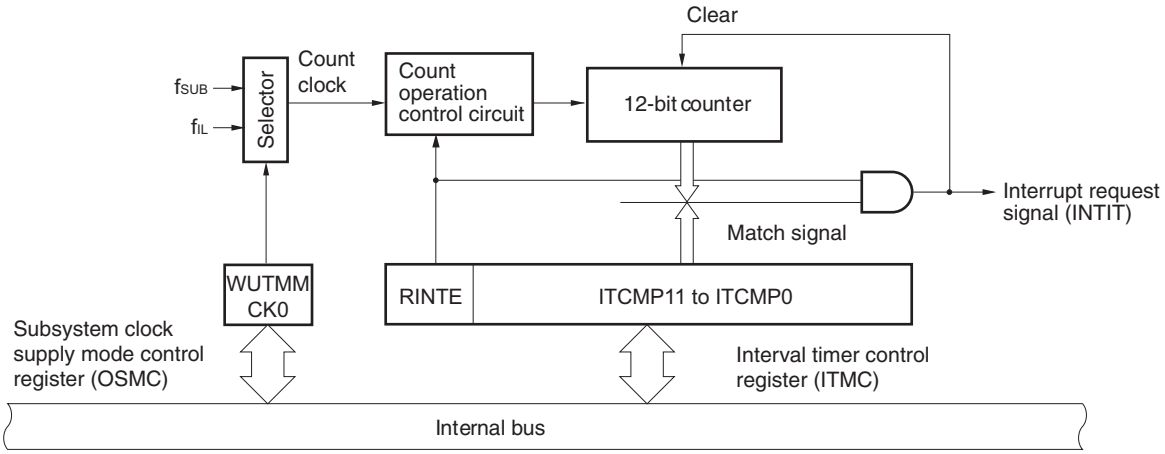
10.2 Configuration of 12-bit Interval Timer

The 12-bit interval timer includes the following hardware.

Table 10-1. Configuration of 12-bit Interval Timer

Item	Configuration
Counter	12-bit counter
Control registers	Peripheral enable register 0 (PER0)
	Subsystem clock supply mode control register (OSMC)
	Interval timer control register (ITMC)

Figure 10-1. Block Diagram of 12-bit Interval Timer



Caution The subsystem clock (f_{SUB}) can be selected as the count clock only for 38-pin products.

10.3 Registers Controlling 12-bit Interval Timer

The 12-bit interval timer is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Subsystem clock supply mode control register (OSMC)
- Interval timer control register (ITMC)

10.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the 12-bit interval timer is used, be sure to set bit 7 (RTCEN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Figure 10-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	3	<2>	1	<0>
PER0	RTCEN	0	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

RTCEN	Control of real-time clock (RTC) and 12-bit interval timer clock supply
0	Stops clock supply. • SFR used by the real-time clock (RTC) and 12-bit interval timer cannot be written. • The real-time clock (RTC) and 12-bit interval timer are in the reset status.
1	Enables clock supply. • SFR used by the real-time clock (RTC) and 12-bit interval timer can be read and written.

Cautions 1. When using the 12-bit interval timer, be sure to first set the RTCEN bit to 1 and then set the following register, while oscillation of the count clock is stable. If RTCEN = 0, writing to the register controlling the 12-bit interval timer is ignored, and, even if the register is read, only the default value is read (except the subsystem clock supply mode control register (OSMC)).

• Interval timer control register (ITMC)

2. Clock supply to peripheral functions other than the real-time clock and 12-bit interval timer can be stopped in STOP mode or HALT mode when the subsystem clock is used, by setting the RTCLPC bit of the subsystem clock supply mode control register (OSMC) to 1.

3. Be sure to clear bits 1, 3, and 6 to "0".

10.3.2 Subsystem clock supply mode control register (OSMC)

The WUTMMCK0 bit can be used to select the 12-bit interval timer or real-time clock operation clock.

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 **CHAPTER 5 CLOCK GENERATOR**.

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

Reset signal generation clears this register to 00H.

Figure 10-3. Format of Subsystem Clock Supply Mode Control Register (OSMC)

Address: F00F3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
OSMC	RTCLPC	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	Selection of operation clock for real-time clock and 12-bit interval timer.
0	Subsystem clock (f_{SUB})
1	Low-speed on-chip oscillator clock (f_{IL})

10.3.3 Interval timer control register (ITMC)

This register is used to set up the starting and stopping of the 12-bit interval timer operation and to specify the timer compare value.

The ITMC register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0FFFH.

Figure 10-4. Format of Interval Timer Control Register (ITMC)

Address: FFF90H After reset: 0FFFH R/W

Symbol	15	14	13	12	11 to 0
ITMC	RINTE	0	0	0	ITMCMP11 to ITMCMP0

RINTE	12-bit interval timer operation control
0	Count operation stopped (count clear)
1	Count operation started

ITMCMP11 to ITMCMP0	Specification of the 12-bit interval timer compare value
001H	These bits generate an interrupt at the fixed cycle (count clock cycles × (ITMCMP setting + 1)).
• • •	
FFFH	
000H	Setting prohibit
Example interrupt cycles when 001H or FFFH is specified for ITMCMP11 to ITMCMP0 • ITMCMP11 to ITMCMP0 = 001H, count clock: when f_{SUB} = 32.768 kHz 1/32.768 [kHz] × (1 + 1) = 0.06103515625 [ms] ≅ 61.03 [μs] • ITMCMP11 to ITMCMP0 = FFFH, count clock: when f_{SUB} = 32.768 kHz 1/32.768 [kHz] × (4095 + 1) = 125 [ms]	

Cautions 1. Before changing the RINTE bit from 1 to 0, use the interrupt mask flag register to disable the INTIT interrupt servicing. When the operation starts (from 0 to 1) again, clear the ITIF flag, and then enable the interrupt servicing.

2. The value read from the RINTE bit is applied one count clock cycle after setting the RINTE bit.

3. When setting the RINTE bit after returned from standby mode and entering standby mode again, confirm that the written value of the RINTE bit is reflected, or wait that more than one clock of the count clock has elapsed after returned from standby mode. Then enter standby mode.

4. Only change the setting of the ITMCMP11 to ITMCMP0 bits when RINTE = 0.
However, it is possible to change the settings of the ITMCMP11 to ITMCMP0 bits at the same time as when changing RINTE from 0 to 1 or 1 to 0.

10.4 12-bit Interval Timer Operation

10.4.1 12-bit interval timer operation timing

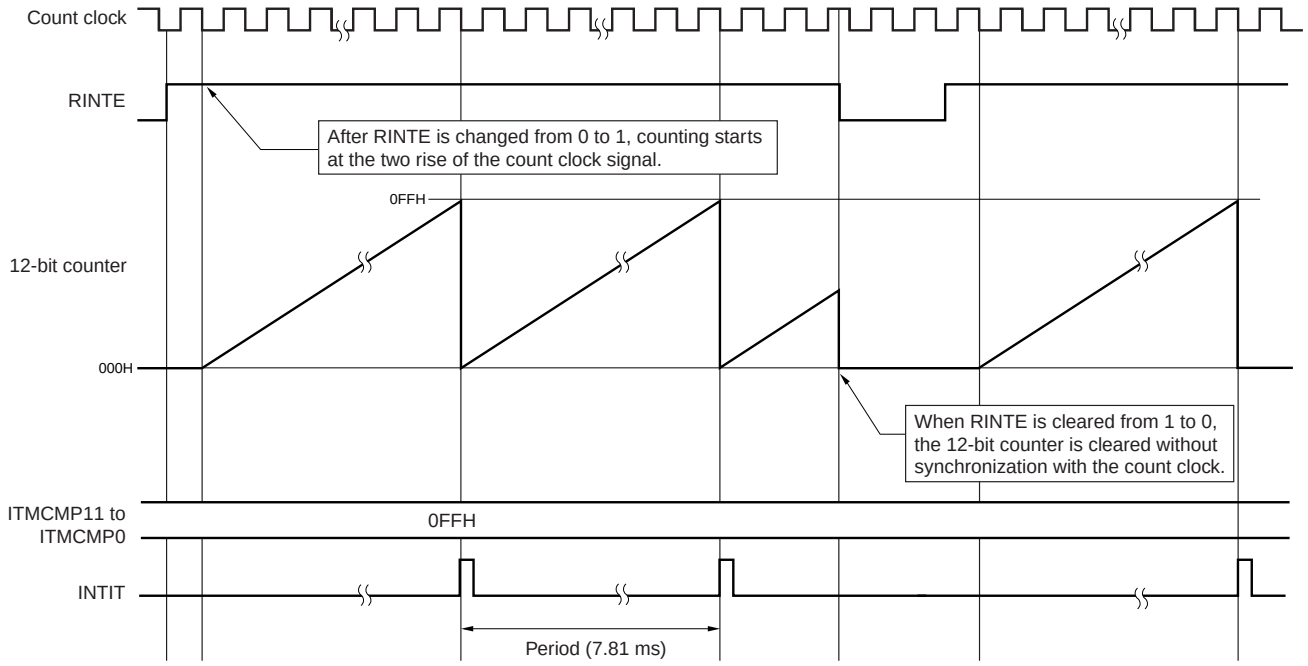
The count value specified for the ITMCMP11 to ITMCMP0 bits is used as an interval to operate a 12-bit interval timer that repeatedly generates interrupt requests (INTIT).

When the RINTE bit is set to 1, the 12-bit counter starts counting.

When the 12-bit counter value matches the value specified for the ITMCMP11 to ITMCMP0 bits, the 12-bit counter value is cleared to 0, counting continues, and an interrupt request signal (INTIT) is generated at the same time.

The basic operation of the 12-bit interval timer is as follows.

Figure 10-5. 12-bit Interval Timer Operation Timing
(ITMCMP11 to ITMCMP0 = 0FFH, count clock: f_{SUB} = 32.768 kHz)

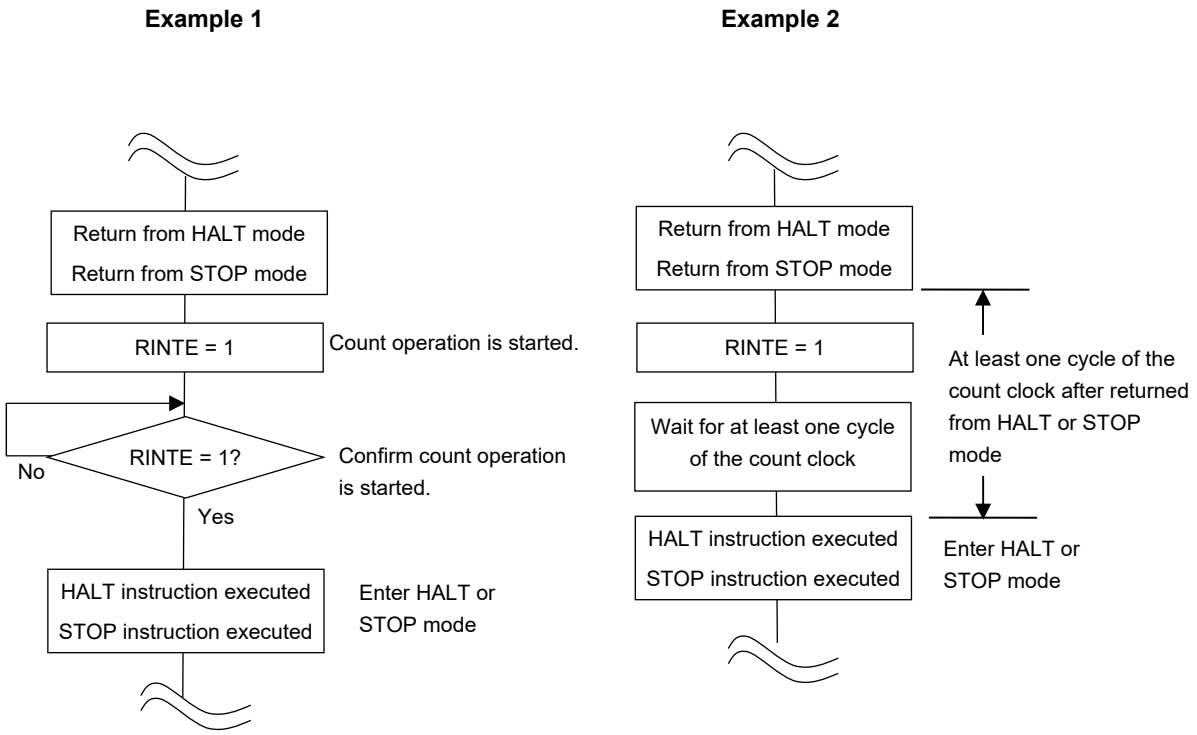


10.4.2 Start of count operation and re-enter to HALT/STOP mode after returned from HALT/STOP mode

When setting the RINTE bit after returned from HALT or STOP mode and entering HALT or STOP mode again, write 1 to the RINTE bit, and confirm the written value of the RINTE bit is reflected or wait for at least one cycle of the count clock. Then, enter HALT or STOP mode.

- After setting RINTE to 1, confirm by polling that the RINTE bit has become 1, and then enter HALT or STOP mode (see **Example 1** in **Figure 10-6**).
- After setting RINTE to 1, wait for at least one cycle of the count clock and then enter HALT or STOP mode (see **Example 2** in **Figure 10-6**).

Figure 10-6. Procedure of Entering to HALT or STOP Mode After Setting RINTE to 1



CHAPTER 11 WATCHDOG TIMER

11.1 Functions of Watchdog Timer

The counting operation of the watchdog timer is set by the option byte (000C0H).

The watchdog timer operates on the low-speed on-chip oscillator clock (f_{IL}).

The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.

Program loop is detected in the following cases.

- If the watchdog timer counter overflows
- If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
- If data other than “ACH” is written to the WDTE register
- If data is written to the WDTE register during a window close period

When a reset occurs due to the watchdog timer, bit 4 (WDTRF) of the reset control flag register (RESF) is set to 1. For details of the RESF register, see **CHAPTER 22 RESET FUNCTION**.

When 75% + 1/2 f_{IL} of the overflow time is reached, an interval interrupt can be generated.

11.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 11-1. Configuration of Watchdog Timer

Item	Configuration
Counter	Internal counter (17 bits)
Control register	Watchdog timer enable register (WDTE)

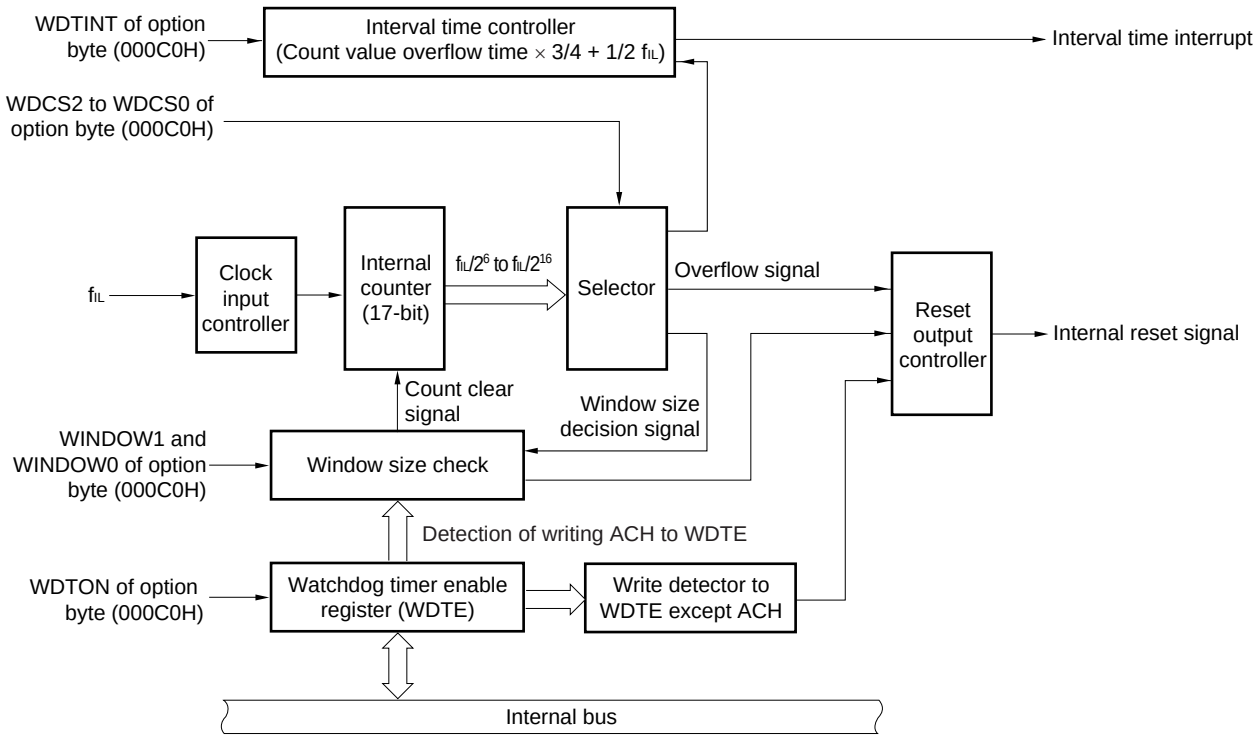
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

Table 11-2. Setting of Option Bytes and Watchdog Timer

Setting of Watchdog Timer	Option Byte (000C0H)
Watchdog timer interval interrupt	Bit 7 (WDTINT)
Window open period	Bits 6 and 5 (WINDOW1, WINDOW0)
Controlling counter operation of watchdog timer	Bit 4 (WDTON)
Overflow time of watchdog timer	Bits 3 to 1 (WDCS2 to WDCS0)
Controlling counter operation of watchdog timer (in HALT/STOP mode)	Bit 0 (WDSTBYON)

Remark For the option byte, see **CHAPTER 27 OPTION BYTE**.

Figure 11-1. Block Diagram of Watchdog Timer



Remark f_{IL}: Low-speed on-chip oscillator clock frequency

11.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

11.3.1 Watchdog timer enable register (WDTE)

Writing “ACH” to the WDTE register clears the watchdog timer counter and starts counting again.

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

Reset signal generation sets this register to 9AH or 1AH[Note].

Figure 11-2. Format of Watchdog Timer Enable Register (WDTE)

Address: FFFABH After reset: 9AH/1AH[Note] R/W

Symbol	7	6	5	4	3	2	1	0
WDTE								

Note The WDTE register reset value differs depending on the WDTON bit setting value of the option byte (000C0H). To operate watchdog timer, set the WDTON bit to 1.

WDTON Bit Setting Value	WDTE Register Reset Value
0 (watchdog timer count operation disabled)	1AH
1 (watchdog timer count operation enabled)	9AH

Cautions 1. If a value other than “ACH” is written to the WDTE register, an internal reset signal is generated.

2. If a 1-bit memory manipulation instruction is executed for the WDTE register, an internal reset signal is generated.

3. The value read from the WDTE register is 9AH/1AH (this differs from the written value (ACH)).

11.4 Operation of Watchdog Timer

11.4.1 Controlling operation of watchdog timer

1. When the watchdog timer is used, its operation is specified by the option byte (000C0H).

   - Enable counting operation of the watchdog timer by setting bit 4 (WDTON) of the option byte (000C0H) to 1 (the counter starts operating after a reset release) (for details, see **CHAPTER 27**).

WDTON	Watchdog Timer Counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

   - Set an overflow time by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H) (for details, see **11.4.2** and **CHAPTER 27**).
   - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (000C0H) (for details, see **11.4.3** and **CHAPTER 27**).

2. After a reset release, the watchdog timer starts counting.
3. By writing “ACH” to the watchdog timer enable register (WDTE) after the watchdog timer starts counting and before the overflow time set by the option byte, the watchdog timer is cleared and starts counting again.
4. After that, write the WDTE register the second time or later after a reset release during the window open period. If the WDTE register is written during a window close period, an internal reset signal is generated.
5. If the overflow time expires without “ACH” written to the WDTE register, an internal reset signal is generated.
   An internal reset signal is generated in the following cases.

   - If a 1-bit manipulation instruction is executed on the WDTE register
   - If data other than “ACH” is written to the WDTE register

Cautions 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.

2. After “ACH” is written to the WDTE register, an error of up to two cycles of f_{IL} may occur before the watchdog timer is cleared.

3. The watchdog timer can be cleared immediately before the count value overflows.

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 option byte (000C0H).

	WDSTBYON = 0	WDSTBYON = 1
In HALT mode	Watchdog timer operation stops.	Watchdog timer operation continues.
In STOP mode		
In SNOOZE mode		

If WDSTBYON = 0, the watchdog timer resumes counting after the HALT or STOP mode is released. At this time, the counter is cleared to 0 and counting starts.
When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed.
Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.
Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.

11.4.2 Setting overflow time of watchdog timer

Set the overflow time of the watchdog timer by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H).

If an overflow occurs, an internal reset signal is generated. The present count is cleared and the watchdog timer starts counting again by writing “ACH” to the watchdog timer enable register (WDTE) during the window open period before the overflow time.

The following overflow times can be set.

Table 11-3. Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer (f_{IL} = 17.25 kHz (MAX.))
0	0	0	$2^6/f_{IL}$ (3.71 ms)
0	0	1	$2^7/f_{IL}$ (7.42 ms)
0	1	0	$2^8/f_{IL}$ (14.84 ms)
0	1	1	$2^9/f_{IL}$ (29.68 ms)
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)
<R> 1	0	1	$2^{13}/f_{IL}$ (474.89 ms) Note
<R> 1	1	0	$2^{14}/f_{IL}$ (949.79 ms) Note
<R> 1	1	1	$2^{16}/f_{IL}$ (3799.18 ms) Note

<R> **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.

Usage conditions that may lead to the generation of an interval interrupt:

- The overflow time of the watchdog timer is set to $2^{13}/f_{IL}$, $2^{14}/f_{IL}$, or $2^{16}/f_{IL}$.
- The interval interrupt is in use (the setting of the WDTINT bit of the relevant option byte is 1).
- 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 (IF0L) to 0.
5. Clear the WDTIMK bit of the interrupt mask flag register 0 (MK0L) to 0.

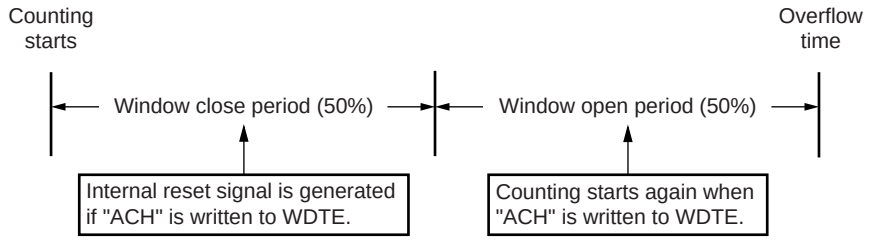
Remark f_{IL}: low-speed on-chip oscillator clock frequency

11.4.3 Setting window open period of watchdog timer

Set the window open period of the watchdog timer by using bits 6 and 5 (WINDOW1, WINDOW0) of the option byte (000C0H). The outline of the window is as follows.

- If “ACH” is written to the watchdog timer enable register (WDTE) during the window open period, the watchdog timer is cleared and starts counting again.
- Even if “ACH” is written to the WDTE register during the window close period, an abnormality is detected and an internal reset signal is generated.

Example: If the window open period is 50%



Caution When data is written to the WDTE register for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

The window open period can be set is as follows.

Table 11-4. Setting Window Open Period of Watchdog Timer

WINDOW1	WINDOW0	Window Open Period of Watchdog Timer
0	0	Setting prohibited
0	1	50%
1	0	75% Note
1	1	100%

Note When the window open period is set to 75%, clearing the counter of the watchdog timer (writing ACH to WDTE) must proceed outside the corresponding period from among those listed below, over which clearing of the counter is prohibited (for example, confirming that the interval timer interrupt request flag (WDTIIF) of the watchdog timer is set).

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{IL} = 17.25 kHz (MAX.))	Period over which clearing the counter is prohibited when the window open period is set to 75%
0	0	0	$2^6/f_{IL}$ (3.71 ms)	1.85 ms to 2.51 ms
0	0	1	$2^7/f_{IL}$ (7.42 ms)	3.71 ms to 5.02 ms
0	1	0	$2^8/f_{IL}$ (14.84 ms)	7.42 ms to 10.04 ms
0	1	1	$2^9/f_{IL}$ (29.68 ms)	14.84 ms to 20.08 ms
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)	56.36 ms to 80.32 ms
1	0	1	$2^{13}/f_{IL}$ (474.89 ms)	237.44 ms to 321.26 ms
1	1	0	$2^{14}/f_{IL}$ (949.79 ms)	474.89 ms to 642.51 ms
1	1	1	$2^{16}/f_{IL}$ (3799.18 ms)	1899.59 ms to 2570.04 ms

Caution When bit 0 (WDSTBYON) of the option byte (000C0H) = 0, the window open period is 100% regardless of the values of the WINDOW1 and WINDOW0 bits.

Remark If the overflow time is set to $2^9/f_{IL}$, the window close time and open time are as follows.

	Setting of Window Open Period		
	50%	75%	100%
Window close time	0 to 20.08 ms	0 to 10.04 ms	None
Window open time	20.08 to 29.68 ms	10.04 to 29.68 ms	0 to 29.68 ms

<When window open period is 50%>

- Overflow time:
  $2^9/f_{IL}$ (MAX.) = 2^9/17.25 kHz = 29.68 ms
- Window close time:
  0 to $2^9/f_{IL}$ (MIN.) $\times$ (1 – 0.5) = 0 to 2^9/12.75 kHz $\times$ 0.5 = 0 to 20.08 ms
- Window open time:
  $2^9/f_{IL}$ (MIN.) $\times$ (1 – 0.5) to $2^9/f_{IL}$ (MAX.) = 2^9/12.75 kHz $\times$ 0.5 to 2^9/17.25 kHz = 20.08 to 29.68 ms

11.4.4 Setting watchdog timer interval interrupt

Depending on the setting of bit 7 (WDTINT) of an option byte (000C0H), an interval interrupt (INTWDTI) can be generated when 75% + 1/2f_{IL} of the overflow time is reached.

Table 11-5. Setting of Watchdog Timer Interval Interrupt

WDTINT	Use of Watchdog Timer Interval Interrupt
0	Interval interrupt is used.
1	Interval interrupt is generated when 75% + 1/2f_{IL} of overflow time is reached.

Caution When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed.
Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.
Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.

Remark The watchdog timer continues counting even after INTWDTI is generated (until ACH is written to the watchdog timer enable register (WDTE)). If ACH is not written to the WDTE register before the overflow time, an internal reset signal is generated.

CHAPTER 12 A/D CONVERTER

The number of analog input channels of the A/D converter differs, depending on the product.

	20-pin	30-pin, 38-pin
Analog input channels	6 ch (ANI0 to ANI2, ANI4, ANI5, ANI8)	11 ch (ANI0 to ANI2, ANI4 to ANI7, ANI16 to ANI19)

Caution Most of the following descriptions in this chapter use the 38-pin as an example.

12.1 Function of A/D Converter

The A/D converter is used to convert analog input signals into digital values, and is configured to control a total of twelve channels of analog inputs, including up to eleven channels of A/D converter analog inputs (ANI0 to ANI2, ANI4 to ANI7, and ANI16 to ANI19) and an internal programmable gain amplifier output signal (PGAOUT). 10-bit or 8-bit resolution can be selected by the ADTYP bit of the A/D converter mode register 2 (ADM2).

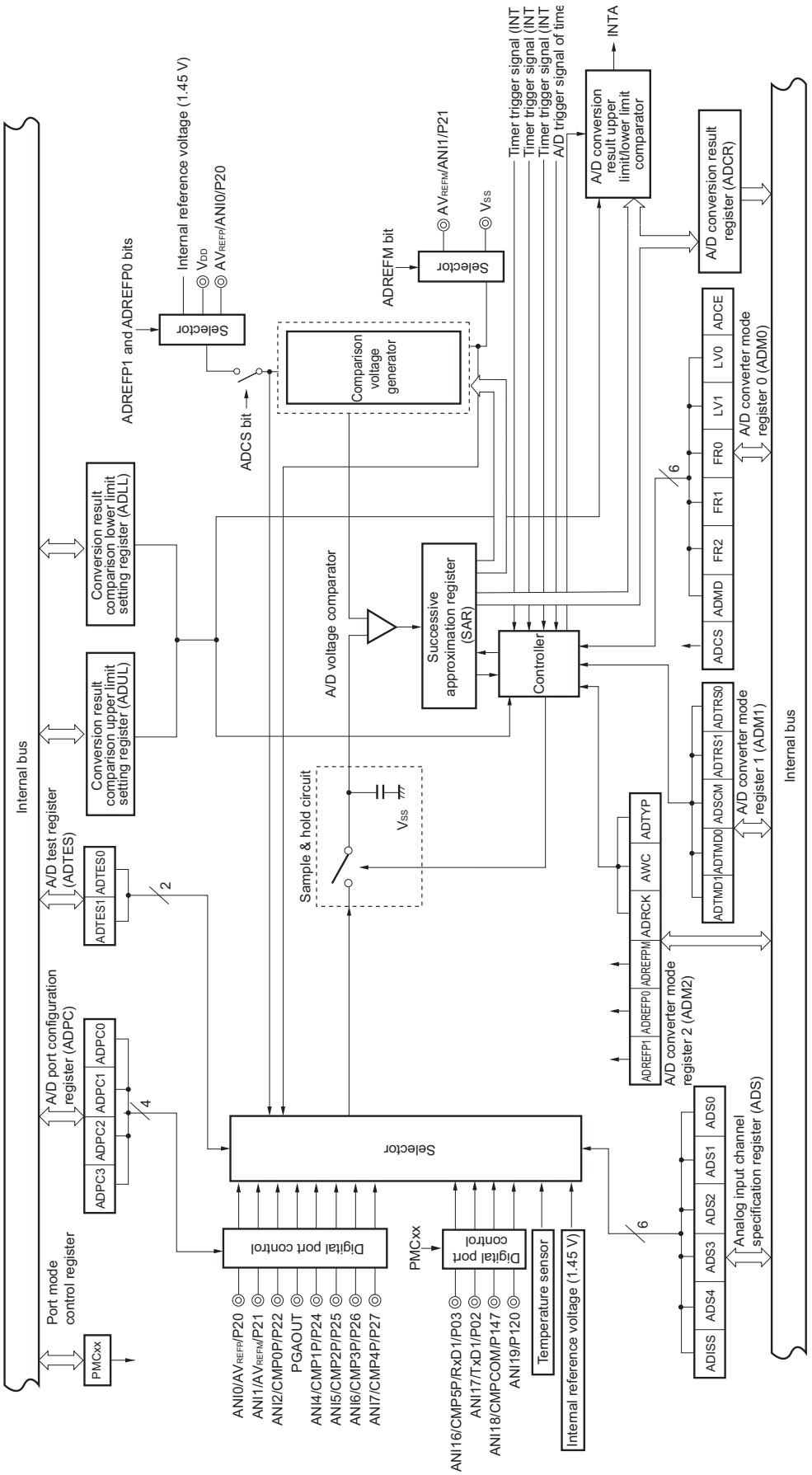
The A/D converter has the following function.

- **10-bit/8-bit resolution A/D conversion**
  10-bit or 8-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from ANI0 to ANI2, ANI4 to ANI7, and ANI16 to ANI19. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated (when in the select mode).

Various A/D conversion modes can be specified by using the mode combinations below.

Trigger mode	Software trigger	Conversion is started by software.
	Hardware trigger no-wait mode	Conversion is started by detecting a hardware trigger.
	Hardware trigger wait mode	The power is turned on by detecting a hardware trigger while the system is off and in the conversion standby state, and conversion is then started automatically after the stabilization wait time passes. When using the SNOOZE mode function, specify the hardware trigger wait mode.
Channel selection mode	Select mode	A/D conversion is performed on the analog input of one selected channel.
	Scan mode	A/D conversion is performed on the analog input of four channels in order.
Conversion operation mode	One-shot conversion mode	A/D conversion is performed on the selected channel once.
	Sequential conversion mode	A/D conversion is sequentially performed on the selected channels until it is stopped by software.
Operation voltage mode	Standard 1 or standard 2 mode	Conversion is done in the operation voltage range of 2.7 V ≤ V_{DD} ≤ 5.5 V.
	Low voltage 1 or low voltage 2 mode	Conversion is done in the operation voltage range of 2.7 V ≤ V_{DD} ≤ 5.5 V. Select this mode for conversion at a low voltage. Because the operation voltage is low, it is internally boosted during conversion.
Sampling time selection	Sampling clock cycles: 7 f_{AD}	The sampling time in standard 1 or low voltage 1 mode is seven cycles of the conversion clock (f_{AD}). Select this mode when the output impedance of the analog input source is high and the sampling time should be long.
	Sampling clock cycles: 5 f_{AD}	The sampling time in standard 2 or low voltage 2 mode is five cycles of the conversion clock (f_{AD}). Select this mode when enough sampling time is ensured (for example, when the output impedance of the analog input source is low).

Figure 12-1. Block Diagram of A/D Converter



Remark Analog input pin for figure 12-1 when a 38-pin product is used.

12.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0 to ANI2, ANI4 to ANI7 and ANI16 to ANI19 pins

These are the analog input pins of the eleven 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) PGAOUT

This is the internal programmable gain amplifier output signal pin. The A/D converter can perform A/D conversion by selecting the output signal of the programmable gain amplifier as the analog input.

(3) Sample & hold circuit

The sample & hold circuit samples each of the analog input voltages sequentially sent from the input circuit, and sends them to the A/D voltage comparator. This circuit also holds the sampled analog input voltage during A/D conversion.

(4) A/D voltage comparator

This A/D voltage comparator compares the voltage generated from the voltage tap of the comparison voltage generator with the analog input voltage. If the analog input voltage is found to be greater than the reference voltage (1/2 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 bit of the SAR is reset.

After that, bit 8 of the SAR register is automatically set, and the next comparison is made. The voltage tap of the comparison voltage generator is selected by the value of bit 9, to which the result has been already set.

Bit 9 = 0: (1/4 AV_{REF})
Bit 9 = 1: (3/4 AV_{REF})

The voltage tap of the comparison voltage generator and the analog input voltage are compared and bit 8 of the SAR register is manipulated according to the result of the comparison.

Analog input voltage ≥ Voltage tap of comparison voltage generator: Bit 8 = 1
Analog input voltage ≤ Voltage tap of comparison voltage generator: Bit 8 = 0

Comparison is continued like this to bit 0 of the SAR register.

When performing A/D conversion at a resolution of 8 bits, the comparison continues until bit 2 of the SAR register.

Remark AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and V_{DD}.

(5) Comparison voltage generator

The comparison voltage generator generates the comparison voltage input from an analog input pin.

(6) Successive approximation register (SAR)

The SAR register is a register that sets voltage tap data whose values from the comparison voltage generator match the voltage values of the analog input pins, 1 bit at a time starting from the most significant bit (MSB).

If data is set in the SAR register all the way to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register (conversion results) are held in the A/D conversion result register (ADCR). When all the specified A/D conversion operations have ended, an A/D conversion end interrupt request signal (INTAD) is generated.

(7) 10-bit A/D conversion result register (ADCR)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCR register holds the A/D conversion result in its higher 10 bits (the lower 6 bits are fixed to 0).

(8) 8-bit A/D conversion result register (ADCRH)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCRH register stores the higher 8 bits of the A/D conversion result.

(9) Controller

This circuit controls the conversion time of an input analog signal that is to be converted into a digital signal, as well as starting and stopping of the conversion operation. When A/D conversion has been completed, this controller generates INTAD through the A/D conversion result upper limit/lower limit comparator.

(10) AV_{REFP} pin

This pin inputs an external reference voltage (AV_{REFP}).

If using AV_{REFP} 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, ANI4 to ANI7, and ANI16 to ANI19 are converted to digital signals based on the voltage applied between AV_{REFP} and the – side reference voltage (AV_{REFM}/V_{SS}).

In addition to AV_{REFP}, it is possible to select V_{DD} or the internal reference voltage (1.45 V) as the + side reference voltage of the A/D converter.

(11) AV_{REFM} pin

This pin inputs an external reference voltage (AV_{REFM}). If using AV_{REFM} as the – side reference voltage of the A/D converter, set the ADREFM bit of the ADM2 register to 1.

In addition to AV_{REFM}, it is possible to select V_{SS} as the – side reference voltage of the A/D converter.

12.3 Registers Controlling A/D Converter

The A/D converter is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- A/D converter mode register 0 (ADM0)
- A/D converter mode register 1 (ADM1)
- A/D converter mode register 2 (ADM2)
- 10-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)
- Analog input channel specification register (ADS)
- Conversion result comparison upper limit setting register (ADUL)
- Conversion result comparison lower limit setting register (ADLL)
- A/D test register (ADTES)
- A/D port configuration register (ADPC)
- Port mode control registers 0, 12, and 14 (PMC0, PMC12, PMC14)
- Port mode registers 0, 2, 12, and 14 (PM0, PM2, PM12, PM14)

12.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the A/D converter is used, be sure to set bit 5 (ADCEN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Figure 12-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	3	<2>	1	<0>
PER0	RTCEN	0	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

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

Cautions 1. When setting the A/D converter, be sure to set the following registers first while the ADCEN bit is set to 1.

If ADCEN = 0, the values of the A/D converter control registers are cleared to their initial values and writing to them is ignored (except for port mode registers 0, 2, 12, and 14 (PM0, PM2, PM12, PM14), port mode control registers 0, 12, and 14 (PMC0, PMC12, PMC14), and A/D port configuration register (ADPC)).

- **A/D converter mode register 0 (ADM0)**
- **A/D converter mode register 1 (ADM1)**
- **A/D converter mode register 2 (ADM2)**
- **10-bit A/D conversion result register (ADCR)**
- **8-bit A/D conversion result register (ADCRH)**
- **Analog input channel specification register (ADS)**
- **Conversion result comparison upper limit setting register (ADUL)**
- **Conversion result comparison lower limit setting register (ADLL)**
- **A/D test register (ADTES)**

2. Be sure to clear the bits 1, 3, and 6 to “0”.

12.3.2 A/D converter mode register 0 (ADM0)

This register sets the conversion time for analog input to be A/D converted, and starts/stops conversion.

The ADM0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 12-3. Format of A/D Converter Mode Register 0 (ADM0)

Address: FFF30H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM0	ADCS	ADMD	FR2 Note 1	FR1 Note 1	FR0 Note 1	LV1 Note 1	LV0 Note 1	ADCE

ADCS	A/D conversion operation control
0	Stops conversion operation [When read] Conversion stopped/standby status
1	Enables conversion operation [When read] While in the software trigger mode: Conversion operation status While in the hardware trigger wait mode: A/D power supply stabilization wait status + conversion operation status

ADMD	Specification of the A/D conversion channel selection mode
0	Select mode
1	Scan mode

ADCE	A/D voltage comparator operation control Note 2
0	Stops A/D voltage comparator operation
1	Enables A/D voltage comparator operation

Notes 1. For details of the FR2 to FR0, LV1, LV0 bits, and A/D conversion, see **Table 12-3 A/D Conversion Time Selection**.

2. While in the software trigger mode or hardware trigger no-wait mode, the operation of the A/D voltage comparator is controlled by the ADCS and ADCE bits, and it takes 1 μs from the start of operation for the operation to stabilize. Therefore, when the ADCS bit is set to 1 after 1 μs or more has elapsed from the time ADCE bit is set to 1, the conversion result at that time has priority over the first conversion result. Otherwise, ignore data of the first conversion.

Cautions 1. Change the ADMD, FR2 to FR0, LV1, and LV0 bits while conversion is stopped (ADCS = 0, ADCE = 0).

2. Do not set the ADCS bit to 1 and the ADCE bit to 0 at the same time.

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 set these bits in the order described in 12.7 A/D Converter Setup Flowchart.

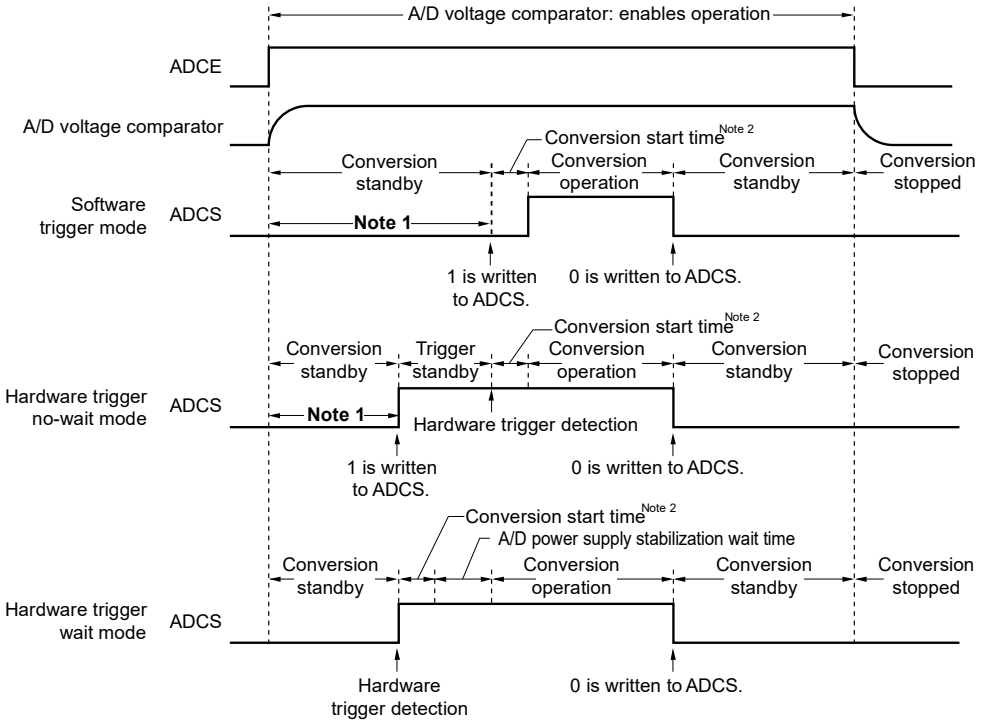
Table 12-1. Settings of ADCS and ADCE Bits

ADCS	ADCE	A/D Conversion Operation
0	0	Conversion stopped state
0	1	Conversion standby state
1	0	Setting prohibited
1	1	Conversion-in-progress state

Table 12-2. Setting and Clearing Conditions for ADCS Bit

A/D Conversion Mode			Set Conditions	Clear Conditions
Software trigger	Select mode	Sequential conversion mode	When 1 is written to ADCS	When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when A/D conversion ends.
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when conversion ends on the specified four channels.
Hardware trigger no-wait mode	Select mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		When 0 is written to ADCS
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		When 0 is written to ADCS
Hardware trigger wait mode	Select mode	Sequential conversion mode	When a hardware trigger is input	When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when A/D conversion ends.
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when conversion ends on the specified four channels.

Figure 12-4. Timing Chart When A/D Voltage Comparator Is Used



Notes 1. While in the software trigger mode or hardware trigger no-wait mode, the time from the rising of the ADCE bit to the falling of the ADCS bit must be 1 μs or longer to stabilize the internal circuit.

2. In starting conversion, the longer will take up to following time

ADM0			Conversion Clock (f_{AD})	Conversion Start Time (Number of f_{CLK} Clock	
FR2	FR1	FR0		Software Trigger Mode/ Hardware Trigger No-wait Mode	Hardware Trigger Wait Mode
0	0	0	f_{CLK}/64	63	1
0	0	1	f_{CLK}/32	31	
0	1	0	f_{CLK}/16	15	
0	1	1	f_{CLK}/8	7	
1	0	0	f_{CLK}/6	5	
1	0	1	f_{CLK}/5	4	
1	1	0	f_{CLK}/4	3	
1	1	1	f_{CLK}/2	1	

However, for the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected.

Cautions 1. If using the hardware trigger wait mode, setting the ADCS bit to 1 is prohibited (but the bit is automatically switched to 1 when the hardware trigger signal is detected). However, it is possible to clear the ADCS bit to 0 to specify the A/D conversion standby status.

2. While in the one-shot conversion mode of the hardware trigger no-wait mode, the ADCS flag is not automatically cleared to 0 when A/D conversion ends. Instead, 1 is retained.

Cautions 3. Only rewrite the value of the ADCE bit when ADCS = 0 (while in the conversion stopped/conversion standby status).

4. To complete A/D conversion, specify at least the following time as the hardware trigger interval:
Hardware trigger no wait mode: 2 f_{CLK} clock + Conversion start time + A/D conversion time
Hardware trigger wait mode: 2 f_{CLK} clock + Conversion start time + A/D power supply stabilization wait time + A/D conversion time

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 12-3. A/D Conversion Time Selection (1/4)

(1) When there is no A/D power supply stabilization wait time

Normal mode 1, 2 (software trigger mode/hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of Conversion Clock Note	Conversion Time	Conversion Time at 10-Bit Resolution				
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	Normal 1	f_{CLK}/64	19 f_{AD} (number of sampling clock: 7 f_{AD})	1216/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	38 μs
0	0	1				f_{CLK}/32		608/f_{CLK}				38 μs	19 μs
0	1	0				f_{CLK}/16		304/f_{CLK}			38 μs	19 μs	9.5 μs
0	1	1				f_{CLK}/8		152/f_{CLK}		38 μs	19 μs	9.5 μs	4.75 μs
1	0	0				f_{CLK}/6		114/f_{CLK}		28.5 μs	14.25 μs	7.125 μs	3.5625 μs
1	0	1				f_{CLK}/5		95/f_{CLK}		23.75 μs	11.875 μs	5.938 μs	2.9688 μs
1	1	0				f_{CLK}/4		76/f_{CLK}		19 μs	9.5 μs	4.75 μs	2.375 μs
1	1	1				f_{CLK}/2		38/f_{CLK}	38 μs	9.5 μs	4.75 μs	2.375 μs	Setting prohibited
0	0	0	0	1	Normal 2	f_{CLK}/64	17 f_{AD} (number of sampling clock: 5 f_{AD})	1088/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	34 μs
0	0	1				f_{CLK}/32		544/f_{CLK}				34 μs	17 μs
0	1	0				f_{CLK}/16		272/f_{CLK}			34 μs	17 μs	8.5 μs
0	1	1				f_{CLK}/8		136/f_{CLK}		34 μs	17 μs	8.5 μs	4.25 μs
1	0	0				f_{CLK}/6		102/f_{CLK}		25.5 μs	12.75 μs	6.375 μs	3.1875 μs
1	0	1				f_{CLK}/5		85/f_{CLK}		21.25 μs	10.625 μs	5.3125 μs	2.6563 μs
1	1	0				f_{CLK}/4		68/f_{CLK}		17 μs	8.5 μs	4.25 μs	2.125 μs
1	1	1				f_{CLK}/2		34/f_{CLK}	34 μs	8.5 μs	4.25 μs	2.125 μs	Setting prohibited

Note These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

Cautions 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 32.6.1 A/D converter characteristics or 33.6.1 A/D converter characteristics.

2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 12-3. A/D Conversion Time Selection (2/4)

(2) When there is no A/D power supply stabilization wait time

Low-voltage mode 1, 2 (software trigger mode/hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of Conversion Clock[Note]	Conversion Time	Conversion Time at 10-Bit Resolution				
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	1	0	Low voltage 1	f_{CLK}/64	19 f_{AD} (number of sampling clock: 7 f_{AD})	1216/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	38 μs
0	0	1				f_{CLK}/32		608/f_{CLK}				38 μs	19 μs
0	1	0				f_{CLK}/16		304/f_{CLK}			38 μs	19 μs	9.5 μs
0	1	1				f_{CLK}/8		152/f_{CLK}		38 μs	19 μs	9.5 μs	4.75 μs
1	0	0				f_{CLK}/6		114/f_{CLK}		28.5 μs	14.25 μs	7.125 μs	3.5625 μs
1	0	1				f_{CLK}/5		95/f_{CLK}		23.75 μs	11.875 μs	5.938 μs	2.9688 μs
1	1	0				f_{CLK}/4		76/f_{CLK}		19 μs	9.5 μs	4.75 μs	2.375 μs
1	1	1				f_{CLK}/2		38/f_{CLK}	38 μs	9.5 μs	4.75 μs	2.375 μs	Setting prohibited
0	0	0	1	1	Low voltage 2	f_{CLK}/64	17 f_{AD} (number of sampling clock: 5 f_{AD})	1088/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	34 μs
0	0	1				f_{CLK}/32		544/f_{CLK}				34 μs	17 μs
0	1	0				f_{CLK}/16		272/f_{CLK}			34 μs	17 μs	8.5 μs
0	1	1				f_{CLK}/8		136/f_{CLK}		34 μs	17 μs	8.5 μs	4.25 μs
1	0	0				f_{CLK}/6		102/f_{CLK}		25.5 μs	12.75 μs	6.375 μs	3.1875 μs
1	0	1				f_{CLK}/5		85/f_{CLK}		21.25 μs	10.625 μs	5.3125 μs	2.6563 μs
1	1	0				f_{CLK}/4		68/f_{CLK}		17 μs	8.5 μs	4.25 μs	2.125 μs
1	1	1				f_{CLK}/2		34/f_{CLK}	34 μs	8.5 μs	4.25 μs	2.125 μs	Setting prohibited

Note These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

Cautions 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 32.6.1 A/D converter characteristics or 33.6.1 A/D converter characteristics.

2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 12-3. A/D Conversion Time Selection (3/4)

(3) When there is A/D power supply stabilization wait time

Normal mode 1, 2 (hardware trigger wait mode[Note 1])

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of A/D Power Supply Stabilization Wait Clock	Number of Conversion Clock[Note 2]	A/D Power Supply Stabilization Wait Time+ Conversion Time	A/D Power Supply Stabilization Wait Time + Conversion Time at 10-Bit Resolution				
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	Normal 1	f_{CLK}/64	8 f_{AD}	19 f_{AD} (number of sampling clock: 7 f_{AD})	1728/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	54 μs
0	0	1				f_{CLK}/32			864/f_{CLK}				54 μs	27 μs
0	1	0				f_{CLK}/16			432/f_{CLK}			54 μs	27 μs	13.5 μs
0	1	1				f_{CLK}/8			216/f_{CLK}		54 μs	27 μs	13.5 μs	6.75 μs
1	0	0				f_{CLK}/6			162/f_{CLK}		40.5 μs	20.25 μs	10.125 μs	5.0625 μs
1	0	1				f_{CLK}/5			135/f_{CLK}		33.75 μs	16.875 μs	8.4375 μs	4.21875 μs
1	1	0				f_{CLK}/4			108/f_{CLK}		27 μs	13.5 μs	6.75 μs	3.375 μs
1	1	1				f_{CLK}/2			54/f_{CLK}	54 μs	13.5 μs	6.75 μs	3.375 μs	Setting prohibited
0	0	0	0	1	Normal 2	f_{CLK}/64	8 f_{AD}	17 f_{AD} (number of sampling clock: 5 f_{AD})	1600/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	50 μs
0	0	1				f_{CLK}/32			800/f_{CLK}				50 μs	25 μs
0	1	0				f_{CLK}/16			400/f_{CLK}			50 μs	25 μs	12.5 μs
0	1	1				f_{CLK}/8			200/f_{CLK}		50 μs	25 μs	12.5 μs	6.25 μs
1	0	0				f_{CLK}/6			150/f_{CLK}		37.5 μs	18.75 μs	9.375 μs	4.6875 μs
1	0	1				f_{CLK}/5			125/f_{CLK}		31.25 μs	15.625 μs	7.8125 μs	3.90625 μs
1	1	0				f_{CLK}/4			100/f_{CLK}		25 μs	12.5 μs	6.25 μs	3.125 μs
1	1	1				f_{CLK}/2			50/f_{CLK}	50 μs	12.5 μs	6.25 μs	3.125 μs	Setting prohibited

Notes 1. For the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected (see **Table 12-3 (1/4)**).

2. These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

Cautions 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 32.6.1 A/D converter characteristics or 33.6.1 A/D converter characteristics. Note that the conversion time (t_{CONV}) does not include the A/D power supply stabilization wait time.

2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

4. When hardware trigger wait mode, specify the conversion time, including the stabilization wait time from the hardware trigger detection.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 12-3. A/D Conversion Time Selection (4/4)

(4) When there is A/D power supply stabilization wait time

Low-voltage mode 1, 2 (hardware trigger wait mode[Note 1])

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of A/D Power Supply Stabilization Wait Clock	Number of Conversion Clock[Note 2]	A/D Power Supply Stabilization Wait Time+ Conversion Time	A/D Power Supply Stabilization Wait Time + Conversion Time at 10-Bit Resolution				
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	1	0	Low voltage 1	f_{CLK}/64	2 f_{AD}	19 f_{AD} (number of sampling clock: 7 f_{AD})	1344/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	42 μs
0	0	1				f_{CLK}/32			672/f_{CLK}				42 μs	21 μs
0	1	0				f_{CLK}/16			336/f_{CLK}			42 μs	21 μs	10.5 μs
0	1	1				f_{CLK}/8			168/f_{CLK}		42 μs	21 μs	10.5 μs	5.25 μs
1	0	0				f_{CLK}/6			126/f_{CLK}		31.5 μs	15.75 μs	7.875 μs	3.9375 μs
1	0	1				f_{CLK}/5			105/f_{CLK}		26.25 μs	13.125 μs	6.5625 μs	3.238125 μs
1	1	0				f_{CLK}/4			84/f_{CLK}		21 μs	10.5 μs	5.25 μs	2.625 μs
1	1	1				f_{CLK}/2			42/f_{CLK}	42 μs	10.5 μs	5.25 μs	2.625 μs	Setting prohibited
0	0	0	1	1	Low voltage 2	f_{CLK}/64	2 f_{AD}	17 f_{AD} (number of sampling clock: 5 f_{AD})	1216/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	38 μs
0	0	1				f_{CLK}/32			608/f_{CLK}				38 μs	19 μs
0	1	0				f_{CLK}/16			304/f_{CLK}			38 μs	19 μs	9.5 μs
0	1	1				f_{CLK}/8			152/f_{CLK}		38 μs	19 μs	9.5 μs	4.75 μs
1	0	0				f_{CLK}/6			114/f_{CLK}		28.5 μs	14.25 μs	7.125 μs	3.5625 μs
1	0	1				f_{CLK}/5			95/f_{CLK}		23.75 μs	11.88 μs	5.938 μs	2.9688 μs
1	1	0				f_{CLK}/4			76/f_{CLK}		19 μs	9.5 μs	4.75 μs	2.375 μs
1	1	1				f_{CLK}/2			38/f_{CLK}	38 μs	9.5 μs	4.75 μs	2.375 μs	Setting prohibited

Notes 1. For the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected (see **Table 12-3 (2/4)**).

2. These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

Cautions 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 32.6.1 A/D converter characteristics or 33.6.1 A/D converter characteristics. Note that the conversion time (t_{CONV}) does not include the A/D power supply stabilization wait time.

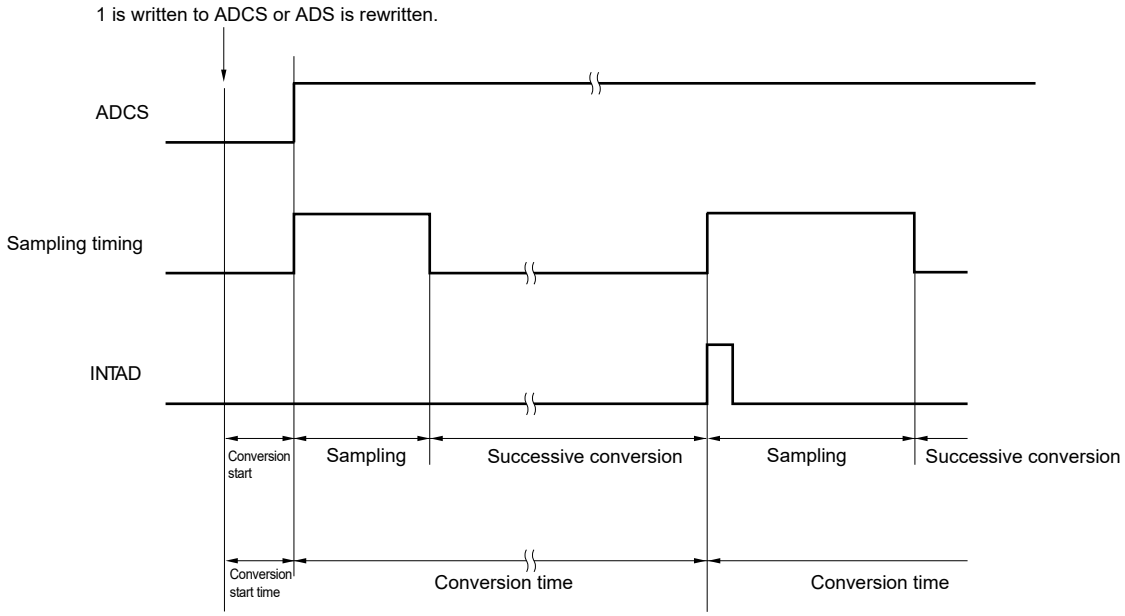
2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

4. When hardware trigger wait mode, specify the conversion time, including the stabilization wait time from the hardware trigger detection.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Figure 12-5. A/D Converter Sampling and A/D Conversion Timing (Example for Software Trigger Mode)



12.3.3 A/D converter mode register 1 (ADM1)

This register is used to specify the A/D conversion trigger, conversion mode, and hardware trigger signal.

The ADM1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 12-6. Format of A/D Converter Mode Register 1 (ADM1)

Address: FFF32H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADM1	ADTMD1	ADTMD0	ADSCM	0	0	0	ADTRS1	ADTRS0

ADTMD1	ADTMD0	Selection of the A/D conversion trigger mode
0	×	Software trigger mode
1	0	Hardware trigger no-wait mode
1	1	Hardware trigger wait mode

ADSCM	Specification of the A/D conversion mode
0	Sequential conversion mode
1	One-shot conversion mode

ADTRS1	ADTRS0	Selection of the hardware trigger signal
0	0	End of timer channel 1 count or capture interrupt signal (INTTM01)
0	1	A/D trigger signal of timer KB0 to KB2[Note]
1	0	Real-time clock interrupt signal (INTRTC)
1	1	12-bit interval timer interrupt signal (INTIT)

Note Use bits 2 and 3 (ADTRG10 and ADTRG11) of peripheral function switch register 0 (PFSEL0) to select one signal from among the A/D trigger signals of timers KB0 to KB2. For details about the PFSEL0 register, see **7.3.16 Peripheral function switch register 0 (PFSEL0)**.

Cautions 1. Rewrite the value of the ADM1 register while conversion is stopped (ADCS = 0, ADCE = 0).

2. To complete A/D conversion, specify at least the following time as the hardware trigger interval:
Hardware trigger no wait mode: 2 f_{CLK} clock + Conversion start time + A/D conversion time
Hardware trigger wait mode: 2 f_{CLK} clock + Conversion start time + A/D power supply stabilization wait time + A/D conversion time

3. In modes other than SNOOZE mode, input of the next INTRTC or INTIT will not be recognized as a valid hardware trigger for up to four f_{CLK} cycles after the first INTRTC or INTIT is input.

Remarks 1. ×: don't care

2. f_{CLK}: CPU/peripheral hardware clock frequency

12.3.4 A/D converter mode register 2 (ADM2)

This register is used to select the + side or – side reference voltage of the A/D converter, check the upper limit and lower limit A/D conversion result values, select the resolution, and specify whether to use the SNOOZE mode.

The ADM2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 12-7. Format of A/D Converter Mode Register 2 (ADM2) (1/2)

Address: F0010H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

ADREFP1	ADREFP0	Selection of the + side reference voltage of the A/D converter
0	0	Supplied from V_{DD}
0	1	Supplied from P20/AV_{REFP}/ANI0
1	0	Supplied from the internal reference voltage (1.45 V)Note
1	1	Setting prohibited

- When ADREFP1 or ADREFP0 bit is rewritten, this must be configured in accordance with the following procedures.
  (1) Set ADCE = 0
  (2) Change the values of ADREFP1 and ADREFP0
  (3) Reference voltage stabilization wait time (A)
  (4) Set ADCE = 1
  (5) Reference voltage stabilization wait time (B)
  When ADREFP1 and ADREFP0 are set to 1 and 0, the setting is changed to A = 5 μs, B = 1 μs.
  When ADREFP1 and ADREFP0 are set to 0 and 0 or 0 and 1, A needs no wait and B = 1 μs.
- When ADREFP1 and ADREFP0 are set to 1 and 0, respectively, A/D conversion cannot be performed on the temperature sensor output voltage and internal reference voltage (1.45 V).
  Be sure to perform A/D conversion while ADISS = 0.

ADREFM	Selection of the – side reference voltage of the A/D converter
0	Supplied from V_{SS}
1	Supplied from P21/AV_{REFM}/ANI1

Note This setting can be used only in HS (high-speed main) mode.

Cautions 1. Rewrite the value of the ADM2 register while conversion is stopped (ADCS = 0, ADCE = 0).

2. Do not set the ADREFP1 bit to 1 when shifting to STOP mode, or to HALT mode while the CPU is operating on the subsystem clock. Also, if the internal reference voltage (ADREFP1 =1, ADREFP0 = 0) is selected, the A/D converter reference voltage (I_{ADREF}) indicated in 32.3.2 or 33.3.2 Supply current characteristics will be added to the current consumption.

3. When using AV_{REFP} and AV_{REFM}, specify ANI0 and ANI1 as the analog input channels and specify input mode by using the port mode register.

Figure 12-7. Format of A/D Converter Mode Register 2 (ADM2) (2/2)

Address: F0010H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

ADRCK	Checking the upper limit and lower limit conversion result values
0	The interrupt signal (INTAD) is output when the ADLL register ≤ the ADCR register ≤ the ADUL register (AREA 1).
1	The interrupt signal (INTAD) is output when the ADCR register < the ADLL register (AREA 2) or the ADUL register < the ADCR register (AREA 3).
Figure 12-8 shows the generation range of the interrupt signal (INTAD) for AREA 1 to AREA 3.	

AWC	Specification of the wakeup function (SNOOZE mode)
0	Do not use the SNOOZE mode function.
1	Use the SNOOZE mode function.

When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and A/D conversion is performed without operating the CPU (the SNOOZE mode).

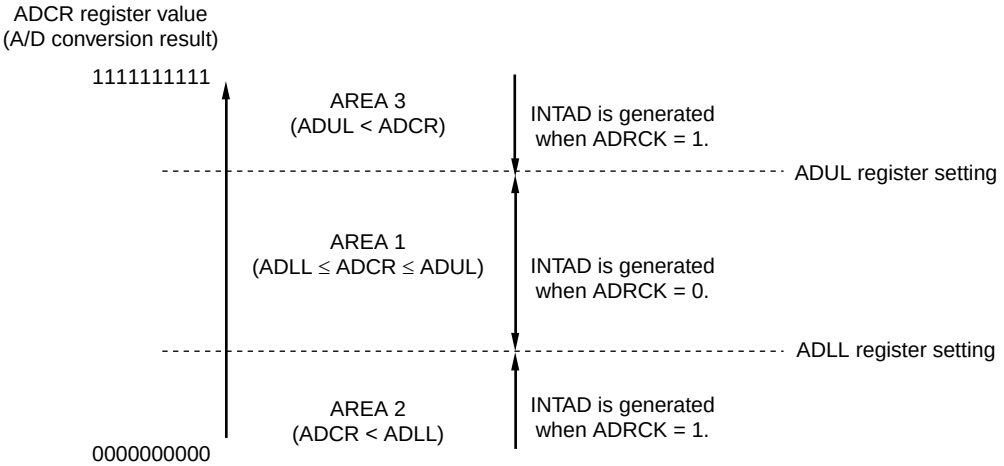
- The SNOOZE mode function can only be specified when the high-speed on-chip oscillator oscillation clock is selected for the CPU/peripheral hardware clock (f_{CLK}). If any other clock is selected, specifying this mode is prohibited.
- Using the SNOOZE mode function in the software trigger mode or hardware trigger no-wait mode is prohibited.
- When using the SNOOZE mode function, specify a hardware trigger interval of at least “shift time to SNOOZE mode[Note] + conversion start time + A/D power supply stabilization wait time + A/D conversion time +2 f_{CLK} clock”
- Even when using 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.

ADTYP	Selection of the A/D conversion resolution
0	10-bit resolution
1	8-bit resolution

Note See “From STOP to SNOOZE” in **21.3.3 SNOOZE mode**.

Caution Rewrite the value of the ADM2 register while conversion is stopped (ADCS = 0, ADCE = 0).

Figure 12-8. ADRCK Bit Interrupt Signal Generation Range



Remark If INTAD does not occur, the A/D conversion result is not stored in the ADCR or ADCRH register.

12.3.5 10-bit A/D conversion result register (ADCR)

This register is a 16-bit register that stores the A/D conversion result. The lower 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR). The higher 8 bits of the conversion result are stored in FFF1FH and the lower 2 bits are stored in the higher 2 bits of FFF1EH[Note].

The ADCR register can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (the value specified by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see **Figure 12-8**), the result is not stored.

Figure 12-9. Format of 10-bit A/D Conversion Result Register (ADCR)

Address: FFF1FH, FFF1EH After reset: 0000H R

Symbol	FFF1FH								FFF1EH							
ADCR											0	0	0	0	0	0

Cautions 1. When 8-bit resolution A/D conversion is selected (when the ADTYP bit of A/D converter mode register 2 (ADM2) is 1) and the ADCR register is read, 0 is read from the lower two bits (bits 7 and 6 of the ADCR register).

2. When the ADCR register is accessed in 16-bit units, the higher 10 bits of the conversion result are read in order starting at bit 15 of the ADCR register..

12.3.6 8-bit A/D conversion result register (ADCRH)

This register is an 8-bit register that stores the A/D conversion result.

The higher 8 bits of 10-bit resolution are stored. [Note]

The ADCRH register can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (the value specified by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see **Figure 12-8**), the result is not stored.

Figure 12-10. Format of 8-bit A/D Conversion Result Register (ADCRH)

Address: FFF1FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADCRH								

Caution When writing to the A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of the ADCRH register may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, and ADPC registers. Using timing other than the above may cause an incorrect conversion result to be read.

12.3.7 Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.

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

Reset signal generation clears this register to 00H.

Figure 12-11. Format of Analog Input Channel Specification Register (ADS)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

O Select mode (ADMD = 0)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P20/ANI0/AV_{REFP} pin
0	0	0	0	0	1	ANI1	P21/ANI1/AV_{REFM} pin
0	0	0	0	1	0	ANI2	P22/ANI2 pin
0	0	0	0	1	1	PGAOUT	PGA output signal 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
1	0	0	0	0	0	–	Temperature sensor output voltage[Note]
1	0	0	0	0	1	–	Internal reference voltage (1.45 V)[Note]
Other than above						Setting prohibited	

O Scan mode (ADMD = 1)

ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel			
					Scan 0	Scan 1	Scan 2	Scan 3
0	0	0	0	0	ANI0	ANI1	ANI2	PGAOUT
0	0	0	0	1	ANI1	ANI2	PGAOUT	ANI4
0	0	0	1	0	ANI2	PGAOUT	ANI4	ANI5
0	0	0	1	1	PGAOUT	ANI4	ANI5	ANI6
0	0	1	0	0	ANI4	ANI5	ANI6	ANI7
Other than above					Setting prohibited			

Note This setting can be used only in HS (high-speed main) mode.

(**Cautions** and **Remark** are listed on the next page.)

Cautions 1. Be sure to clear bits 5 and 6 to 0.

2. Set a channel to be set the analog input by ADPC and PMCx registers in the input mode by using port mode registers 0, 2, 12, and 14 (PM0, PM2, PM12, PM14).

3. Do not set the pin that is set by the A/D port configuration register (ADPC) as digital I/O by the ADS register.

4. Do not set the pin that is set by port mode control register 0, 12, or 14 (PMC0, PMC12, PMC14) as digital I/O by the ADS register.

5. Rewrite the value of the ADISS bit while conversion is stopped (ADCS = 0, ADCE = 0).

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.

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.

8. If ADISS is set to 1, the internal reference voltage (1.45 V) cannot be used for the + side reference voltage.

After the ADISS bit is set to 1, the initial conversion result cannot be used. For the setting flow, see 12.7.4 Setup when temperature sensor output voltage/internal reference voltage is selected (example for software trigger mode and one-shot conversion mode).

9. When PGAOUT is selected as analog input, set the ADS register after setting up PGA operations (see CHAPTER 13 PROGRAMMABLE GAIN AMPLIFIER).

10. Do not set the ADISS bit to 1 when shifting to STOP mode, or to HALT mode while the CPU is operating on the subsystem clock. Also, if the ADREFP1 bit is set to 1, the A/D converter reference voltage current (I_{ADREF}) indicated in 32.3.2 or 33.3.2 Supply current characteristics will be added to the current consumption.

Remark ×: don't care

12.3.8 Conversion result comparison upper limit setting register (ADUL)

This register is used to specify the setting for checking the upper limit of the A/D conversion results.

The A/D conversion results and ADUL register value are compared, and interrupt signal (INTAD) generation is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in **Figure 12-8**).

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

Reset signal generation sets this register to FFH.

Figure 12-12. Format of Conversion Result Comparison Upper Limit Setting Register (ADUL)

Address: F0011H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
ADUL	ADUL7	ADUL6	ADUL5	ADUL4	ADUL3	ADUL2	ADUL1	ADUL0

12.3.9 Conversion result comparison lower limit setting register (ADLL)

This register is used to specify the setting for checking the lower limit of the A/D conversion results.

The A/D conversion results and ADLL register value are compared, and interrupt signal (INTAD) generation is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in **Figure 12-8**).

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

Reset signal generation clears this register to 00H.

Figure 12-13. Format of Conversion Result Comparison Lower Limit Setting Register (ADLL)

Address: F0012H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADLL	ADLL7	ADLL6	ADLL5	ADLL4	ADLL3	ADLL2	ADLL1	ADLL0

Cautions 1. When A/D conversion with 10-bit resolution is selected, the eight higher-order bits of the 10-bit A/D conversion result register (ADCR) are compared with the values in the ADUL and ADLL registers.

2. Only write new values to the ADUL and ADLL registers while conversion is stopped (ADCS = 0, ADCE = 0).

3. The setting of the ADUL registers must be greater than that of the ADLL register.

12.3.10 A/D test register (ADTES)

This register is used to select the + side reference voltage or – side reference voltage for the converter, an analog input channel (ANIxx), the temperature sensor output voltage, the internal reference voltage (1.45 V), or PGAOUT as the target for A/D conversion.

When using this register to test the converter, set as follows.

- For zero-scale measurement, select the – side reference voltage as the target for conversion.
- For full-scale measurement, select the + side reference voltage as the target for conversion.

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

Reset signal generation clears this register to 00H.

Figure 12-14. Format of A/D Test Register (ADTES)

Address: F0013H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	0	ADTES1	ADTES0

ADTES1	ADTES0	A/D conversion target
0	0	ANIxx/temperature sensor output voltage[Note] /internal reference voltage (1.45 V)[Note]/ PGAOUT (This is specified using the analog input channel specification register (ADS).)
1	0	The – side reference voltage (selected by the ADREFM bit of the ADM2 register)
1	1	The + side reference voltage (selected by the ADREFP1 or ADREFP0 bit of the ADM2 register)
Other than above		Setting prohibited

Note The temperature sensor output voltage and internal reference voltage (1.45 V) can be selected only in the HS (high-speed main) mode.

12.3.11 Registers controlling port function of analog input pins

Set up the registers for controlling the functions of the ports shared with the analog input pins of the A/D converter (port mode registers (PMxx), port mode control registers (PMCxx), and A/D port configuration register (ADPC)). For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.6 Port mode control registers (PMCxx)**, and **4.3.7 A/D port configuration register (ADPC)**.

When using the ANI0 to ANI2, ANI4 to ANI17, and PGAOUT pins (internal pins) for analog input of the A/D converter, set the port mode register (PMxx) bit corresponding to each port to 1 and select analog input through the A/D port configuration register (ADPC).

When using the ANI16 to ANI19 pins for analog input of the A/D converter, set the port mode register (PMxx) bit and port mode control register (PMCxx) bit corresponding to each port to 1.

12.4 A/D Converter Conversion Operations

The A/D converter conversion operations are described below.

<1> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<2> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the sampled voltage is held until the A/D conversion operation has ended.

<3> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) AV_{REF} by the tap selector.

<4> The voltage difference between the series resistor string voltage tap and sampled voltage is compared by the voltage comparator. If the analog input is greater than (1/2) AV_{REF}, the MSB bit of the SAR register remains set to 1. If the analog input is smaller than (1/2) AV_{REF}, the MSB bit is reset to 0.

<5> Next, bit 8 of the SAR register is automatically set to 1, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 9, as described below.

- Bit 9 = 1: (3/4) AV_{REF}
- Bit 9 = 0: (1/4) AV_{REF}

The voltage tap and sampled voltage are compared and bit 8 of the SAR register is manipulated as follows.

- Sampled voltage ≥ Voltage tap: Bit 8 = 1
- Sampled voltage < Voltage tap: Bit 8 = 0

<6> Comparison is continued in this way up to bit 0 of the SAR register.

<7> Upon completion of the comparison of 10 bits, an effective digital result value remains in the SAR register, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH) and then latched[Note 1].
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated[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.

Notes 1. If the A/D conversion result is outside the A/D conversion result range specified by the ADRCK bit and the ADUL and ADLL registers (see Figure 12-8), the A/D conversion result interrupt request signal is not generated and no A/D conversion results are stored in the ADCR and ADCRH registers.

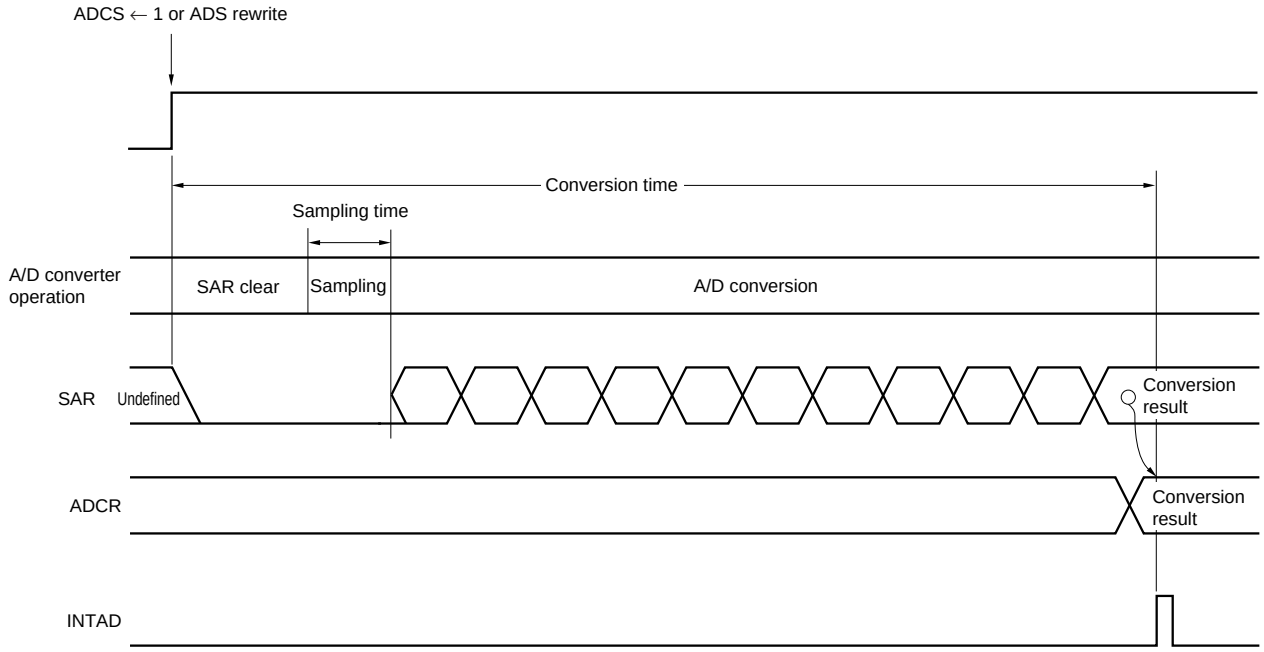
2. While in the sequential conversion mode, the ADCS flag is not automatically cleared to 0. This flag is not automatically cleared to 0 while in the one-shot conversion mode of the hardware trigger no-wait mode, either. Instead, 1 is retained.

Remarks 1. Two types of the A/D conversion result registers are available.

- ADCR register (16 bits): Store 10-bit A/D conversion value
- ADCRH register (8 bits): Store 8-bit A/D conversion value

2. AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and V_{DD}.

Figure 12-15. Conversion Operation of A/D Converter (Software Trigger Mode)



In one-shot conversion mode, the ADCS bit is automatically cleared to 0 after completion of A/D conversion.

In sequential conversion mode, A/D conversion operations proceed continuously until the software clears bit 7 (ADCS) of the A/D converter mode register 0 (ADM0) to 0.

Writing to the analog input channel specification register (ADS) during A/D conversion interrupts the current conversion after which A/D conversion of the analog input specified by the ADS register proceeds. Data from the A/D conversion that was in progress are discarded.

Reset signal generation clears the A/D conversion result register (ADCR, ADCRH) to 0000H or 00H.

12.5 Input Voltage and Conversion Results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI2, ANI4 to ANI7, ANI16 to ANI19, PGAOUT) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$SAR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5\right)$$

$$ADCR = SAR \times 64$$

or

$$\left(\frac{ADCR}{64} - 0.5\right) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < \left(\frac{ADCR}{64} + 0.5\right) \times \frac{AV_{REF}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

V_{AIN}: Analog input voltage

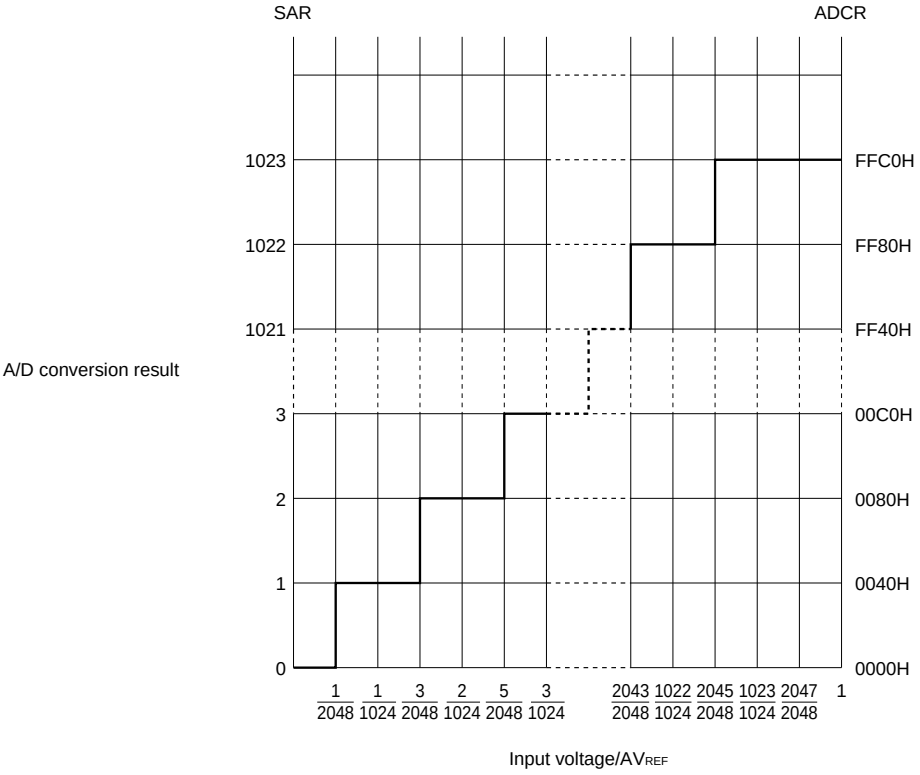
AV_{REF}: AV_{REF} pin voltage

ADCR: A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 12-16 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 12-16. Relationship Between Analog Input Voltage and A/D Conversion Result



Remark AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and V_{DD}.

12.6 A/D Converter Operation Modes

The operation of each A/D converter mode is described below. In addition, the procedure for specifying each mode is described in **12.7 A/D Converter Setup Flowchart**.

12.6.1 Software trigger mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

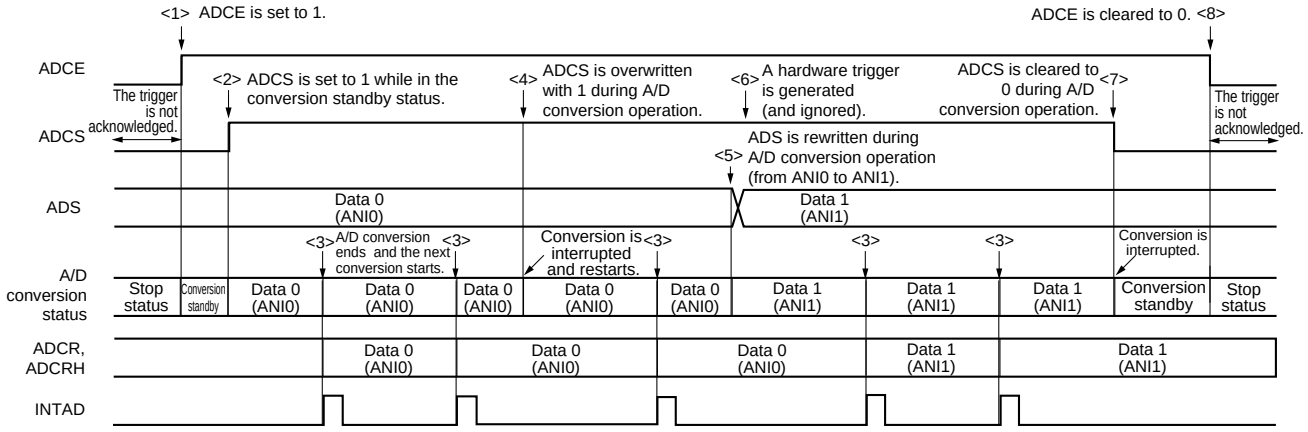
<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 12-17. Example of Software Trigger Mode (Select Mode, Sequential Conversion Mode) Operation Timing



12.6.2 Software trigger mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.

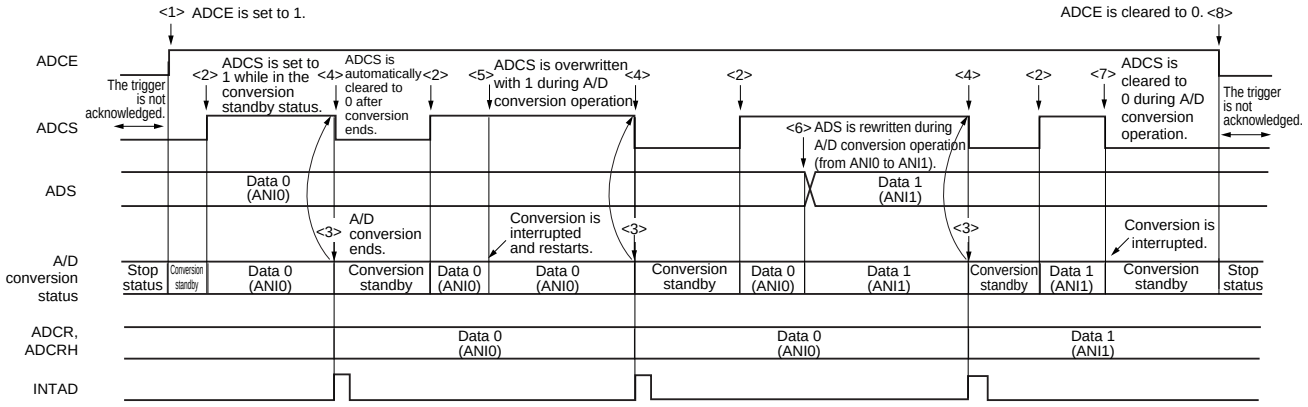
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby status.

Figure 12-18. Example of Software Trigger Mode (Select Mode, One-shot Conversion Mode) Operation Timing



12.6.3 Software trigger mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts (until all four channels are finished).

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

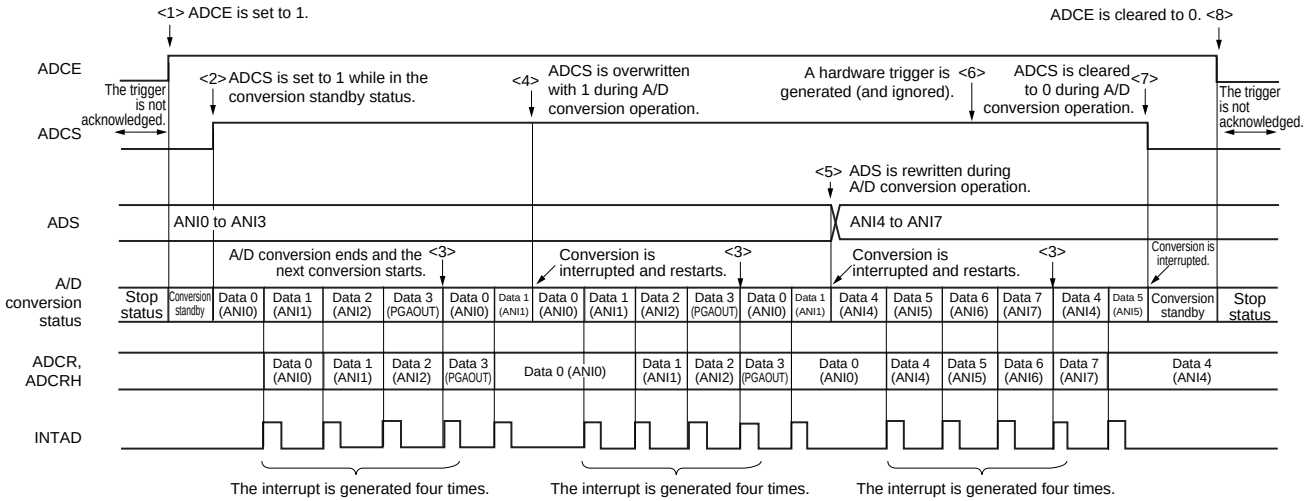
<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<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 system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 12-19. Example of Software Trigger Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



12.6.4 Software trigger mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion of the four channels ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.

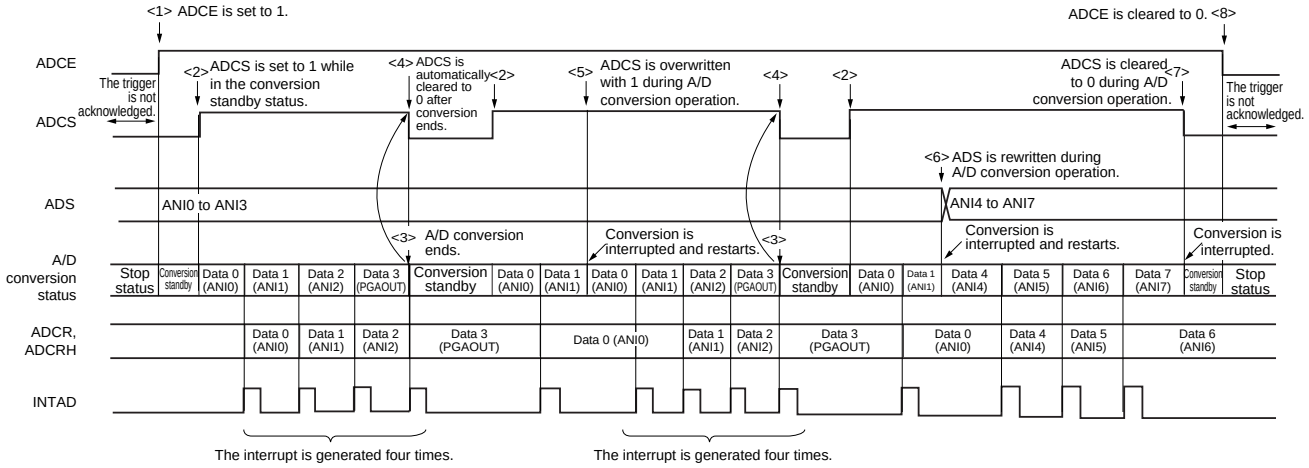
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is 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 system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby status.

Figure 12-20. Example of Software Trigger Mode (Scan Mode, One-shot Conversion Mode) Operation Timing



12.6.5 Hardware trigger no-wait mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).

<4> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

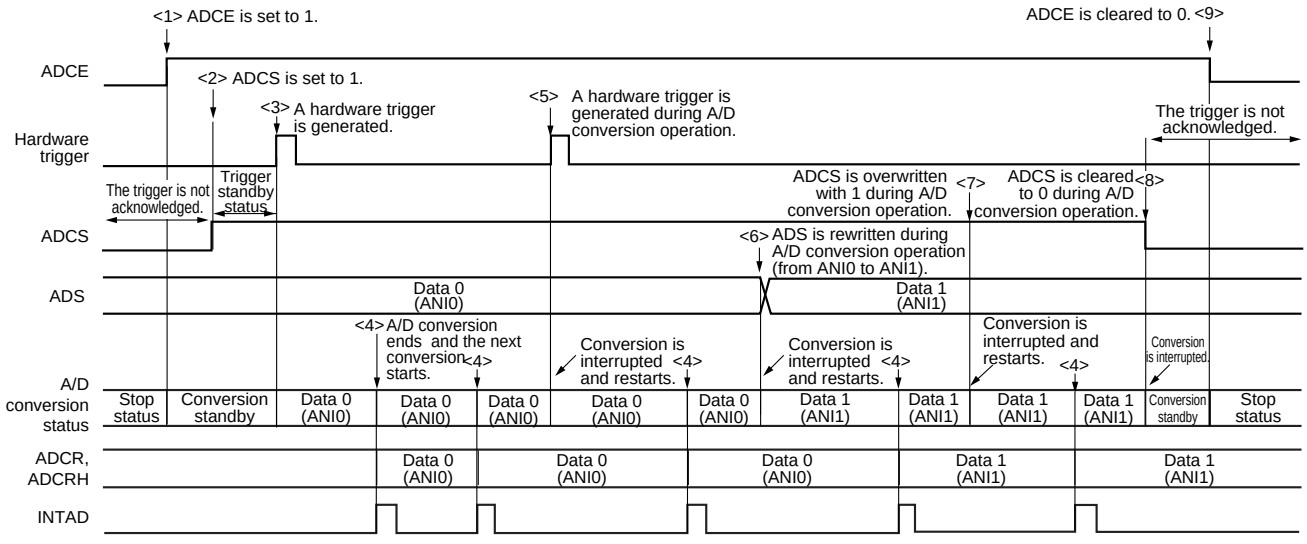
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<9> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 12-21. Example of Hardware Trigger No-wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



12.6.6 Hardware trigger no-wait mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).

<4> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<5> After A/D conversion ends, the ADCS bit remains set to 1, and the system enters the A/D conversion standby status.

<6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

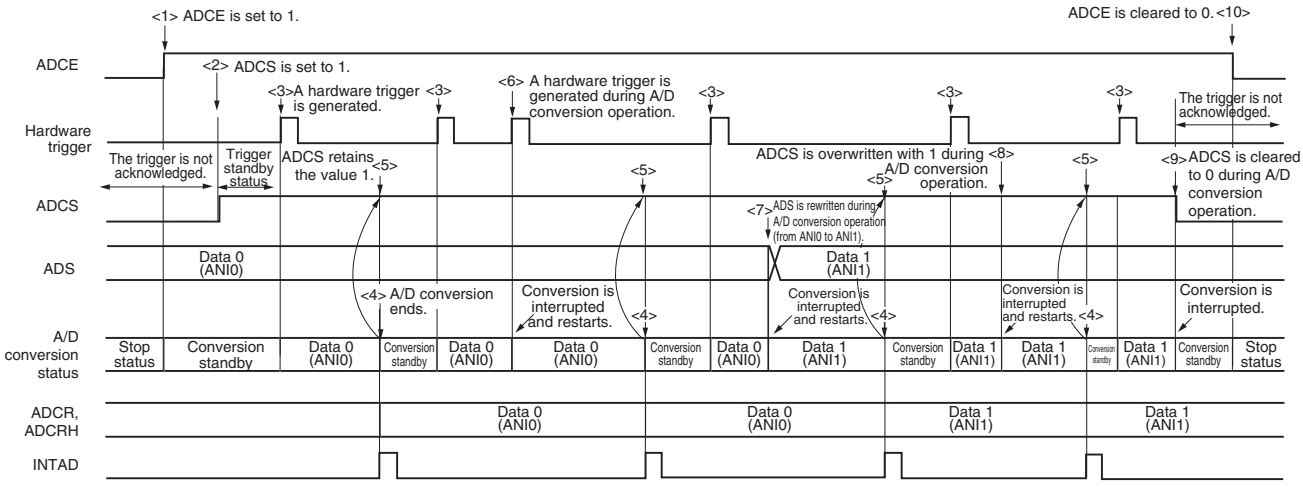
<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<10> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 12-22. Example of Hardware Trigger No-wait Mode (Select Mode, One-shot Conversion Mode) Operation Timing



12.6.7 Hardware trigger no-wait mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<4> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

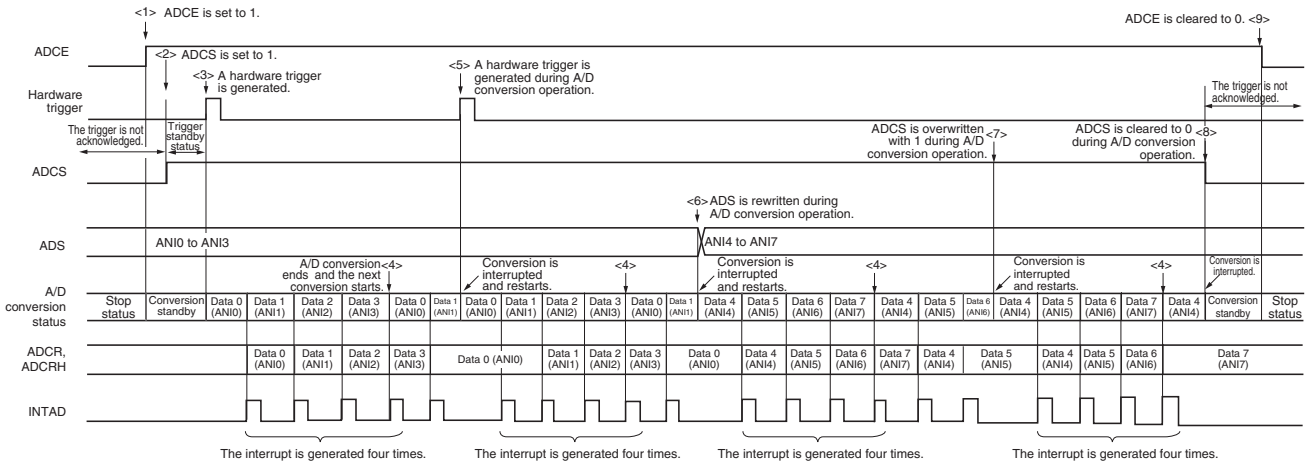
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<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 system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<9> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 12-23. Example of Hardware Trigger No-wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



12.6.8 Hardware trigger no-wait mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<4> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<5> After A/D conversion of the four channels ends, the ADCS bit remains set to 1, and the system enters the A/D conversion standby status.

<6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

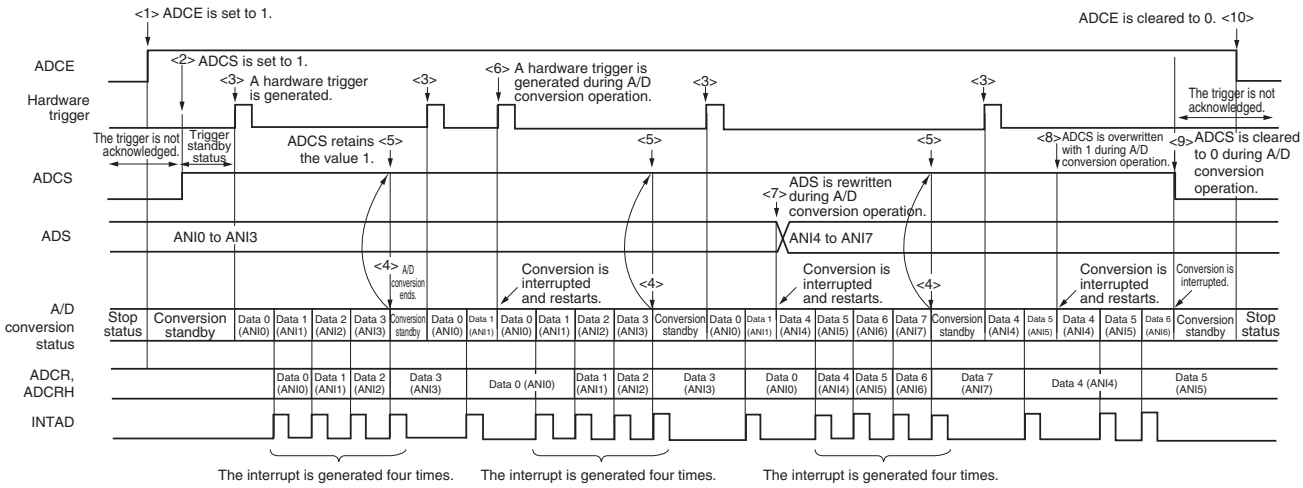
<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel. The partially converted data is discarded.

<9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<10> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 12-24. Example of Hardware Trigger No-wait Mode (Scan Mode, One-shot Conversion Mode) Operation Timing



12.6.9 Hardware trigger wait mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the hardware trigger standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input.

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts. (At this time, no hardware trigger is necessary.)

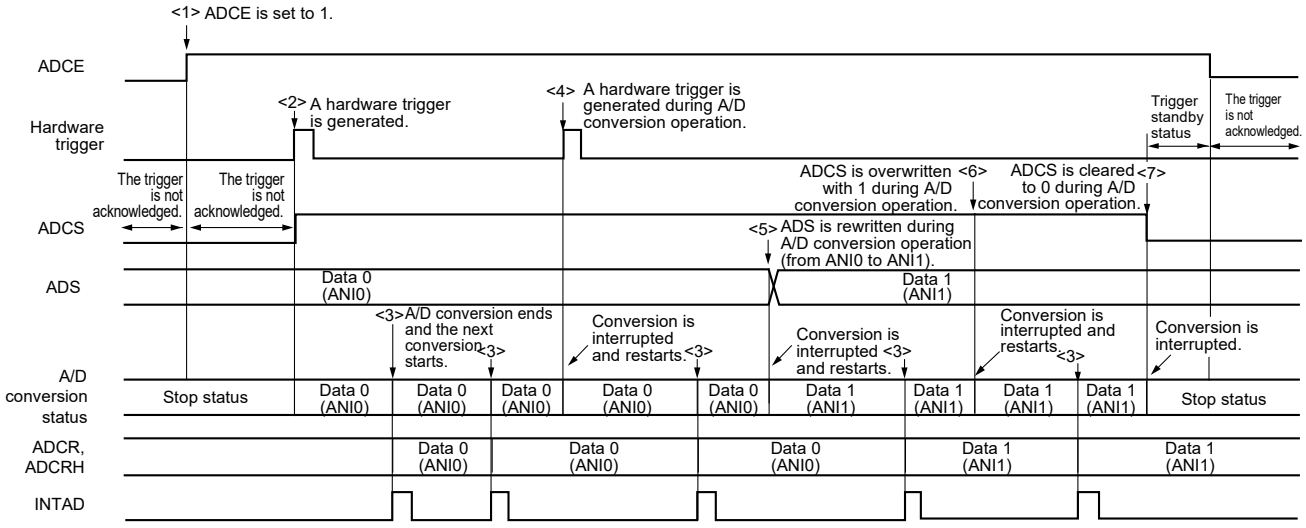
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<6> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 12-25. Example of Hardware Trigger Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



12.6.10 Hardware trigger wait mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the hardware trigger standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input.

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the stop status.

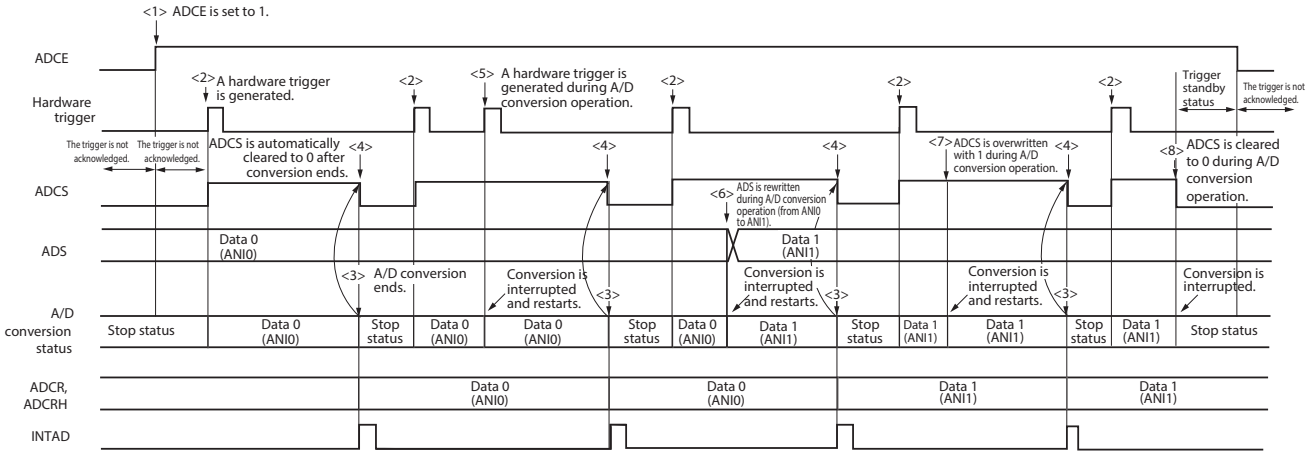
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is initialized.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 12-26. Example of Hardware Trigger Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



12.6.11 Hardware trigger wait mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts.

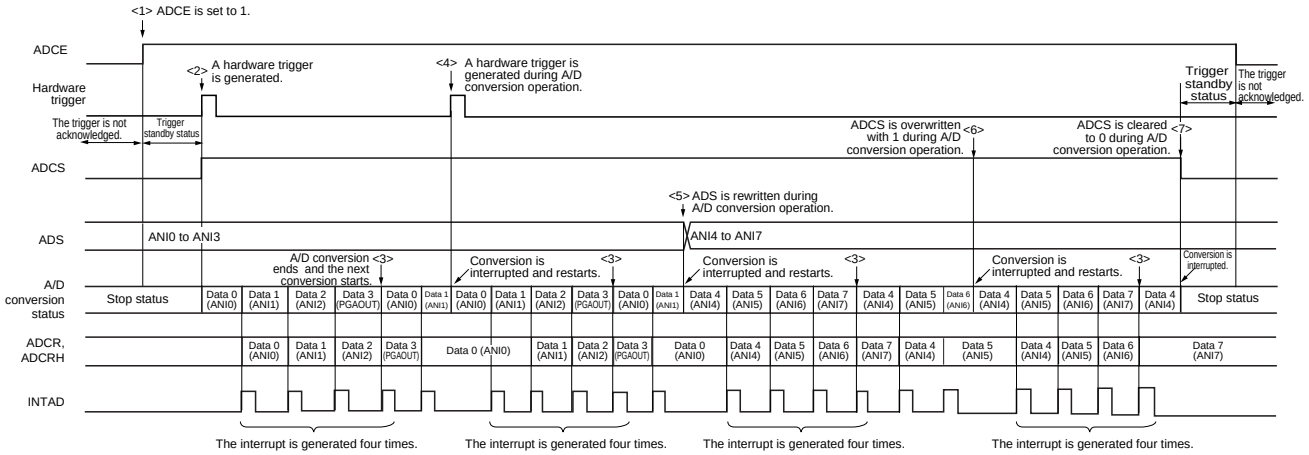
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<5> When the value of the ADS register is 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 system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 12-27. Example of Hardware Trigger Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



12.6.12 Hardware trigger wait mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the stop status.

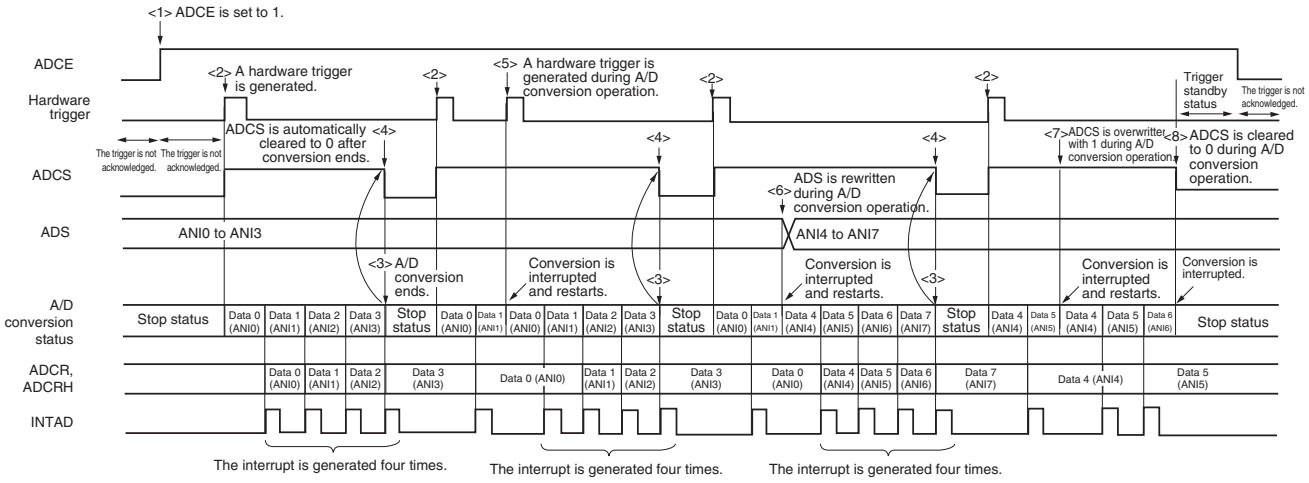
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is 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 system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 12-28. Example of Hardware Trigger Wait Mode (Scan Mode, One-shot Conversion Mode) Operation Timing

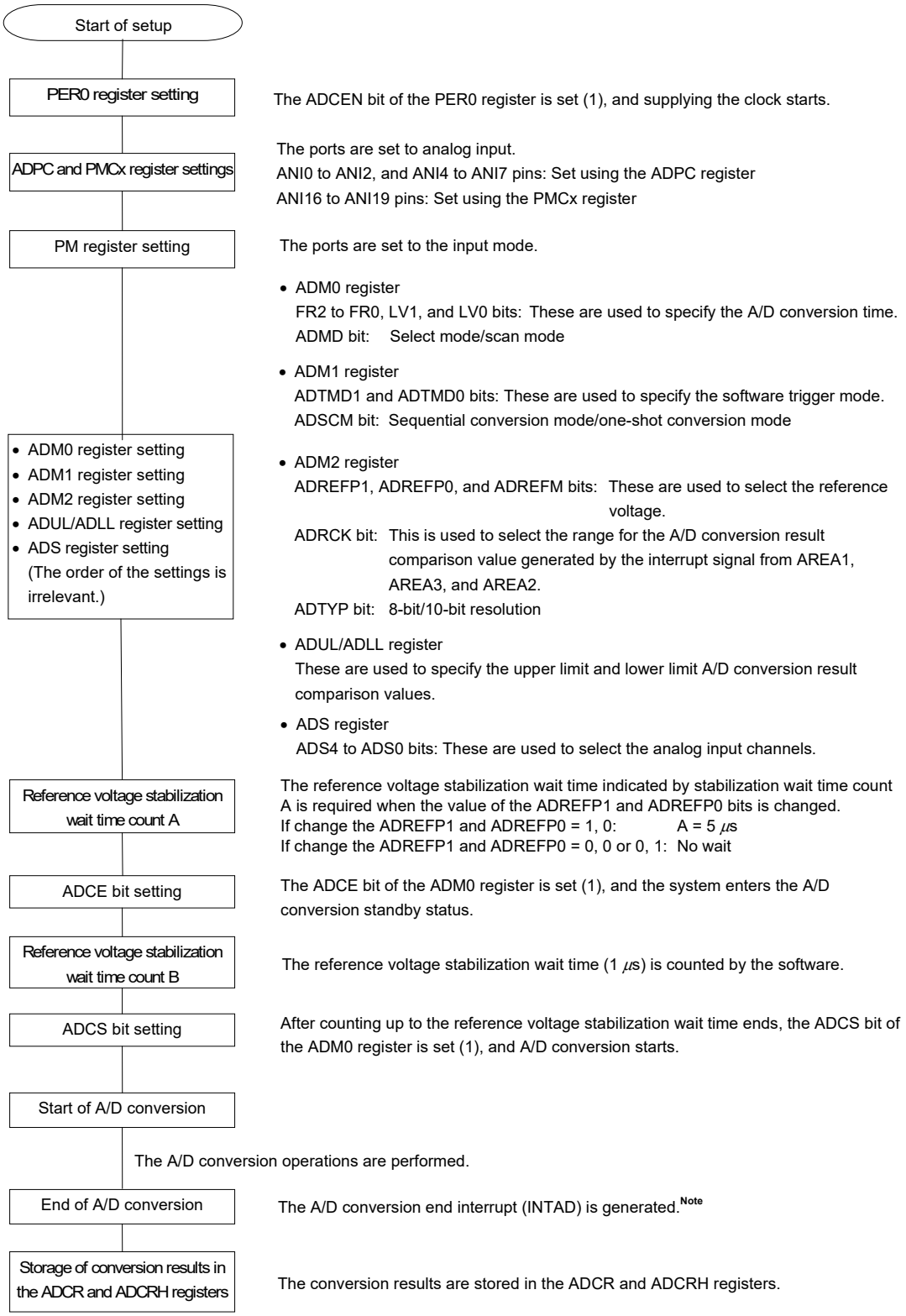


12.7 A/D Converter Setup Flowchart

The A/D converter setup flowchart in each operation mode is described below.

12.7.1 Setting up software trigger mode

Figure 12-29. Setting up Software Trigger Mode

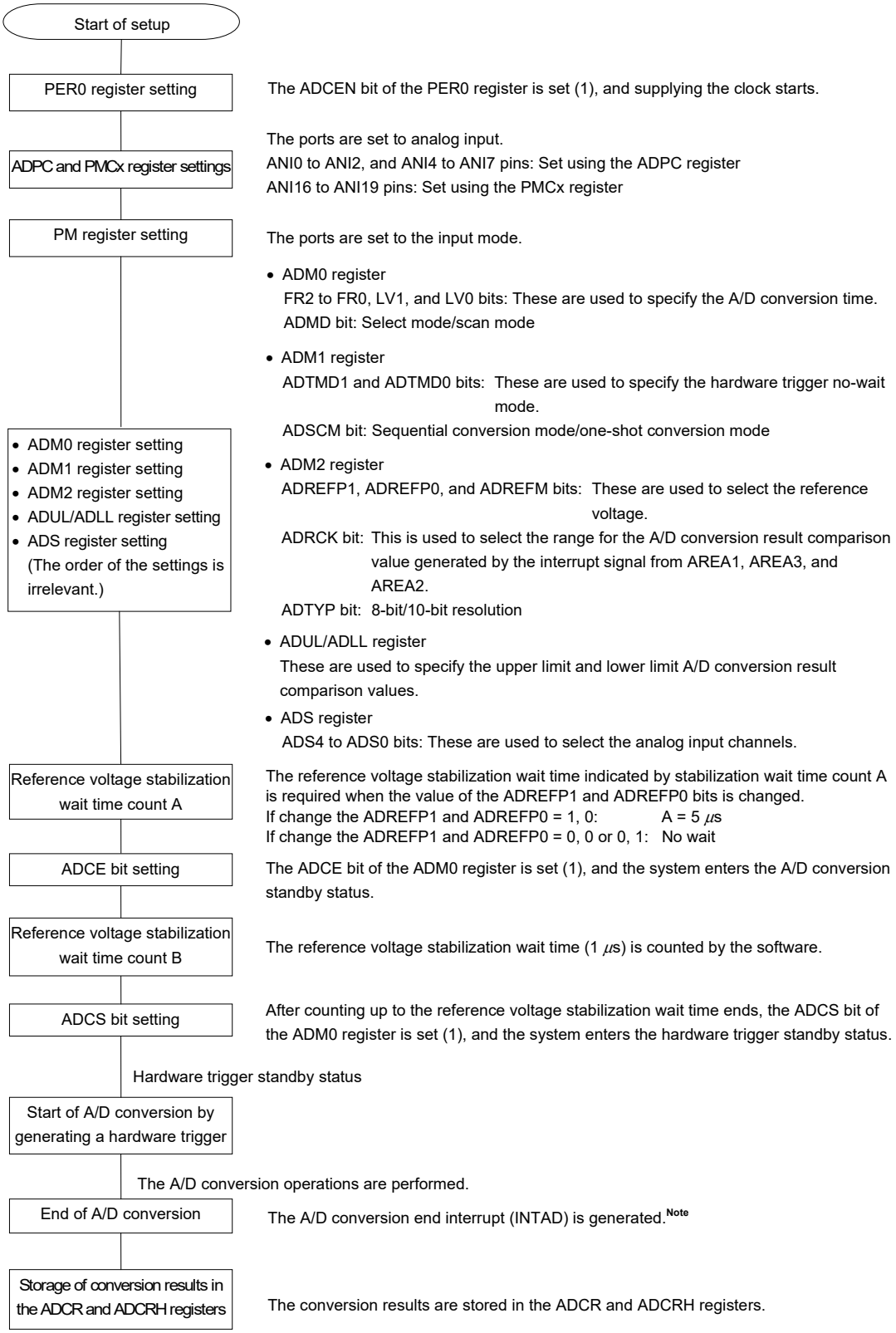


Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR register.

Caution When PGAOUT is selected as analog input, set the ADS register after setting up PGA operations (see CHAPTER 13 PROGRAMMABLE GAIN AMPLIFIER).

12.7.2 Setting up hardware trigger no-wait mode

Figure 12-30. Setting up Hardware Trigger No-wait Mode

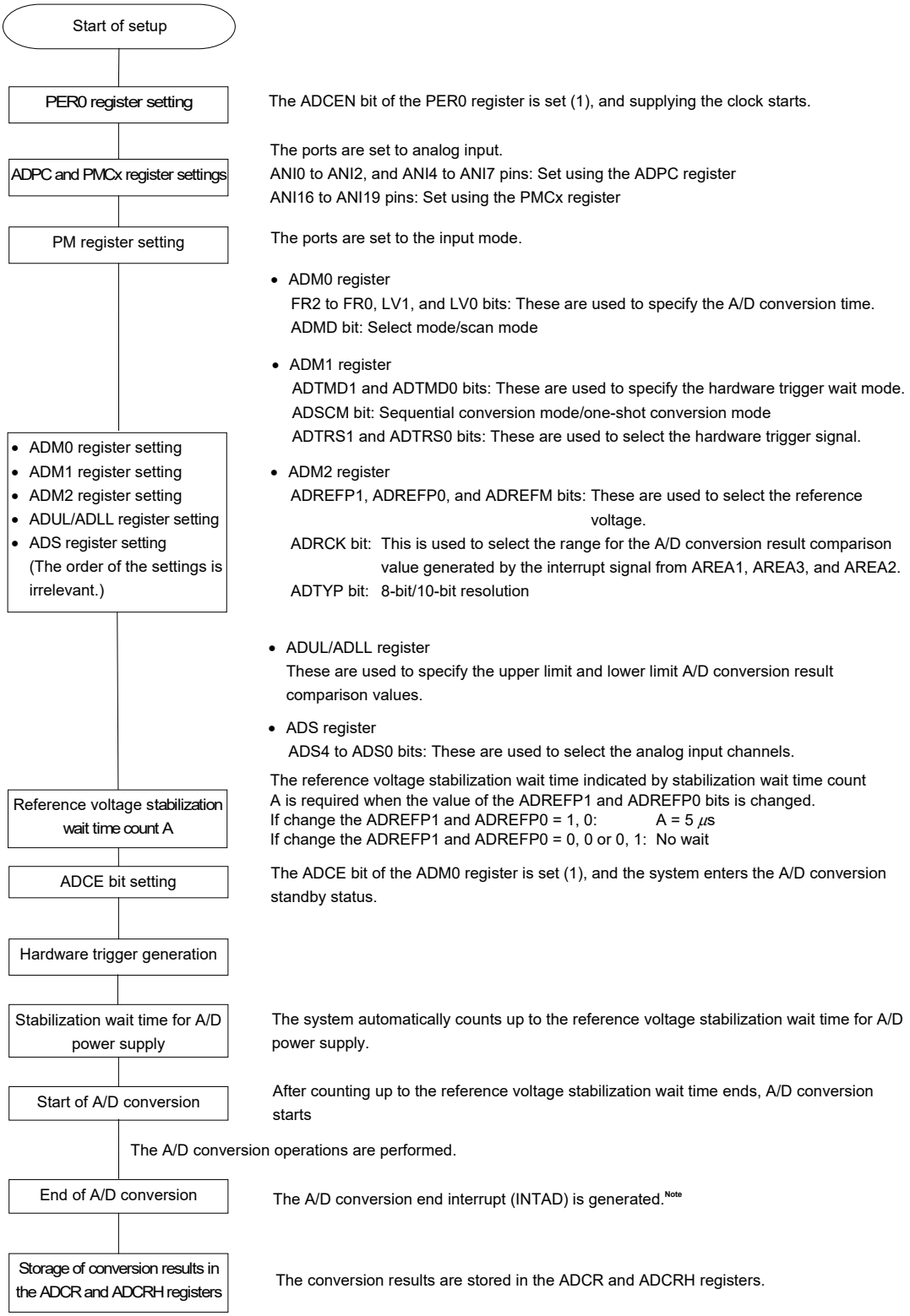


Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR register.

Caution When PGAOUT is selected as analog input, set the ADS register after setting up PGA operations (see CHAPTER 13 PROGRAMMABLE GAIN AMPLIFIER).

12.7.3 Setting up hardware trigger wait mode

Figure 12-31. Setting up Hardware Trigger Wait Mode

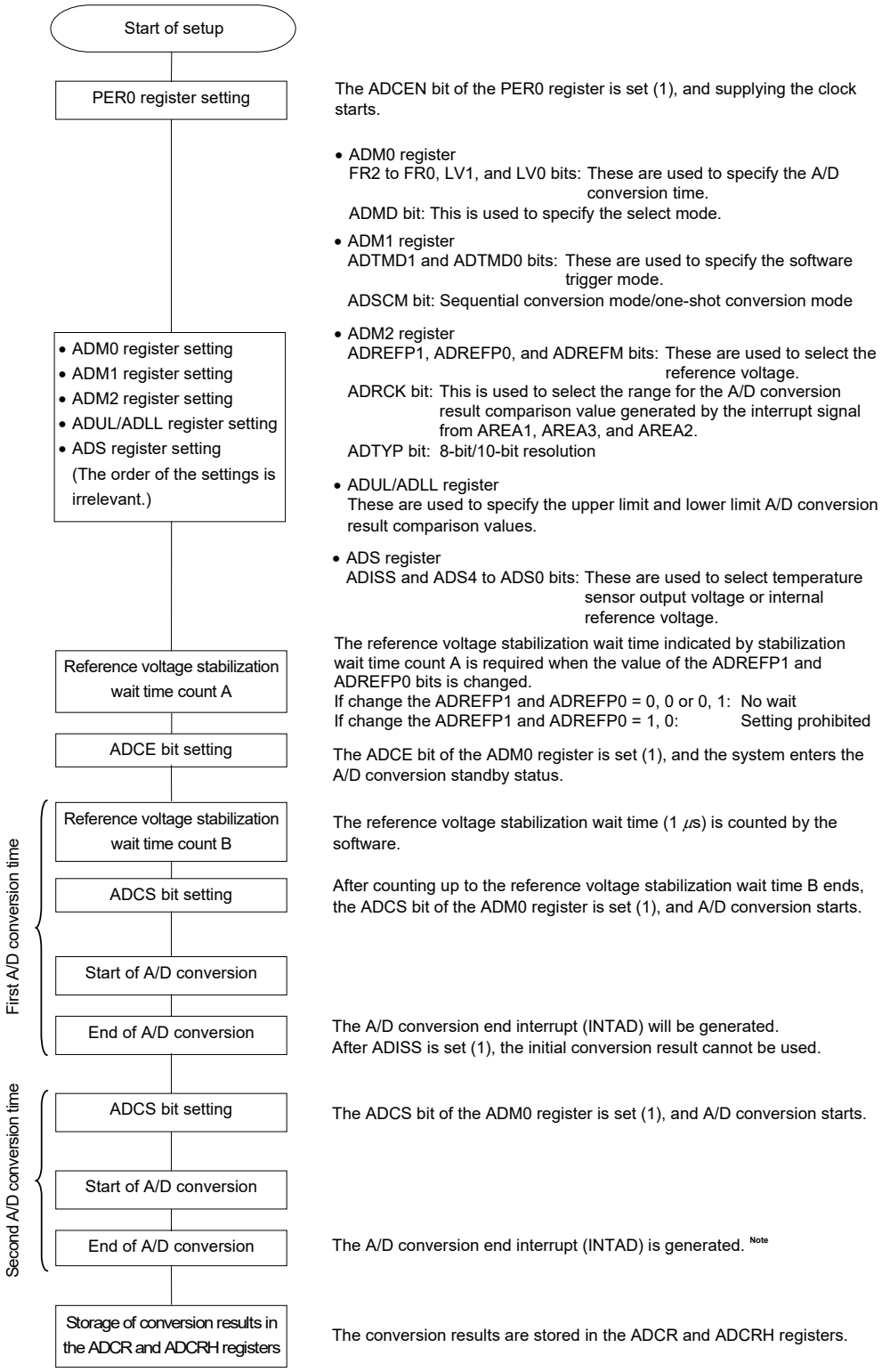


Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR register.

Caution When PGAOUT is selected as analog input, set the ADS register after setting up PGA operations (see CHAPTER 13 PROGRAMMABLE GAIN AMPLIFIER).

12.7.4 Setup when temperature sensor output voltage/internal reference voltage is selected (example for software trigger mode and one-shot conversion mode)

Figure 12-32. Setup When Temperature Sensor Output Voltage/Internal Reference Voltage Is Selected

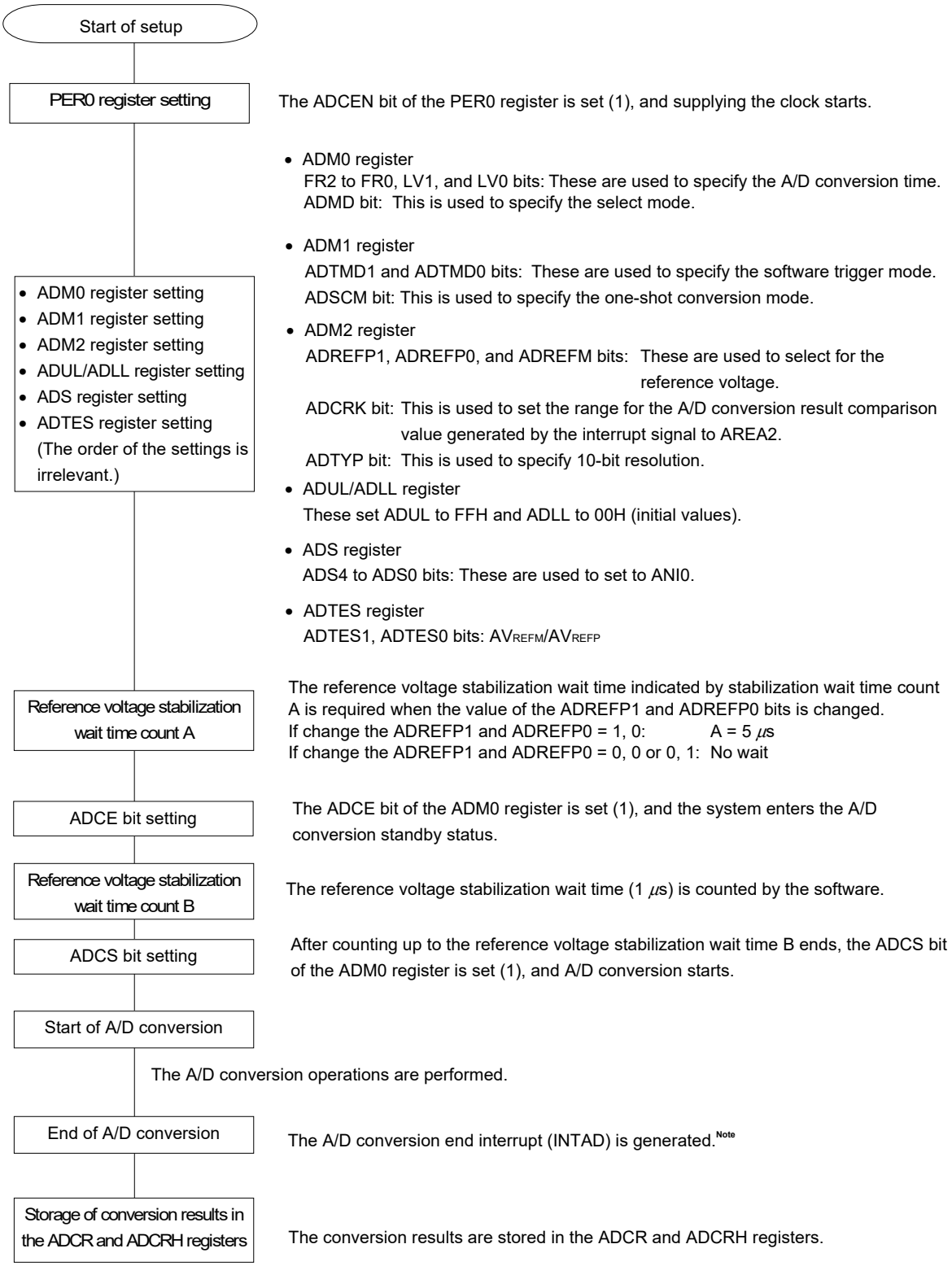


Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR register.

Caution This setting can be used only in HS (high-speed main) mode.

12.7.5 Setting up test mode

Figure 12-33. Setting up Test Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR register.

Caution For the procedure for testing the A/D converter, see 25.3.8 A/D test function.

12.8 SNOOZE Mode Function

In the SNOOZE mode, A/D conversion is triggered by inputting a hardware trigger in the STOP mode. Normally, A/D conversion is stopped while in the STOP mode, but, by using the SNOOZE mode function, A/D conversion can be performed without operating the CPU. This is effective for reducing the operating current.

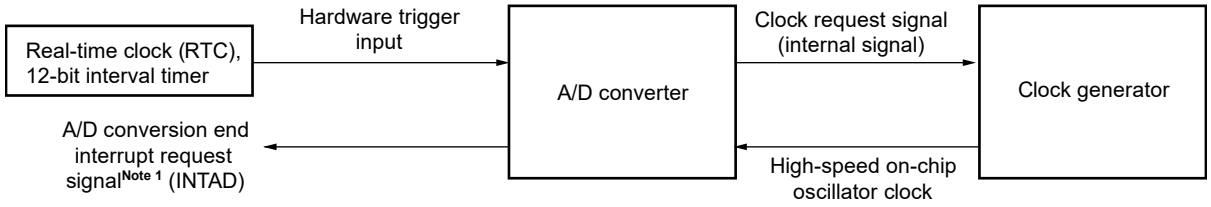
If the A/D conversion result range is specified using the ADUL and ADLL registers, A/D conversion results can be judged at a certain interval of time in SNOOZE mode. Using this function enables power supply voltage monitoring and input key judgment based on A/D inputs.

In the SNOOZE mode, only the following two conversion modes can be used:

- Hardware trigger wait mode (select mode, one-shot conversion mode)
- Hardware trigger wait mode (scan mode, one-shot conversion mode)

Caution That the SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for f_{CLK}.

Figure 12-34. Block Diagram When Using SNOOZE Mode Function



When using the SNOOZE mode function, the initial setting of each register is specified before switching to the STOP mode. (For details about these settings, see **12.7.3 Setting up hardware trigger wait mode Note 2**.) Just before move to STOP mode, bit 2 (AWC) of A/D converter mode register 2 (ADM2) is set to 1. After the initial settings are specified, bit 0 (ADCE) of A/D converter mode register 0 (ADM0) is set to 1.

If a hardware trigger is input after switching to the STOP mode, the high-speed on-chip oscillator clock is supplied to the A/D converter. After supplying this clock, the system automatically counts up to the A/D power supply stabilization wait time, and then A/D conversion starts.

The SNOOZE mode operation after A/D conversion ends differs depending on whether an interrupt signal is generated Note 1.

Notes 1. Depending on the setting of the A/D conversion result comparison function (ADRCK bit, ADUL/ADLL register), there is a possibility of no interrupt signal being generated.

2. Be sure to set the ADM1 register to E2H.

Remark The hardware trigger has INTTM01, INTRTC, INTIT, and the A/D trigger signal of timer KB0 to KB2. Specify the hardware trigger by using the A/D Converter Mode Register 1 (ADM1).

(1) If an interrupt is generated after A/D conversion ends

If the A/D conversion result value is inside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit and ADUL/ADLL register), the A/D conversion end interrupt request signal (INTAD) is generated.

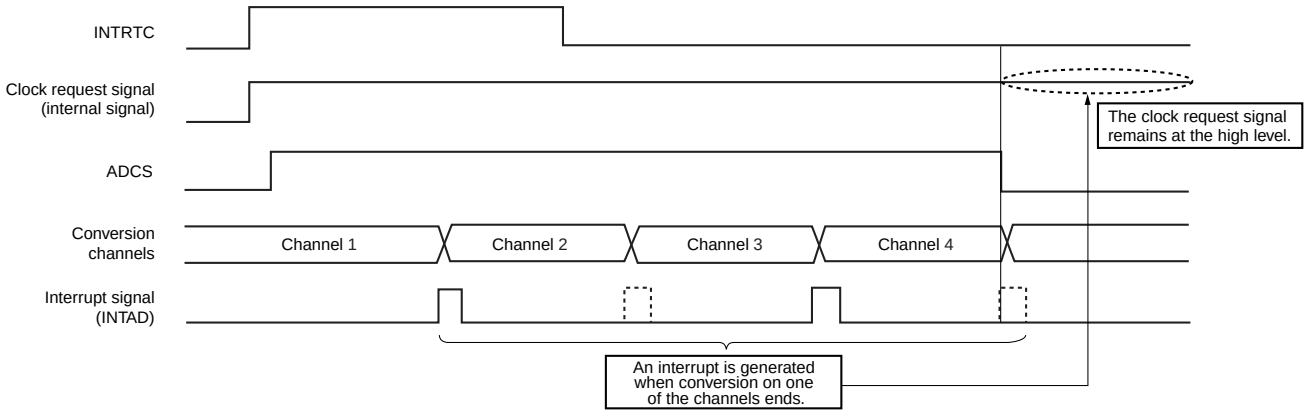
- While in the select mode

  When A/D conversion ends and an A/D conversion end interrupt request signal (INTAD) is generated, the A/D converter returns to normal operation mode from SNOOZE mode. At this time, be sure to clear bit 2 (AWC = 0: SNOOZE mode release) of the A/D converter mode register 2 (ADM2). If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent SNOOZE or normal operation mode.

- While in the scan mode

  If even one A/D conversion end interrupt request signal (INTAD) is generated during A/D conversion of the four channels, the clock request signal remains at the high level, and the A/D converter switches from the SNOOZE mode to the normal operation mode. At this time, be sure to clear bit 2 (AWC = 0: SNOOZE mode release) of A/D converter mode register 2 (ADM2) to 0. If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent SNOOZE or normal operation mode.

Figure 12-35. Operation Example When Interrupt Is Generated After A/D Conversion Ends (While in Scan Mode)



(2) If no interrupt is generated after A/D conversion ends

If the A/D conversion result value is outside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit and ADUL/ADLL register), the A/D conversion end interrupt request signal (INTAD) is not generated.

- While in the select mode

  If the A/D conversion end interrupt request signal (INTAD) is not generated after A/D conversion ends, the clock request signal (an internal signal) is automatically set to the low level, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

- While in the scan mode

  If the A/D conversion end interrupt request signal (INTAD) is not generated even once during A/D conversion of the four channels, the clock request signal (an internal signal) is automatically set to the low level after A/D conversion of the four channels ends, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

Figure 12-36. Operation Example When No Interrupt Is Generated After A/D Conversion Ends (While in Scan Mode)

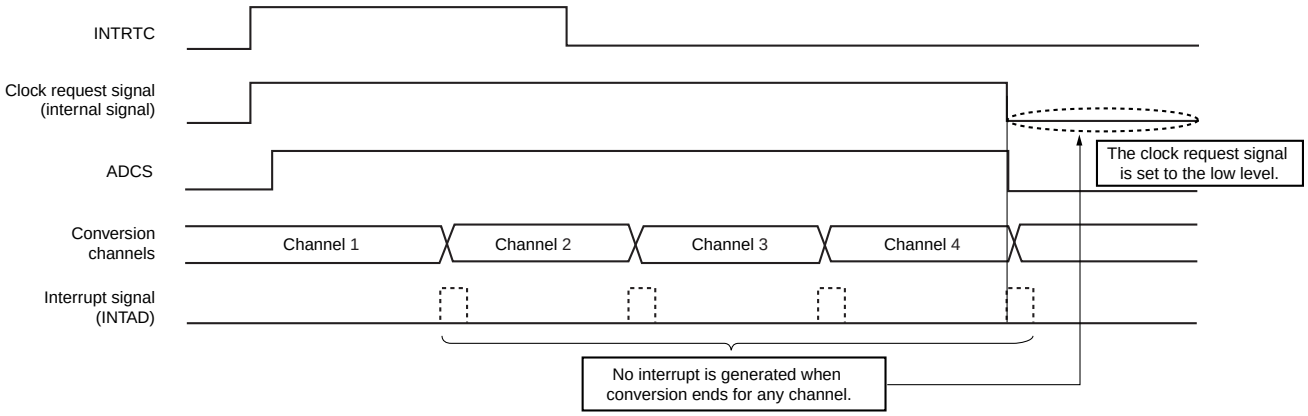
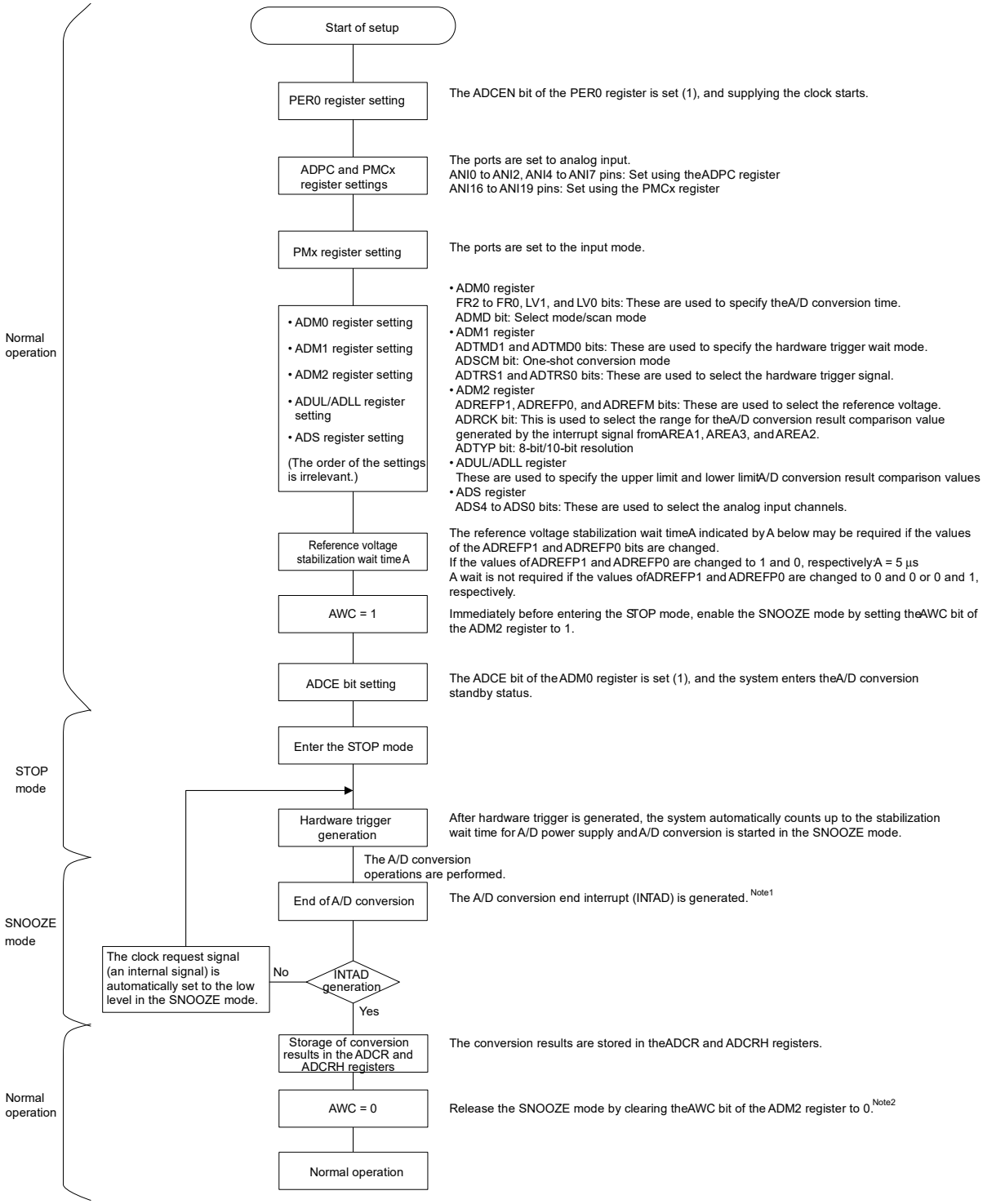


Figure 12-37. Flowchart for Setting up SNOOZE Mode



Notes 1. If the A/D conversion end interrupt request signal (INTAD) is not generated by setting ADRCK bit and ADUL/ADLL register, the result is not stored in the ADCR and ADCRH registers.
The system enters the STOP mode again. If a hardware trigger is input later, A/D conversion operation is again performed in the SNOOZE mode.

2. If the AWC bit is left set to 1, A/D conversion will not start normally in spite of the subsequent SNOOZE or normal operation mode. Be sure to clear the AWC bit to 0.

12.9 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 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 10 bits.

$1LSB = 1/2^{10} = 1/1024$
$= 0.098\%FSR$

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.
Zero-scale error, full-scale error, integral linearity error, and differential linearity errors that are combinations of these express the overall error.
Note that the quantization error is not included in the overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, a ±1/2LSB 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 12-38. Overall Error

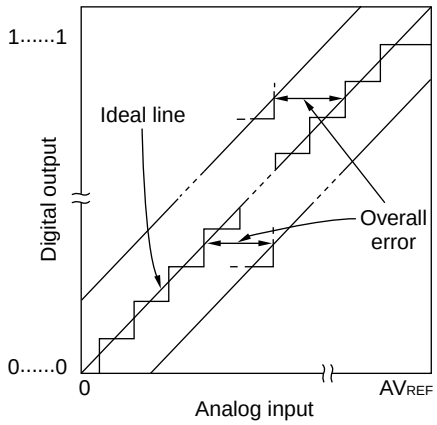
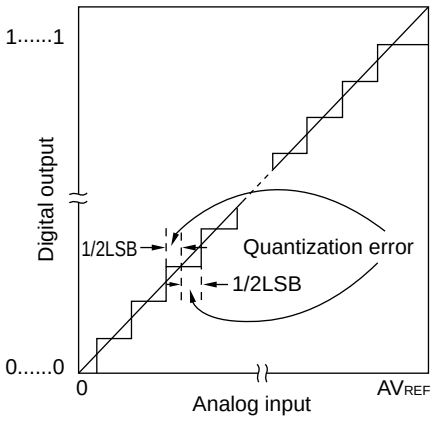


Figure 12-39. Quantization Error



(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.

Figure 12-40. Zero-Scale Error

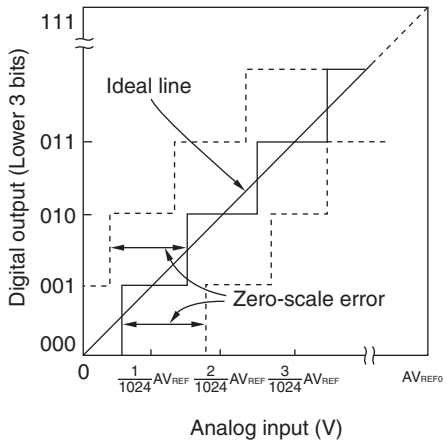


Figure 12-41. Full-Scale Error

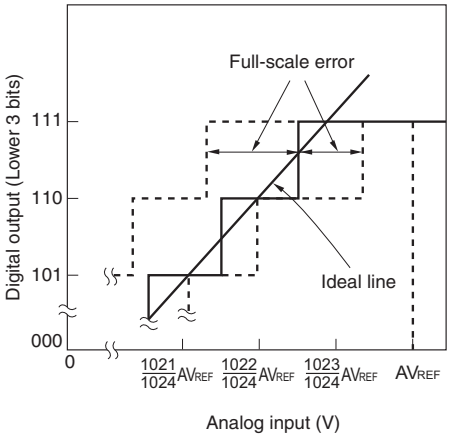


Figure 12-42. Integral Linearity Error

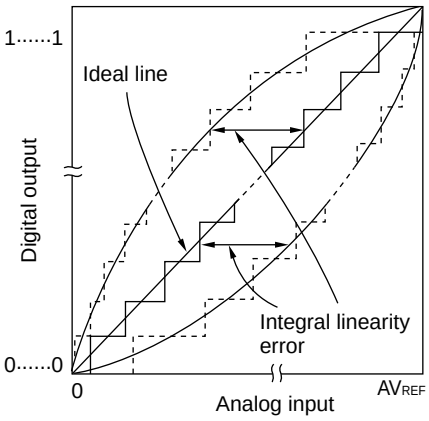
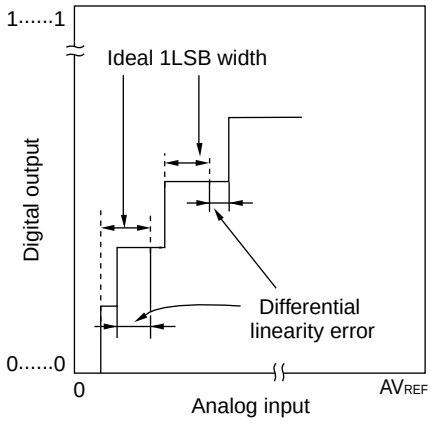


Figure 12-43. Differential Linearity Error



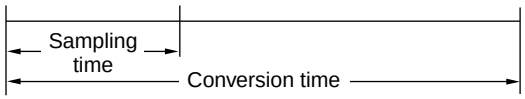
(8) Conversion time

This expresses the time from the start of sampling to when the digital output is obtained.

The sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.



12.10 Cautions for A/D Converter

(1) Operating current in STOP mode

Shift to STOP mode after stopping the A/D converter (by setting bit 7 (ADCS) of A/D converter mode register 0 (ADM0) to 0). The operating current can be reduced by setting bit 0 (ADCE) of the ADM0 register to 0 at the same time.

To restart from the standby status, clear bit 0 (ADIF) of interrupt request flag register 1H (IF1H) to 0 and start operation.

(2) Input range of ANI0 to ANI2, ANI4 to ANI7, ANI16 to ANI19, and PGAOUT pins

Observe the rated range of the ANI0 to ANI2, ANI4 to ANI7, ANI16 to ANI19, and PGAOUT pins input voltage. 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.

Caution Internal reference voltage (1.45 V) can be used only in HS (high-speed main) mode.

(3) Conflicting operations

<1> Conflict between the A/D conversion result register (ADCR, ADCRH) write and the ADCR or ADCRH register read by instruction upon the end of conversion

The ADCR or ADCRH register read has priority. After the read operation, the new conversion result is written to the ADCR or ADCRH registers.

<2> Conflict between the ADCR or ADCRH register write and the A/D converter mode register 0 (ADM0) write, the analog input channel specification register (ADS), or A/D port configuration register (ADPC) write upon the end of conversion

The ADM0, ADS, or ADPC registers write has priority. The ADCR or ADCRH register write is not performed, nor is the conversion end interrupt signal (INTAD) generated.

(4) Noise countermeasures

To maintain the 10-bit resolution, attention must be paid to noise input to the AV_{REFP}, V_{DD}, ANI0 to ANI2, ANI4 to ANI7, and ANI16 to ANI19 pins.

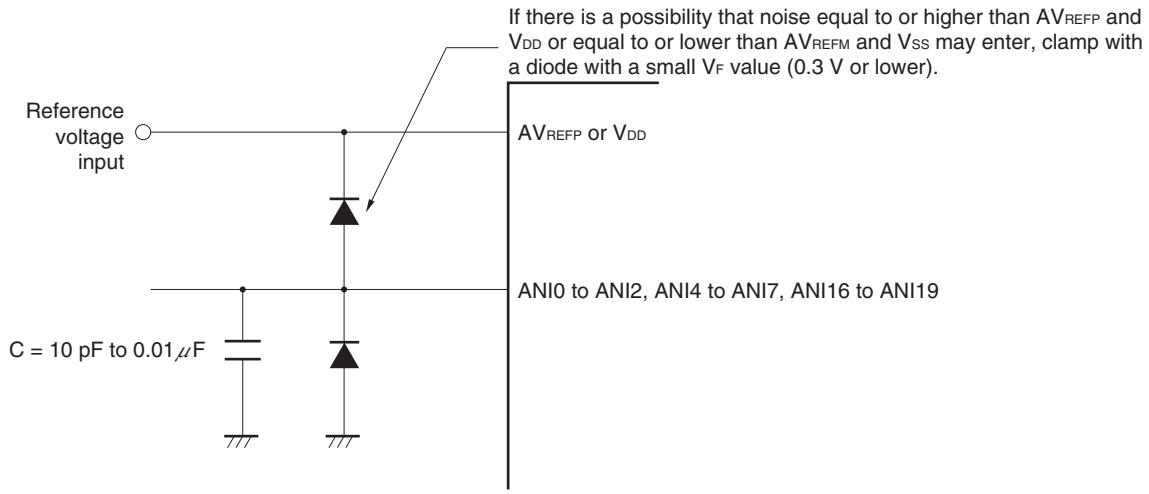
<1> Connect a capacitor with a low equivalent resistance and a good frequency response (capacitance of about 0.01 μF) via the shortest possible run of relatively thick wiring to the power supply.

<2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting an external capacitor as shown in **Figure 12-44** is recommended.

<3> Do not switch these pins with other pins during conversion.

<4> The accuracy is improved if the HALT mode is set immediately after the start of conversion.

Figure 12-44. Analog Input Pin Connection



(5) Analog input (ANIn) pins

<1> The analog input pins (ANI0 to ANI2, ANI4 to ANI7, ANI16 to ANI19) are also used as input port pins (P20 to P22, P24 to P27, P03, P02, P147, P120).
When A/D conversion is performed with any of the ANI0 to ANI2, ANI4 to ANI7 and ANI16 to ANI19 pins selected, do not change to output value P20 to P22, P24 to P27, P03, P02, P147 and P120 while conversion is in progress; otherwise the conversion resolution may be degraded.

<2> If a pin adjacent to a pin that is being A/D converted is used as a digital I/O port pin, the A/D conversion result might differ from the expected value due to a coupling noise. Be sure to prevent such a pulse from being input or output. 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 (ANIn) pins

This A/D converter charges a sampling capacitor for sampling during sampling time.
Therefore, only a leakage current flows when sampling is not in progress, and a current that charges the capacitor flows during sampling. Consequently, the input impedance fluctuates depending on whether sampling is in progress, and on the other states.
To make sure that sampling is effective, however, we recommend using the converter with analog input sources that have output impedances no greater than 1 kΩ. If a source has a higher output impedance, lengthen the sampling time or connect a larger capacitor (with a value of about 0.1 μF) to the pin from among ANI0 to ANI2, ANI4 to ANI7, and ANI16 to ANI19 to which the source is connected (see **Figure 12-44**). The sampling capacitor may be being charged while the setting of the ADCS bit is 0 and immediately after sampling is restarted and so is not defined at these times. Accordingly, the state of conversion is undefined after charging starts in the next round of conversion after the value of the ADCS bit has been 1 or when conversion is repeated. Thus, to secure full charging regardless of the size of fluctuations in the analog signal, ensure that the output impedances of the sources of analog inputs are low or secure sufficient time for the completion of conversion.

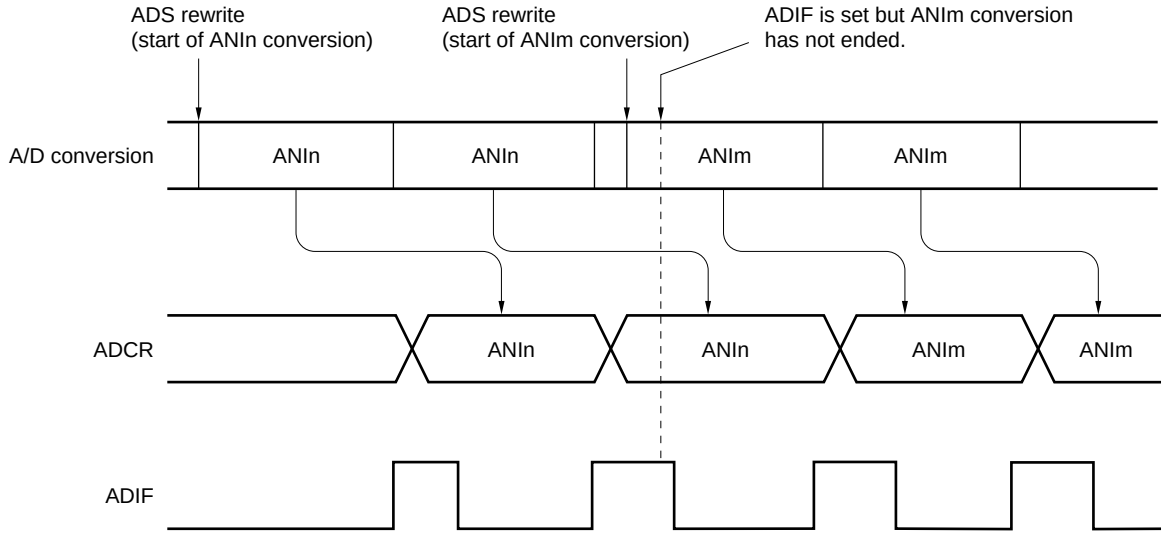
(7) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the analog input channel specification register (ADS) is changed.

Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and ADIF flag for the pre-change analog input may be set just before the ADS register rewrite. Caution is therefore required since, at this time, when ADIF flag is read immediately after the ADS register rewrite, ADIF flag is set despite the fact A/D conversion for the post-change analog input has not ended.

When A/D conversion is stopped and then resumed, clear ADIF flag before the A/D conversion operation is resumed.

Figure 12-45. Timing of A/D Conversion End Interrupt Request Generation



(8) Conversion results just after A/D conversion start

While in the software trigger mode or hardware trigger no-wait mode, the first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 1 μs after the ADCE bit was set to 1. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.

(9) A/D conversion result register (ADCR, ADCRH) read operation

When a write operation is performed to A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), A/D port configuration register (ADPC), and port mode control register (PMC), the contents of the ADCR and ADCRH registers may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, ADPC, or PMC register. Using a timing other than the above may cause an incorrect conversion result to be read.

(10) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 12-46. Internal Equivalent Circuit of ANIn Pin

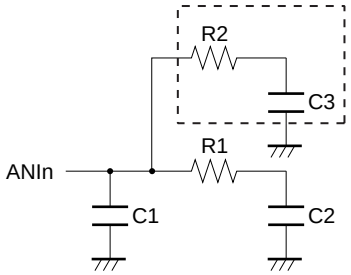


Table 12-4. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{REFP}, V_{DD}	ANIn Pins	R1 [kΩ]	C1 [pF]	C2 [pF]
3.6 V ≤ V_{DD} ≤ 5.5 V	ANI0 to ANI2, ANI4 to ANI7	14	8	2.5
	ANI16 to ANI19	18	8	7.0
2.7 V ≤ V_{DD} < 3.6 V	ANI0 to ANI2, ANI4 to ANI7	39	8	2.5
	ANI16 to ANI19	53	8	7.0

AV_{REFP}, V_{DD}	ANIn Pins	R2/C3	MIN.	TYP.	MAX.
2.7 V ≤ V_{DD} < 5.5 V	Other than ANI0, ANI1, ANI17, ANI19 pins	R2 [kΩ]	2		3200
		C3 [pF]			2

Remark The resistance and capacitance values shown in Table 12-4 are not guaranteed values.

(11) Starting the A/D converter

Start the A/D converter after the AV_{REFP} and V_{DD} voltages stabilize.

(12) Cautions when using a PGA

The A/D converter is affected by noise input from the signal source or analog power supply (AV_{REFP}/AV_{REFM}).

This effect can be very large when using a PGA, which uses a particularly small voltage as the signal source. When using a PGA, therefore, be sure to evaluate it thoroughly by taking measures to reduce the amount of noise entering the A/D converter from peripheral circuits and adjusting the A/D conversion timing and conversion time.

CHAPTER 13 PROGRAMMABLE GAIN AMPLIFIER

A circuit of programmable gain amplifier is incorporated in RL78/I1A.
The number of analog input channels corresponding to programmable gain amplifier differs, depending on the product.

	20-pin	30-pin, 38-pin
Analog input channels	4 ch (ANI2/CMP0P, ANI4/CMP1P, ANI5/CMP2P, ANI18/(CMP3P)/(CMPCOM))	6 ch (ANI2/CMP0P, ANI4/CMP1P to ANI7/CMP4P, ANI16/CMP5P, ANI18/CMPCOM)

Caution Most of the following descriptions in this chapter use the 38-pin products as an example.

13.1 Functions of Programmable Gain Amplifier

The programmable gain amplifier is provided with the following functions.

- Programmable gain amplifier input can be selected from among the seven CMP0P-CMP5P, CMPCOM pins.
- One among four amplification factors can be selected.
- The output signal of a programmable gain amplifier can be set as the analog input of an A/D converter.

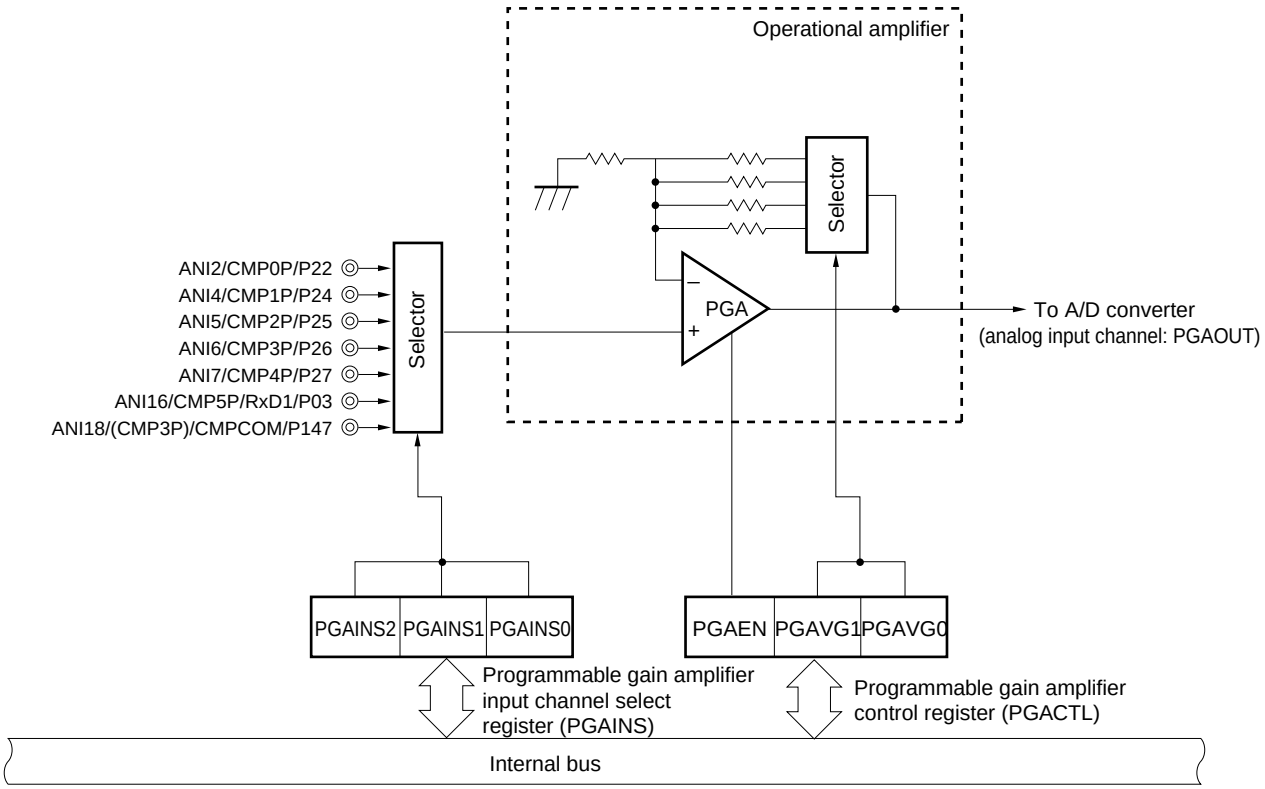
13.2 Configuration of Programmable Gain Amplifier

The programmable gain amplifier includes the following hardware.

Table 13-1. Configuration of Programmable Gain Amplifier

Item	Configuration
Programmable gain amplifier main unit	Analog input channel MAX. 6 ch
Programmable gain amplifier input	CMP0P-CMP5P, CMPCOM pins
Control registers	Peripheral enable register 2 (PER2) Programmable gain amplifier control register (PGACTL) Programmable gain amplifier input channel select register (PGAINS) A/D port configuration register (ADPC) Port mode control registers 0, 14 (PMC0, PMC14) Port mode registers 0, 2, 14 (PM0, PM2, PM14) Comparator and PGA internal reference voltage control register (CVRCTL)

Figure 13-1. Block Diagram of Operational Amplifier



13.3 Registers Used in Operational Amplifier

The operational amplifier uses the following eight registers.

- Peripheral enable register 2 (PER2)
- Programmable gain amplifier control register (PGACTL)
- Programmable gain amplifier input channel select register (PGAINS)
- A/D port configuration register (ADPC)
- Port mode control registers 0, 14 (PMC0, PMC14)
- Port mode registers 0, 2, 14 (PM0, PM2, PM14)
- Comparator and PGA internal reference voltage control register (CVRCTL)

13.3.1 Peripheral enable register 2 (PER2)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the programmable gain amplifier is used, be sure to set bit 7 (PGACMPEN) of this register to 1.

The PER2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 13-2. Format of Peripheral Enable Register 2 (PER2)

Address: F0509H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER2	PGACMPEN	TKBPA2EN	TKBPA1EN	TKBPA0EN	TKC0EN	TKB2EN[Note]	TKB1EN	TKB0EN

PGACMPEN	Comparator/programmable gain amplifier input clock control
0	Stops supply of input clock. • SFR used by the comparator or programmable gain amplifier cannot be written. • The comparator or programmable gain amplifier is in the reset status.
1	Supplies input clock. • SFR used by the comparator or programmable gain amplifier can be read/written.

Note 30-pin products and 38-pin products only.

13.3.2 Programmable gain amplifier control register (PGACTL)

PGACTL controls the operations of programmable gain amplifier.

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

Reset signal generation clears this register to 00H.

Figure 13-3. Format of Programmable Gain Amplifier Control Register (PGACTL)

Address: F0550H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
PGACTL	PGAEN	0	0	0	0	0	PGAVG1	PGAVG0

PGAEN	Programmable gain amplifier operation control
0	Stops operation of programmable gain amplifier.
1	Enables operation of programmable gain amplifier.

PGAVG1	PGAVG0	Programmable gain amplifier amplification factor selection
0	0	×4
0	1	×8
1	0	×16
1	1	×32

Cautions 1. When using the programmable gain amplifier, use the ADPC register to select the CMP0P/ANI2/P22, CMP1P/ANI4/P24, CMP2P/ANI5/P25, CMP3P/ANI6/P26, CMP4P/ANI7/P27 pins, and PGAOUT pin (internal pin) as analog inputs. Use the PMC0 and PMC14 registers to select the CMP5P/ANI16/RxD1/P03 and CMPCOM/ANI18/P147 pins as analog inputs.

2. When using as digital inputs the pins of port 0 and port 2, which are not used with the programmable gain amplifier, when the programmable gain amplifier is used, make sure that the input levels of digital input ports are fixed to prevent degradation of the A/D conversion accuracy.

3. Set the amplification factor before enabling (PGAEN = 1) the operation of the programmable gain amplifier. Changing the amplification factor setting in the operation enabled state (PGAEN = 1) is prohibited.

4. For the programmable gain amplifier, an operation stabilization wait time (5 μs when the amplification factor is set to x4 or x8, or 10 μs when set to x16 or x32) is required after setting PGAEN = 1.

13.3.3 Programmable gain amplifier input channel select register (PGAINS)

This register selects the input channel for the programmable gain amplifier.

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

Reset signal generation clears this register to 00H.

Figure 13-4. Format of Programmable Gain Amplifier Input Channel Select Register (PGAINS)

Address: F0551H After reset: 00H 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 for input to programmable gain amplifier
0	0	0	ANI2/CMP0P
0	0	1	ANI4/CMP1P
0	1	0	ANI5/CMP2P
0	1	1	ANI6/CMP3P
1	0	0	ANI7/CMP4P
1	0	1	ANI16/CMP5P
1	1	0	ANI18/CMPCOM/(CMP3P Note)
Other than above			Setting prohibited

Note Selected by the comparator input switch control register (CMPSEL) (20-pin products only)

Caution Set the PGAINS register during stop operation of the programmable gain amplifier (PGAEN = 0).

13.3.4 A/D port configuration register (ADPC)

This register switches the ANI0/P20, ANI1/P21, ANI2/CMP0P/P22, ANI4/CMP1P/P24 to ANI7/CMP4P/P27 pins, and PGAOUT pin (internal pin) to digital I/O of port or analog input of A/D converter, programmable gain amplifier, or comparator.

When using the programmable gain amplifier or comparator, use the ADPC register to select the ANI2/CMP0P/P22, ANI4/CMP1P/P24 to ANI7/CMP4P/P27 pins, and PGAOUT pin (internal pin) as analog inputs.

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

Reset signal generation sets this register to 00H.

Figure 13-5. Format of A/D Port Configuration Register (ADPC)

Address: F0076H After reset: 00H R/W

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

ADPC3	ADPC2	ADPC1	ADPC0	Analog input (A)/digital I/O (D) switching							
				ANI7/ CMP4P/P27	ANI6/ CMP3P/P26	ANI5/ CMP2P/P25	ANI4/ CMP1P/P24	PGAOUT[Note]	ANI2/ CMP0P/P22	ANI1/P21/ AV_{REFM}	ANI0/P20/ AV_{REFP}
0	0	0	0	A	A	A	A	A	A	A	A
0	0	0	1	D	D	D	D	D	D	D	D
0	0	1	0	D	D	D	D	D	D	D	A
0	0	1	1	D	D	D	D	D	D	A	A
0	1	0	0	D	D	D	D	D	A	A	A
0	1	0	1	D	D	D	D	A	A	A	A
0	1	1	0	D	D	D	A	A	A	A	A
0	1	1	1	D	D	A	A	A	A	A	A
1	0	0	0	D	A	A	A	A	A	A	A
1	0	0	1	A	A	A	A	A	A	A	A
1	1	1	1	A	A	A	A	A	A	A	A
Other than above				Setting prohibited							

Note This is an internal output pin for the programmable gain amplifier. When output signals from the programmable gain amplifier is used as an analog input channel for the A/D converter, set ADPC as 0000B or 0101B or above.

Cautions 1. Set the channel used for A/D conversion to the input mode by using port mode register 2 (PM2).

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

13.3.5 Port mode control registers 0, 14 (PMC0, PMC14)

These registers are used to set the digital I/O/analog input of port 0 or 14 in 1-bit units.

To use the ANI16/CMP5P/RxD1/P03 or ANI18/CMPCOM/P147 pin as an analog input pin, set the PMC03 or PMC147 bit to 1.

The PMC0 and PMC14 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 13-6. Format of Port Mode Control Registers 0, 14 (PMC0, PMC14)

Address: F0060H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PMC0	1	1	1	1	PMC03	PMC02	1	1

Address: F006EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PMC14	PMC147	1	1	1	1	1	1	1

PMCmn	Pmn pin digital I/O/analog input selection (mn = 02, 03, 147)
0	Digital I/O (alternate function other than analog input)
1	Analog input

13.3.6 Port mode registers 0, 2, 14 (PM0, PM2, PM14)

When using the ANI2/CMP0P/P22, ANI4/CMP1P/P24 to ANI7/CMP4P/P27, ANI16/CMP5P/RxD1/P03, or ANI18/CMPCOM/P147 pin for an analog input port, set the PM22, PM24 to PM27, PM03, or PM147 bit to 1. The output latches of P22, P24 to P27, P03, and P147 at this time may be 0 or 1.

If the PM22, PM24 to PM27, PM03, and PM147 bits are set to 0, they cannot be used as analog input port pins.

The PM0, PM2, and PM14 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Caution When a pin is set as an analog input port, not the pin level but "0" is always read despite PMxx = 1.

Figure 13-7. Format of Port Mode Registers 0, 2, 14 (PM0, PM2, PM14) (38-pin Products)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	PM06	PM05	1	PM03	PM02	1	1

Address: FFF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	1	PM22	PM21	PM20

Address: FFF2EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM14	PM147	1	1	1	1	1	1	1

PMmn	Pmn pin I/O mode selection (m = 0, 2, 14; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bits 0, 1, 4, and 7 of the PM0 register, bit 3 of the PM2 register, bits 0 to 6 of the PM14 register, and bit 7 of the PM20 register to "1".
For 30- and 20-pin products, the following bits must be set to output mode (specified by setting the port registers and port mode registers to 0) by software after reset is cancelled.
30-pin products: Bits 5 and 6 of the PM0 register
20-pin products: Bits 2, 3, 5, and 6 of the PM0 register, and bits 6 and 7 of the PM2 register

Remark The figure shown above presents the format of port mode registers 0, 2, and 14 of the 38-pin products. For the format of the port mode register of other products, see **Table 4-3 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx Registers and Bits Mounted on Each Product**.

The functions of the ANI2/CMP0P/P22, ANI4/CMP1P/P24 to ANI7/CMP4P/P27 pins, and PGAOUT/P23 pin (internal pin) can be selected by using the A/D port configuration register (ADPC) and PM2 register.

Table 13-2. Setting Functions of ANI2/CMP0P/P22, ANI4/CMP1P/P24 to ANI7/CMP4P/P27 Pins, and PGAOUT/P23 Pin (Internal Pin)

ADPC	PM2	ANI2/CMP0P/P22, ANI4/CMP1P/P24 to ANI7/CMP4P/P27 Pins, and PGAOUT/P23 Pin (Internal Pin)
Digital I/O selection	Input mode	Digital input
	Output mode	Digital output
Analog input selection	Input mode	Analog input
	Output mode	Setting prohibited

The functions of the ANI16/CMP5P/RxD1/P03 and ANI18/CMPCOM/P147 pins can be selected by using the port mode control registers 0, 14 (PMC0, PMC14), PM0, and PM14 registers.

Table 13-3. Setting Functions of ANI16/CMP5P/RxD1/P03 and ANI18/CMPCOM/P147 Pins

PMC0, PMC14	PM0, PM14	ANI16/CMP5P/RxD1/P03 and ANI18/CMPCOM/P147 Pins
Digital I/O selection	Input mode	Digital input
	Output mode	Digital output
Analog input selection	Input mode	Analog input
	Output mode	Setting prohibited

13.3.7 Comparator and PGA internal reference voltage control register (CVRCTL)

This register is used to control the internal reference voltage of the comparator, and to select the sources of the internal reference voltage and the comparator/GND of programmable gain amplifier.

The internal reference voltage is enabled or stopped by using the CVREm bit.

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

Reset signal generation clears this register to 00H.

Figure 13-8. Format of Comparator Internal Reference Voltage Control Register (CVRCTL)

Address: F0560 H After reset: 00H R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
CVRCTL	0	0	CVRVS1	CVRVS0	0	CVRE2	CVRE1	CVRE0

CVRVS1	GND selection of internal reference voltage and programmable gain amplifier
0	V_{SS}
1	AV_{REFM}

CVRVS0	Power supply selection of internal reference voltage
0	V_{DD}
1	AV_{REFP}

CVREm	Internal reference voltage (DAm) generation operation control
0	Stops operation.
1	Enables operation.

Cautions 1. The stabilization wait time (10 μs) is required after setting the CVREm bit.

2. Rewrite the CVRVS1 and CVRVS0 bits during stop operation of the internal reference voltage (CVREm = 0). Rewrite the CVRVS1 bit while the programmable gain amplifier operation is stopped (PGAEN = 0 in the PGACTL register).

Remark m = 0 to 2

13.4 Operation of Programmable Gain Amplifier

The analog voltage input from the CMP0P to CMP5P and CMPCOM pins is 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.

The procedure for starting operation of the programmable gain amplifier is described below.

<1> Use the ADPC, PMC0, and PMC14 registers to set the pins (CMP0P to CMP5P and CMPCOM) to be used in the programmable gain amplifier as analog inputs.
<2> Use the PM0, PM2, and PM14 registers to set the pins (CMP0P to CMP5P and CMPCOM) to be used in the programmable gain amplifier to input mode.
<3> Use the PGAVG0 and PGAVG1 bits to select the gain (×4, ×8, ×16, and ×32).
<4> Use the PGAINS0 to PGAINS2 bits to select the pin for input to the programmable gain amplifier.
<5> Use the ADS register to select PGAOUT as the analog input channel.
<6> Set (1) the PGAEN bit and enable operation of the programmable gain amplifier.

13.5 Setting Procedure of Programmable Gain Amplifier

Figure 13-9. Operation Setting Flow Chart of Programmable Gain Amplifier (PGA)
(Using PGA Output as Analog Input of A/D Converter)

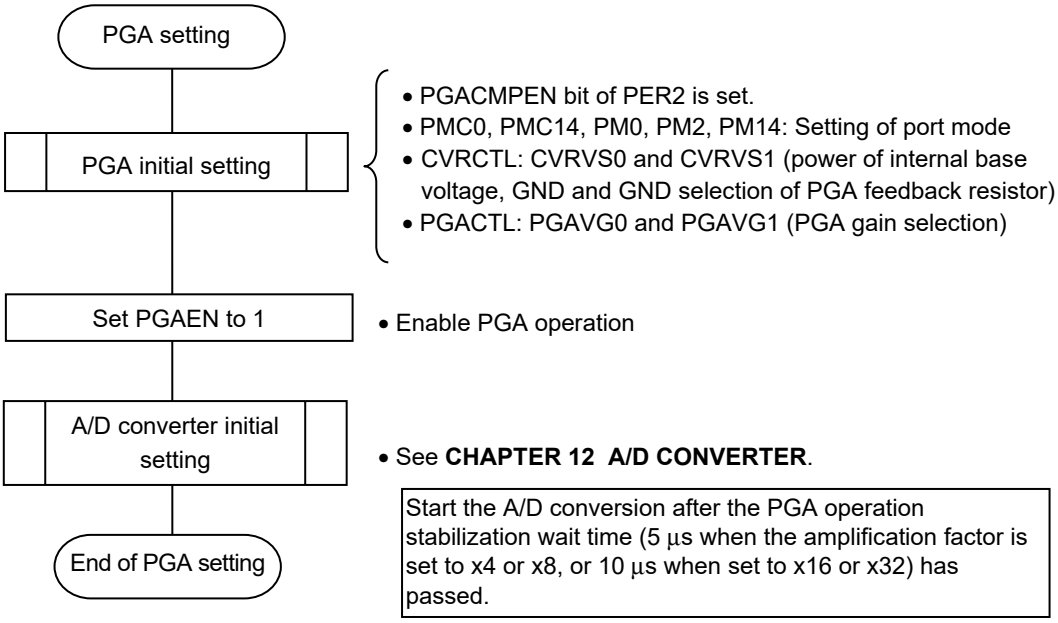
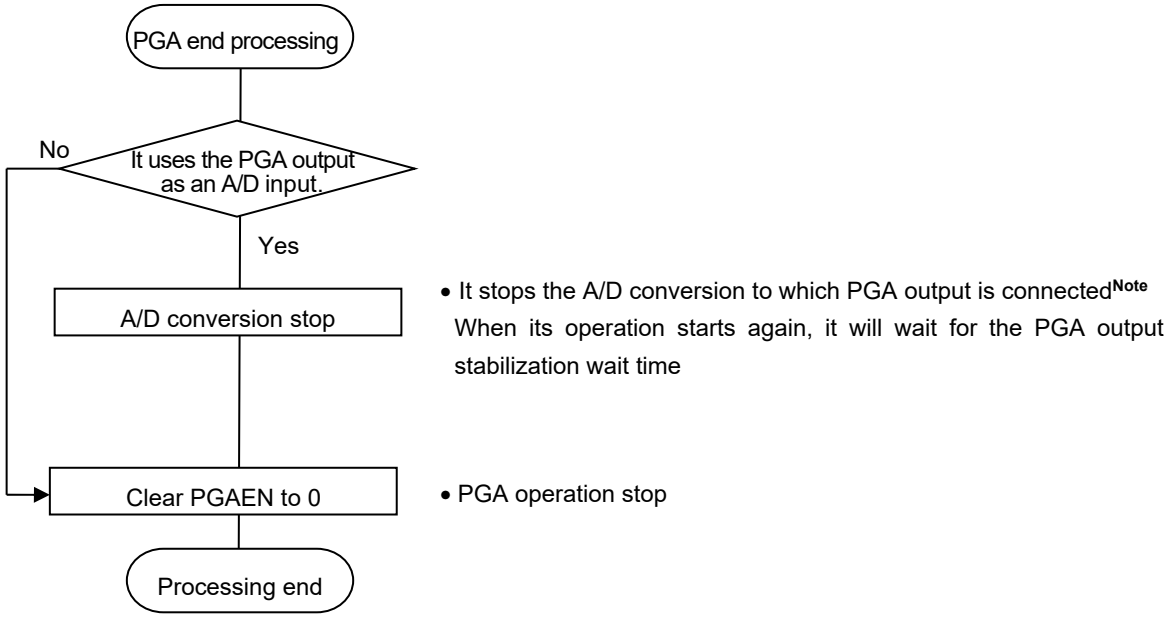


Figure 13-10. Operation End Flow Chart of Programmable Gain Amplifier



- When A/D to which the PGA output is connected is switched or its operation is restarted, the operational setting flow should be performed after executing the PGA termination flow.

Note Conversion of the A/D to which the PGA output is not connected is possible.

CHAPTER 14 COMPARATOR

The number of channels of the comparator differs, depending on the product.

	20-pin	30-pin, 38-pin
Channels (analog input channels)	4 ch (ANI2/CMP0P, ANI4/CMP1P, ANI5/CMP2P, ANI18/(CMP3P)/(CMPCOM))	6 ch (ANI2/CMP0P, ANI4/CMP1P to ANI7/CMP4P, ANI16/CMP5P, P147/CMPCOM/ANI18))

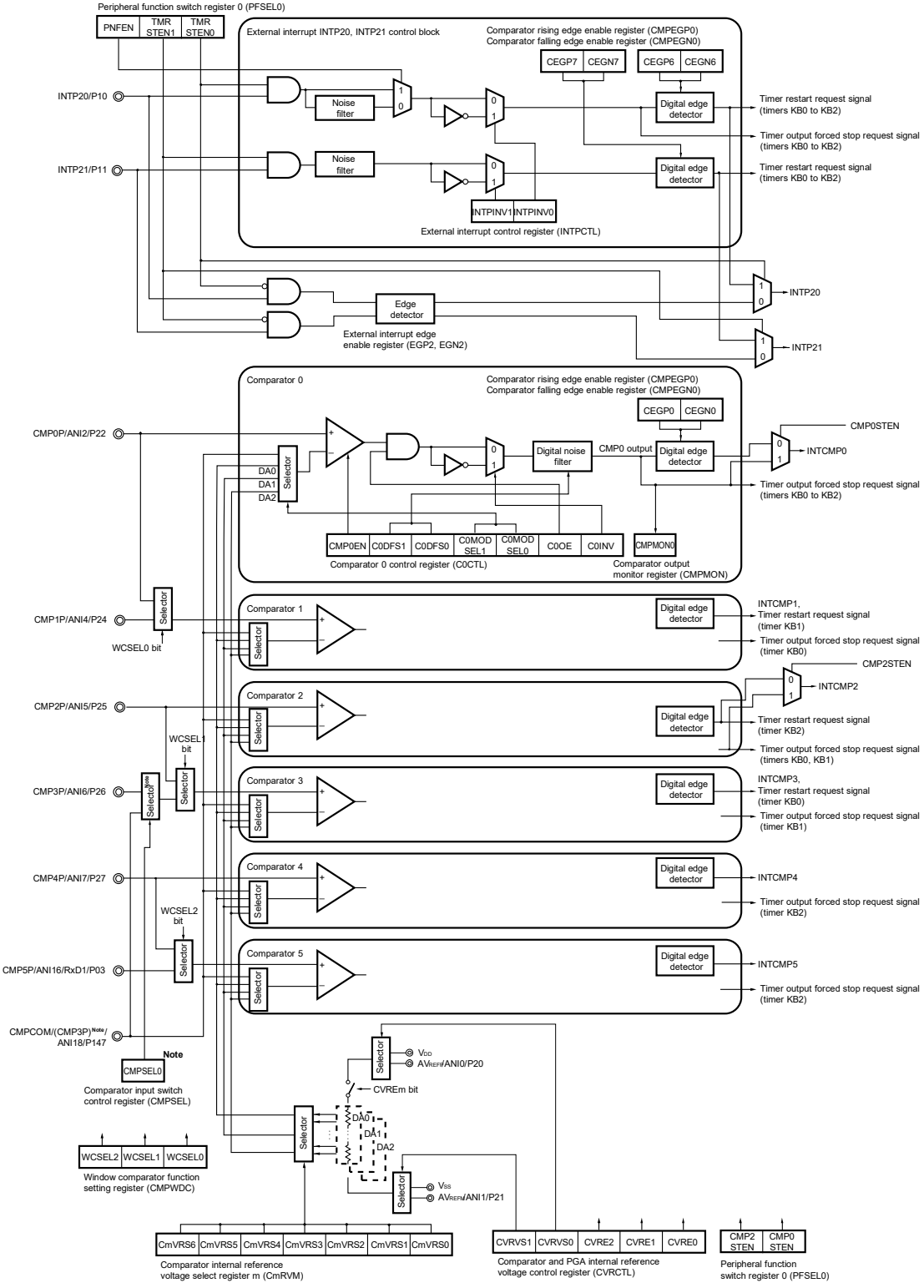
Caution Most of the following descriptions in this chapter use the 38-pin products as an example.

14.1 Functions of Comparator

The comparator is provided with the following functions.

- Comparators are equipped with six channels (comparators 0 to 5).
- The following reference voltages can be selected.
  <1> Internal reference voltage: 3 (256 resolution based on V_{DD}/AV_{REFP} and V_{SS}/AV_{REFM})
  <2> Input voltage from external reference voltage input pin (CMPCOM)
- An interrupt signal can be generated by detecting the valid edge of the comparator output. The valid edge can be set by using the CEGPn and CEGNn bits (n = 0 to 5).
- The comparator output can be used as the PWM output of 16-bit timers KB0, KB1, and KB2 and a timer counter reset (see **CHAPTER 7 16-BIT TIMERS KB0, KB1, AND KB2**).
- The elimination width of the noise elimination digital filter can be selected.
- Window comparator function (comparator 0 and 1, comparator 2 and 3, comparator 4 and 5)
  When one input voltage is selected for two comparators, this function compares the single input voltage to two reference voltages.

Figure 14-1. Block Diagram of Comparator



Note 20-pin products only. ANI16/CMP3P/P26 is selected by default for 30- and 38-pin products.

Caution When INTP20, INTP21, and comparator are used as the timer KB forced output stop function 2 or timer KB restart function, see 14. 5 Caution for Using Timer KB Simultaneous Operation Function.

Remark m = 0 to 2

14.2 Configuration of Comparator

The comparator includes the following hardware.

Table 14-1. Configuration of Comparator

Item	Configuration
Comparator main unit	Comparator MAX. 6 ch
Programmable gain amplifier input	Internal reference voltage 3 and external CAPCOM input
Noise filter	Noise elimination digital filter
Control registers	Peripheral enable register 2 (PER2) Comparator n control register (CnCTL) Comparator and PGA internal reference voltage control register (CVRCTL) Comparator internal reference voltage select register m (CmRVM) Comparator rising edge enable register0 (CMPEGP0) Comparator falling edge enable register0 (CMPEGN0) Comparator output monitor register (CMPMON) Window comparator function setting register (CMPWDC) Comparator input switch control register (CMPSEL) (20-pin products only) External interrupt control register (INTPCTL) Peripheral function switch register 0 (PFSEL0) A/D port configuration register (ADPC) Port mode control registers 0, 14 (PMC0, PMC14) Port mode registers 0, 2, 14 (PM0, PM2, PM14)

14.3 Registers Controlling Comparator

The comparator uses the following 14 registers.

- Peripheral enable register 2 (PER2)
- Comparator n control register (CnCTL)
- Comparator and PGA internal reference voltage control register (CVRCTL)
- Comparator internal reference voltage select register m (CmRVM)
- Comparator rising edge enable register 0(CMPEGP0)
- Comparator falling edge enable register0 (CMPEGN0)
- Comparator output monitor register (CMPMON)
- Window comparator function setting register (CMPWDC)
- Comparator input switch control register (CMPSEL) (20-pin products only)
- External interrupt control register (INTPCTL)
- A/D port configuration register (ADPC)
- Peripheral function switch register 0 (PFSEL0)
- Port mode control registers 0, 14 (PMC0, PMC14)
- Port mode registers 0, 2, 14 (PM0, PM2, PM14)

Remark n = 0 to 5, m = 0 to 2

14.3.1 Peripheral enable register 2 (PER2)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the comparator is used, be sure to set bit 7 (PGACMPEN) of this register to 1.

The PER2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14-2. Format of Peripheral Enable Register 2 (PER2)

Address: F0509H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER2	PGACMPEN	TKBPA2EN	TKBPA1EN	TKBPA0EN	TKC0EN	TKB2EN[Note]	TKB1EN	TKB0EN

PGACMPEN	Comparator/programmable gain amplifier input clock control
0	Stops supply of input clock. • SFR used by the comparator or programmable gain amplifier cannot be written. • The comparator or programmable gain amplifier is in the reset status.
1	Supplies input clock. • SFR used by the comparator or programmable gain amplifier can be read/written.

Note 30-pin products and 38-pin products only.

14.3.2 Comparator n control register (CnCTL)

This register is used to control the operation of comparator n, enable or disable comparator output, reverse the output, and set the noise elimination width and reference voltage.

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

Reset signal generation clears this register to 00H.

Remark n = 0 to 5

Figure 14-3. Format of Comparator n Control Register (CnCTL) (1/2)

Address: F0552H (C0CTL), F0553H (C1CTL), F0554H (C2CTL) After reset: 00H R/W
F0555H (C3CTL), F0556H (C4CTL), F0557H (C5CTL)

Symbol	<7>	6	5	4	3	2	<1>	0
CnCTL	CMPnEN	CnDFS1	CnDFS0	CnMODSEL1	CnMODSEL0	0	CnOE	CnINV

CMPnEN	Comparator n operation control
0	Stops operation. Comparator n output signal is low level.
1	Enables operation. Enables input to the comparator n.

Figure 14-3. Format of Comparator n Control Register (CnCTL) (2/2)

Address: F0552H (C0CTL), F0553H (C1CTL), F0554H (C2CTL), F0555H (C3CTL), F0556H (C4CTL), F0557H (C5CTL) After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	<1>	0
CnCTL	CMPnEN	CnDFS1	CnDFS0	CnMODSEL1	CnMODSEL0	0	CnOE	CnINV

CnDFS1	CnDFS0	Noise elimination width setting
0	0	Noise filter unused
0	1	$2^3/f_{CLK}$, $2^3/f_{PLL}$ Note (When f_{PLL} = 64 MHz, 125 to 187.5 ns)
1	0	$2^4/f_{CLK}$, $2^4/f_{PLL}$ Note (When f_{PLL} = 64 MHz, 250 to 375 ns)
1	1	$2^5/f_{CLK}$, $2^5/f_{PLL}$ Note (When f_{PLL} = 64 MHz, 500 to 750 ns)

CnMODSEL1	CnMODSEL0	Reference voltage selection
0	0	Internal reference voltage: DA0
0	1	Internal reference voltage: DA1
1	0	Internal reference voltage: DA2
1	1	External reference voltage: CMPCOM

CnOE	Enabling or disabling of comparator output
0	Disables comparator output (disables output of timer forced output stop request signal (output signal = fixed to low level))
1	Enables comparator output (enables output of timer forced output stop request signal)

CnINV	Output reversal setting
0	Forward
1	Reverse

Note When PLLON = 1 in the PLL control register (PLLCTL), f_{PLL} is supplied.

Cautions
1. **Rewrite the CnDFS1, CnDFS0, CnMODSEL1, CnMODSEL0, and CnINV bits after setting the comparator n output to the disabled state (CnOE = 0). Rewrite the CnMODSEL1 and CnMODSEL0 bits during stop operation of the comparator (CMPnEN = 0).**
2. **With the noise elimination width, an extra CPU/peripheral hardware clock frequency (f_{CLK}) or PLL output clock (f_{PLL}) may be eliminated from the setting value.**
3. **If the comparator output noise interval is within "set noise elimination width + 1 clock", an illegal waveform may be output.**
4. **To use the internal reference voltage, enable (CVREn = 1) operation of the internal reference voltage before enabling (CMPnEN = 1) the comparator operation.**
5. **An operation stabilization wait time (1 μs at 3.3 to 5.5 V operation or 3 μs at 2.7 to 3.3 V operation) is required after comparator operation is enabled (CMPnEN = 1).**

Remarks
1. f_{CLK}: CPU/peripheral hardware clock frequency
2. n = 0 to 5

14.3.3 Comparator and PGA internal reference voltage control register (CVRCTL)

This register is used to control the internal reference voltage of the comparator, and to select the sources of the internal reference voltage and the comparator/GND of programmable gain amplifier.

The internal reference voltage is enabled or stopped by using the CVREm bit.

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

Reset signal generation clears this register to 00H.

Figure 14-4. Format of Comparator Internal Reference Voltage Control Register (CVRCTL)

Address: F0560 H After reset: 00H R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
CVRCTL	0	0	CVRVS1	CVRVS0	0	CVRE2	CVRE1	CVRE0

CVRVS1	GND selection of internal reference voltage and programmable gain amplifier
0	V_{SS}
1	AV_{REFM}

CVRVS0	Power supply selection of internal reference voltage
0	V_{DD}
1	AV_{REFP}

CVREm	Internal reference voltage (DAm) generation operation control
0	Stops operation.
1	Enables operation.

Cautions 1. The stabilization wait time (10 μs) is required after setting the CVREm bit.

2. Rewrite the CVRVS1 and CVRVS0 bits during stop operation of the internal reference voltage (CVREm = 0). Rewrite the CVRVS1 bit while the programmable gain amplifier operation is stopped (PGAEN = 0 in the PGACTL register).

Remark m = 0 to 2

14.3.4 Comparator internal reference voltage select register m (CmRVM)

This register is used to set the internal reference voltage level of comparator.

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

Reset signal generation clears this register to 00H.

Figure 14-5. Format of Comparator Internal Reference Voltage Select Register m (CmRVM)

Address: 0561H (C0RVM), F0562H (C1RVM), F0563H (C2RVM) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CmRVM	CmVRS7	CmVRS6	CmVRS5	CmVRS4	CmVRS3	CmVRS2	CmVRS1	CmVRS0

CmVRS7 to CmVRS0	Internal reference voltage (DAm) level setting
00000000	((AV$_{REFP}$ or V$_{DD}$)/256) × 0
00000001	((AV$_{REFP}$ or V$_{DD}$)/256) × 1
00000010	((AV$_{REFP}$ or V$_{DD}$)/256) × 2
.	.
.	.
.	.
11111101	((AV$_{REFP}$ or V$_{DD}$)/256) × 253
11111110	((AV$_{REFP}$ or V$_{DD}$)/256) × 254
11111111	((AV$_{REFP}$ or V$_{DD}$)/256) × 255

Caution Rewrite the CmRVM register while the internal reference voltage operation is stopped (CVREm = 0).

Remark m = 0 to 2

14.3.5 Comparator rising edge enable register 0 (CMPEGP0), comparator falling edge enable register 0 (CMPEGN0)

These registers are used to set the valid edges of comparator n detection interrupt signal (INTCMPn) and external interrupts (INTP20, INTP21).

The CMPEGP0 and CMPEGN0 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14-6. Format of Comparator Rising Edge Enable Register 0 (CMPEGP0) and Comparator Falling Edge Enable Register 0 (CMPEGN0)

Address: F0558 H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMPEGP0	CEGP7	CEGP6	CEGP5	CEGP4	CEGP3	CEGP2	CEGP1	CEGP0

Address: F0559 H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMPEGN0	CEGN7	CEGN6	CEGN5	CEGN4	CEGN3	CEGN2	CEGN1	CEGN0

CEGP7	CEGN7	INTP21 pin valid edge selection
0	0	Edge detection disabled (disables output of timer restart signal (output signal = fixed to low level))
0	1	Falling edge (enables output of timer restart signal)
1	0	Rising edge (enables output of timer restart signal)
1	1	Both rising and falling edges (enables output of timer restart signal)

CEGP6	CEGN6	INTP20 pin valid edge selection
0	0	Edge detection disabled (disables output of timer restart signal (output signal = fixed to low level))
0	1	Falling edge (enables output of timer restart signal)
1	0	Rising edge (enables output of timer restart signal)
1	1	Both rising and falling edges (enables output of timer restart signal)

CEGPn	CEGNn	INTCMPn valid edge selection
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Caution The valid edge setting is set for the signal that was non-reverse- or reverse-rotated from the comparator detect signal using the CnINV bit of the CnCTL register.

Remark n = 0 to 5

14.3.6 Comparator output monitor register (CMPMON)

This register indicates the level of the timer forced output stop request signal, which is a comparator output signal.

The CMPMON register can only be read by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14-7. Format of Comparator Output Monitor Register (CMPMON)

Address: F055AH After reset: 00H R

Symbol	7	6	<5>	<4>	<4>	<2>	<1>	<0>
CMPMON	0	0	CMPMON5	CMPMON4	CMPMON3	CMPMON2	CMPMON1	CMPMON0

CMPMONn	Comparator n output level (n = 0 to 5)
0	Timer forced output stop request signal is low level
1	Timer forced output stop request signal is high level

14.3.7 Window comparator function setting register (CMPWDC)

The window comparator function selects a single input voltage for two comparators so that the input voltage can be compared to two reference voltages. The CMPWDC register controls the selection of input signals when using this window comparator function.

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

Reset signal generation clears this register to 00H.

Figure 14-8. Format of Window Comparator Function Setting Register (CMPWDC)

Address: F0564H After reset: 00H R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
CMPWDC	0	0	0	0	0	WCSEL2	WCSEL1	WCSEL0

WCSEL2	+ side input selection for comparator 5
0	CMP5P
1	CMP4P
When WCSEL2 = 1 is set: input voltage from CMP4 is compared to reference voltages by comparator 4 and comparator 5	

WCSEL1	+ side input selection for comparator 3
0	CMP3P
1	CMP2P
When WCSEL1 = 1 is set: input voltage from CMP2 is compared to reference voltages by comparator 2 and comparator 3	

WCSEL0	+ side input selection for comparator 1
0	CMP1P
1	CMP0P
When WCSEL0 = 1 is set: input voltage from CMP0 is compared to reference voltages by comparator 1 and comparator 0	

Caution Rewrite the CMPWDC register during stop operation of the comparator (CMPnEN = 0).

14.3.8 Comparator input switch control register (CMPSEL) (20-pin products only)

This register is used to set the input signal for comparator 3.

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

Reset signal generation clears this register to 00H.

Figure 14-9. Format of Comparator Input Switch Control Register (CMPSEL)

Address: F0565H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMPSEL	0	0	0	0	0	0	0	CMPSEL0

CMPSEL0	+ side input control for comparator 3
0	Comparator 3 is not used in 20-pin products.
1	(CMP3P)/CMPCOM/ANI18/P147 pin

Caution Rewrite the CMPSEL register during stop operation of the comparator (CMP3EN = 0).

14.3.9 External interrupt control register (INTPCTL)

This register is used to set inverted output of external interrupts INTP20 and INTP21 for the timer output forced stop.

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

Reset signal generation clears this register to 00H.

Figure 14-10. Format of External Interrupt Control Register (INTPCTL)

Address: F055BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
INTPCTL	0	0	0	0	0	0	INTPINV1	INTPINV0

INTPINV1	Inverted output setting of INTP21 signal
0	Non-inverted output of INTP21 signal
1	Inverted output of INTP21 signal

INTPINV0	Inverted output setting of INTP20 signal
0	Non-inverted o output of INTP20 signal
1	Inverted output of INTP20 signal

Caution Rewrite the INTPCTL register when edge detection is prohibited for INTP20 and INTP21 (CEGPn and CEGNn = 0, 0) (n = 6, 7).

14.3.10 A/D port configuration register (ADPC)

This register switches the ANI0/P20, ANI1/P21, ANI2/CMP0P/P22, and ANI4/CMP1P/P24 to ANI7/CMP4P/P27 pins to digital I/O of port or analog input of A/D converter, programmable gain amplifier, or comparator.

When using the programmable gain amplifier or comparator, use the ADPC register to select the CMP0P/ANI2/P22 and CMP1P/ANI4/P24 to CMP4P/ANI7/P27 pins as analog inputs.

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

Reset signal generation sets this register to 00H.

Figure 14-11. Format of A/D Port Configuration Register (ADPC)

Address: F0076H After reset: 00H R/W

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

ADPC3	ADPC2	ADPC1	ADPC0	Analog input (A)/digital I/O (D) switching							
				ANI7/ CMP4P/P27	ANI6/ CMP3P/P26	ANI5/ CMP2P/P25	ANI4/ CMP1P/P24	PGAOUT[Note]	ANI2/ CMP0P/P22	ANI1/P21/ AV_{REFM}	ANI0/P20/ AV_{REFP}
0	0	0	0	A	A	A	A	A	A	A	A
0	0	0	1	D	D	D	D	D	D	D	D
0	0	1	0	D	D	D	D	D	D	D	A
0	0	1	1	D	D	D	D	D	D	A	A
0	1	0	0	D	D	D	D	D	A	A	A
0	1	0	1	D	D	D	D	A	A	A	A
0	1	1	0	D	D	D	A	A	A	A	A
0	1	1	1	D	D	A	A	A	A	A	A
1	0	0	0	D	A	A	A	A	A	A	A
1	0	0	1	A	A	A	A	A	A	A	A
1	1	1	1	A	A	A	A	A	A	A	A
Other than above				Setting prohibited							

Note This is an internal output pins for the programmable gain amplifier. When output signals from the programmable gain amplifier are used as an analog input channel for the A/D converter, set ADPC as 0000B or 0101B or above.

Caution Set the channel used for A/D conversion to the input mode by using port mode register 2 (PM2).

14.3.11 Peripheral function switch register 0 (PFSEL0)

Bits 0 and 1 of the PFSEL0 register use external interrupts INTP20 and INTP21 either for PWM control of 16-bit timers KB0, KB1, and KB2 or for clearing STOP mode.

Bit 4 selects whether or not to use the INTP20 noise filter. When INTP20 is used by forced output stop function 2, select to not use the noise filter in order to speed up the reaction time between trigger input and stopping output.

Bits 5 and 6 are used to select the detection interrupt functions for CMP0 and CMP2. For details about the other bits, see **7.3.16 Peripheral function switch register 0 (PFSEL0)**.

PFSEL0 can be written by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14-12. Format of Peripheral Function Switch Register 0 (PFSEL0)

Address: F05C6H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	3	2	<1>	<0>
PFSEL0	0	CMP2STEN	CMP0STEN	PNFEN	ADTRG11	ADTRG10	TMRSTEN1	TMRSTEN0

CMP2STEN	Comparator 2 detection interrupt (INTCMP2) input signal switching[Note 1]
0	Signal via digital edge detect circuit is selected. STOP mode release is disabled.
1	Forced output stop request signal is selected. STOP mode release is enabled, but only when not using noise filter. (Can be set when operating in low-power RTC mode (RTCLPC = 1 in the OSMC register))

CMP0STEN	Comparator 0 detection interrupt (INTCMP0) input signal switching[Note 1]
0	Signal via digital edge detect circuit is selected. STOP mode release is disabled.
1	Forced output stop request signal is selected. STOP mode release is enabled, but only when not using noise filter. (Can be set when operating in low-power RTC mode (RTCLPC = 1 in the OSMC register))

PNFEN	Use/Do not use external interrupt INTP20 noise filter
0	Use noise filter
1	Do not use noise filter

TMRSTEN1	External interrupt INTP21 function switching[Note 2]
0	External interrupt function is selected. (STOP mode release is enabled, but cannot be used for timer restart function)
1	Timer restart function is selected. (STOP mode release is disabled, but can be used for timer restart function)

TMRSTEN0	External interrupt INTP20 function switching[Note 2]
0	External interrupt function is selected. (STOP mode release is enabled, but cannot be used for timer restart function)
1	Timer restart function/forced output stop function 2 is selected. (STOP mode release is disabled, but can be used for timer restart function)

(The **Notes** and **Remark** are listed on the next page.)

Notes 1. When the interrupt for CMP0 and CMP2 is used, adopt a function used with the interrupt input signal.
When the CMP0 and CMP2 are used as a trigger of the timer KB forced output stop function, set CMPnSTEN = 1.
When the CMP2 is used as a trigger of the timer restart function for timer KB, set CMP2STEN = 0.
For details, see **14.5 Caution for Using Timer KB Simultaneous Operation Function**.

2. When INTP20 and INTP21 are used as a trigger of the timer KB forced output stop function 2 or timer restart function, see **14.5 Caution for Using Timer KB Simultaneous Operation Function**.

Remark n = 0, 2

14.3.12 Port mode control registers 0, 14 (PMC0, PMC14)

These registers are used to set the digital I/O/analog input of port 0 or 14 in 1-bit units.

To use the CMP5P/ANI16/RxD1/P03 or CMPCOM/ANI18/P147 pin as an analog input pin, set the PMC03 or PMC147 bit to 1.

The PMC0 and PMC14 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 14-13. Format of Port Mode Control Registers 0, 14 (PMC0, PMC14)

Address: F0060H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PMC0	1	1	1	1	PMC03	PMC02	1	1

Address: F006EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PMC14	PMC147	1	1	1	1	1	1	1

PMCmn	Pmn pin digital I/O/analog input selection (m = 02, 03, 147)
0	Digital I/O (alternate function other than analog input)
1	Analog input

14.3.13 Port mode registers 0, 2, 14 (PM0, PM2, PM14)

When using the CMP0P/ANI2/P22, CMP1P/ANI4/P24 to CMP4P/ANI7/P27, CMP5P/ANI16/RxD1/P03, or CMPCOM/ANI18/P147 pin for an analog input port, set the PM22, PM24 to PM27, PM03, or PM147 bit to 1. The output latches of P22, P24 to P27, P03, and P147 at this time may be 0 or 1.

If the PM22, PM24 to PM27, PM03, and PM147 bits are set to 0, they cannot be used as analog input port pins.

The PM0, PM2, and PM14 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Caution When a pin is set as an analog input port, not the pin level but "0" is always read despite PMxx = 1.

Figure 14-14. Format of Port Mode Registers 0, 2, 14 (PM0, PM2, PM14)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	PM06	PM05	1	PM03	PM02	1	1

Address: FFF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	1	PM22	PM21	PM20

Address: FFF2EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM14	PM147	1	1	1	1	1	1	1

PMmn	Pmn pin I/O mode selection (m = 0, 2, 14; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bits 0, 1, 4, and 7 of the PM0 register, bit 3 of the PM2 register, and bits 0 to 6 of the PM14 register to "1".

For 30- and 20-pin products, the following bits must be set to output mode (specified by setting the port registers and port mode registers to 0) by software after reset is cancelled.

30-pin products: Bits 5 and 6 of the PM0 register

20-pin products: Bits 2, 3, 5, and 6 of the PM0 register, and bits 6 and 7 of the PM2 register

The functions of the CMP0P/ANI2/P22 and CMP1P/ANI4/P24 to CMP4P/ANI7/P27 pins can be selected by using the A/D port configuration register (ADPC) and PM2 register.

Table 14-2. Setting Functions of CMP0P/ANI2/P22 and CMP1P/ANI4/P24 to CMP4P/ANI7/P27 Pins

ADPC	PM2	CMP0P/ANI2/P22 and CMP1P/ANI4/P24 to CMP4P/ANI7/P27 Pins
Digital I/O selection	Input mode	Digital input
	Output mode	Digital output
Analog input selection	Input mode	Analog input
	Output mode	Setting prohibited

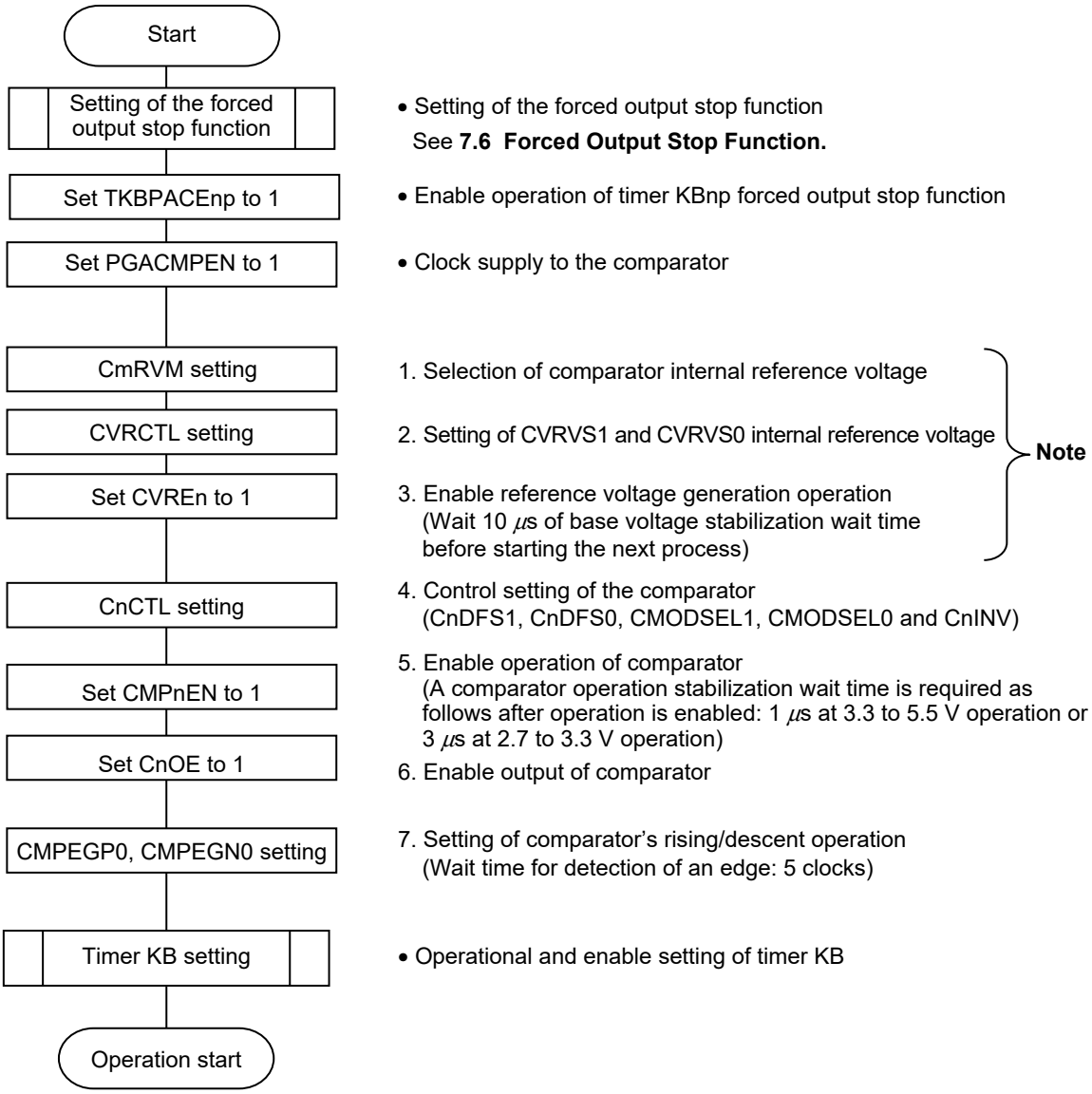
The functions of the CMP5P/ANI16/RxD1/P03 and CMPCOM/ANI18/P147 pins can be selected by using the port mode control registers 0, 14 (PMC0, PMC14), PM0, and PM14 registers.

Table 14-3. Setting Functions of CMP5P/ANI16/RxD1/P03 and CMPCOM/ANI18/P147 Pins

PMC0, PMC14	PM0, PM14	CMP5P/ANI16/RxD1/P03 and CMPCOM/ANI18/P147 Pins
Digital I/O selection	Input mode	Digital input
	Output mode	Digital output
Analog input selection	Input mode	Analog input
	Output mode	Setting prohibited

14.4 Setting Procedure of Comparator

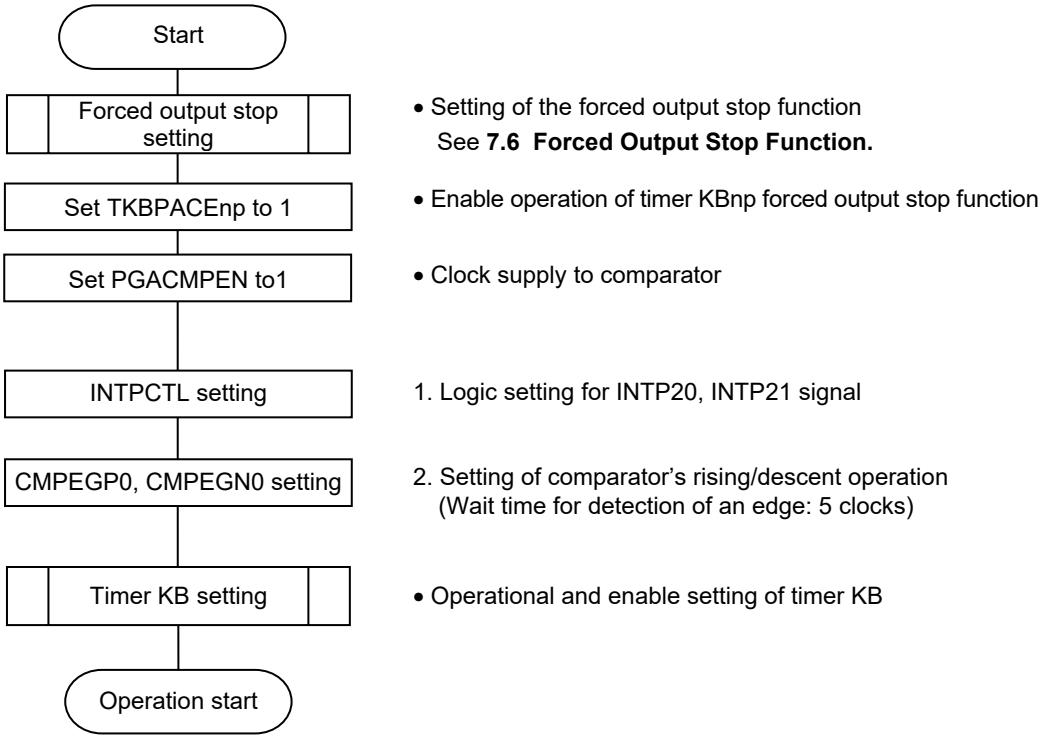
Figure 14-15. Operation Setting Flow Chart 1 of Comparator (CMP)
(Using Timer Forced Output Stop Request Signal by INTCMPn, CMPn)



Note It is not required, when the external pin CMPCOM is used for base voltage.

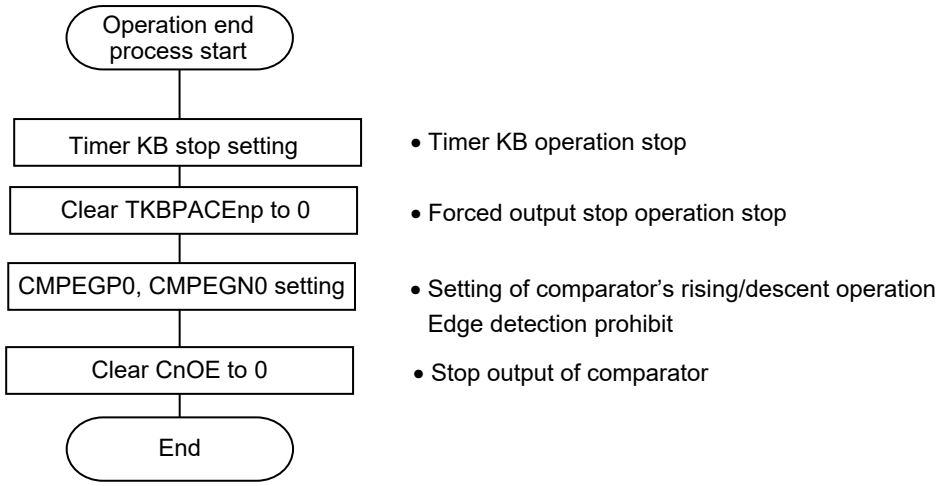
Caution Above 1. to 7. should be set under INTCMP process prohibition state.

Figure 14-16. Operation Setting Flow Chart 2 of Comparator (CMP)
<R> **(Using Timer Forced Output Stop Request Signal by INTP2m (Using Edge Circuit only))**



Caution Above 1. to 2. should be set under INTCMP process prohibition state.

Figure 14-17. Operation End Flow Chart of Comparator



14.5 Caution for Using Timer KB Simultaneous Operation Function

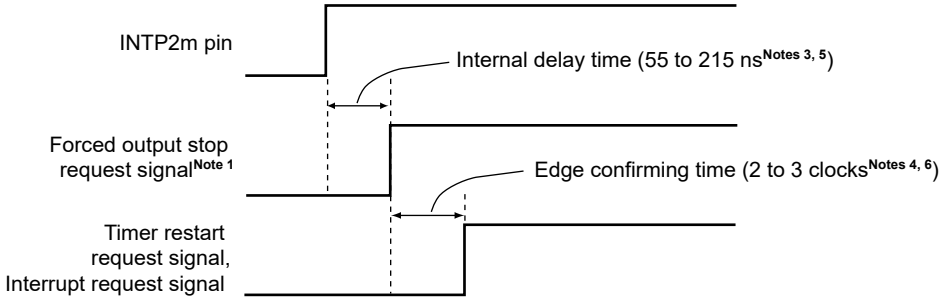
In addition to their use as an external interrupt input, the INTP2m pin output and the comparator output signal can be used as a trigger for functions that operate simultaneously with timer KB, such as the forced output stop function and timer restart function. The settings in peripheral function switch register 0 (PFSEL0) and the edge selection registers must be specified according to the function used. The width of the active signal required until each function starts operating differs.

When using INTP2m or the comparator output signal, see Tables 14-4 to 14-6 to specify the necessary register settings, and configure external circuits so that the required active signal width is assured.

Table 14-4. Relationship of INTP2m Function, Register Settings, and Active Signal Width

Function	Peripheral Enable Register Setting	Edge Setting Registers	Necessary Active Signal Width to Operate Each Function		
			Interrupt	Forced Output Stop	Timer Restart
External interrupt (STOP release is enabled)	TMRSTENm = 0	EGPn, EGNn	Up to 1 μs	–	–
Forced output stop[Note 1]	TMRSTENm = 1	CEGPp, CEGNp [Note 2]	55 to 215 ns[Note 3] + 2 to 3 clocks[Note 4]	55 to 215 ns[Notes 3, 5]	–
Timer restart	TMRSTENm = 1	CEGPp, CEGNp	55 to 215 ns[Note 3] + 2 to 3 clocks[Note 4]	–	55 to 215 ns[Note 3] + 2 to 3 clocks[Notes 4, 6]

Figure 14-18. Generation Timing of Forced Output Stop Signal and Timer Restart Request Signal by INTP2m



Notes 1. Only INTP20 can be used as a trigger for forced output stop function 2.

2. The active level of the forced output stop function 2 is high. Edge selection is only applied to detection of an interrupt signal.

3. 5 to 15 ns when noise filtering on INTP20 is disabled (PNFEN = 1)

4. For f_{CLK} or f_{PLL} (when PLLON = 1)

5. An additional output delay time (10 to 40 ns) is required from when forced output stop function 2 starts operating to when the level of the timer KB output changes.

6. Until the timer restart function starts operating, an additional clock cycle is required after the timer restart request signal is received, and an additional output delay time (10 to 40 ns) is required until the level of the timer KB output changes.

Remark m = 0, 1 n = 20, 21 p = 7, 6

Table 14-5. Relationship of Comparator 0 and 2 Functions, Register Settings, and Active Signal Width

Function	Peripheral Enable Register Setting	Edge Setting Registers	Necessary Active Signal width to Operate Each Function		
			Interrupt	Forced Output Stop	Timer Restart
External interrupt (STOP release is enabled[Note 1])	CMPnSTEN = 1	Rising edge only [Note 2]	Up to 150 ns[Note 3]	–	–
External interrupt (STOP release is disabled)	CMPnSTEN = 0	CEGPn, CEGNn	Up to 150 ns[Note 3] + 2 to 3 clocks[Notes 4, 5]	–	–
Forced output stop	CMPnSTEN = 1	**Note 6**	Up to 150 ns[Note 3]	Up to 150 ns[Notes 3, 7]	–
Timer restart	CMPnSTEN = 0	CEGPn, CEGNn	Up to 150 ns[Note 3] + 2 to 3 clocks[Notes 4, 5]	–	Up to 150 ns[Note 3] + 2 to 3 clocks[Notes 4, 5]

Figure 14-19. Generation Timing of Forced Output Stop Request Signal by Comparator 0 and 2 (CMPnSTEN = 1)

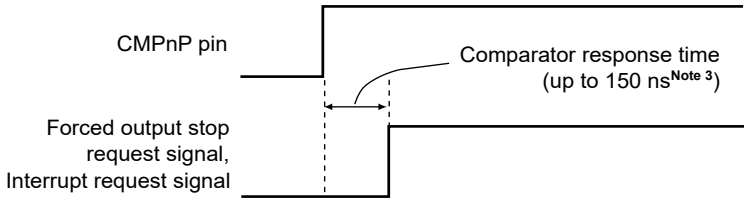
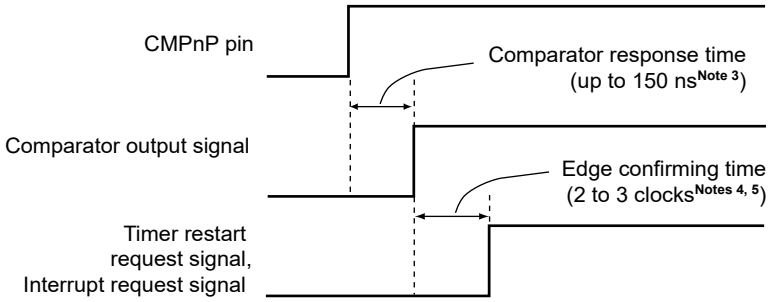


Figure 14-20. Generation Timing of Timer Restart Request Signal by Comparator 0 and 2 (CMPnSTEN = 0)



Notes 1. When noise filtering is set to “0, 0” by using the CnDFS1 and CnDFS0 bits in the comparator n control register (CnCTL)

2. To change the level of the edge direction, invert the comparator output signal by using the CnINV bit in the comparator n control register (CnCTL).

3. This is the time required when noise filtering is set to “0, 0” by using the CnDFS1 and CnDFS0 bits in the comparator n control register (CnCTL).
If a setting other than “0, 0” is specified, the specified noise elimination width is added.

4. For f_{CLK} or f_{PLL} (when PLLON = 1)

5. Until the timer restart function starts operating, an additional clock cycle is required after the timer restart request signal is received, and an additional output delay time (10 to 40 ns) is required until the level of the timer KB output changes.

6. The active level of the forced output stop function is high.

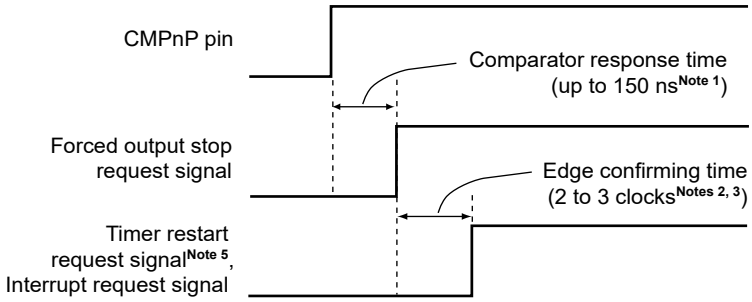
7. An additional output delay time (10 to 40 ns) is required from when forced output stop function starts operating to when the level of the timer KB output changes.

Remark n = 0, 2

Table 14-6. Relationship of Comparator 1, 3, 4, and 5 Functions, Register Settings, and Active Signal Width

Function	Peripheral Enable Register Setting	Edge Setting Registers	Necessary Active Signal Width to Operate Each Function		
			Interrupt	Forced Output Stop	Timer Restart
External interrupt (STOP release is disabled)	–	CEGPn, CEGNn	Up to 150 ns[Note 1] + 2 to 3 clocks[Notes 2, 3]	–	–
Forced output stop	–	**Note 4**	Up to 150 ns[Note 2] + 2 to 3 clocks[Notes 3, 4]	Up to 150 ns[Notes 2, 5]	–
Timer restart[Note 6]	–	CEGPn, CEGNn	Up to 150 ns[Note 2] + 2 to 3 clocks[Notes 3, 4]	–	Up to 150 ns[Note 2] + 2 to 3 clocks[Notes 3, 4]

Figure 14-21. Generation Timing of Forced Output Stop Request Signal and Timer Restart Request Signal by Comparator 1, 3, 4, and 5



Notes 1. When noise filtering is set to “0, 0” by using the CnDFS1 and CnDFS0 bits in the comparator n control register (CnCTL). If a setting other than “0, 0” is specified, the specified noise elimination width is added.

2. For f_{CLK} or f_{PLL} (when PLLON = 1)

3. Until the timer restart function starts operating, an additional clock cycle is required after the timer restart request signal is received, and an additional output delay time (10 to 40 ns) is required until the level of the timer KB output changes.

4. The active level of the forced output stop function is high.

5. An additional output delay time (10 to 40 ns) is required from when forced output stop function starts operating to when the level of the timer KB output changes.

6. The timer restart function can be used for comparator 1 and 3 only.

Remark n = 1, 3 to 5

CHAPTER 15 SERIAL ARRAY UNIT 0

Serial array unit 0 has up to four serial channels. Each channel can achieve Simplified SPI (CSI [Note]) and UART communication.

Function assignment of each channel supported by the RL78/I1A is as shown below.

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

- 20-pin products

Channel	Used as Simplified SPI (CSI)	Used as UART
0	–	UART0 (supporting LIN-bus, DMX512)
1	–	
2	–	–
3	–	

- 30-pin products

Channel	Used as Simplified SPI (CSI)	Used as UART
0	–	UART0 (supporting LIN-bus, DMX512)
1	–	
2	–	UART1
3	–	

- 38-pin products

Channel	Used as Simplified SPI (CSI)	Used as UART
0	CSI00	UART0 (supporting LIN-bus, DMX512)
1	–	
2	–	UART1
3	–	

In 38-pin products, when “UART0” is used for channel 0 and 1, CSI00 cannot be used, but “UART1” can be used for channel 2 and 3.

Caution Most of the following descriptions in this chapter use the units and channels of the 38-pin products as an example.

15.1 Functions of Serial Array Unit 0

Each serial interface supported by the RL78/I1A has the following features.

15.1.1 Simplified SPI (CSI00)

Data is transmitted or received in synchronization with the serial clock (SCK) output from the master channel. Simplified SPI communication is clocked communication performed by using three communication 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 **15.5 Operation of Simplified SPI (CSI00) Communication.**

[Data transmission/reception]

- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable

[Clock control]

- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel
- Maximum transfer rate[Note]

  During master communication: Max. $f_{CLK}/2$

  During slave communication: Max. $f_{MCK}/6$

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

In addition, CSI00 (channel 0) supports the SNOOZE mode. When SCK00 pin input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible.

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T_A = –40 to +105°C)** or **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T_A = –40 to +125°C)**.

15.1.2 UART (UART0, UART1)

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel). The LIN-bus and DMX512 can be implemented by using timer array unit with an external interrupt (INTP0).

For details about the settings, see **15.6 Operation of UART (UART0, UART1) Communication**.

[Data transmission/reception]

- Data length of 7, 8, or 9 bits[Note]
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

In addition, UART0 reception (channel 1) supports the SNOOZE mode. When RxD0 pin input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible.

The LIN-bus and DMX512 are accepted in UART0 (channel 0, 1).

[LIN-bus functions]

- Wakeup signal detection
- Break field (BF) detection
- Sync field measurement, baud rate calculation

} Using the external interrupt (INTP0) and timer array unit0

[DMX512 functions]

- BREAK signal detection
- Pulse width detection

} Using the external interrupt (INTP0) and timer array unit 0

Note Only following UARTs can be specified for the 9-bit data length.

15.2 Configuration of Serial Array Unit 0

The serial array unit 0 includes the following hardware.

Table 15-1. Configuration of Serial Array Unit 0

Item	Configuration
Shift register	8 bits or 9 bits[Note 1]
Buffer register	Lower 8 bits or 9 bits of serial data register mn (SDRmn)[Notes 1, 2]
Serial clock I/O	SCK00 pin (for Simplified SPI)
Serial data input	SI00 pin (for Simplified SPI), RxD0 pin (for UART supporting LIN-bus and DMX512), RxD1 pin (for UART)
Serial data output	SO00 pin (for Simplified SPI), TxD0 pin (for UART supporting LIN-bus and DMX512), TxD1 pin (for UART)
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Serial clock select register m (SPSm) • Serial channel enable status register m (SEm) • Serial channel start register m (SSm) • Serial channel stop register m (STm) • Serial output enable register m (SOEm) • Serial output register m (SOm) • Serial output level register m (SOLm) • Serial standby control register 0 (SSC0) • Input switch control register (ISC) • Noise filter enable register 0 (NFEN0)
	<Registers of each channel> • Serial data register mn (SDRmn) • Serial mode register mn (SMRmn) • Serial communication operation setting register mn (SCRmn) • Serial status register mn (SSRmn) • Serial flag clear trigger register mn (SIRmn)
	• Port input mode registers 0, 1 (PIM0, PIM1) • Port output mode registers 0, 1 (POM0, POM1) • Port mode registers 0, 1 (PM0, PM1) • Port registers 0, 1 (P0, P1)

Notes 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, mn = 02, 03: lower 8 bits

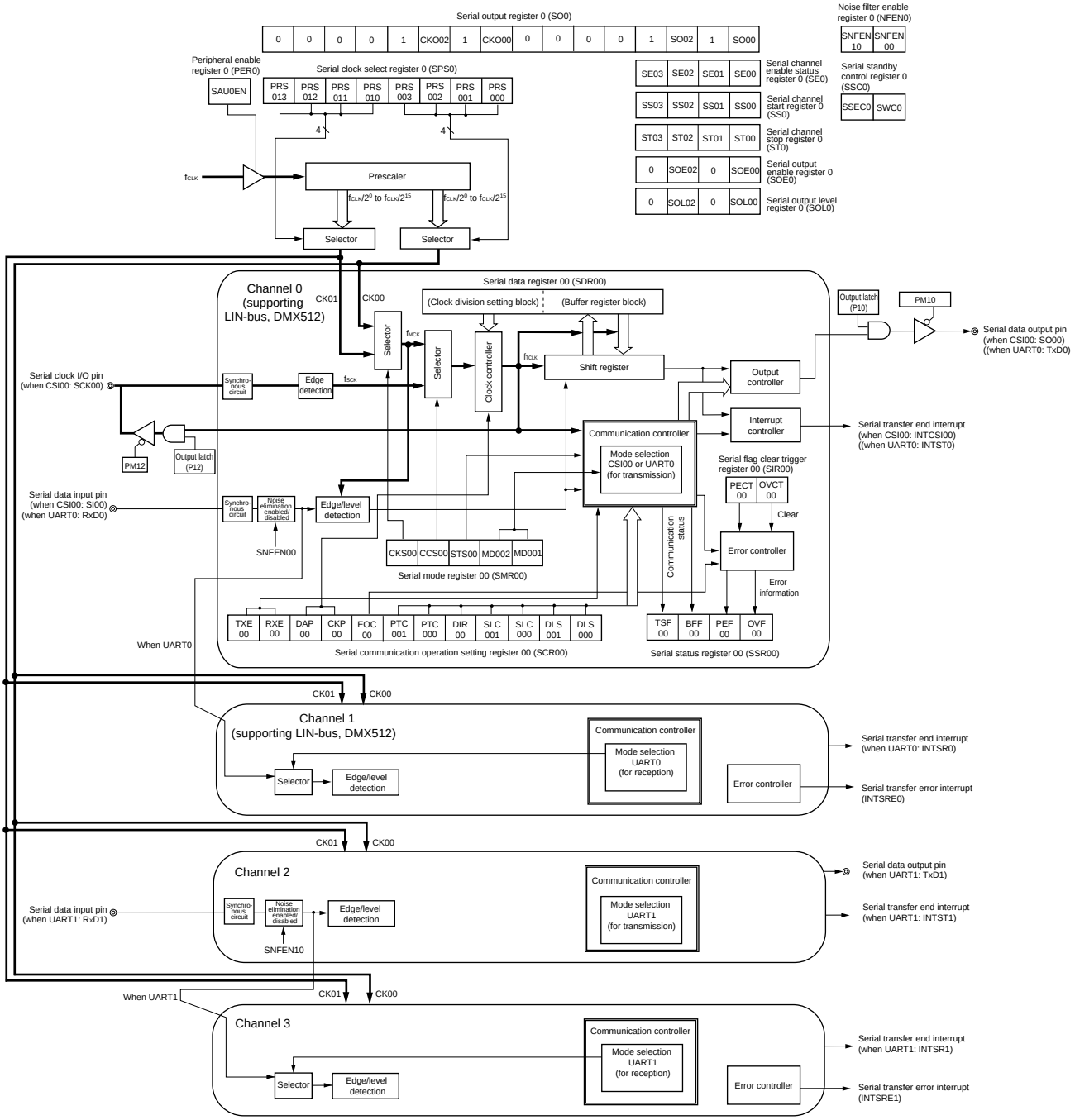
2. The lower 8 bits of serial data register mn (SDRmn) can be read or written as the following SFR, depending on the communication mode.

- CSIp communication … SIOp (CSIp data register)
- UARTq reception … RxDq (UARTq receive data register)
- UARTq transmission … TxDq (UARTq transmit data register)

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00),
q: UART number (q = 0, 1)

Figure 15-1 shows the block diagram of the serial array unit 0.

Figure 15-1. Block Diagram of Serial Array Unit 0



15.2.1 Shift register

This is an 9-bit register that converts parallel data into serial data or vice versa.

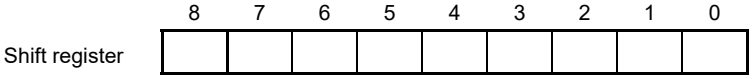
In case of the UART communication of nine bits of data, nine bits (bits 0 to 8) are used[Note 1].

During reception, it converts data input to the serial pin into parallel data.

When data is transmitted, the value set to this register is output as serial data from the serial output pin.

The shift register cannot be directly manipulated by program.

To read or write the shift register, use the lower 8/9 bits of serial data register mn (SDRmn).



15.2.2 Lower 8/9 bits of the serial data register mn (SDRmn)

The SDRmn register is the transmit/receive data register (16 bits) of channel n. Bits 8 to 0 (lower 9 bits)[Note 1] or bits 7 to 0 (lower 8 bits) function as a transmit/receive buffer register, and bits 15 to 9 are used as a register that sets the division ratio of the operation clock (f_{MCK}).

When data is received, parallel data converted by the shift register is stored in the lower 8/9 bits. When data is to be transmitted, set transmit to be transferred to the shift register to the lower 8/9 bits.

The data stored in the lower 8/9 bits of this register is as follows, depending on the setting of bits 0 and 1 (DLSmn0, DLSmn1) of serial communication operation setting register mn (SCRmn), regardless of the output sequence of the data.

- 7-bit data length (stored in bits 0 to 6 of SDRmn register)
- 8-bit data length (stored in bits 0 to 7 of SDRmn register)
- 9-bit data length (stored in bits 0 to 8 of SDRmn register)[Note 1]

The SDRmn register can be read or written in 16-bit units.

The lower 8/9 bits of the SDRmn register can be read or written[Note 2] in 8-bit units 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)

Reset signal generation clears the SDRmn register to 0000H.

Notes 1. Only following UARTs can be specified for the 9-bit data length.

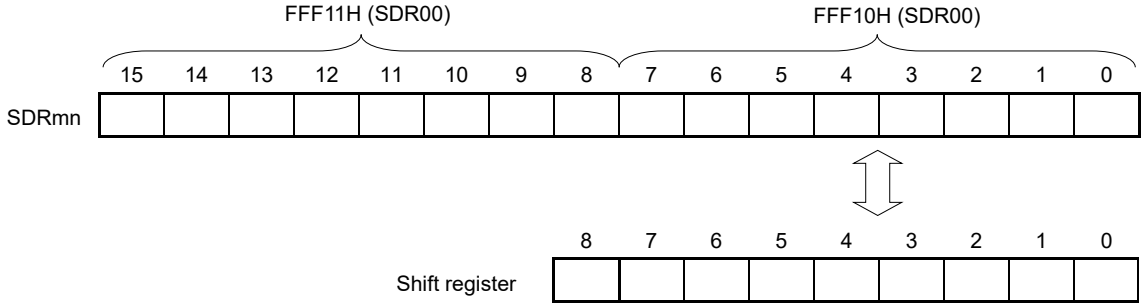
2. When operation is stopped (SEmn = 0), do not rewrite SDRmn[7:0] by an 8-bit memory manipulation instruction (SDRmn[15:9] are all cleared to 0).

Remarks 1. After data is received, “0” is stored in bits 0 to 8 in bit portions that exceed the data length.

2. m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00), q: UART number (q = 0, 1)

Figure 15-2. Format of Serial Data Register mn (SDRmn) (mn = 00, 01)

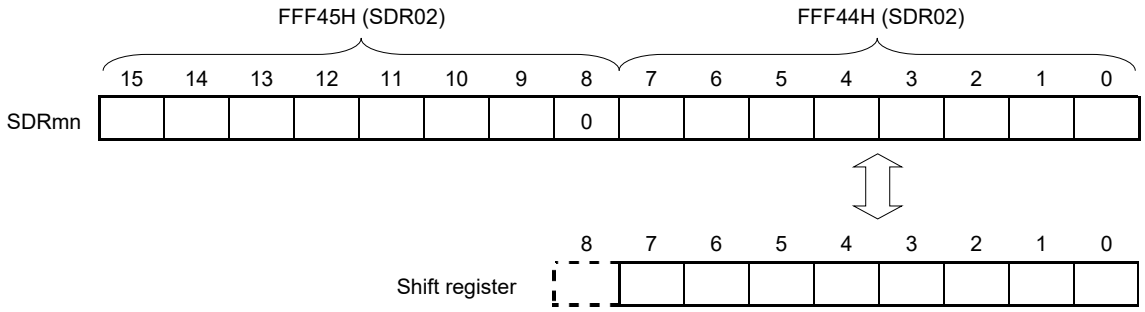
Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01) After reset: 0000H R/W



Remark For the function of the higher 7 bits of the SDRmn register, see **15.3 Registers Controlling Serial Array Unit 0**.

Figure 15-3. Format of Serial Data Register mn (SDRmn) (mn = 02, 03)

Address: FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03) After reset: 0000H 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 **15.3 Registers Controlling Serial Array Unit 0**.

15.3 Registers Controlling Serial Array Unit 0

Serial array unit 0 is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Serial clock select register m (SPSm)
- Serial mode register mn (SMRmn)
- Serial communication operation setting register mn (SCRmn)
- Serial data register mn (SDRmn)
- Serial flag clear trigger register mn (SIRmn)
- Serial status register mn (SSRmn)
- Serial channel start register m (SSm)
- Serial channel stop register m (STm)
- Serial channel enable status register m (SEm)
- Serial output enable register m (SOEm)
- Serial output level register m (SOLm)
- Serial output register m (SOm)
- Serial standby control register 0 (SSC0)
- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port input mode registers 0, 1 (PIM0, PIM1)
- Port output mode registers 0, 1 (POM0, POM1)
- Port mode registers 0, 1 (PM0, PM1)
- Port registers 0, 1 (P0, P1)

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

15.3.1 Peripheral enable register 0 (PER0)

PER0 is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When serial array unit 0 is used, be sure to set bit 2 (SAU0EN) of this register to 1.

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

Reset signal generation clears the PER0 register to 00H.

Figure 15-4. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCEN	0	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

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

Cautions 1. When setting serial array unit m, be sure to first set the following registers with the SAU0EN bit set to 1. If SAU0EN = 0, control registers of serial array unit 0 become default values and writing to them is ignored (except for the input switch control register (ISC), noise filter enable register 0 (NFEN0), port input mode registers 0, 1 (PIM0, PIM1), port output mode registers 0, 1 (POM0, POM1), port mode registers 0, 1 (PM0, PM1), and port registers 0, 1 (P0, P1)).

- **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 0 (SSC0)**

2. Be sure to clear bits 1, 3, and 6 to "0".

15.3.2 Serial clock select register m (SPSm)

The SPSm register is a 16-bit register that is used to select two types of operation clocks (CKm0, CKm1) that are commonly supplied to each channel. CKm1 is selected by bits 7 to 4 of the SPSm register, and CKm0 is selected by bits 3 to 0.

Rewriting the SPSm register is prohibited when the register is in operation (when SEmn = 1).

The SPSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SPSm register can be set with an 8-bit memory manipulation instruction with SPSmL.

Reset signal generation clears the SPSm register to 0000H.

Figure 15-5. Format of Serial Clock Select Register m (SPSm)

Address: F0126H, F0127H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPSm	0	0	0	0	0	0	0	0	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Section of operation clock (CKmk)[Note]					
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 32 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz	32 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	4 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz	2 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz	1 MHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	250 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	62.5 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	15.6 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz	3.91 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	153 kHz	305 Hz	610 Hz	977 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 0 (SAU0).

Caution Be sure to clear bits 15 to 8 to "0".

Remarks 1. f_{CLK}: CPU/peripheral hardware clock frequency
f_{SUB}: Subsystem clock frequency

2. m: Unit number (m = 0)

3. k = 0, 1

15.3.3 Serial mode register mn (SMRmn)

The SMRmn register is a register that sets an operation mode of channel n. It is also used to select an operation clock (f_{MCK}), specify whether the serial clock (f_{SCK}) may be input or not, set a start trigger, an operation mode (Simplified SPI (CSI) or UART), and an interrupt source. This register is also used to invert the level of the receive data only in the UART mode.

Rewriting the SMRmn register is prohibited when the register is in operation (when SEmn = 1). However, the MDmn0 bit can be rewritten during operation.

The SMRmn register can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets the SMRmn register to 0020H.

Figure 15-6. Format of Serial Mode Register mn (SMRmn) (1/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03) After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSmn Note	0	SISmn0 Note	1	0	0	0	MDmn1	MDmn0

CKSmn	Selection of operation clock (f_{MCK}) of channel n
0	Operation clock CKm0 set by the SPSm register
1	Operation clock CKm1 set by the SPSm register
Operation clock (f_{MCK}) is used by the edge detector. In addition, depending on the setting of the CCSmn bit and the higher 7 bits of the SDRmn register, a transfer clock (f_{TCLK}) is generated.	

CCSmn	Selection of transfer clock (f_{TCLK}) of channel n
0	Divided operation clock f_{MCK} specified by the CKSmn bit
1	Clock input f_{SCK} from the SCKp pin (slave transfer in Simplified SPI (CSI) mode)
Transfer clock f_{TCLK} is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCSmn = 0, the division ratio of operation clock (f_{MCK}) is set by the higher 7 bits of the SDRmn register.	

STSmn Note	Selection of start trigger source
0	Only software trigger is valid (selected for Simplified SPI (CSI) and UART transmission).
1	Valid edge of the RxDq pin (selected for UART reception)
Transfer is started when the above source is satisfied after 1 is set to the SSm register.	

Note The SMR01 and SMR03 registers only.

Caution Be sure to clear bits 13 to 9, 7, 4, and 3 (or bits 13 to 6, 4, and 3 for the SMR00 or SMR02 register) to "0". Be sure to set bit 5 to "1".

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00), q: UART number (q = 0, 1)

Figure 15-6. Format of Serial Mode Register mn (SMRmn) (2/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03) After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn Note	0	SIS mn0 Note	1	0	0	0	MD mn1	MD mn0

SIS mn0	Controls inversion of level of receive data of channel n in UART mode
0	Falling edge is detected as the start bit. The input communication data is captured as is.
1	Rising edge is detected as the start bit. The input communication data is inverted and captured.

MD mn1	Setting of operation mode of channel n
0	Simplified SPI (CSI) mode
1	UART mode

MD mn0	Selection of interrupt source of channel n
0	Transfer end interrupt
1	Buffer empty interrupt (Occurs when data is transferred from the SDRmn register to the shift register.)
For successive transmission, the next transmit data is written by setting the MDmn0 bit to 1 when SDRmn data has run out.	

Note The SMR01 and SMR03 registers only.

Caution Be sure to clear bits 13 to 9, 7, 4, and 3 (or bits 13 to 6, 4, and 3 for the SMR00 or SMR02 register) to "0". Be sure to set bit 5 to "1".

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00), q: UART number (q = 0, 1)

15.3.4 Serial communication operation setting register mn (SCRmn)

The SCRmn register is a communication operation setting register of channel n. It is used to set a data transmission/reception mode, phase of data and clock, whether an error signal is to be masked or not, parity bit, start bit, stop bit, and data length.

Rewriting the SCRmn register is prohibited when the register is in operation (when SEmn = 1).

The SCRmn register can be set by a 16-bit memory manipulation instruction.

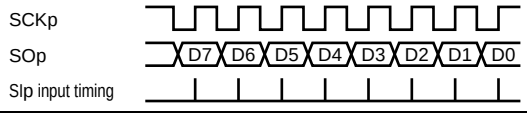
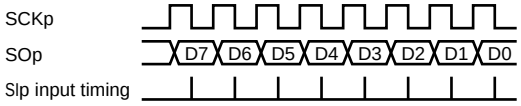
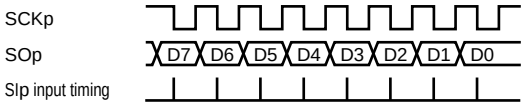
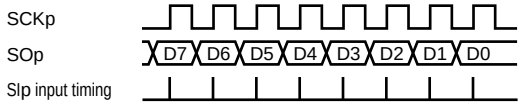
Reset signal generation sets the SCRmn register to 0087H.

Figure 15-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (1/2)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TxEmn	RxEmn	DAPmn	CKPmn	0	EOCmn	PTCmn1	PTCmn0	DIRmn	0	SLCmn1 Note 1	SLCmn0	0	1	DLSmn1 Note 2	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
Be sure to set DAPmn, CKPmn = 0, 0 in the UART mode.			

EOCmn	Mask control of error interrupt signal (INTSREx (x = 0 to 3))
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 and during UART transmission Note 3.	

Notes 1. The SCR00 and SCR02 registers only.

2. The SCR00 and SCR01 registers only.

3. When using CSIp not with EOCmn = 0, error interrupt INTSREn may be generated.

Caution Be sure to clear bits 3, 6, and 11 to "0". (Also clear bit 5 of the SCR01 or SCR03 register to 1.) Be sure to set bit 2 to "1". (Also set bit 1 of the SCR02 or SCR03 register.)

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00)

Figure 15-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (2/2)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TxEmn	RxEmn	DAPmn	CKPmn	0	EOCmn	PTCmn1	PTCmn0	DIRmn	0	SLCmn1[Note 1]	SLCmn0	0	1	DLSmn1[Note 2]	DLSmn0

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 3].	No parity judgment
1	0	Outputs even parity.	Judged as even parity.
1	1	Outputs odd parity.	Judges as odd parity.
Be sure to set PTCmn1, PTCmn0 = 0, 0 in the Simplified SPI (CSI) mode.			

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.

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 only)
1	1	Setting prohibited
When the transfer end interrupt is selected, the interrupt is generated when all stop bits have been completely transferred. Set 1 bit (SLCmn1, SLCmn0 = 0, 1) during UART reception. Set no stop bit (SLCmn1, SLCmn0 = 0, 0) in the Simplified SPI (CSI) mode. Set 1 bit (SLCmn1, SLCmn0 = 0, 1) or 2 bits (SLCmn1, SLCmn0 = 1, 0) during UART transmission.		

DLSmn1[Note 2]	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)
Other than above		Setting prohibited

Notes 1. The SCR00 and SCR02 registers only.

2. The SCR00 and SCR01 registers only.

3. 0 is always added regardless of the data contents.

Caution Be sure to clear bits 3, 6, and 11 to "0". (Also clear bit 5 of the SCR01 or SCR03 register to 1.) Be sure to set bit 2 to "1". (Also set bit 1 of the SCR02 or SCR03 register.)

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00)

15.3.5 Higher 7 bits of the serial data register mn (SDRmn)

The SDRmn register is the transmit/receive data register (16 bits) of channel n. Bits 8 to 0 (lower 9 bits) of SDR00 and SDR01 or bits 7 to 0 (lower 8 bits) of SDR02 and SDR03 function as a transmit/receive buffer register, and bits 15 to 9 (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 operating clock by the bits 15 to 9 (higher 7 bits) of the SDRmn register is used as the transfer clock.

If the CCSmn bit is set to 1, set bits 15 to 9 (higher 7 bits) of SDRmn to 0000000B. The input clock f_{SCK} (slave transfer in Simplified SPI (CSI) mode) from the SCKp pin is used as the transfer clock.

The lower 8/9 bits of the SDRmn register function as a transmit/receive buffer register. During reception, the parallel data converted by the shift register is stored in the lower 8/9 bits, and during transmission, the data to be transmitted to the shift register is set to the lower 8/9 bits.

The SDRmn register can be read or written in 16-bit units.

However, the higher 7 bits can be written or read only when the operation is stopped (SEmn = 0). During operation (SEmn = 1), a value is written only to the lower 8/9 bits of the SDRmn register. When the SDRmn register is read during operation, the higher 7 bits are always read as 0.

Reset signal generation clears the SDRmn register to 0000H.

Figure 15-8. Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01) After reset: 0000H R/W

	FFF11H (SDR00)								FFF10H (SDR00)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn																

Address: FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03) After reset: 0000H R/W

	FFF44H (SDR02)								FFF45H (SDR02)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn								0								

SDRmn[15:9]							Transfer clock setting by dividing the operating clock (f_{MCK})
0	0	0	0	0	0	0	$f_{MCK}/2$
0	0	0	0	0	0	1	$f_{MCK}/4$
0	0	0	0	0	1	0	$f_{MCK}/6$
0	0	0	0	0	1	1	$f_{MCK}/8$
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
1	1	1	1	1	1	0	$f_{MCK}/254$
1	1	1	1	1	1	1	$f_{MCK}/256$

(**Cautions** and **Remarks** are listed on the next page.)

Cautions 1. Be sure to clear bit 8 of the SDR02 and SDR03 to "0".

2. Setting SDRmn[15:9] = (0000000B, 0000001B) is prohibited when UART is used.

3. When operation is stopped (SEmn = 0), do not rewrite SDRmn[7:0] by an 8-bit memory manipulation instruction (SDRmn[15:9] are all cleared to 0).

Remarks 1. For the function of the lower 8/9 bits of the SDRmn register, see **15.2 Configuration of Serial Array Unit 0**.

2. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

15.3.6 Serial flag clear trigger register mn (SIRmn)

The SIRmn register is a trigger register that is used to clear each error flag of channel n.

When each bit (FECTmn, PECTmn, OVCTmn) of this register is set to 1, the corresponding bit (FEFmn, PEFmn, OVFmn) of serial status register mn is cleared to 0. Because the SIRmn register is a trigger register, it is cleared immediately when the corresponding bit of the SSRmn register is cleared.

The SIRmn register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SIRmn register can be set with an 8-bit memory manipulation instruction with SIRmnL.

Reset signal generation clears the SIRmn register to 0000H.

Figure 15-9. Format of Serial Flag Clear Trigger Register mn (SIRmn)

Address: F0108H, F0109H (SIR00) to F010EH, F010FH (SIR03) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FECTmn[Note]	PECTmn	OVCTmn

FECTmn	Clear trigger of framing error 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 The SIR01 and SIR03 registers only.

Caution Be sure to clear bits 15 to 3 (or bits 15 to 2 for the SIR00 or SIR02 register) to "0".

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

2. When the SIRmn register is read, 0000H is always read.

15.3.7 Serial status register mn (SSRmn)

The SSRmn register is a register that indicates the communication status and error occurrence status of channel n. The errors indicated by this register are a framing error, parity error, and overrun error.

The SSRmn register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSRmn register can be set with an 8-bit memory manipulation instruction with SSRmnL.

Reset signal generation clears the SSRmn register to 0000H.

Figure 15-10. Format of Serial Status Register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSFmn	BFFmn	0	0	FEFmn[Note]	PEFmn	OVFmn

TSFmn	Communication status indication flag of channel n
0	Communication is stopped or suspended.
1	Communication is in progress.
<Clear conditions> • The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is suspended). • Communication ends. <Set condition> • Communication starts.	

BFFmn	Buffer register status indication flag of channel n
0	Valid data is not stored in the SDRmn register.
1	Valid data is stored in the SDRmn register.
<Clear conditions> • Transferring transmit data from the SDRmn register to the shift register ends during transmission. • Reading receive data from the SDRmn register ends during reception. • The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is enabled). <Set conditions> • Transmit data is written to the SDRmn register while the TxEmn bit of the SCRmn register is set to 1 (transmission or transmission and reception mode in each communication mode). • Receive data is stored in the SDRmn register while the RxEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode). • A reception error occurs.	

Note The SSR01 and SSR03 registers only.

Caution When the Simplified SPI (CSI) is performing reception operations in the SNOOZE mode (SWCm = 1), the BFFmn flag will not change.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 15-10. Format of Serial Status Register mn (SSRmn) (2/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEFm n[Note]	PEF mn	OVF mn

FEFmn[Note]	Framing error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception).
<Clear condition> • 1 is written to the FECTmn bit of the SIRmn register. <Set condition> • A stop bit is not detected when UART reception ends.	

PEFmn	Parity error detection flag of channel n
0	No error occurs.
1	A Parity error occurs (during UART reception).
<Clear condition> • 1 is written to the PECTmn bit of the SIRmn register. <Set condition> • The parity of the transmit data and the parity bit do not match when UART reception ends (parity error).	

OVFmn	Overrun error detection flag of channel n
0	No error occurs.
1	An error occurs
<Clear condition> • 1 is written to the OVCTmn bit of the SIRmn register. <Set condition> • Even though receive data is stored in the SDRmn register, that data is not read and transmit data or the next receive data is written while the RxEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode). • Transmit data is not ready for slave transmission or transmission and reception in Simplified SPI (CSI) mode.	

Note The SSR01 and SSR03 registers only.

Cautions 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 (OVEmn = 1) is detected.

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), n: Channel number (n = 0 to 3)

15.3.8 Serial channel start register m (SSm)

The SSm register is a trigger register that is used to enable starting communication/count by each channel.

When 1 is written a bit of this register (SSmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is set to 1 (Operation is enabled). Because the SSmn bit is a trigger bit, it is cleared immediately when SEmn = 1.

The SSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSm register can be set with an 1-bit or 8-bit memory manipulation instruction with SSmL.

Reset signal generation clears the SSm register to 0000H.

Figure 15-11. Format of Serial Channel Start Register m (SSm)

Address: F0122H, F0123H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3	SSm2	SSm1	SSm0

SSmn	Operation start trigger of channel n
0	No trigger operation
1	Sets the SEmn bit to 1 and enters the communication wait status[Note].

Note If set the SSmn = 1 to during a communication operation, will wait status to stop the communication. At this time, holding status value of control register and shift register, SCKmn and SOmn pins, and FEFmn, PEFmn, OVFmn flags.

Cautions 1. Be sure to clear bits 15 to 4 to "0".

2. For the UART reception, set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after 4 or more f_{MCK} clocks have elapsed.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

2. When the SSm register is read, 0000H is always read.

15.3.9 Serial channel stop register m (STm)

The STm register is a trigger register that is used to enable stopping communication/count by each channel.

When 1 is written a bit of this register (STmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is cleared to 0 (operation is stopped). Because the STmn bit is a trigger bit, it is cleared immediately when SEmn = 0.

The STm register can set written by a 16-bit memory manipulation instruction.

The lower 8 bits of the STm register can be set with a 1-bit or 8-bit memory manipulation instruction with STmL.

Reset signal generation clears the STm register to 0000H.

Figure 15-12. Format of Serial Channel Stop Register m (STm)

Address: F0124H, F0125H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STm	0	0	0	0	0	0	0	0	0	0	0	0	STm3	STm2	STm1	STm0

STmn	Operation stop trigger of channel n
0	No trigger operation
1	Clears the SEmn bit to 0 and stops the communication operation[Note].

Note Holding status value of the control register and shift register, the SCKmn and SOmn pins, and FEFmn, PEFmn, OVFmn flags.

Caution Be sure to clear bits 15 to 4 to "0".

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

2. When the STm register is read, 0000H is always read.

15.3.10 Serial channel enable status register m (SEm)

The SEm register indicates whether data transmission/reception operation of each channel is enabled or stopped.

When 1 is written a bit of serial channel start register m (SSm), the corresponding bit of this register is set to 1. When 1 is written a bit of serial channel stop register m (STm), the corresponding bit is cleared to 0.

Channel n that is enabled to operate cannot rewrite by software the value of the CKOmn bit (serial clock output of channel n) of serial output register m (SOm) to be described below, and a value reflected by a communication operation is output from the serial clock pin.

Channel n that stops operation can set the value of the CKOmn bit of the SOm register by software and output its value from the serial clock pin. In this way, any waveform, such as that of a start condition/stop condition, can be created by software.

The SEm register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the SEm register can be set with a 1-bit or 8-bit memory manipulation instruction with SEmL.

Reset signal generation clears the SEm register to 0000H.

Figure 15-13. Format of Serial Channel Enable Status Register m (SEm)

Address: F0120H, F0121H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEm	0	0	0	0	0	0	0	0	0	0	0	0	SEm3	SEm2	SEm1	SEm0

SEmn	Indication of operation enable/stop status of channel n
0	Operation stops
1	Operation is enabled.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

15.3.11 Serial output enable register m (SOEm)

The SOEm register is a register that is used to enable or stop output of the serial communication operation of each channel.

Channel n that enables serial output cannot rewrite by software the value of the SOmn bit of serial output register m (SOm) to be described below, and a value reflected by a communication operation is output from the serial data output pin.

For channel n, whose serial output is stopped, the SOmn bit value of the SOm register can be set by software, and that value can be output from the serial data output pin. In this way, any waveform of the start condition and stop condition can be created by software.

The SOEm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SOEm register can be set with a 1-bit or 8-bit memory manipulation instruction with SOEmL.

Reset signal generation clears the SOEm register to 0000H.

Figure 15-14. Format of Serial Output Enable Register m (SOEm)

Address: F012AH, F012BH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2	0	SOEm0

SOEmn	Serial output enable/stop of channel n
0	Stops output by serial communication operation.
1	Enables output by serial communication operation.

Caution Be sure to clear bits 15 to 3 and 1 to "0".

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)

15.3.12 Serial output register m (SOm)

The SOm register is a buffer register for serial output of each channel.

The value of the SOmn bit of this register is output from the serial data output pin of channel n.

The value of the CKOmn bit of this register is output from the serial clock output pin of channel n.

The SOmn bit of this register can be rewritten by software only when serial output is disabled (SOEmn = 0). When serial output is enabled (SOEmn = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.

The CKOmn bit of this register can be rewritten by software only when the channel operation is stopped (SEmn = 0). While channel operation is enabled (SEmn = 1), rewriting by software is ignored, and the value of the CKOmn bit can be changed only by a serial communication operation.

To use 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.

Reset signal generation clears the SOm register to 0F0FH.

Figure 15-15. Format of Serial Output Register m (SOm)

Address: F0128H, F0129H After reset: 0F0FH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2	1	CKOm0	0	0	0	0	1	SOm2	1	SOm0

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 Be sure to set bits 11, 9, 3, and 1 to “1”. And be sure to clear bits 15 to 12 and 7 to 4 to “0”.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)

15.3.13 Serial output level register m (SOLm)

The SOLm register is a register that is used to set inversion of the data output level of each channel.

This register can be set only in the UART mode. Be sure to set 0 for corresponding bit in the Simplified SPI (CSI) mode.

Inverting channel n by using this register is reflected on pin output only when serial output is enabled (SOEmn = 1). When serial output is disabled (SOEmn = 0), the value of the SOmn bit is output as is.

Rewriting the SOLm register is prohibited when the register is in operation (when SEmn = 1).

The SOLm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SOLm register can be set with an 8-bit memory manipulation instruction with SOLmL.

Reset signal generation clears the SOLm register to 0000H.

Figure 15-16. Format of Serial Output Level Register m (SOLm)

Address: F0134H, F0135H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOLm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOLm2	0	SOLm0

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.

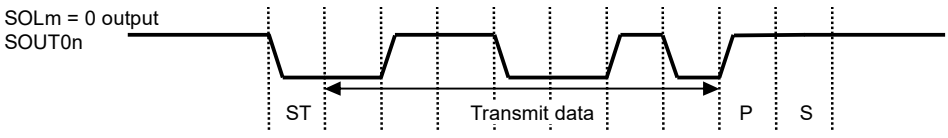
Caution Be sure to clear bits 15 to 3 and 1 to "0".

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)

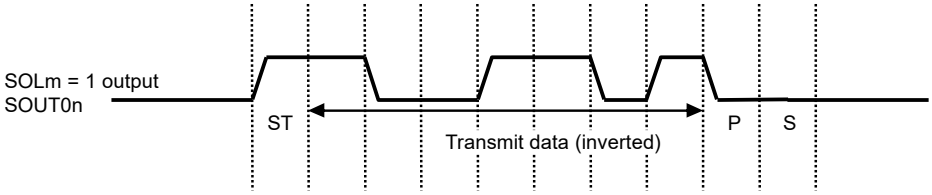
Figure 15-17 shows examples in which the level of transmit data is reversed during UART transmission.

Figure 15-17. Examples of Reverse Transmit Data

(a) Non-reverse Output (SOLmn = 0)



(b) Reverse Output (SOLmn = 1)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)

15.3.14 Serial standby control register 0 (SSC0)

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 SSC0 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSC0 register can be set with an 8-bit memory manipulation instruction with SSC0L.

Reset signal generation clears the SSC0 register to 0000H.

Caution The maximum transfer rate in the SNOOZE mode is as follows.

- **When using CSI00: Up to 1 Mbps**
- **When using UART0: 4800 bps only**

Figure 15-18. Format of Serial Standby Control Register 0 (SSC0)

Address: F0138H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSC0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSEC0	SWC0

SSEC0	Selection of whether to enable or stop the generation of communication error interrupts in the SNOOZE mode
0	Enable the generation of error interrupts (INTSRE0).
1	Stop the generation of error interrupts (INTSRE0).
• The SSEC0 bit can be set to 1 or 0 only when both the SWC0 and EOC0n bits are set to 1 during UART reception in the SNOOZE mode. In other cases, clear the SSEC0 bit to 0. • Setting SSEC0, SWC0 = 1, 0 is prohibited.	

SWC0	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 A/D conversion is performed without operating the CPU (the SNOOZE mode). • The SNOOZE mode function can only be specified when the high-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (f_{CLK}). If any other clock is selected, specifying this mode is prohibited. • Even when using SNOOZE mode, be sure to set the SWC0 bit to 0 in normal operation mode and change it to 1 just before shifting to STOP mode. Also, be sure to change the SWC0 bit to 0 after returning from STOP mode to normal operation mode.	

Figure 15-19. Interrupt in UART Reception Operation in SNOOZE Mode

EOC0n Bit	SSEC0 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.

15.3.15 Input switch control register (ISC)

The ISC1 and ISC0 bits of the ISC register are used to realize a LIN-bus and DMX512 communication operation by UART0 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 and a BREAK 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, the pulse width of the sync field, and an input signal width of DMX512 communication can be measured by the timer.

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

Reset signal generation clears the ISC register to 00H.

Figure 15-20. Format of Input Switch Control Register (ISC)

Address: F0073H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	ISC1	ISC0

ISC1	Switching channel 7 input of timer array unit
0	Do not use a timer input signal for channel 7.
1	Input signal of the RxD0 pin is used as timer input (LIN-bus: detects the wakeup signal, measures the low width of the break field and the pulse width of the sync field, DMX512: measures the input signal width).

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 (LIN-bus: wakeup signal detection, DMX512: BREAK signal detection).

Caution Be sure to clear bits 7 to 2 to "0".

15.3.16 Noise filter enable register 0 (NFEN0)

The NFEN0 register is used to set whether the noise filter can be used for the input signal from the serial data input pin to each channel.

Disable the noise filter of the pin used for Simplified SPI (CSI) 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 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.

The NFEN0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the NFEN0 register to 00H.

Figure 15-21. Format of Noise Filter Enable Register 0 (NFEN0)

Address: F0070H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN0	0	0	0	0	0	SNFEN10	0	SNFEN00

SNFEN10	Use of noise filter of RxD1 pin (RxD1/CMP5P/ANI16/P03)
0	Noise filter OFF
1	Noise filter ON
Set SNFEN10 to 1 to use the RxD1 pin. Clear SNFEN10 to 0 to use this other than RxD1 pin.	

SNFEN00	Use of noise filter of RxD0 pin (SI00/RxD0/TKCO01/INTP21/SDAA0/(TI07)/(DALIRxD4)/(TxRx4)/P11)
0	Noise filter OFF
1	Noise filter ON
Set the SNFEN00 bit to 1 to be used as the RxD0 pin. Clear the SNFEN00 bit to 0 to be used as this other than RxD0 pin.	

Caution Be sure to clear bits 7 to 3 and 1 to "0".

15.3.17 Registers controlling port functions of serial I/O pins

When using the serial array unit set the registers that control the port functions multiplexed on the target channel (port mode register (PMxx), port register (Pxx), port input mode register (PIMxx), port output mode register (POMxx), port mode control register (PMCxx)).

For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.2 Port registers (Pxx)**, **4.3.4 Port input mode registers (PIMxx)**, **4.3.5 Port output mode registers (POMxx)**, and **4.3.6 Port mode control registers (PMCxx)**.

When using the ports (such as P02/TxD1/ANI17, P12/SCK00/(TKCO03)) to be shared with the serial data output pin or serial clock output pin for serial data output or serial clock output, set the corresponding bit in the port mode control register (PMCxx) and port mode register (PMxx) to 0. And set the corresponding bit in the port register (Pxx) to 1.

Example: When using P02/TxD1/ANI17 for serial data output or serial clock output
Set the PMC02 bit of the port mode control register 0 to 0.
Set the PM02 bit of the port mode register 0 to 0.
Set the P02 bit of the port register 0 to 1.

When using the ports (such as P03/RxD1/CMP5P/ANI16, P12/SCK00/(TKCO03)) to be shared with the serial data input pin or serial clock input pin for serial data input or serial clock input, set the corresponding bit in the port mode register (PMxx) to 1. And set the corresponding bit in the port mode control register (PMCxx) to 0. In this case, the corresponding bit in the port register (Pxx) can be set to 0 or 1.

Example: When using P12/SCK00/(TKCO03) for serial data input or serial clock input
Set the PMC12 bit of port mode control register 1 to 0.
Set the PM12 bit of port mode register 1 to 1.
Set the P12 bit of port register 1 to 0 or 1.

The PM0 and PM1 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets the PM0 and PM1 registers to FFH.

The port mode registers (PMxx), port registers (Pxx), port mode control register (PMCxx), port output mode registers (POMxx), and port input mode registers (PIMxx) to be set differ depending on the product. For details, see **4.5.3 Register setting examples for used port and alternate functions**.

15.4 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.

15.4.1 Stopping the operation per unit

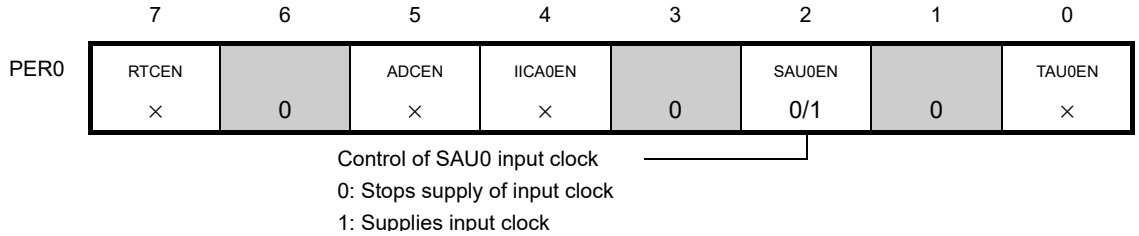
The stopping of the operation per unit is set by using peripheral enable register 0 (PER0).

The PER0 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

To stop the operation of serial array unit 0, set bit 2 (SAU0EN) to “0”.

Figure 15-22. Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation per Unit

(a) Peripheral enable register 0 (PER0) … Set only the bit of SAUm to be stopped to 0.



Cautions 1. If SAU0EN = 0, writing to a control register of serial array unit 0 is ignored, and, even if the register is read, only the default value is read

Note that this does not apply to the following registers.

- **Input switch control register (ISC)**
- **Noise filter enable register 0 (NFEN0)**
- **Port input mode registers 0, 1 (PIM0, PIM1)**
- **Port output mode registers 0, 1 (POM0, POM1)**
- **Port mode control registers 0 (PMC0)**
- **Port mode registers 0, 1 (PM0, PM1)**
- **Port registers 0, 1 (P0, P1)**

2. Be sure to clear bits 1, 3, and 6 to “0”.

Remark ☐: Setting disabled (set to the initial value)

×: 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

15.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

Figure 15-23. Each Register Setting When Stopping the Operation by Channels

(a) Serial channel stop register m (STm) … This register is a trigger register that is used to enable stopping communication/count by each channel.

STm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	STm3 0/1	STm2 0/1	STm1 0/1	STm0 0/1

1: Clears the SEmn bit to 0 and stops the communication operation

* Because the STmn bit is a trigger bit, it is cleared immediately when SEmn = 0.

(b) Serial Channel Enable Status Register m (SEm) … This register indicates whether data transmission/reception operation of each channel is enabled or stopped.

SEm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SEm3 0/1	SEm2 0/1	SEm1 0/1	SEm0 0/1

0: Operation stops

* The SEm register is a read-only status register. The serial channel operation is stopped by using the STm register.
With a channel whose operation is stopped, the value of the CKOmn bit of the SOm register can be set by software.

(c) Serial output enable register m (SOEm) … This register is a register that is used to enable or stop output of the serial communication operation of each channel.

SOEm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	0	SOEm0 0/1

0: Stops output by serial communication operation

* For channel n, whose serial output is stopped, the SOmn bit value of the SOm register can be set by software.

(d) Serial output register m (SOm) …This register is a buffer register for serial output of each channel.

SOm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	CKOm2 0/1	1	CKOm0 0/1	0	0	0	0	1	SOm2 0/1	1	SOm0 0/1

1: Serial clock output value is “1” 1: Serial data output value is “1”

* When using pins corresponding to each channel as port function pins, set the corresponding CKOmn, SOmn bits to “1”.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

2. : Setting disabled (set to the initial value), 0/1: Set to 0 or 1 depending on the usage of the user

15.5 Operation of Simplified SPI (CSI00) Communication

This is a clocked communication function that uses three lines: serial clock (SCK) and serial data (SI and SO) lines.

[Data transmission/reception]

- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable

[Clock control]

- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel
- Maximum transfer rate[Note]
  - During master communication : Max. $f_{CLK}/2$
  - During slave communication: Max. $f_{MCK}/6$

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

In addition, CSIs of following channels supports the SNOOZE mode. When SCK input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only following CSIs can be specified.

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T_A = –40 to +105°C)** or **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T_A = –40 to +125°C)**.

The channels supporting Simplified SPI (CSI00) are channels 0.

- 20-pin products

Channel	Used as Simplified SPI (CSI)	Used as UART
0	–	UART0 (supporting LIN-bus, DMX512)
1	–	
2	–	–
3	–	

- 30-pin products

Channel	Used as Simplified SPI (CSI)	Used as UART
0	–	UART0 (supporting LIN-bus, DMX512)
1	–	
2	–	UART1
3	–	

- 38-pin products

Channel	Used as Simplified SPI (CSI)	Used as UART
0	CSI00	UART0 (supporting LIN-bus, DMX512)
1	–	
2	–	UART1
3	–	

Simplified SPI (CSI00) performs the following seven types of communication operations.

- Master transmission (See **15.5.1**.)
- Master reception (See **15.5.2**.)
- Master transmission/reception (See **15.5.3**.)
- Slave transmission (See **15.5.4**.)
- Slave reception (See **15.5.5**.)
- Slave transmission/reception (See **15.5.6**.)
- SNOOZE mode function (See **15.5.7**.)

15.5.1 Master transmission

Master transmission is that the RL78 microcontroller outputs a transfer clock and transmits data to another device.

Simplified SPI	CSI00
Target channel	Channel 0 of SAU0
Pins used	SCK00, SO00
Interrupt	INTCSI00
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	None
Transfer data length	7 or 8 bits
Transfer rate	Max. $f_{CLK}/2$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz]**Note** f_{CLK}: System clock frequency
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts from the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse
Data direction	MSB or LSB first

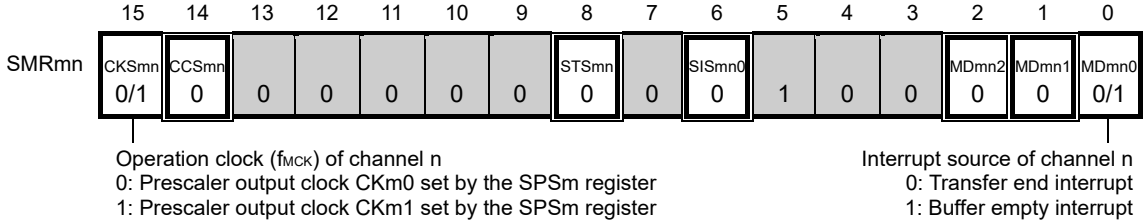
Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C)** or **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T_A = −40 to +125°C)**).

Remark m: Unit number (m = 0), n: Channel number (n = 0)

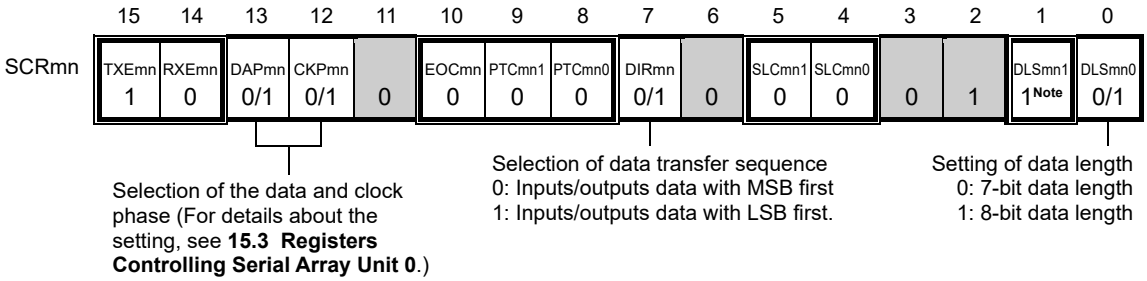
(1) Register setting

Figure 15-24. Example of Contents of Registers for Master Transmission of Simplified SPI (CSI00)

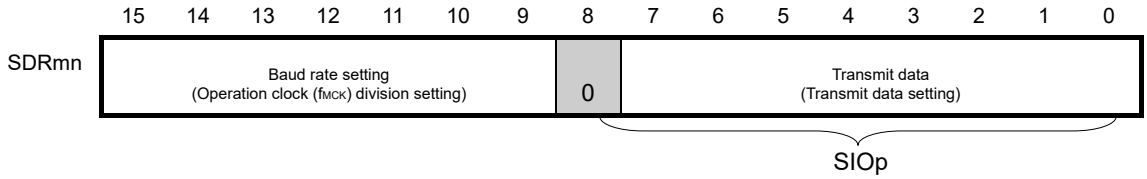
(a) Serial mode register mn (SMRmn)



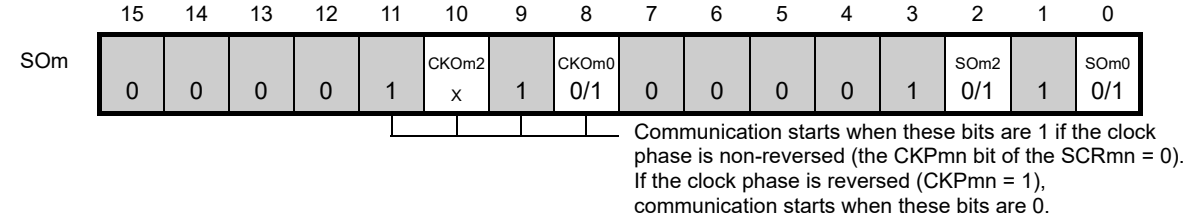
(b) Serial communication operation setting register mn (SCRmn)



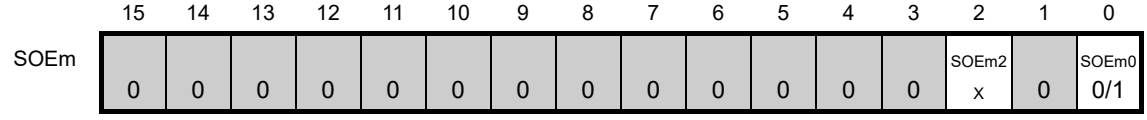
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



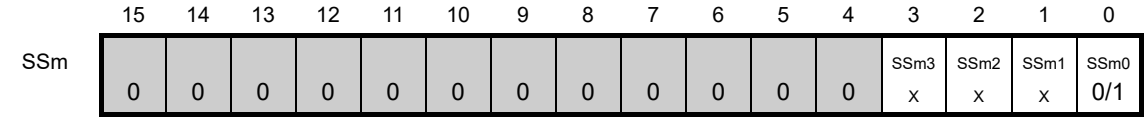
(d) Serial output register m (SOm) … Sets only the bits of the target channel.



(e) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.



(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.



Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

2. ☐ : Setting is fixed in the Simplified SPI (CSI) master transmission mode,

▒ : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 15-25. Initial Setting Procedure for Master Transmission

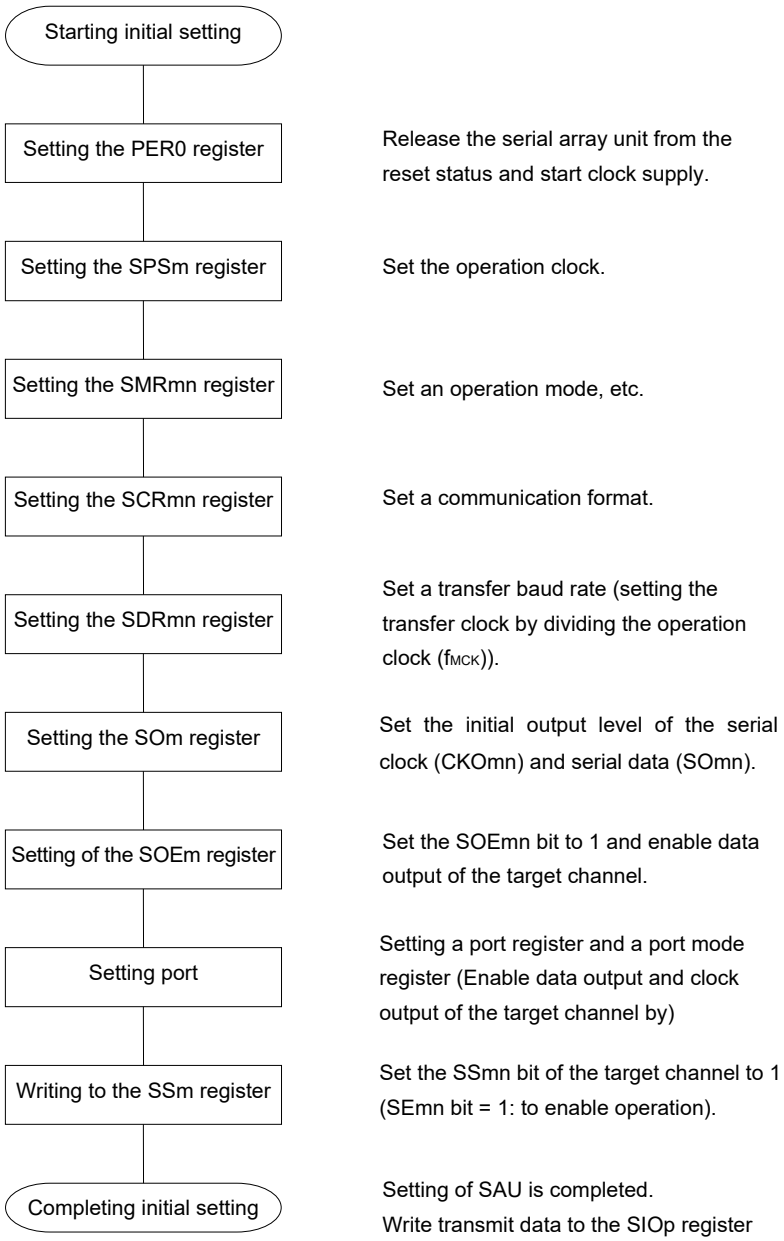


Figure 15-26. Procedure for Stopping Master Transmission

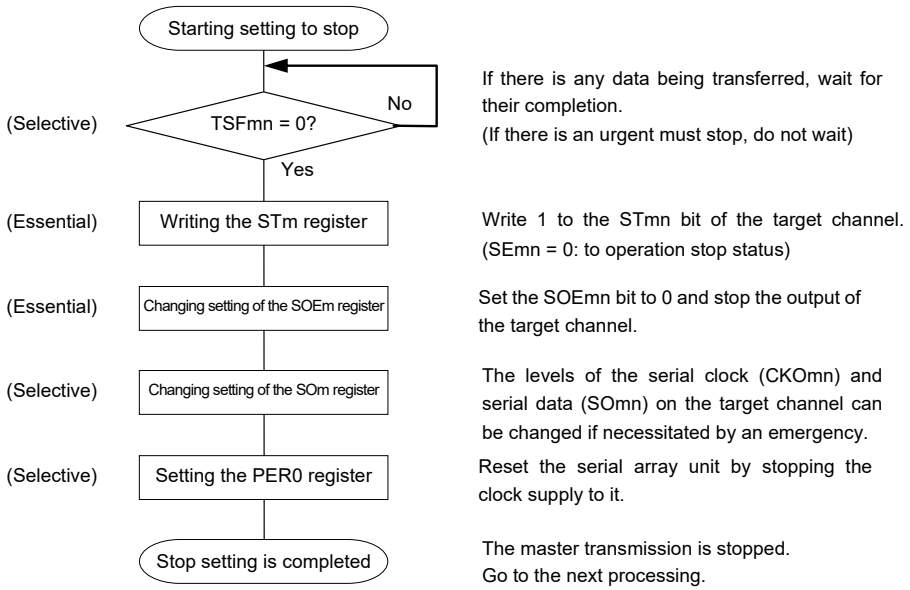
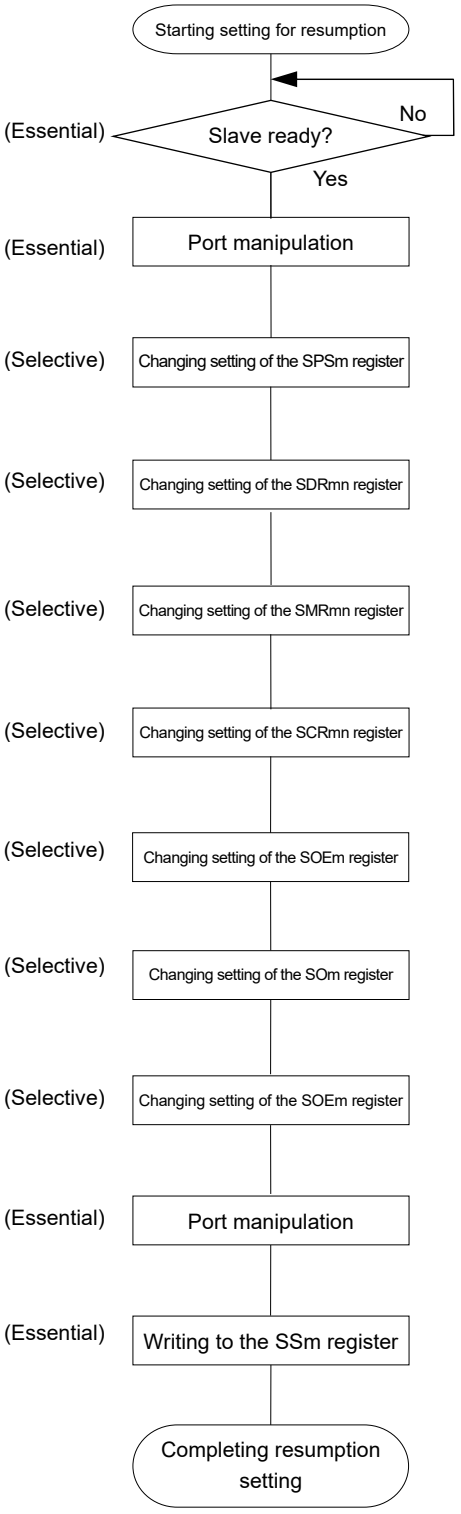


Figure 15-27. Procedure for Resuming Master Transmission



Wait until stop the communication target (slave) or communication operation completed

Disable data output and clock output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

Set the SOEmn bit to 0 to stop output from the target channel.

Set the initial output level of the serial clock (CKOmn) and serial data (SOmn).

Set the SOEmn bit to 1 and enable output from the target channel.

Enable data output and clock output of the target channel by setting a port register and a port mode register.

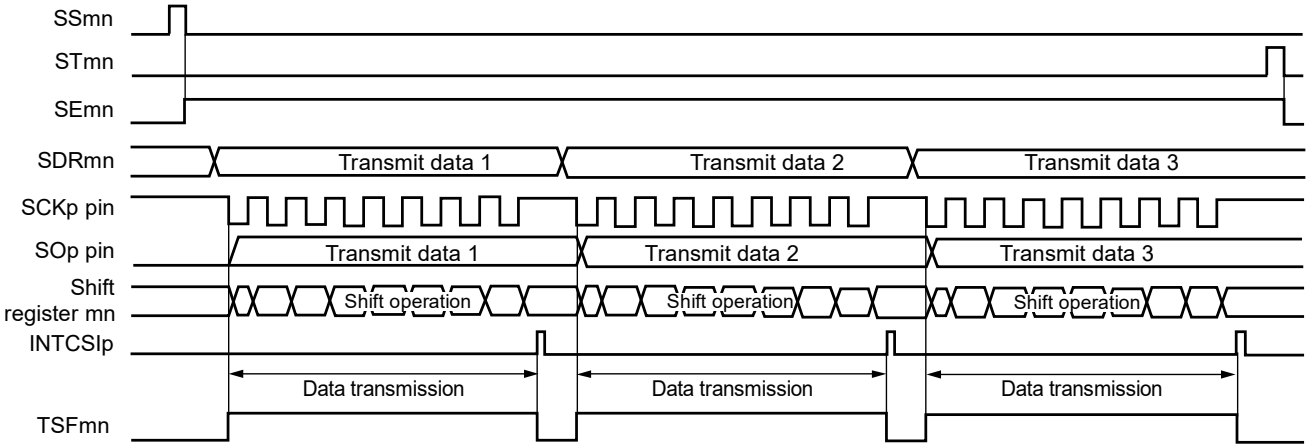
Set the SSmn bit of the target channel to 1 (SEmn = 1: to enable operation).

Setting is completed
Sets transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and start communication.

Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

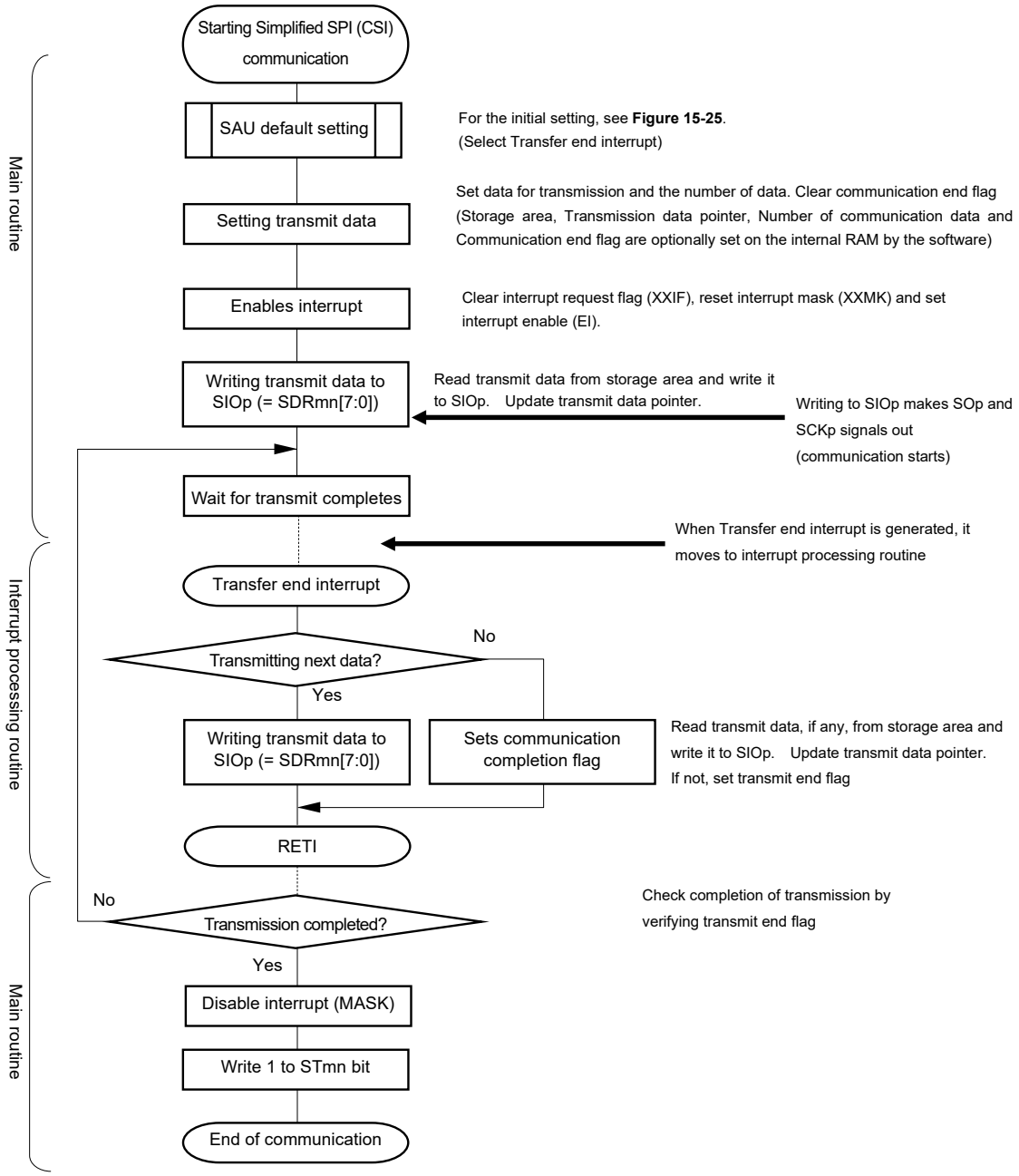
(3) Processing flow (in single-transmission mode)

Figure 15-28. Timing Chart of Master Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



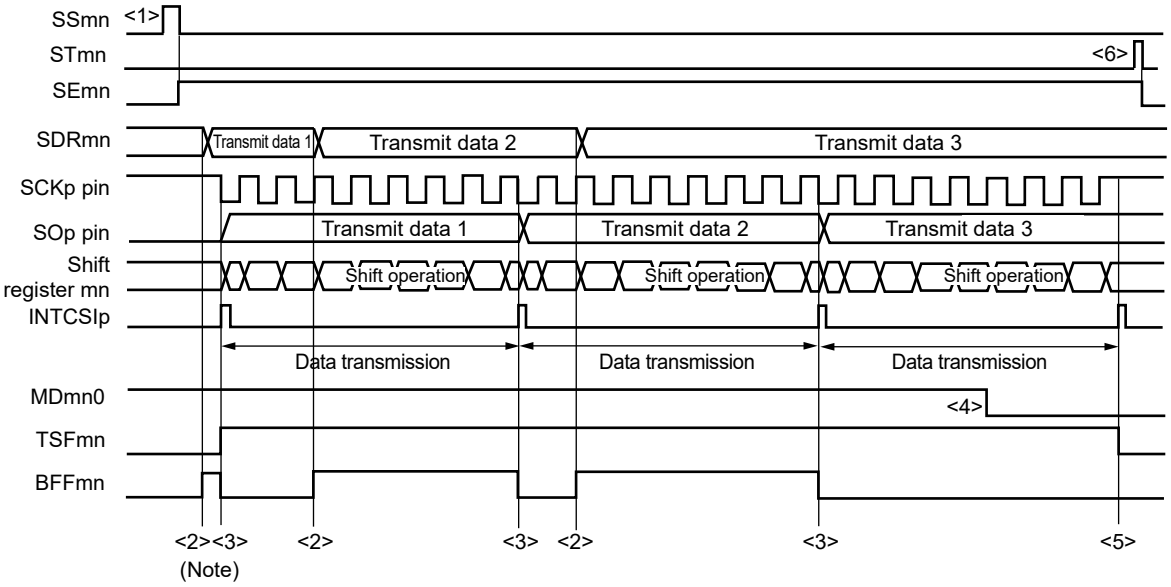
Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 15-29. Flowchart of Master Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 15-30. Timing Chart of Master Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

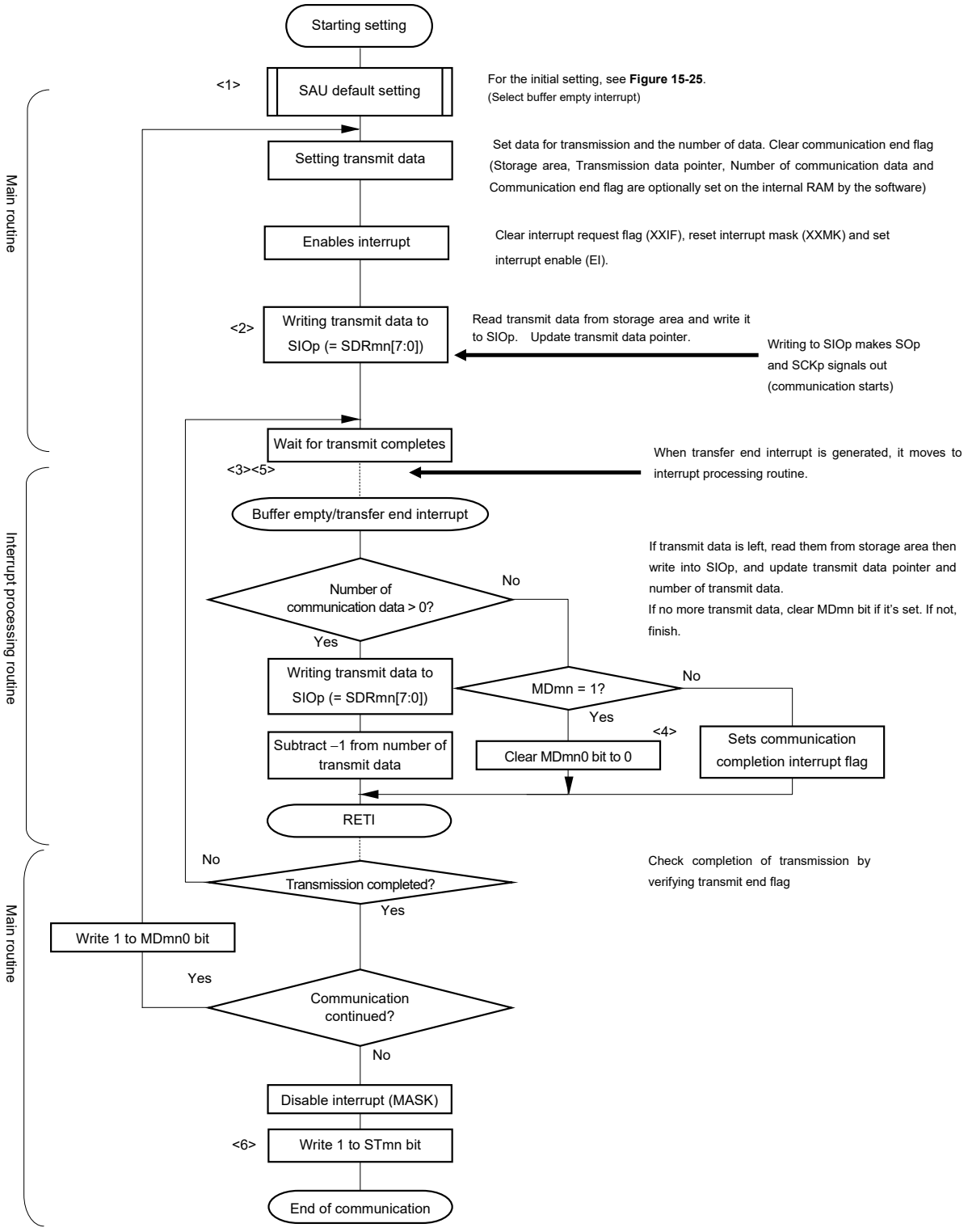


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 15-31. Flowchart of Master Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 15-30 Timing Chart of Master Transmission (in Continuous Transmission Mode)**.

15.5.2 Master reception

Master reception is that the RL78 microcontroller outputs a transfer clock and receives data from other device.

Simplified SPI	CSI00	CSI01	CSI10	CSI11	CSI20	CSI21	CSI30	CSI31
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	Channel 2 of SAU1	Channel 3 of SAU1
Pins used	SCK00, SI00	SCK01, SI01	SCK10, SI10	SCK11, SI11	SCK20, SI20	SCK21, SI21	SCK30, SI30	SCK31, SI31
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21	INTCSI30	INTCSI31
	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] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK}: System clock frequency							
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data input starts from the start of the operation of the serial clock. • DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.							
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse							
Data direction	MSB or LSB first							

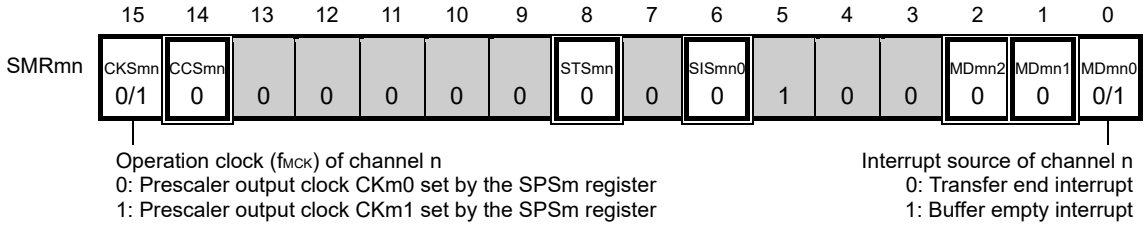
Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C)** or **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T_A = −40 to +125°C)**).

Remark m: Unit number (m = 0), n: Channel number (n = 0)

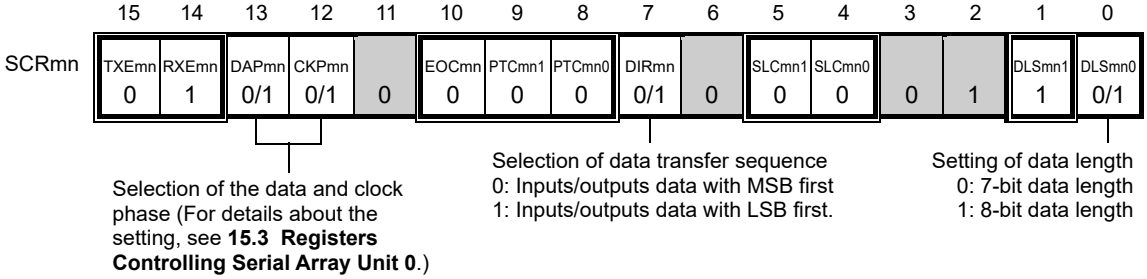
(1) Register setting

Figure 15-32. Example of Contents of Registers for Master Reception of Simplified SPI (CSI00) (1/2)

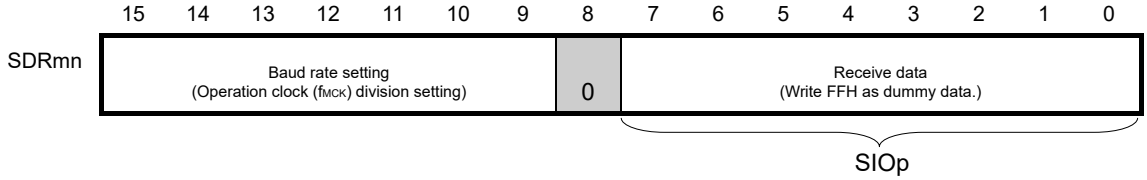
(a) Serial mode register mn (SMRmn)



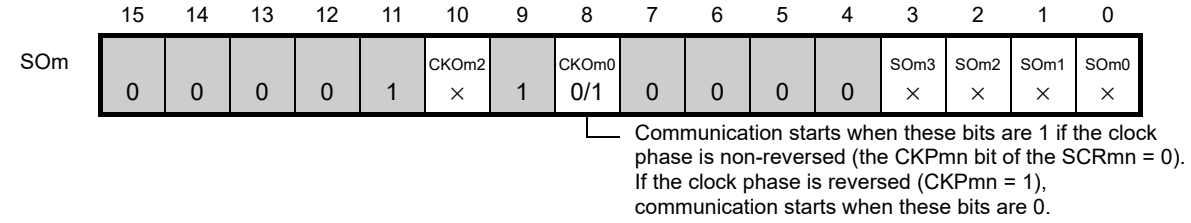
(b) Serial communication operation setting register mn (SCRmn)



(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



(d) Serial output register m (SOm) … Sets only the bits of the target channel.



Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

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

Figure 15-32. Example of Contents of Registers for Master Reception of Simplified SPI (CSI00) (2/2)

(e) Serial output enable register m (SOEm) …The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	0	SOEm0 ×

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 0/1

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

2. ☐ : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 15-33. Initial Setting Procedure for Master Reception

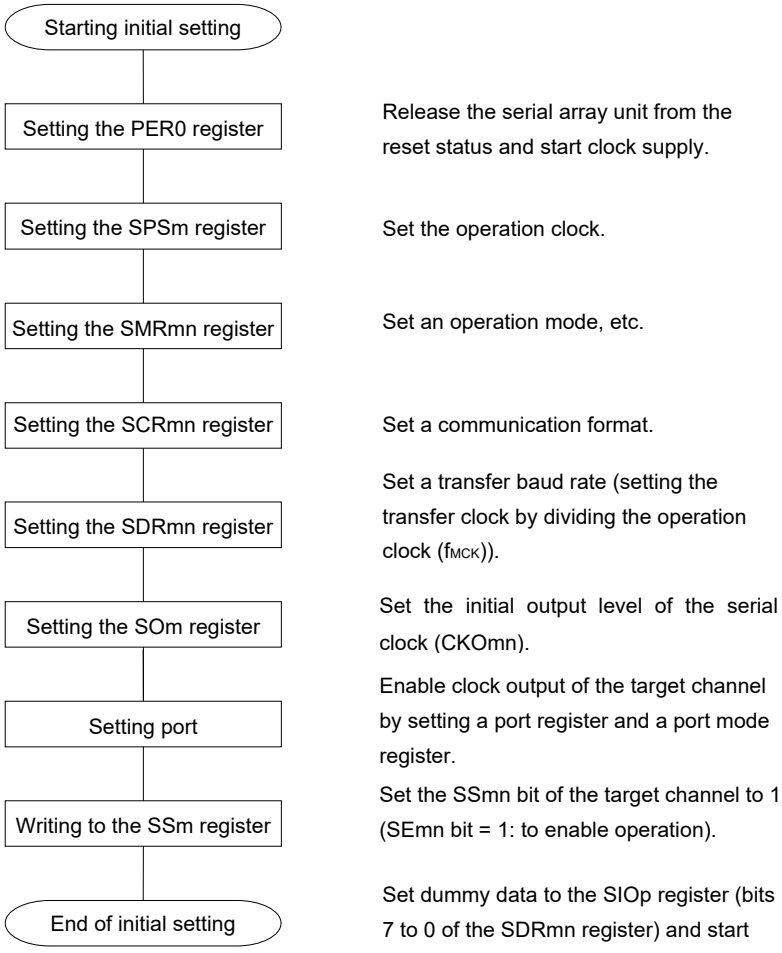


Figure 15-34. Procedure for Stopping Master Reception

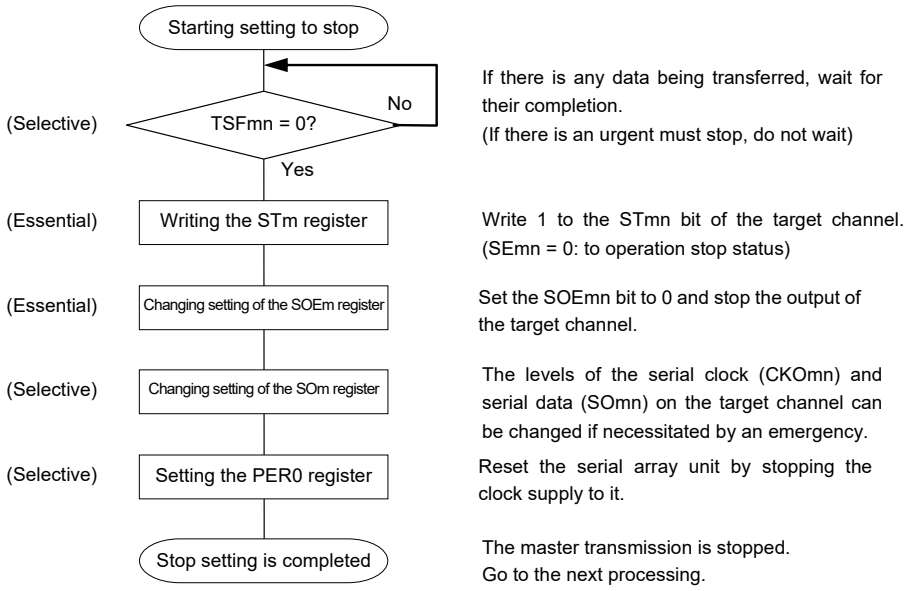
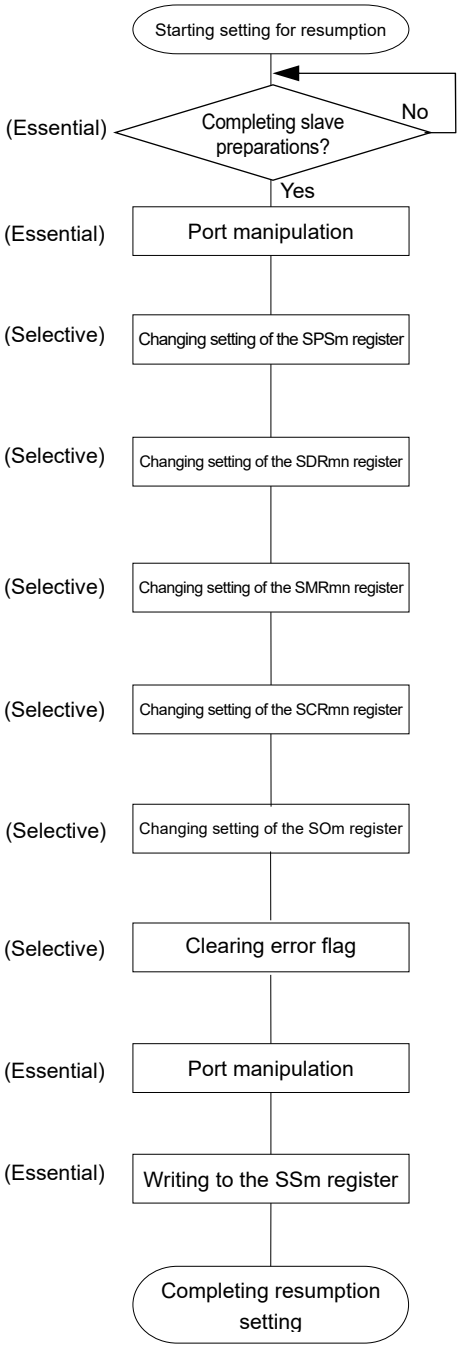


Figure 15-35. Procedure for Resuming Master Reception



Wait until stop the communication target (slave) or communication operation completed

Disable clock output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

Set the initial output level of the serial clock (CKOmn).

If the OVF flag remain set, clear this using serial flag clear trigger register mn (SIRmn).

Enable clock output of the target channel by setting a port register and a port mode register.

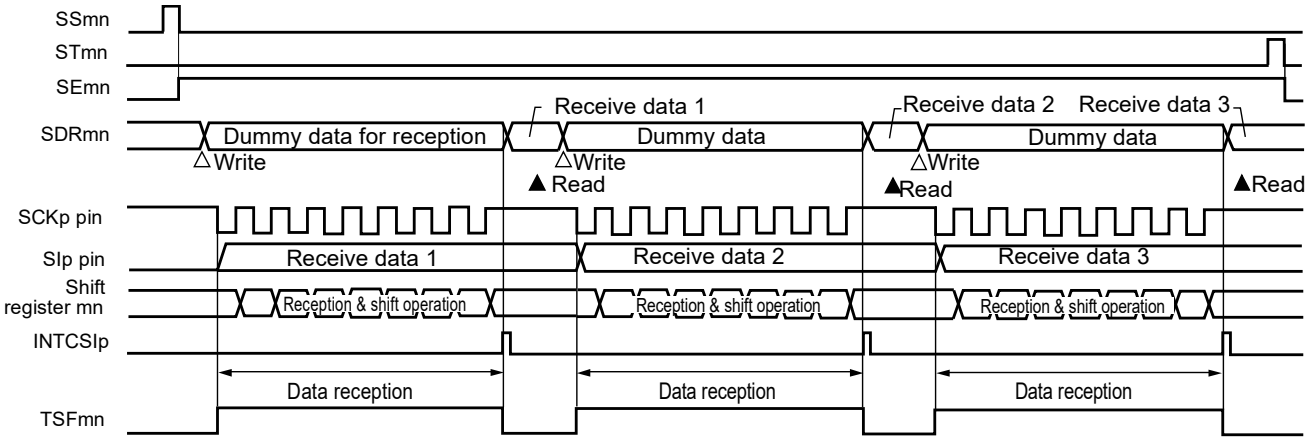
Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation).

Setting is completed
Sets dummy data to the SIOp register (bits 7 to 0 of the SDRmn register) and start communication.

Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

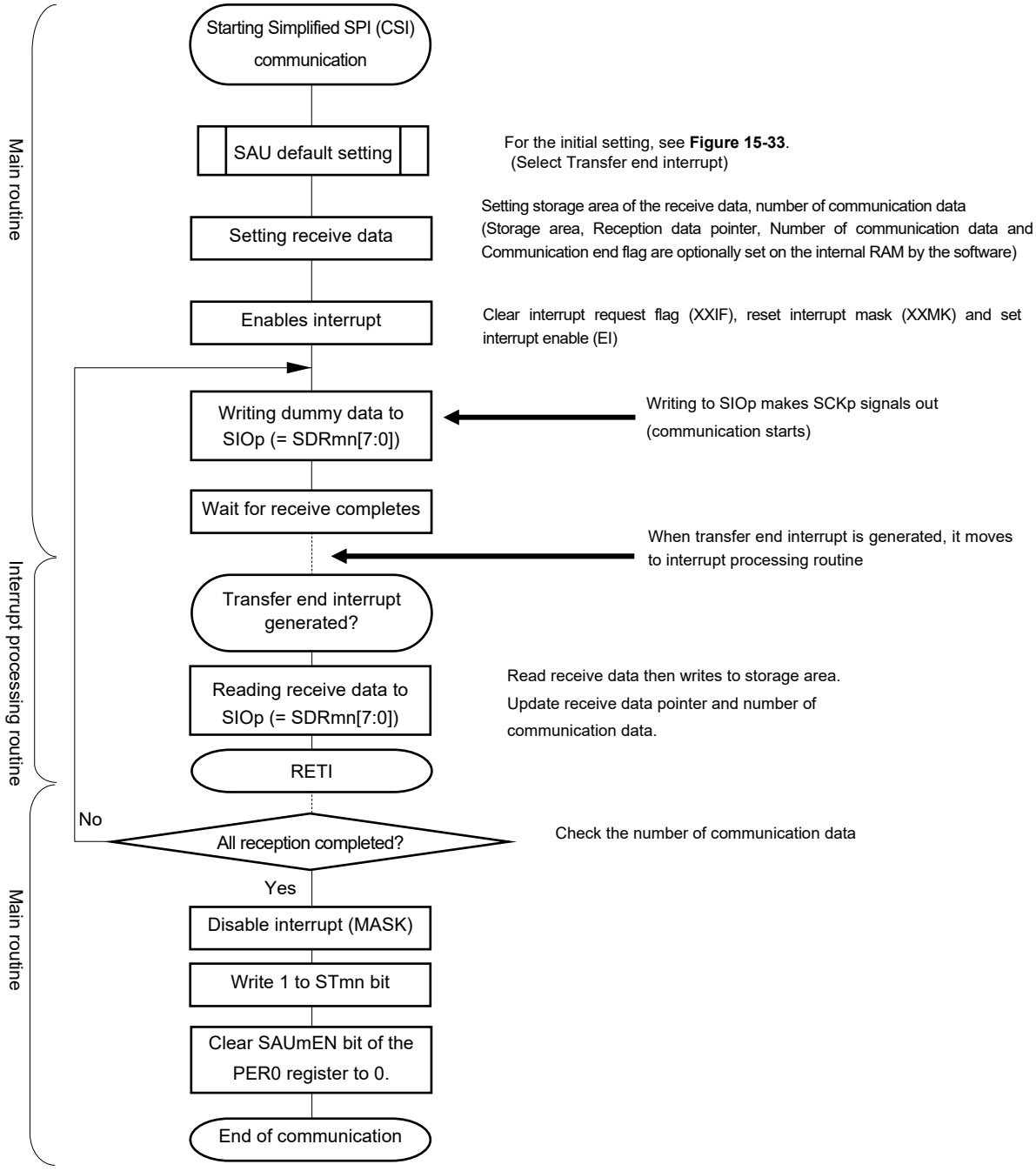
(3) Processing flow (in single-reception mode)

Figure 15-36. Timing Chart of Master Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



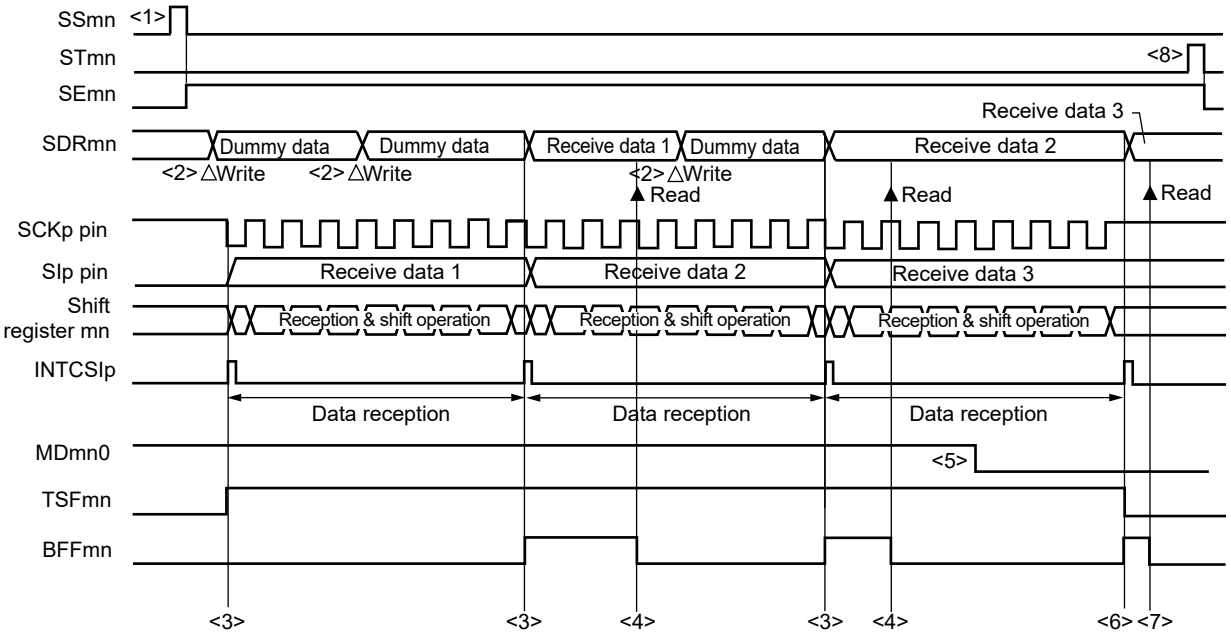
Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 15-37. Flowchart of Master Reception (in Single-Reception Mode)



(4) Processing flow (in continuous reception mode)

Figure 15-38. Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)

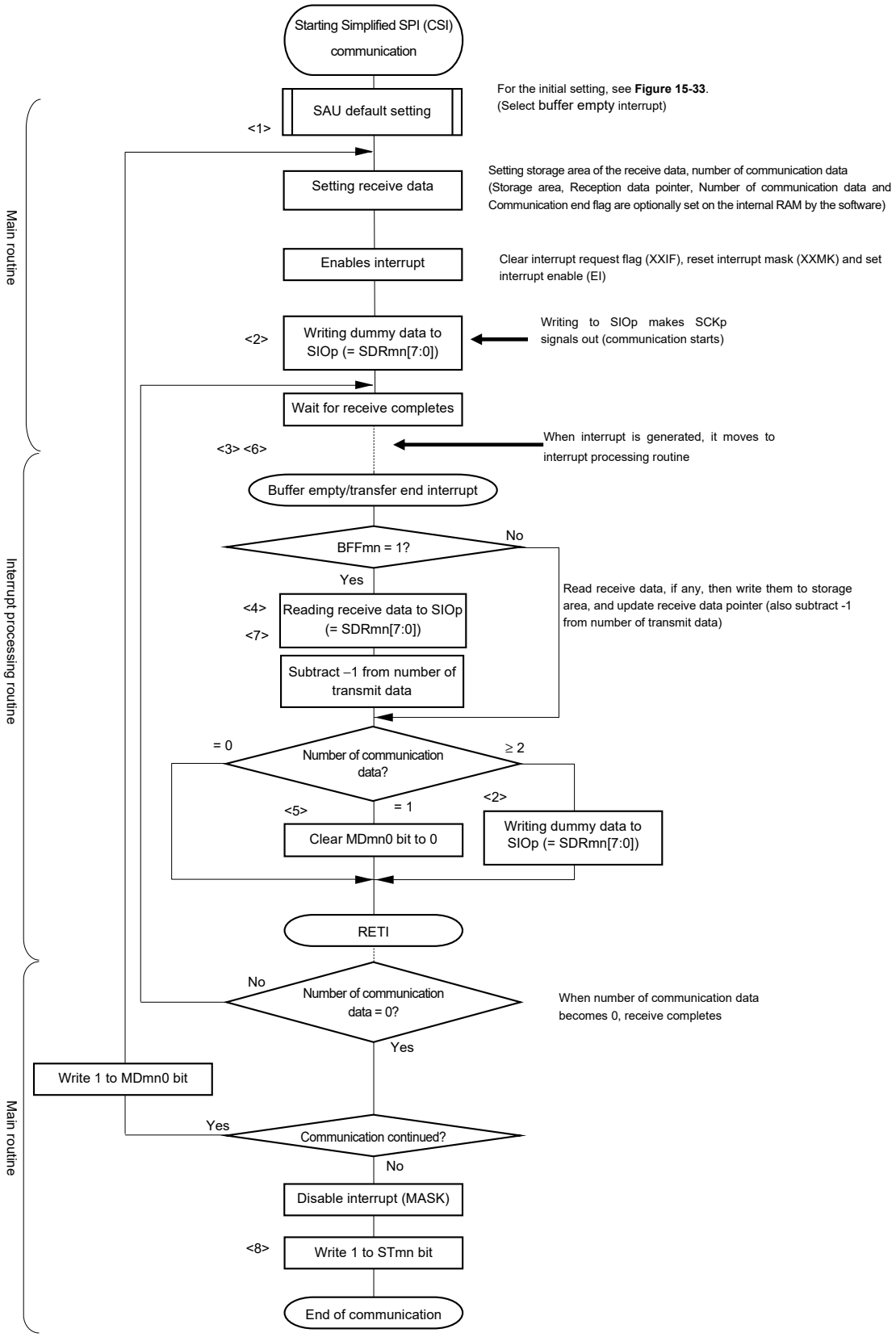


Caution The MDmn0 bit can be rewritten even during operation.
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 15-39 Flowchart of Master Reception (in Continuous Reception Mode)**.

2. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 15-39. Flowchart of Master Reception (in Continuous Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 15-38 Timing Chart of Master Reception (in Continuous Reception Mode)**.

15.5.3 Master transmission/reception

Master transmission/reception is that the RL78 microcontroller outputs a transfer clock and transmits/receives data to/from other device.

Simplified SPI	CSI00
Target channel	Channel 0 of SAU0
Pins used	SCK00, SI00, SO00
Interrupt	INTCSI00
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	Overrun error detection flag (OVFmn) only
Transfer data length	7 or 8 bits
Transfer rate	Max. $f_{CLK}/2$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz]**Note** f_{CLK}: System clock frequency
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data I/O starts at the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse
Data direction	MSB or LSB first

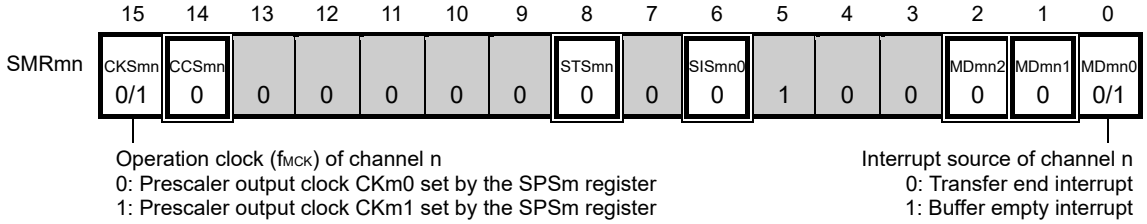
Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C)** or **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T_A = −40 to +125°C)**).

Remark m: Unit number (m = 0), n: Channel number (n = 0)

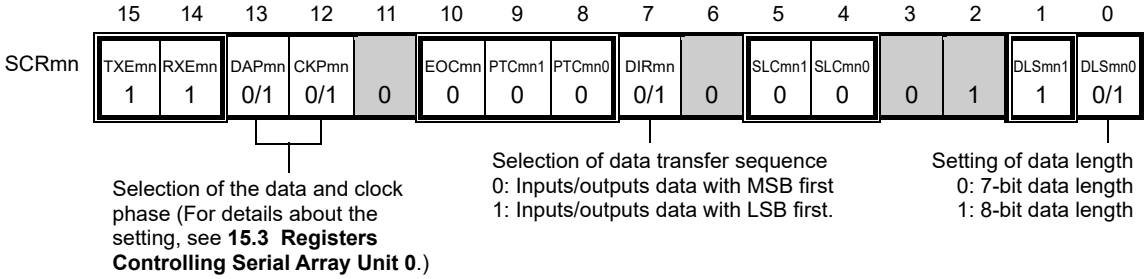
(1) Register setting

Figure 15-40. Example of Contents of Registers for Master Transmission/Reception of Simplified SPI (CSI00) (1/2)

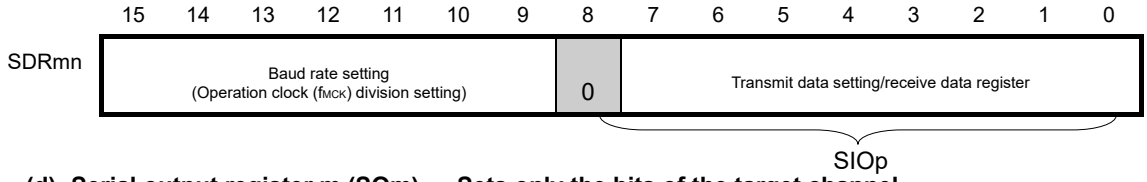
(a) Serial mode register mn (SMRmn)



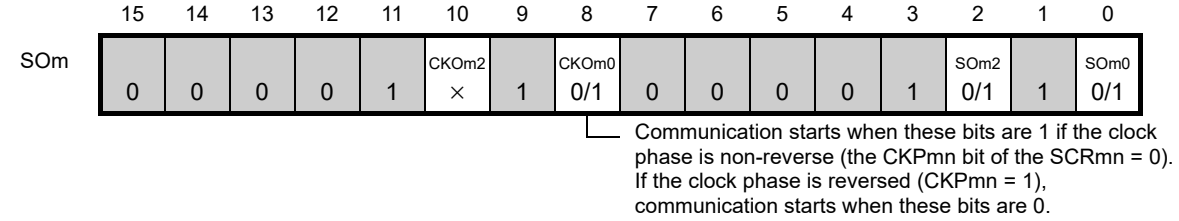
(b) Serial communication operation setting register mn (SCRmn)



(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



(d) Serial output register m (SOm) … Sets only the bits of the target channel.



Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

2. ☐: Setting is fixed in the Simplified SPI (CSI) master transmission/reception mode

▒: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

Figure 15-40. Example of Contents of Registers for Master Transmission/Reception of Simplified SPI (CSI00) (2/2)

(e) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	0	SOEm0 0/1

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 0/1

Note Serial array unit 0 only.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

2. ▢ : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 15-41. Initial Setting Procedure for Master Transmission/Reception

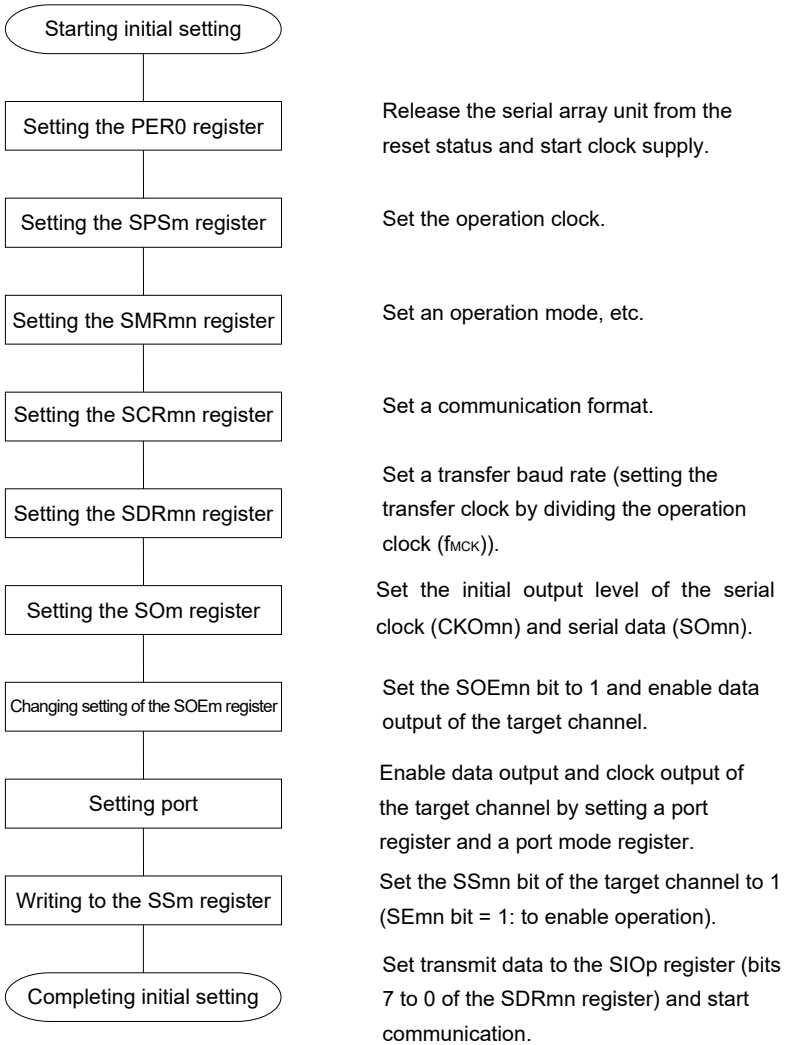


Figure 15-42. Procedure for Stopping Master Transmission/Reception

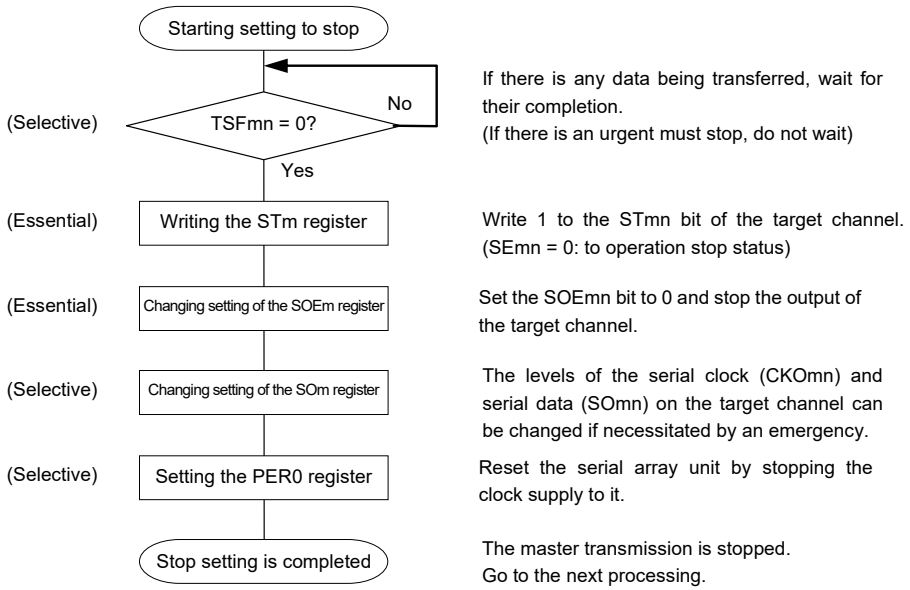
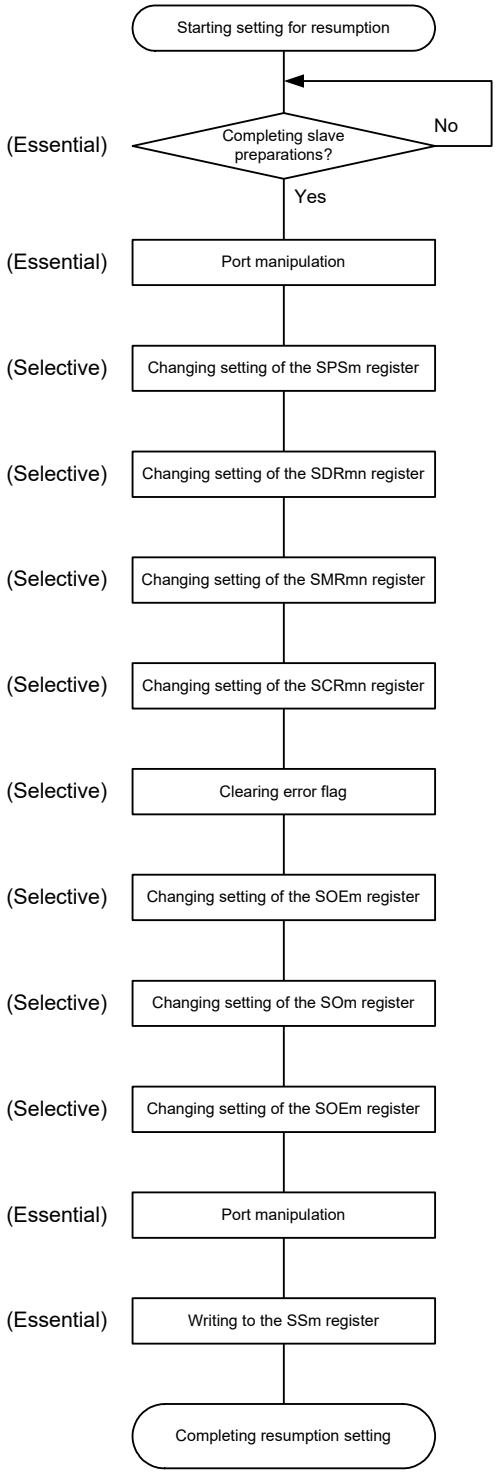


Figure 15-43. Procedure for Resuming Master Transmission/Reception



Wait until stop the communication target (slave) or communication operation completed.

Disable data output and clock output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

If the OVF flag remains set, clear this using serial flag clear trigger register mn (SIRmn).

Set the SOEmn bit to 0 to stop output from the target channel.

Set the initial output level of the serial clock (CKOmn) and serial data (SOmn).

Set the SOEmn bit to 1 and enable output from the target channel.

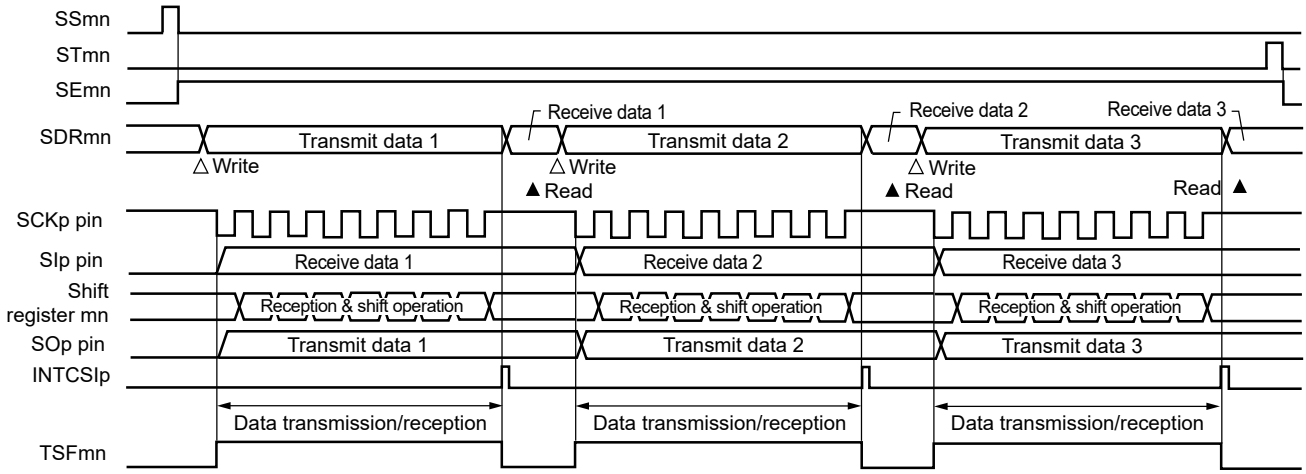
Enable data output and clock output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 and set the SEmn bit to 1 (to enable operation).

Sets transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and start communication.

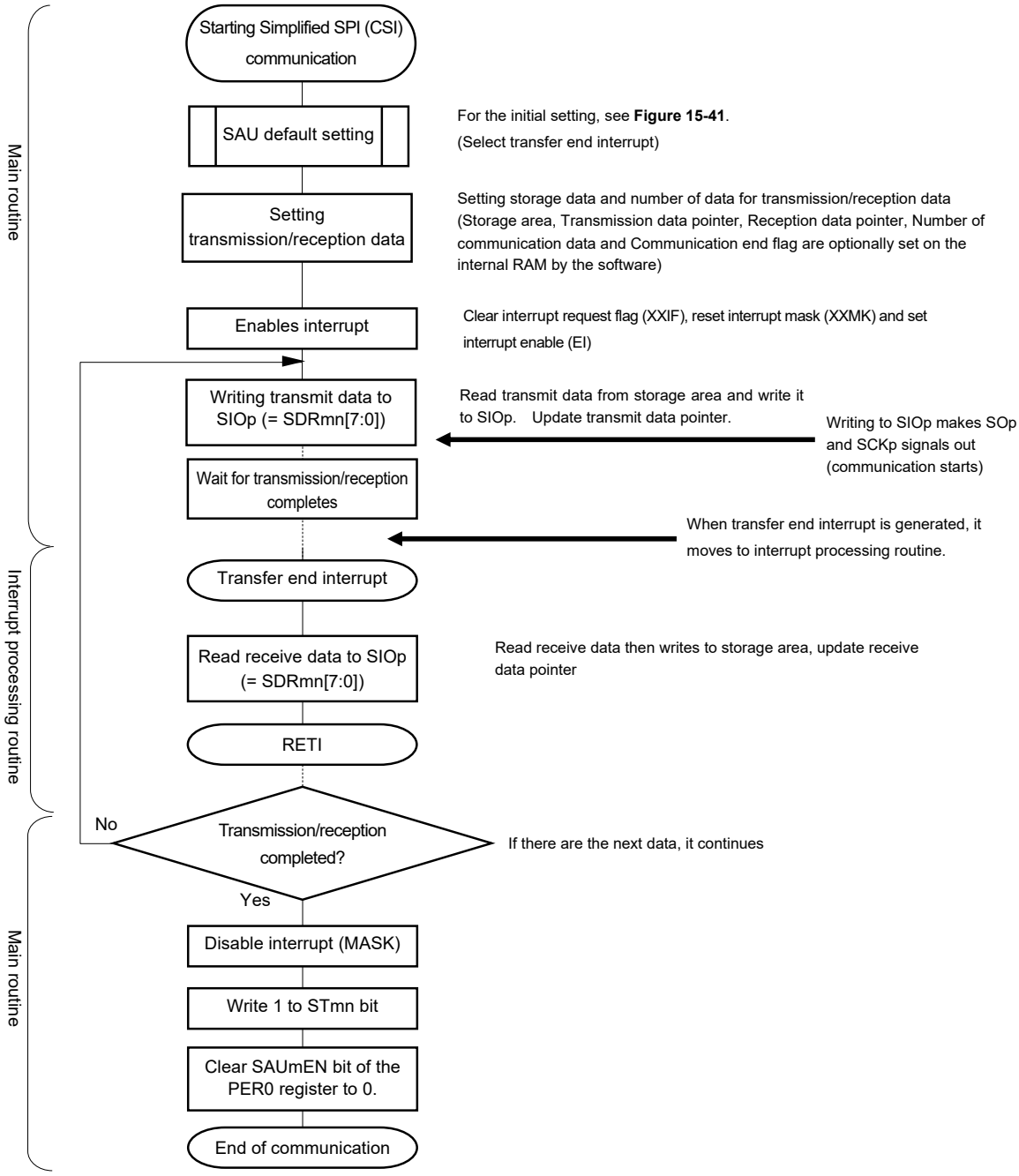
(3) Processing flow (in single-transmission/reception mode)

Figure 15-44. Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



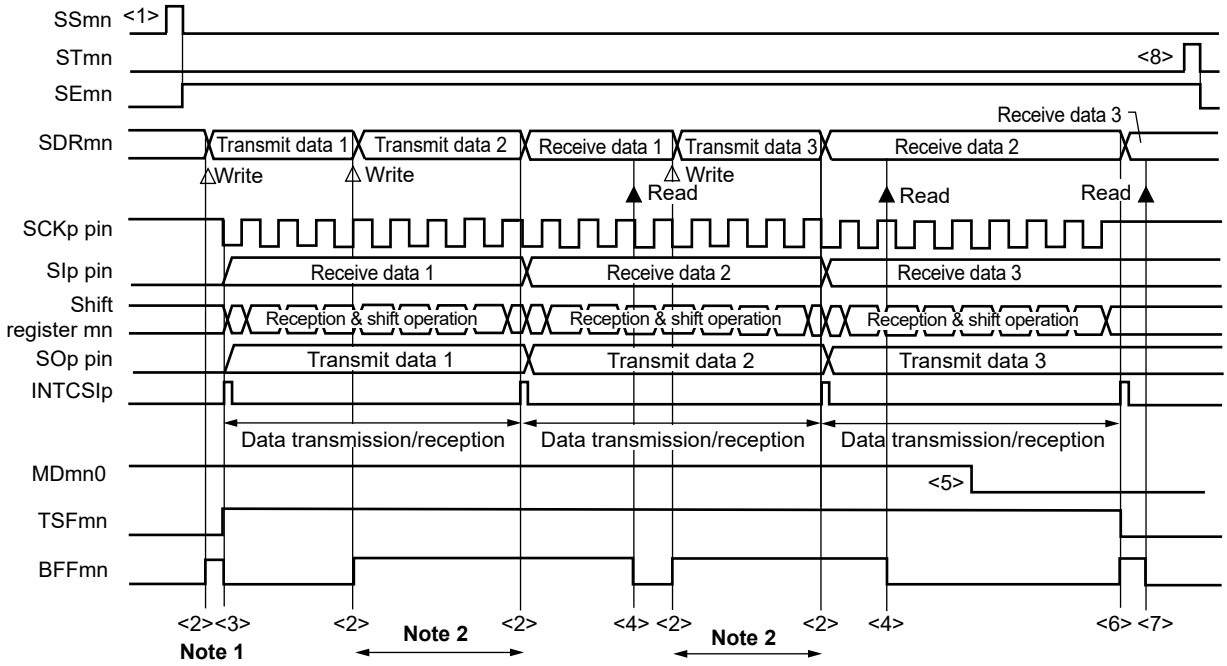
Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 15-45. Flowchart of Master Transmission/Reception (in Single-Transmission/Reception Mode)



(4) Processing flow (in continuous transmission/reception mode)

Figure 15-46. Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Notes 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.

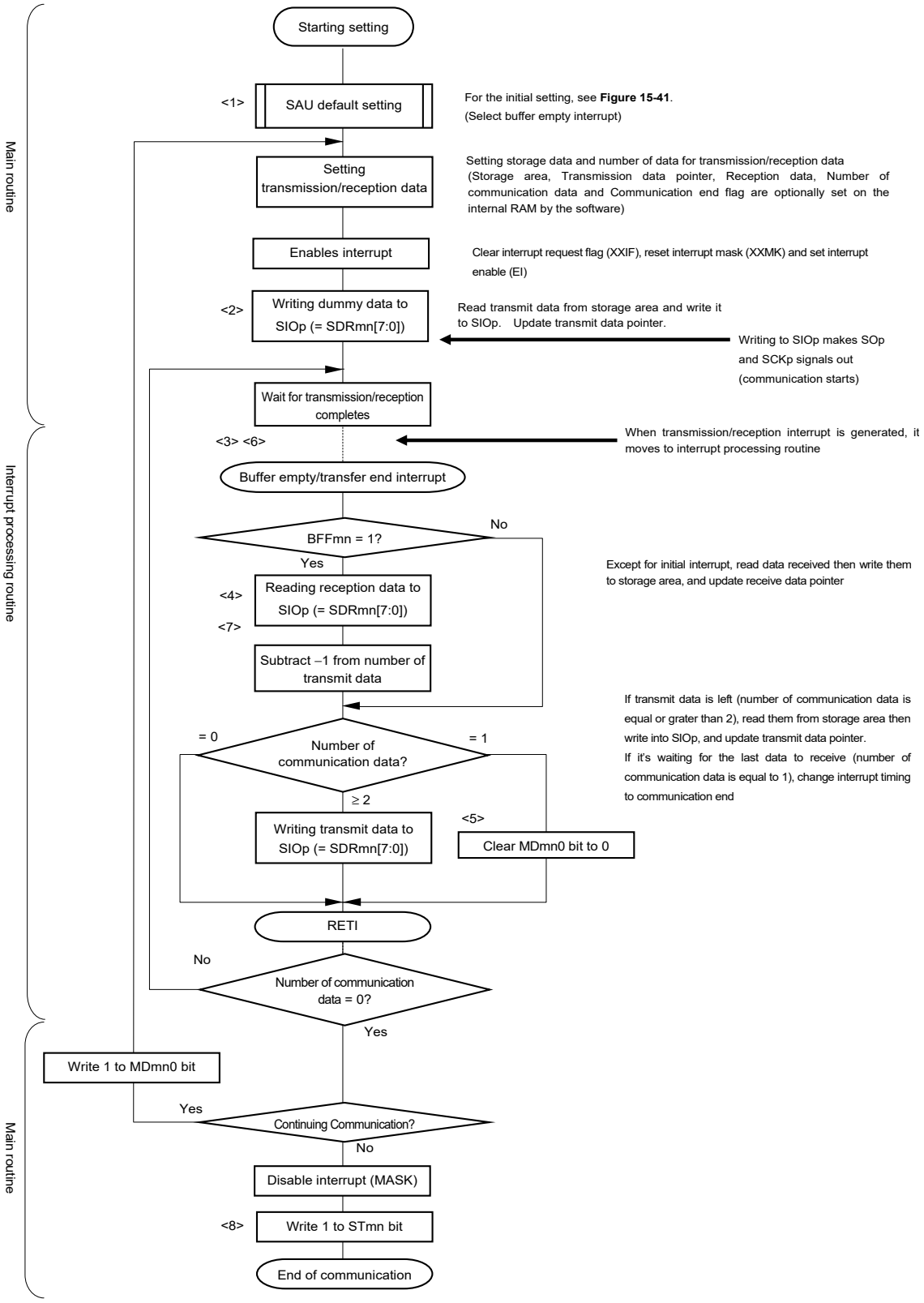
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

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.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 15-47 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 15-47. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 15-46 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

15.5.4 Slave transmission

Slave transmission is that the RL78 microcontroller transmits data to another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00
Target channel	Channel 0 of SAU0
Pins used	SCK00, SO00
Interrupt	INTCSI00
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	Overrun error detection flag (OVFmn) only
Transfer data length	7 or 8 bits
Transfer rate	Max. f_{MCK}/6 [Hz]Notes 1, 2.
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts from the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse
Data direction	MSB or LSB first

Notes 1. Because the external serial clock input to the SCK00, pin is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz].

2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C)** or **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T_A = −40 to +125°C)**).

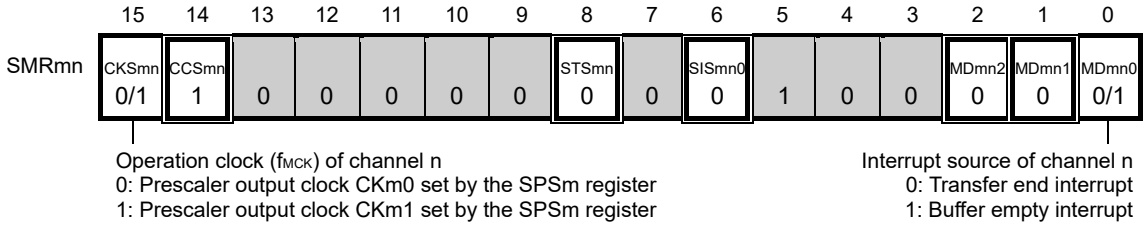
Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{SCK}: Serial clock frequency

2. m: Unit number (m = 0), n: Channel number (n = 0)

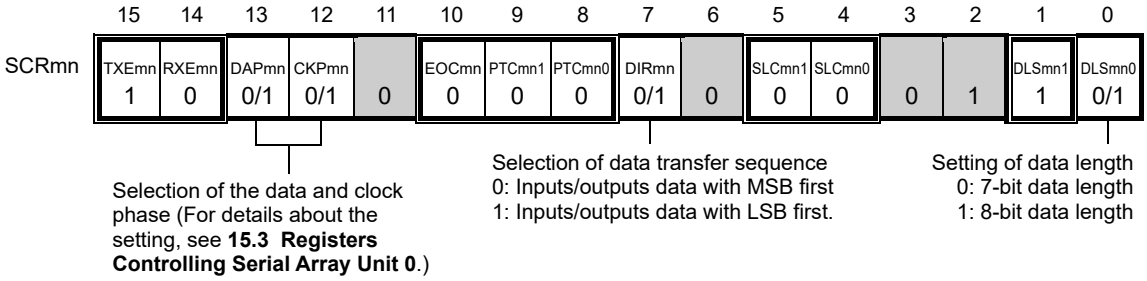
(1) Register setting

Figure 15-48. Example of Contents of Registers for Slave Transmission of Simplified SPI (CSI00) (1/2)

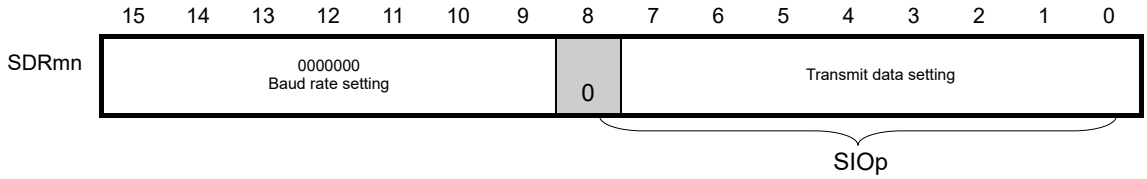
(a) Serial mode register mn (SMRmn)



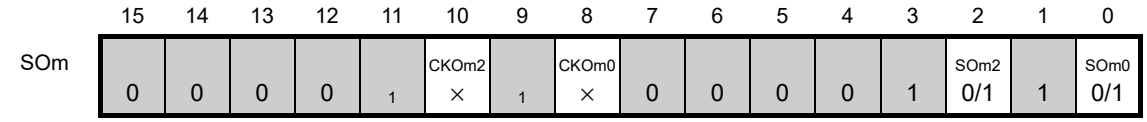
(b) Serial communication operation setting register mn (SCRmn)



(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



(d) Serial output register m (SOm) … Sets only the bits of the target channel.



Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

2. ☐: Setting is fixed in the Simplified SPI (CSI) slave transmission mode,

■: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

Figure 15-48. Example of Contents of Registers for Slave Transmission of Simplified SPI (CSI00) (2/2)

(e) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	0	SOEm0 0/1

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 0/1

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

2. ☐: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 15-49. Initial Setting Procedure for Slave Transmission

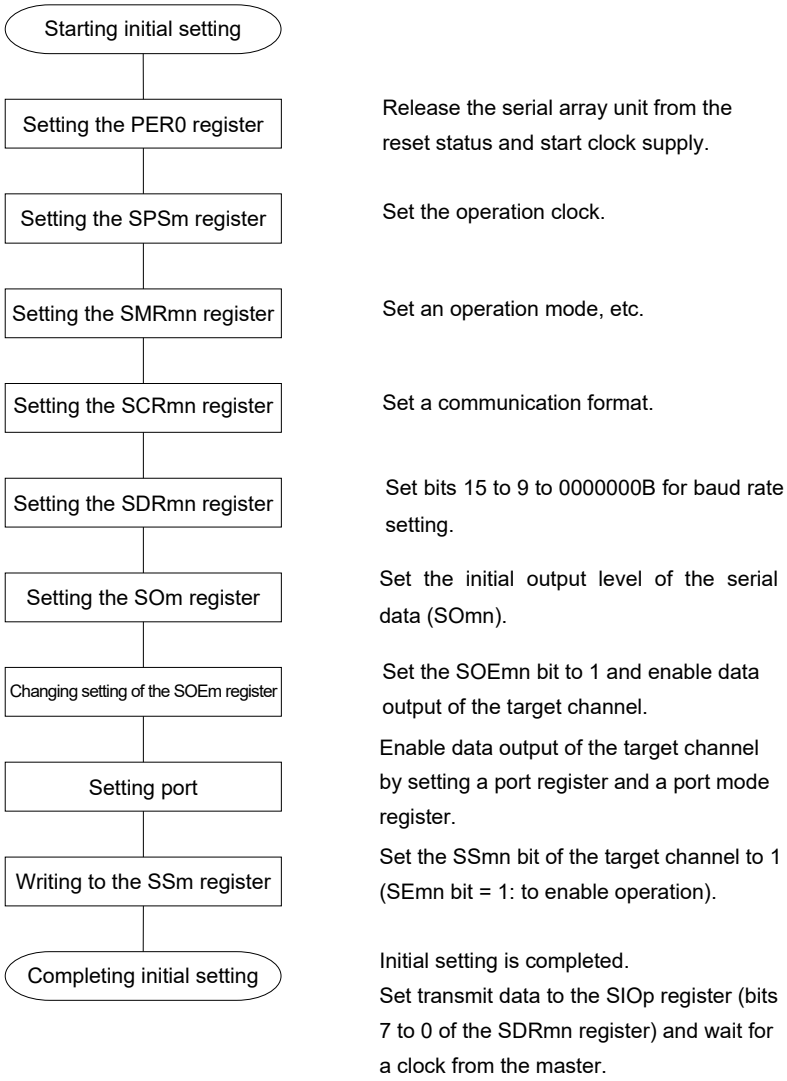


Figure 15-50. Procedure for Stopping Slave Transmission

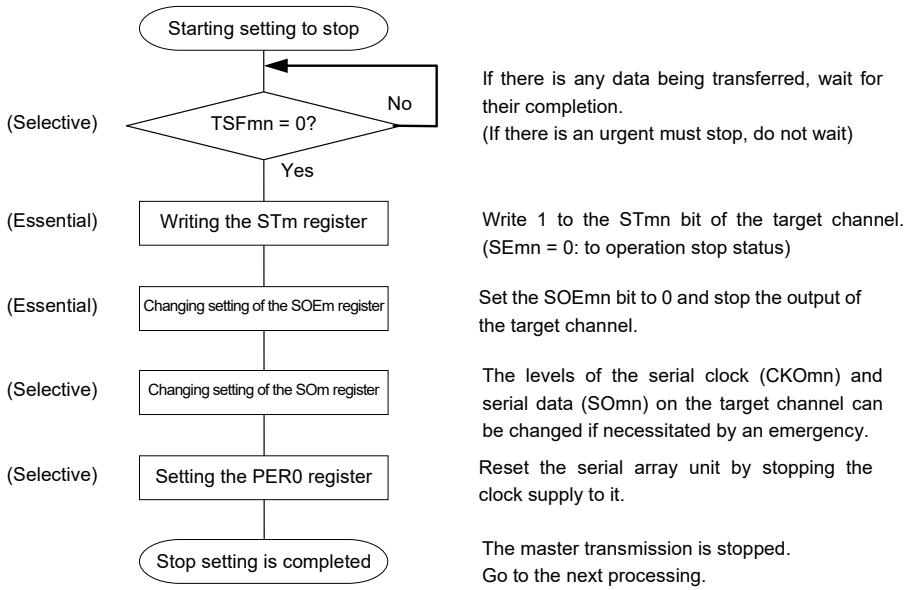
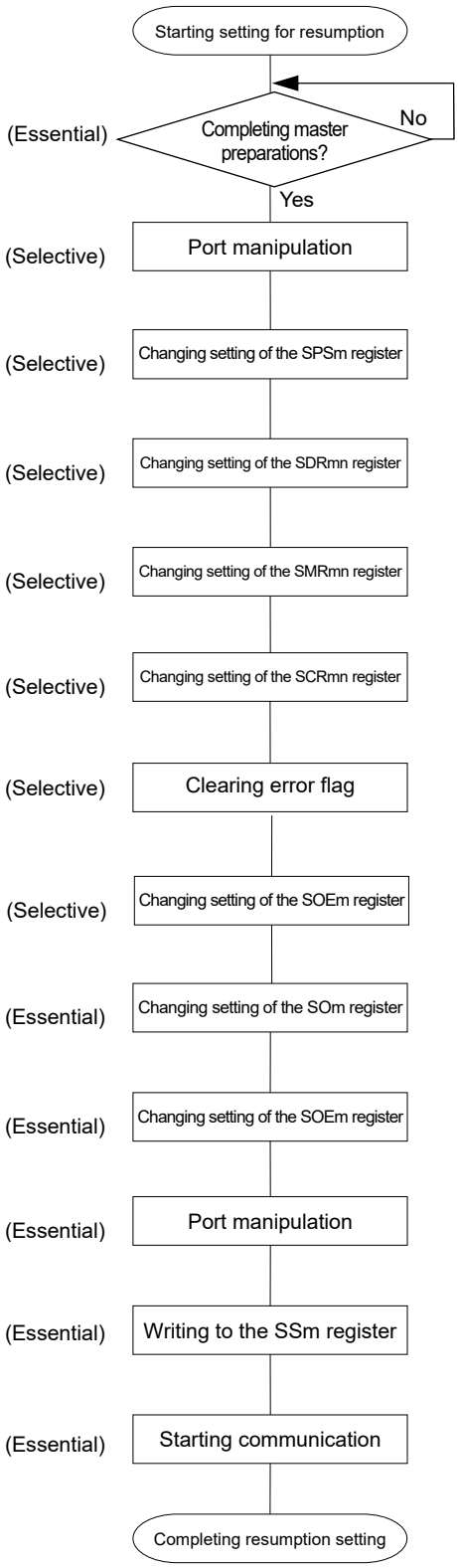


Figure 15-51. Procedure for Resuming Slave Transmission



Wait until stop the communication target (master)

Disable data output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

If the OVF flag remain set, clear this using serial flag clear trigger register mn (SIRmn).

Set the SOEmn bit to 0 to stop output from the target channel.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to 1 and enable output from the target channel.

Enable data output of the target channel by setting a port register and a port mode register.

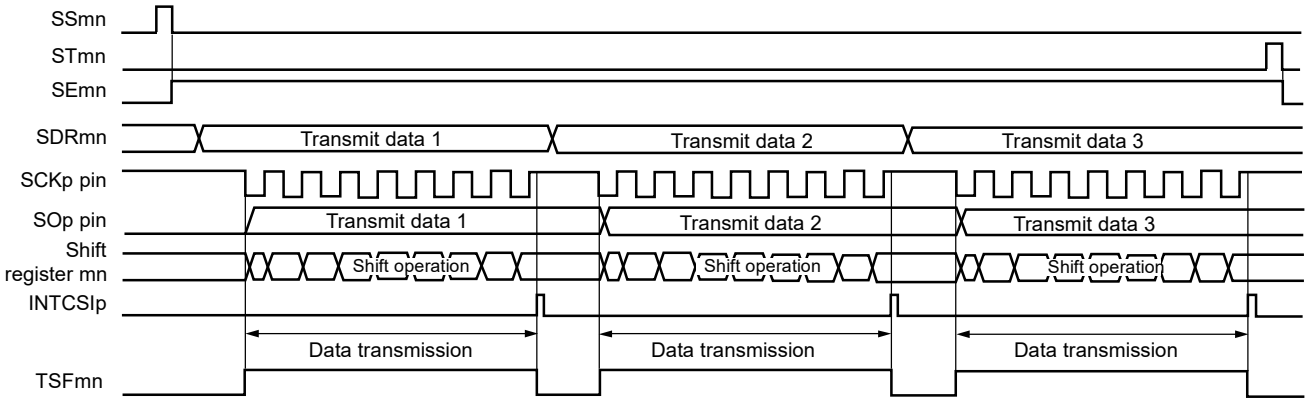
Set the SSmn bit of the target channel to 1 (SEmn = 1: to enable operation).

Sets transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and wait for a clock from the master.

Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

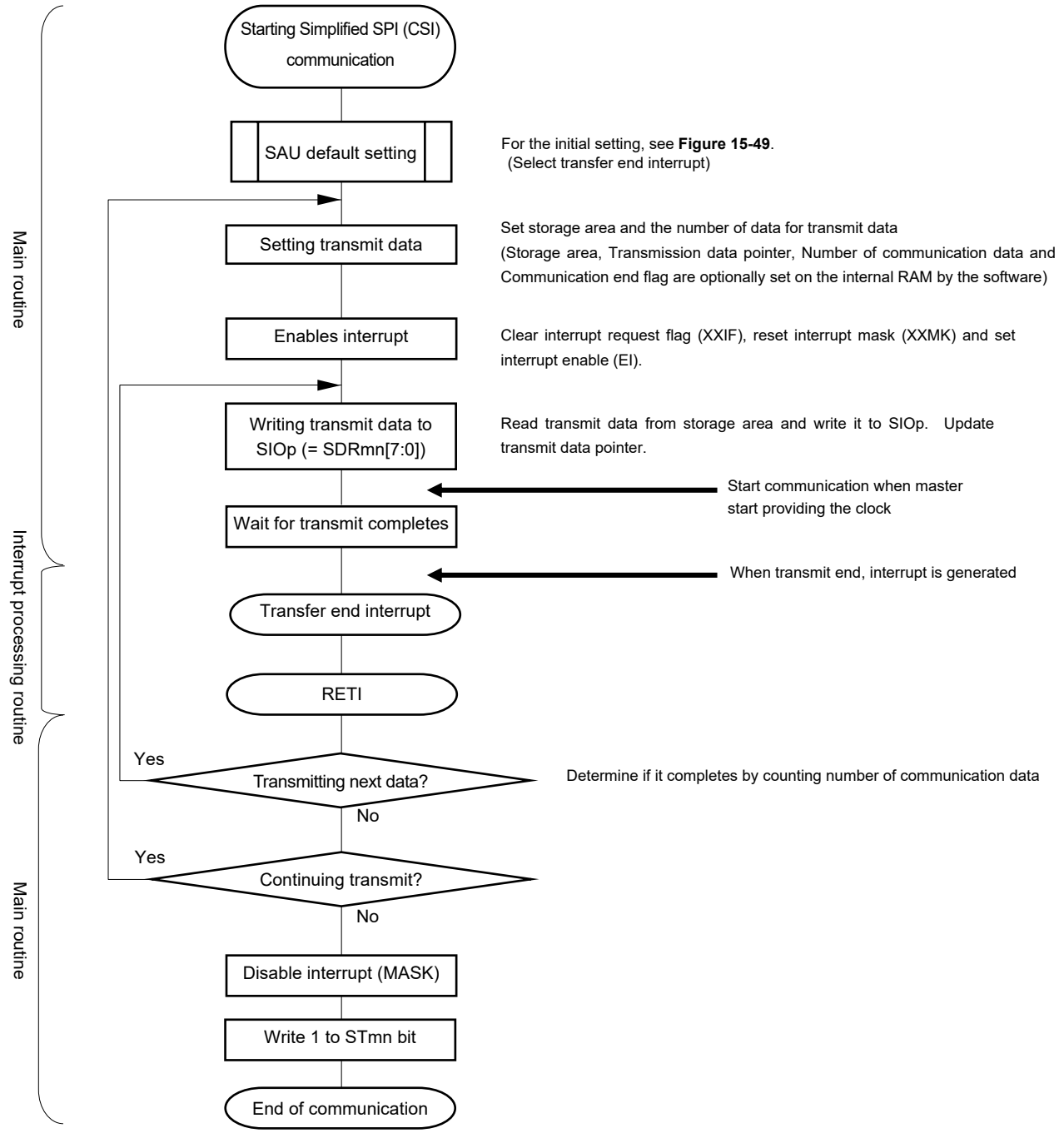
(3) Processing flow (in single-transmission mode)

Figure 15-52. Timing Chart of Slave Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



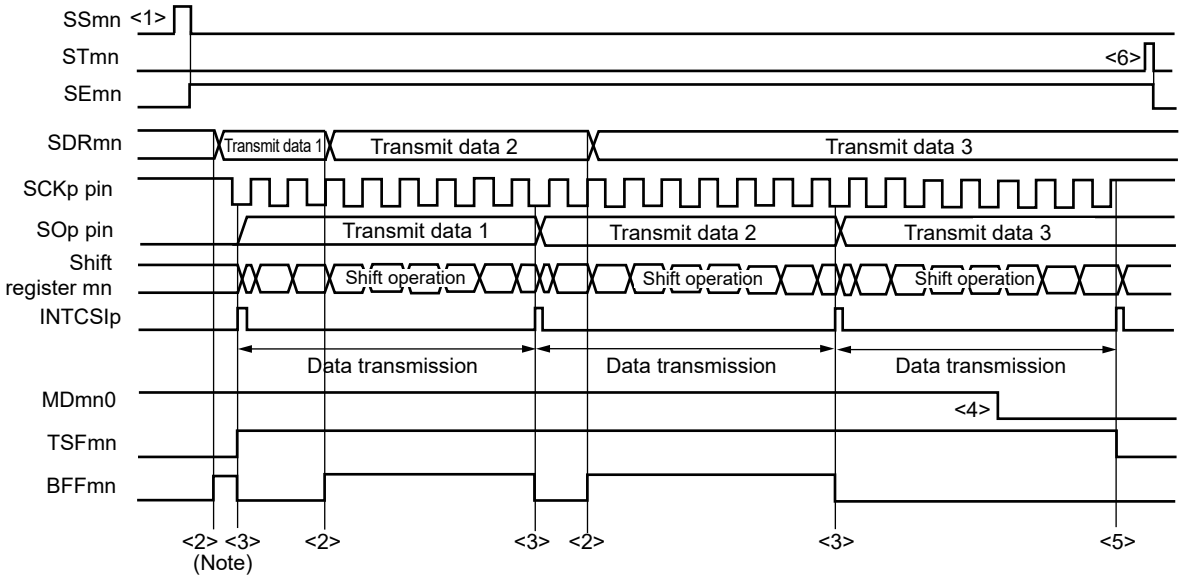
Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 15-53. Flowchart of Slave Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 15-54. Timing Chart of Slave Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

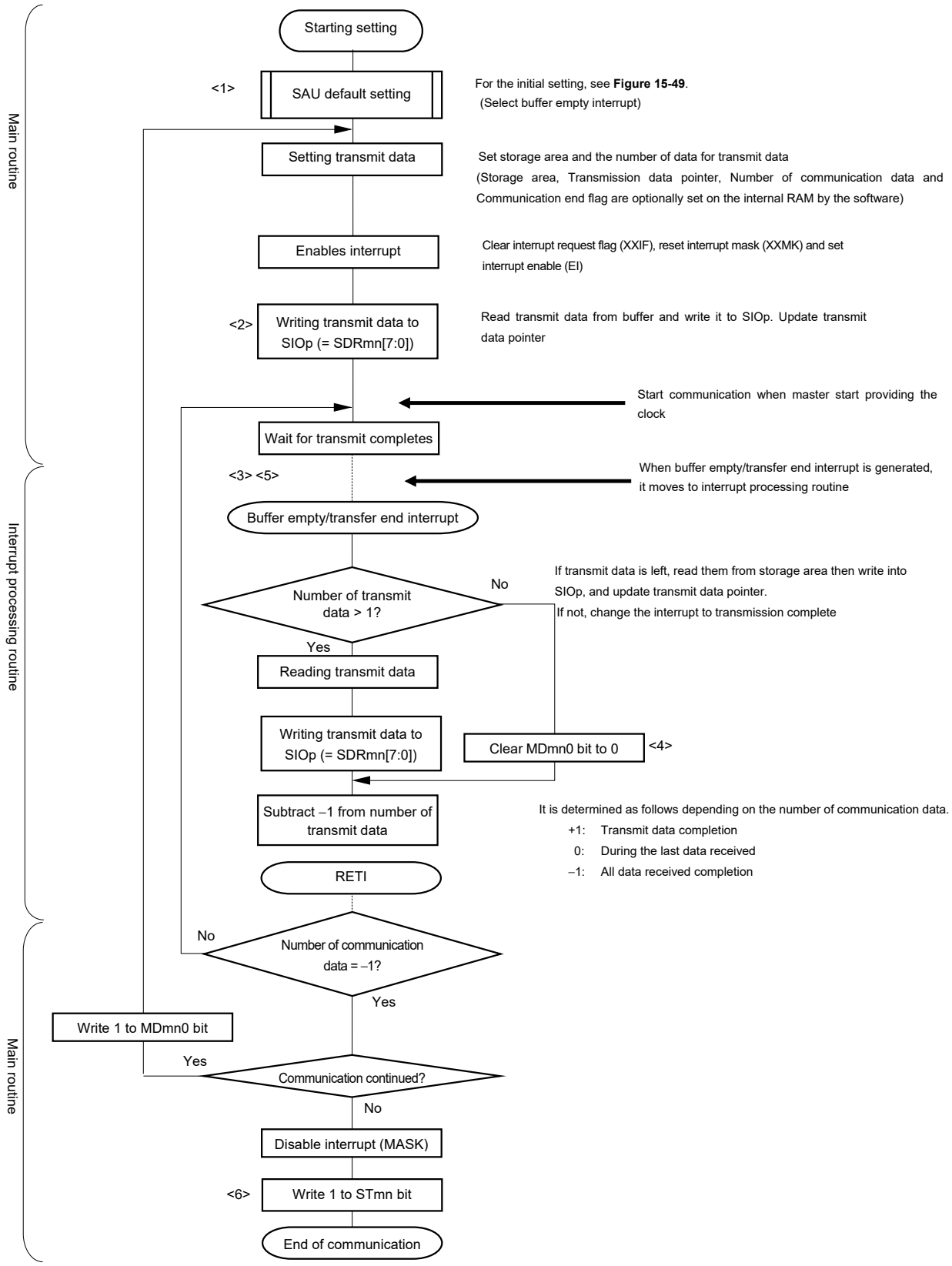


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 15-55. Flowchart of Slave Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 15-54 Timing Chart of Slave Transmission (in Continuous Transmission Mode)**.

15.5.5 Slave reception

Slave reception is that the RL78 microcontroller receives data from another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00
Target channel	Channel 0 of SAU0
Pins used	SCK00, SI00
Interrupt	INTCSI00
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	Overrun error detection flag (OVFmn) only
Transfer data length	7 or 8 bits
Transfer rate	Max. f_{MCK}/6 [Hz]Notes 1, 2
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data input starts from the start of the operation of the serial clock. • DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse
Data direction	MSB or LSB first

Notes 1. Because the external serial clock input to the SCK00 pin is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz].

2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C)** or **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T_A = −40 to +125°C)**).

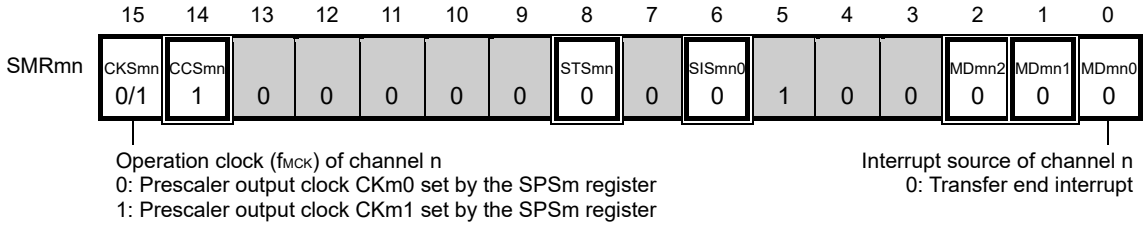
Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{SCK}: Serial clock frequency

2. m: Unit number (m = 0), n: Channel number (n = 0)

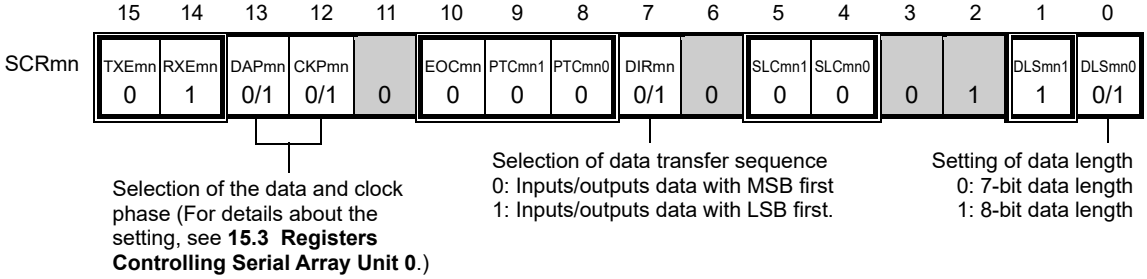
(1) Register setting

Figure 15-56. Example of Contents of Registers for Slave Reception of Simplified SPI (CSI00) (1/2)

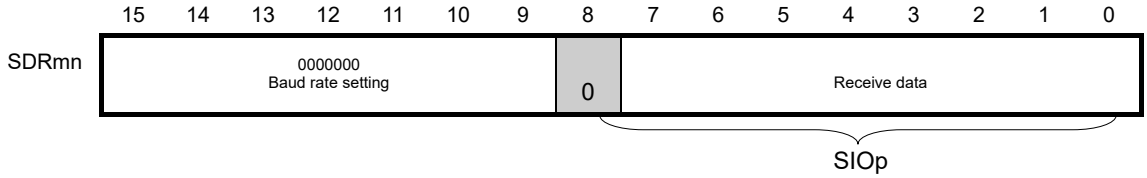
(a) Serial mode register mn (SMRmn)



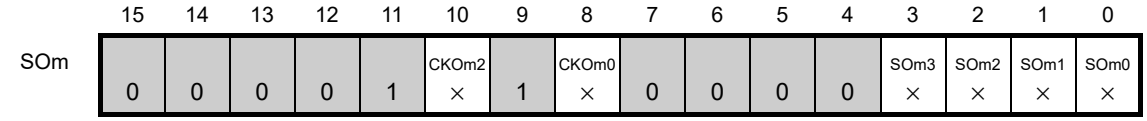
(b) Serial communication operation setting register mn (SCRmn)



(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



(d) Serial output register m (SOm) …The Register that not used in this mode.



Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

2. ☐: Setting is fixed in the Simplified SPI (CSI) slave transmission mode,

▒: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

Figure 15-56. Example of Contents of Registers for Slave Reception of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) (2/2)

(e) Serial output enable register m (SOEm) …The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	0	SOEm0 ×

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 0/1

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

2. ▒ : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 15-57. Initial Setting Procedure for Slave Reception

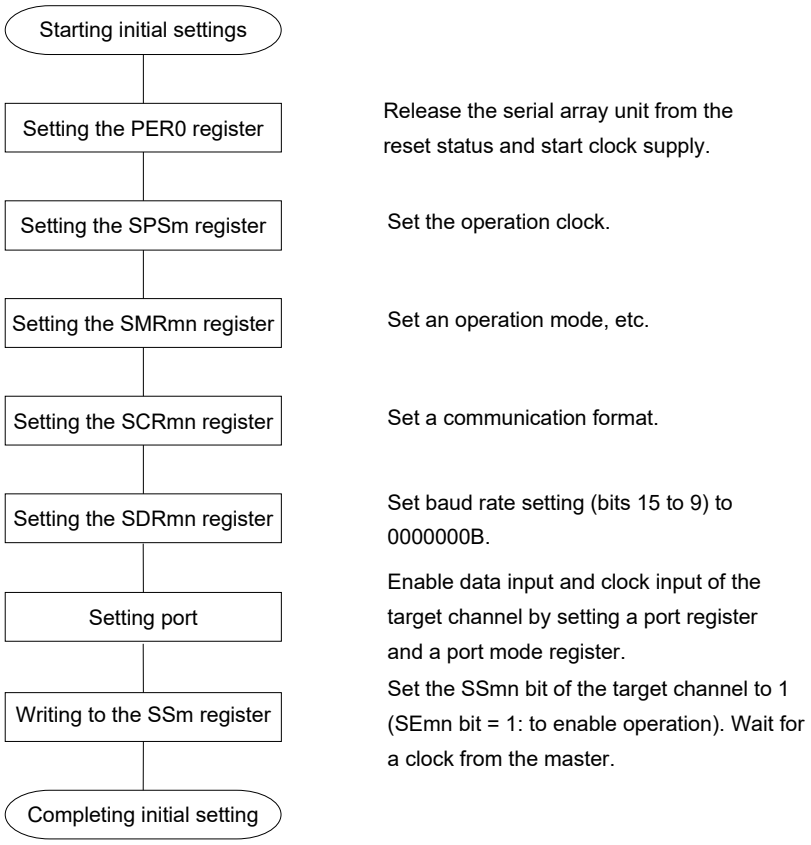


Figure 15-58. Procedure for Stopping Slave Reception

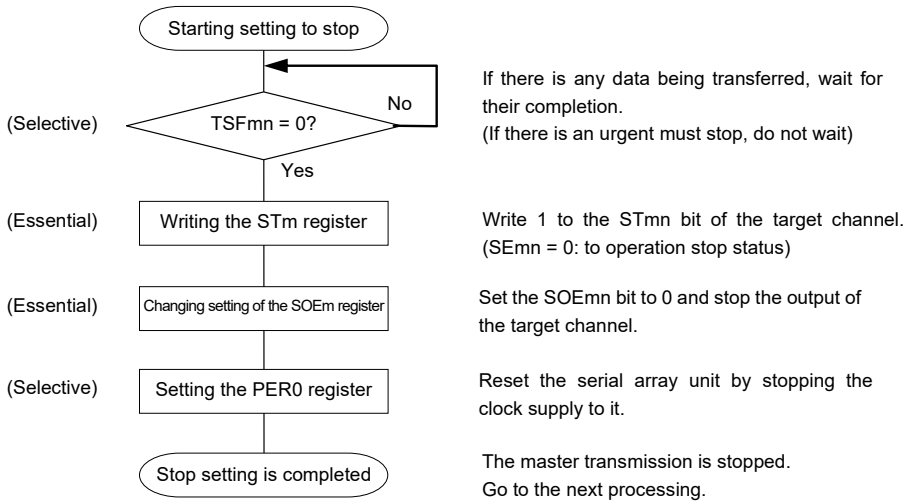
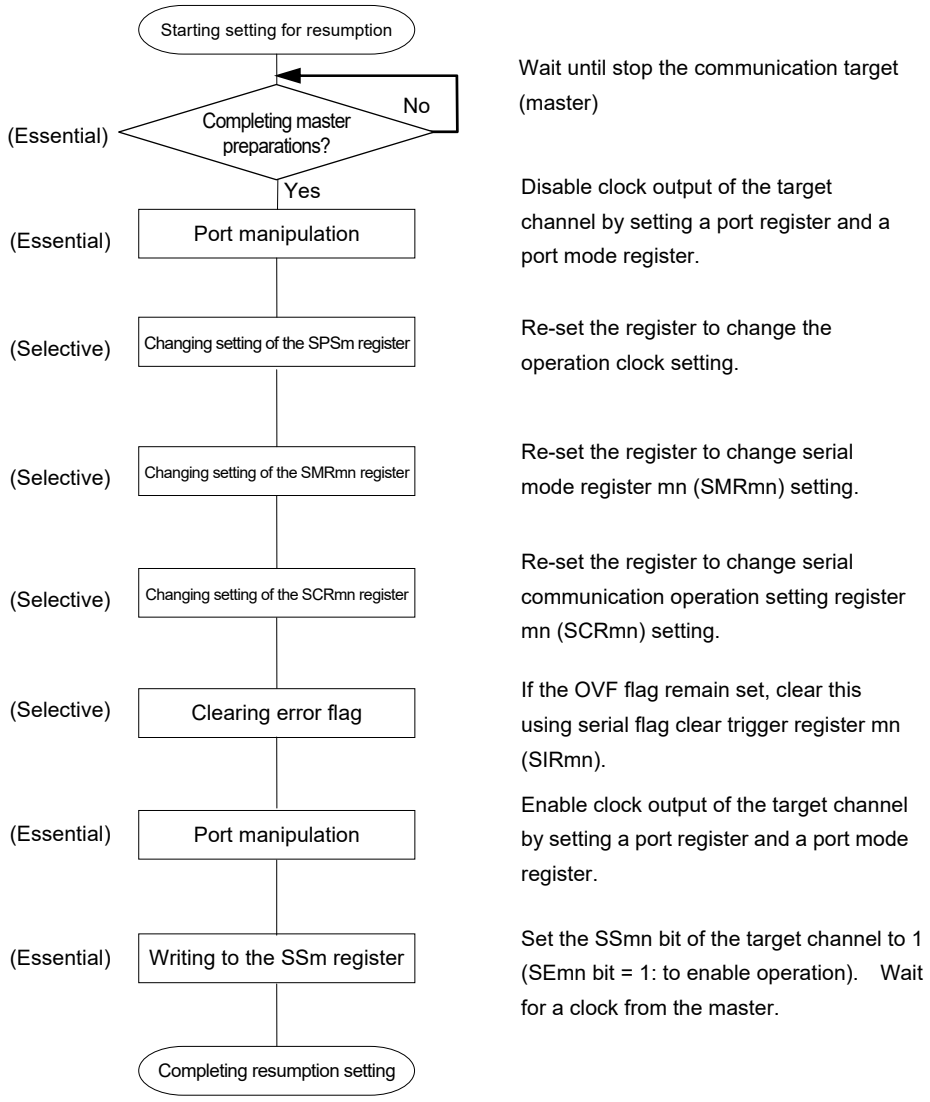


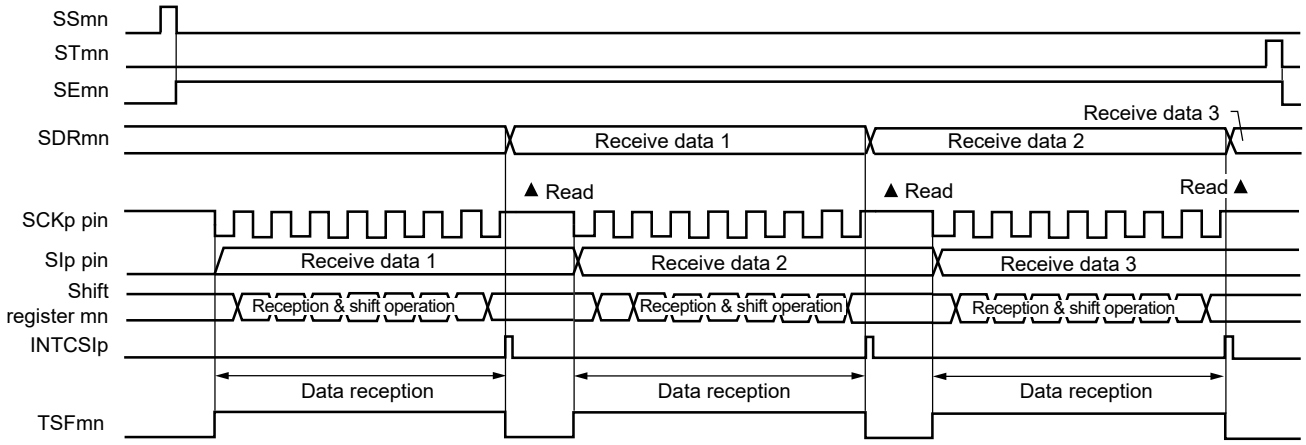
Figure 15-59. Procedure for Resuming Slave Reception



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

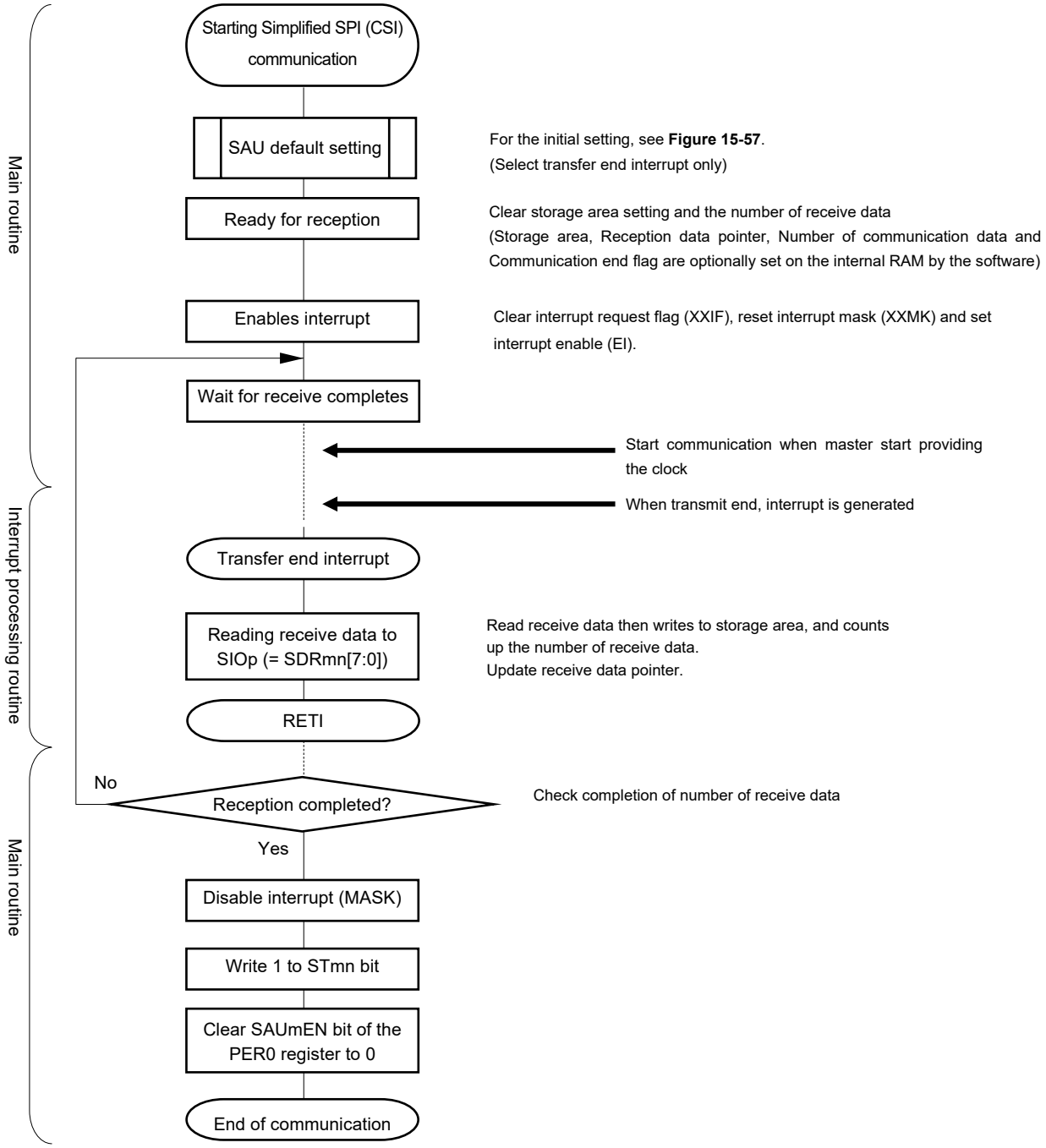
(3) Processing flow (in single-reception mode)

Figure 15-60. Timing Chart of Slave Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 15-61. Flowchart of Slave Reception (in Single-Reception Mode)



15.5.6 Slave transmission/reception

Slave transmission/reception is that the RL78 microcontroller transmits/receives data to/from another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00
Target channel	Channel 0 of SAU0
Pins used	SCK00, SI00, SO00
Interrupt	INTCSI00
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	Overrun error detection flag (OVFmn) only
Transfer data length	7 or 8 bits
Transfer rate	Max. f_{MCK}/6 [Hz]Notes 1, 2.
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data I/O starts from the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse
Data direction	MSB or LSB first

Notes 1. Because the external serial clock input to the SCK00 pin is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz].

2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C)** or **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T_A = −40 to +125°C)**).

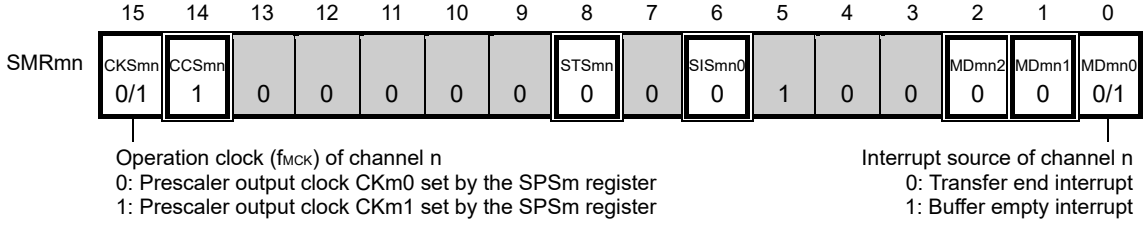
Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: Serial clock frequency

2. m: Unit number (m = 0), n: Channel number (n = 0)

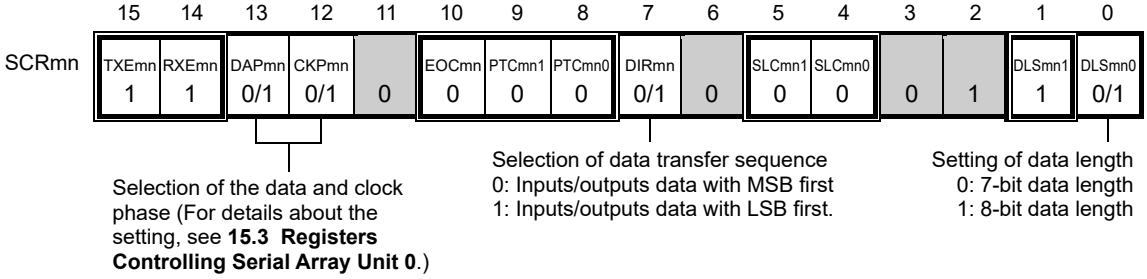
(1) Register setting

Figure 15-62. Example of Contents of Registers for Slave Transmission/Reception of Simplified SPI (CSI00) (1/2)

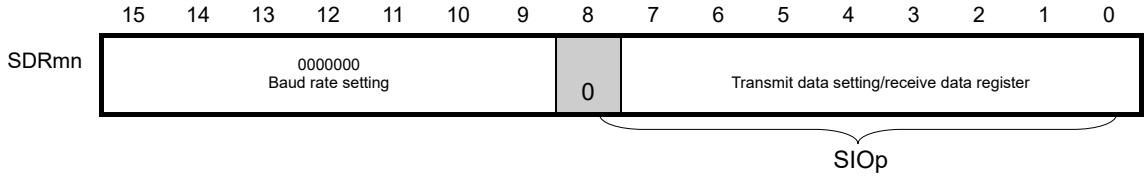
(a) Serial mode register mn (SMRmn)



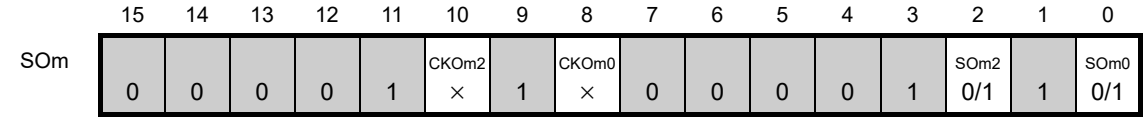
(b) Serial communication operation setting register mn (SCRmn)



(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



(d) Serial output register m (SOm) … Sets only the bits of the target channel.



Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

2. ☐: Setting is fixed in the Simplified SPI (CSI) slave transmission/reception mode,

■: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

Figure 15-62. Example of Contents of Registers for Slave Transmission/Reception of Simplified SPI (CSI00) (2/2)

(e) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	15	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	0	SOEm0 0/1

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	15	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 0/1

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

2. ☐ : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 15-63. Initial Setting Procedure for Slave Transmission/Reception

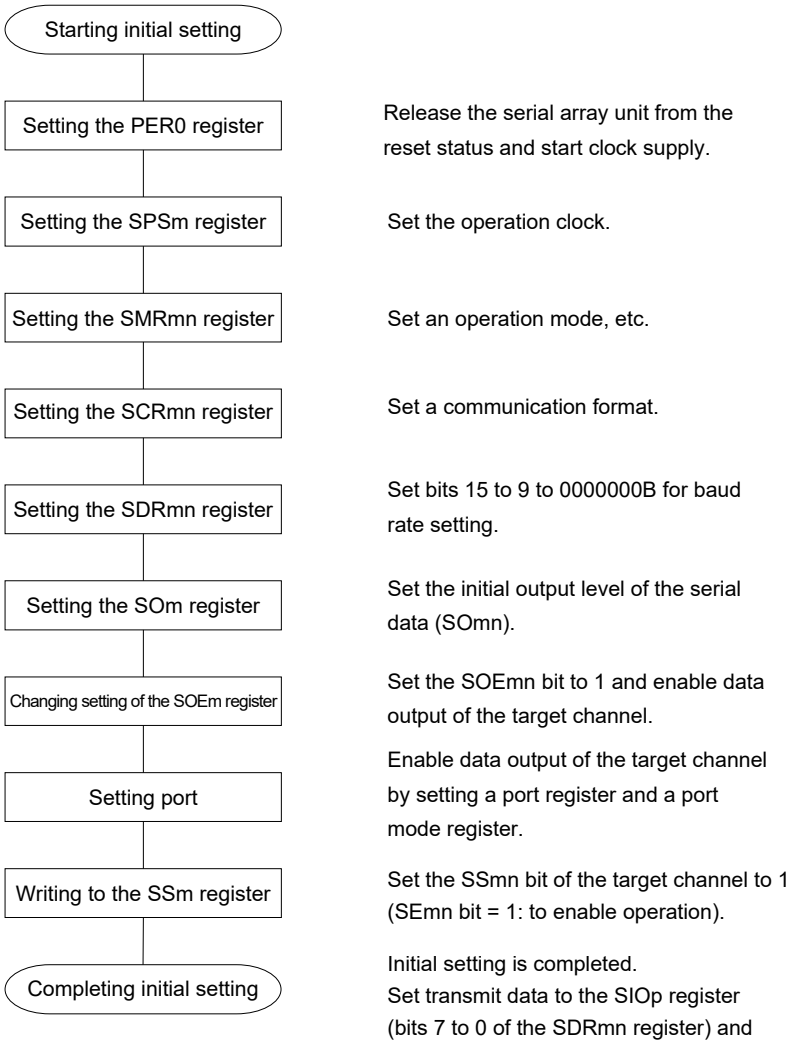


Figure 15-64. Procedure for Stopping Slave Transmission/Reception

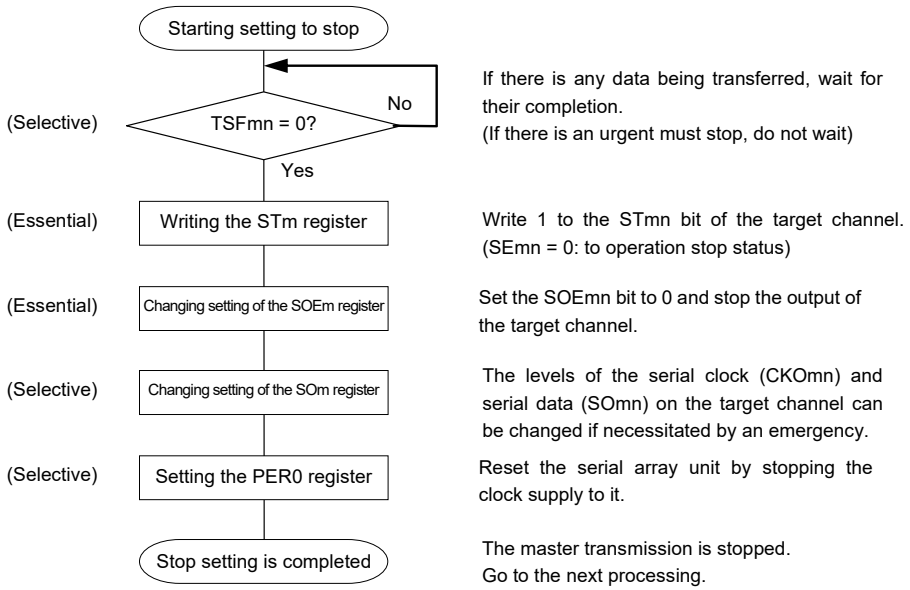
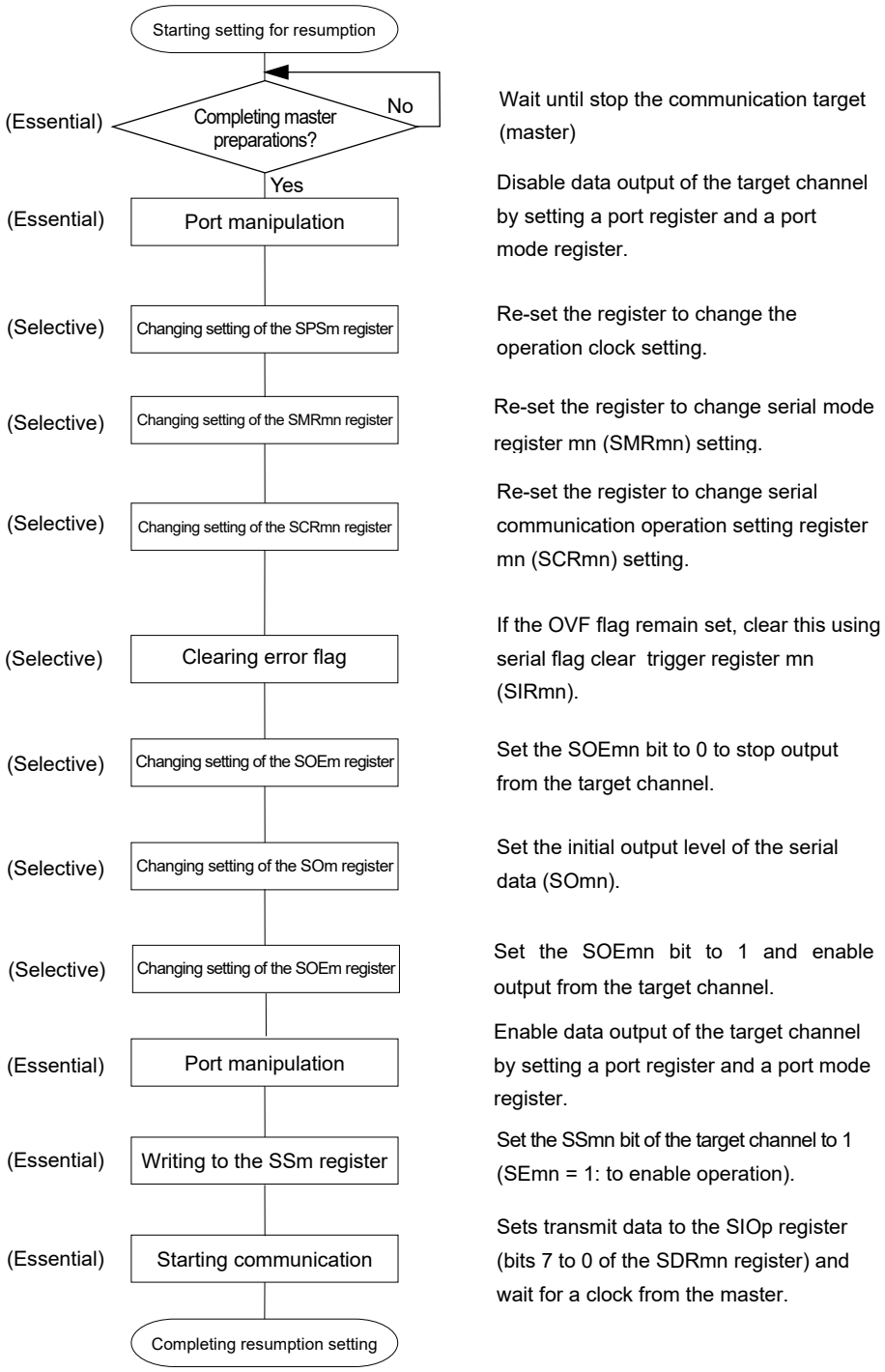


Figure 15-65. Procedure for Resuming Slave Transmission/Reception

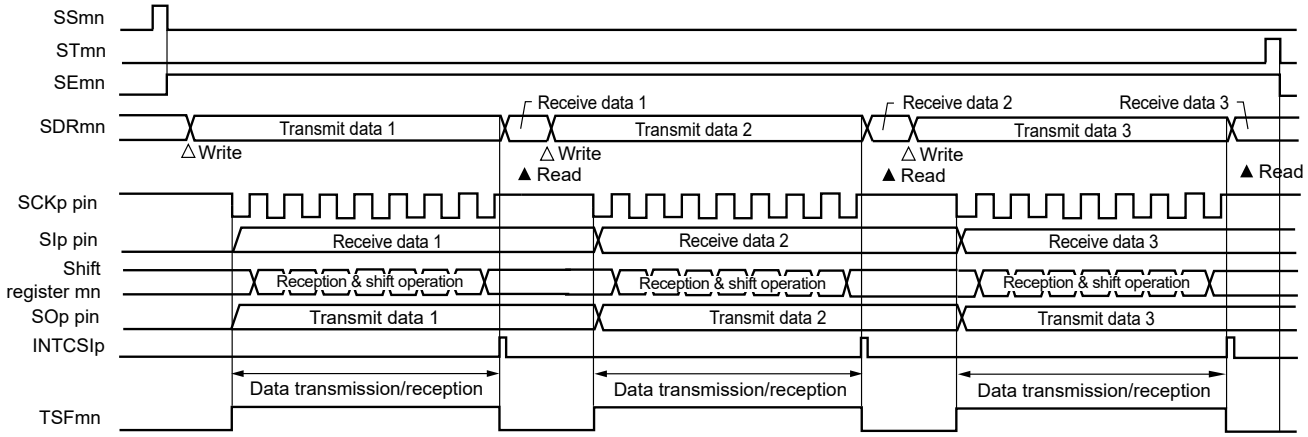


Cautions 1. Be sure to set transmit data to the SIOp register before the clock from the master is started.

2. If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

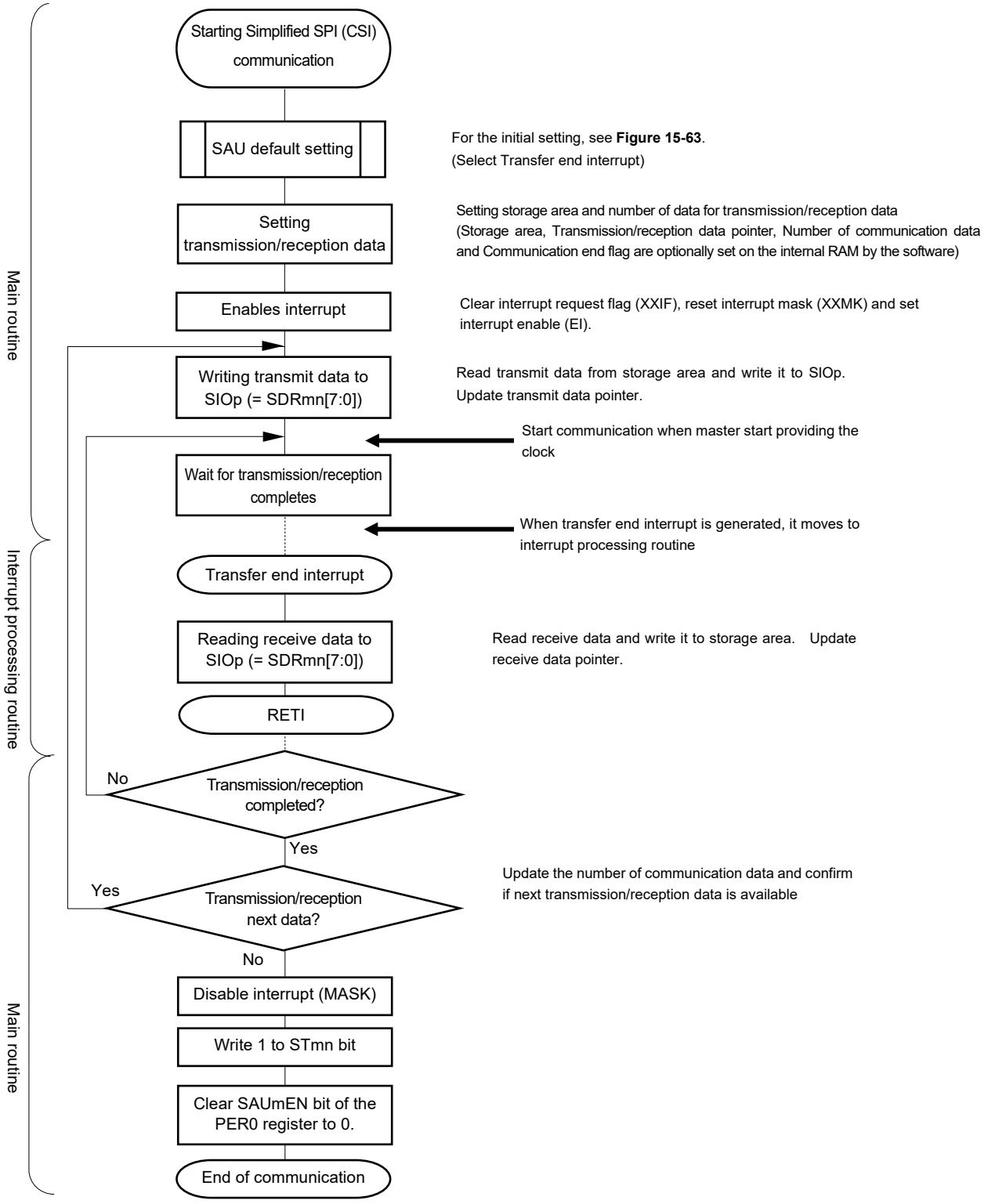
(3) Processing flow (in single-transmission/reception mode)

Figure 15-66. Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

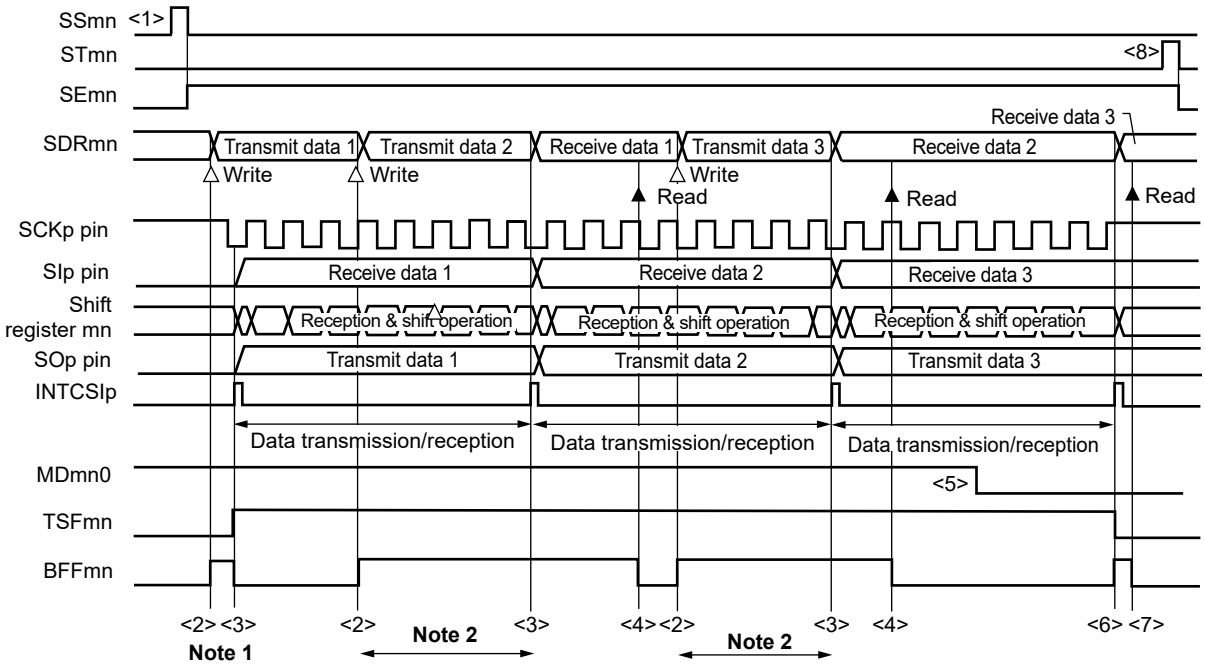
Figure 15-67. Flowchart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)



Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

(4) Processing flow (in continuous transmission/reception mode)

Figure 15-68. Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Notes 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.

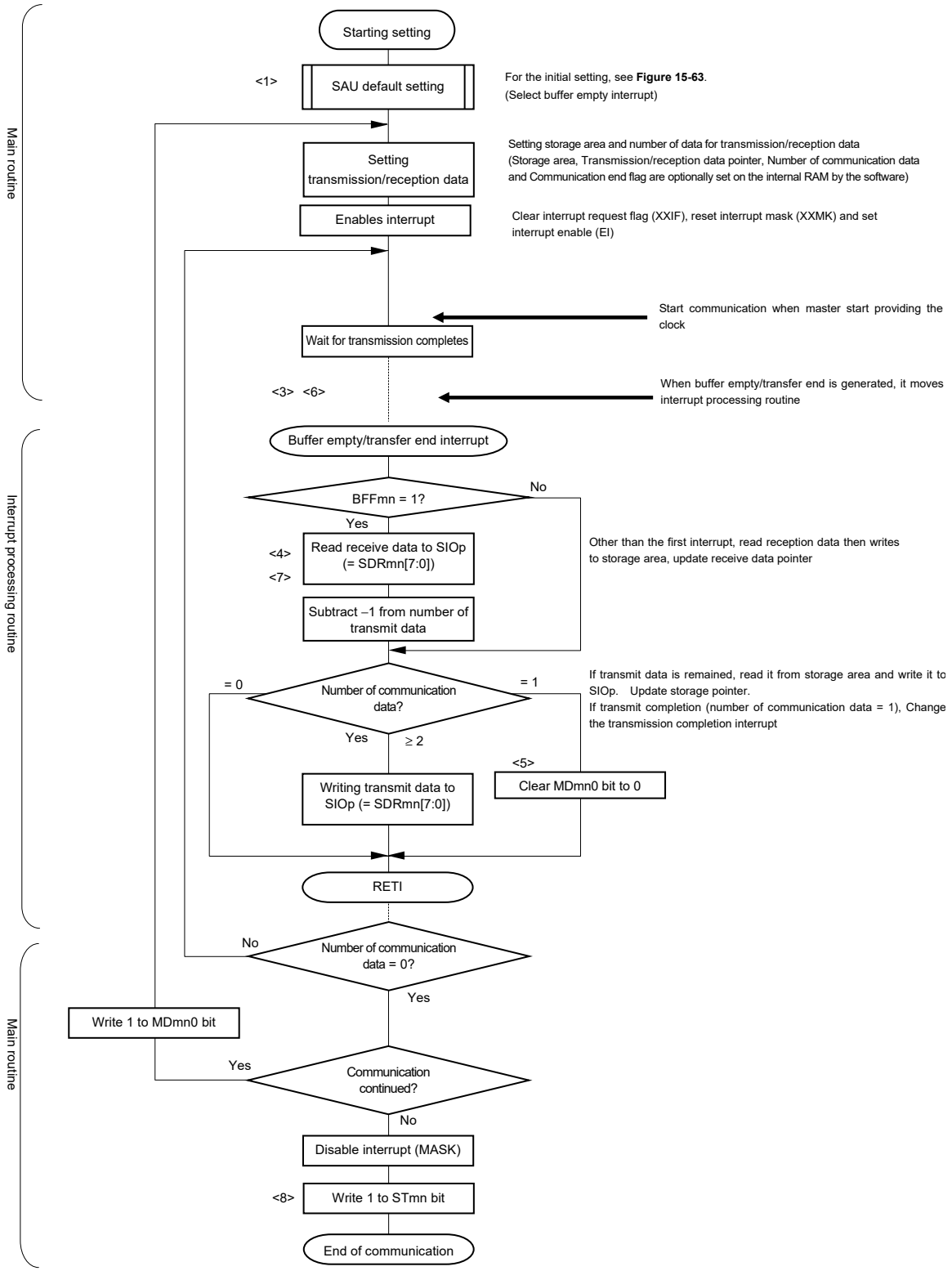
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

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.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 15-69 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 15-69. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 15-68 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

15.5.7 SNOOZE mode function

SNOOZE mode makes Simplified SPI (CSI) operate reception by SCKp pin input detection while the STOP mode. Normally Simplified SPI (CSI) stops communication in the STOP mode. But, using the SNOOZE mode makes reception Simplified SPI (CSI) operate unless the CPU operation by detecting 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 15-71 Flowchart of SNOOZE Mode Operation (Once Startup)** and **Figure 15-73 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 just 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.

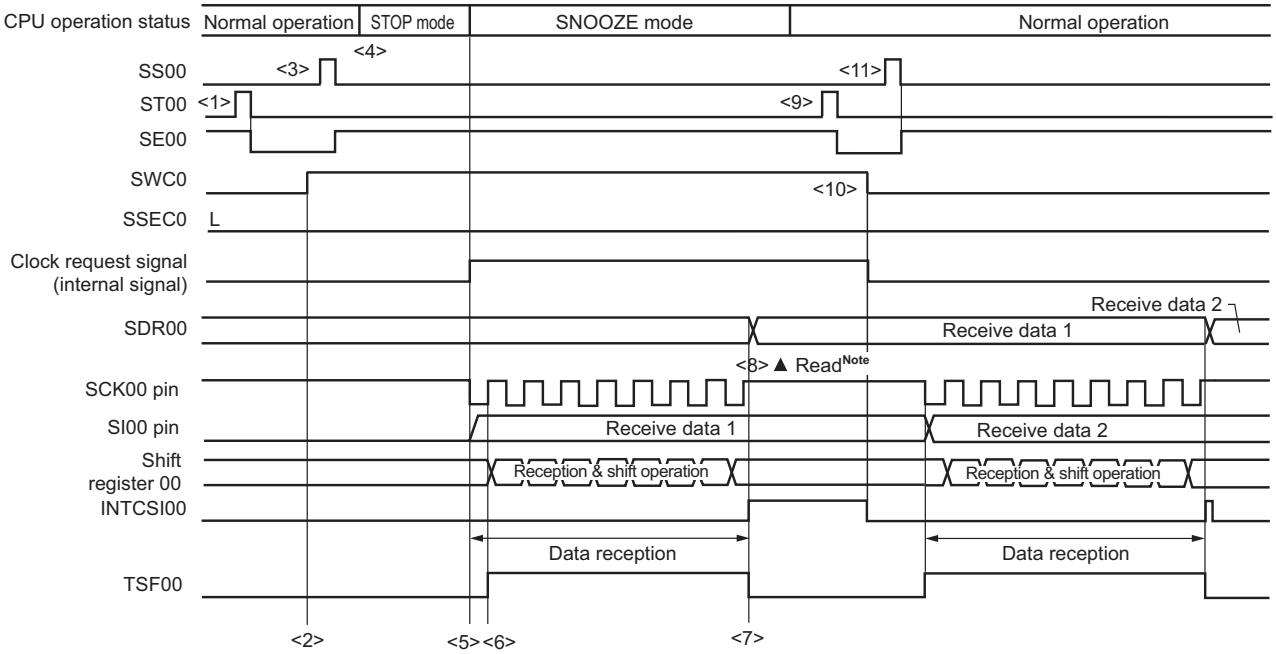
After a transition to the STOP mode, is switched to the SNOOZE mode upon detection of a valid edge of the SCKp pin. The Simplified SPI (CSI) starts reception operations after input to a serial clock of the SCKp.

Cautions 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for f_{CLK}.

2. The maximum transfer rate when using CSIp in the SNOOZE mode is 1 Mbps.

(1) SNOOZE mode operation (once startup)

Figure 15-70. 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 edge of the SCKp pin input is detected.

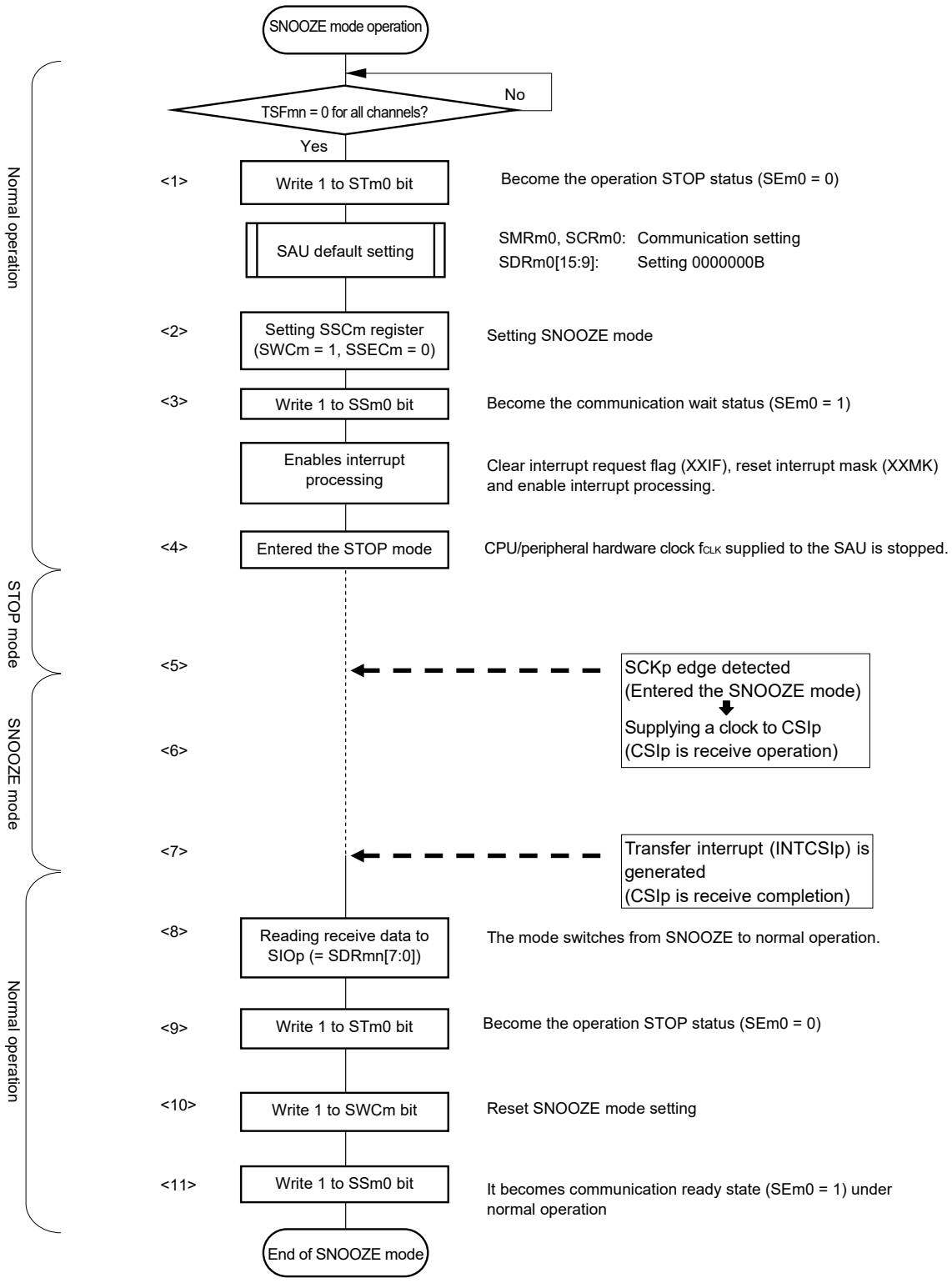
Cautions 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (clear the SEm0 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

2. When SWCm = 1, the BFFm1 and OVFm1 flags will not change.

Remarks 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 15-71 Flowchart of SNOOZE Mode Operation (Once Startup)**.

2. m: Unit number (m = 0), p: CSI number (p = 00)

Figure 15-71. Flowchart of SNOOZE Mode Operation (Once Startup)

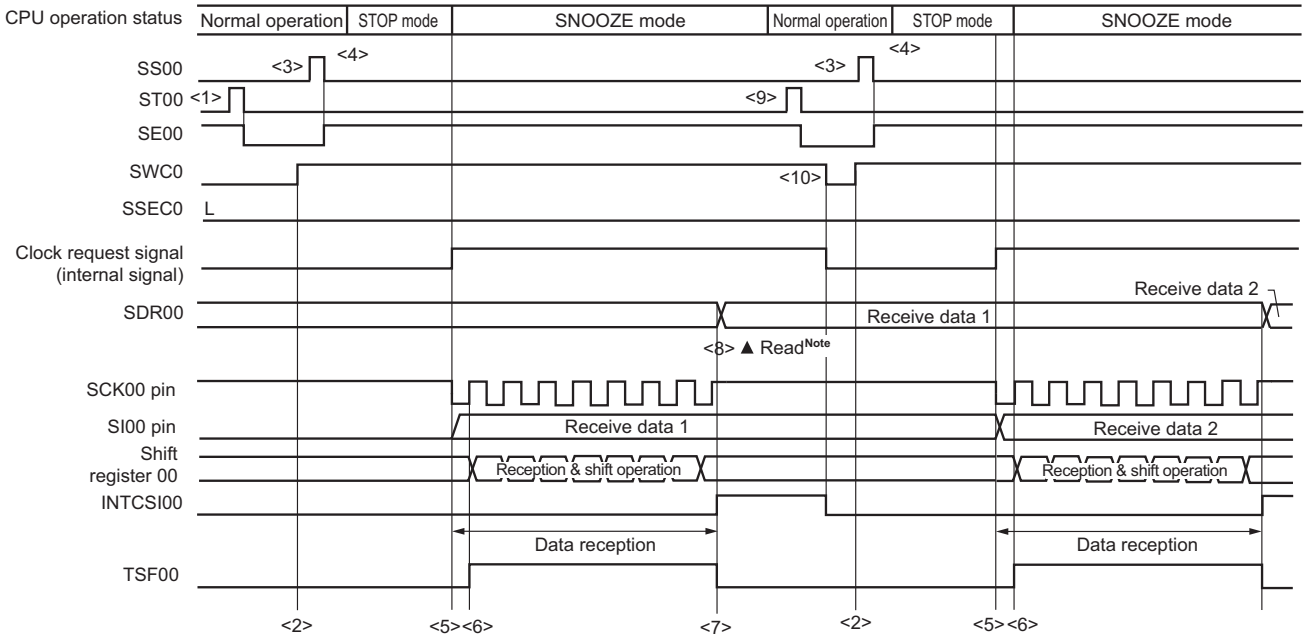


Remarks 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 15-70 Timing Chart of SNOOZE Mode Operation (Once Startup)**.

2. m: Unit number (m = 0), p: CSI number (p = 00)

(2) SNOOZE mode operation (continuous startup)

Figure 15-72. Timing Chart of SNOOZE Mode Operation (Continuous Startup) (Type 1: DAPmn = 0, CKPmn = 0)

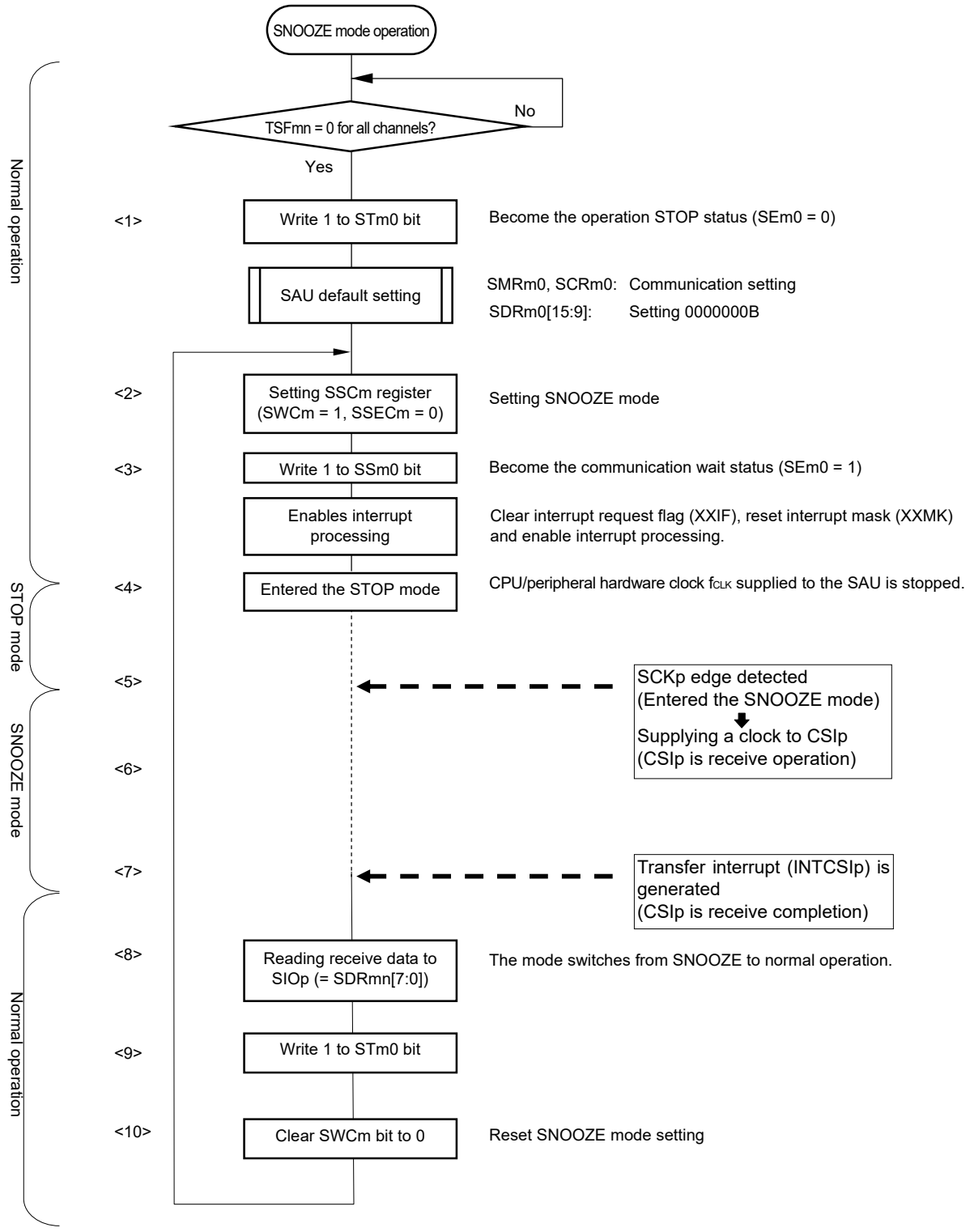


Note Only read received data while SWCm = 1 and before the next edge of the SCKp pin input is detected.

Cautions 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (clear the SEm0 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE release).
2. When SWCm = 1, the BFFm1 and OVFm1 flags will not change.

Remarks 1. <1> to <10> in the figure correspond to <1> to <10> in **Figure 15-73 Flowchart of SNOOZE Mode Operation (Continuous Startup)**.
2. m: Unit number (m = 0), p: CSI number (p = 00)

Figure 15-73. Flowchart of SNOOZE Mode Operation (Continuous Startup)



Remarks 1. <1> to <10> in the figure correspond to <1> to <10> in **Figure 15-72 Timing Chart of SNOOZE Mode Operation (Continuous Startup)**.

2. m: Unit number (m = 0), p: CSI number (p = 00)

15.5.8 Calculating transfer clock frequency

The transfer clock frequency for Simplified SPI (CSI00) communication can be calculated by the following expressions.

(1) Master

(Transfer clock frequency) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [Hz]

(2) Slave

(Transfer clock frequency) = {Frequency of serial clock (SCK) supplied by master}[Note] [Hz]

Note The permissible maximum transfer clock frequency is f_{MCK}/6.

Remark The value of SDRmn[15:9] is the value of bits 15 to 9 of serial data register mn (SDRmn) (0000000B to 1111111B) and therefore is 0 to 127.

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 15-2. Selection of Operation Clock For Simplified SPI

SMRmn Register	SPSm Register								Operation Clock (f_{MCK})[Note]	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 32 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	32 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	16 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	8 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	4 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	2 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	1 MHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	500 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	250 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	125 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz
	X	X	X	X	1	1	0	0	$f_{CLK}/2^{12}$	7.81 kHz
	X	X	X	X	1	1	0	1	$f_{CLK}/2^{13}$	3.91 kHz
	X	X	X	X	1	1	1	0	$f_{CLK}/2^{14}$	1.95 kHz
	X	X	X	X	1	1	1	1	$f_{CLK}/2^{15}$	977 Hz
1	0	0	0	0	X	X	X	X	f_{CLK}	32 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	16 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	8 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	4 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	2 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	1 MHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	500 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	250 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	125 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	62.5 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	31.25 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	15.63 kHz
	1	1	0	0	X	X	X	X	$f_{CLK}/2^{12}$	7.81 kHz
	1	1	0	1	X	X	X	X	$f_{CLK}/2^{13}$	3.91 kHz
	1	1	1	0	X	X	X	X	$f_{CLK}/2^{14}$	1.95 kHz
	1	1	1	1	X	X	X	X	$f_{CLK}/2^{15}$	977 Hz
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).

Remarks 1. X: Don't care

2. m: Unit number (m = 0), n: Channel number (n = 0)

15.5.9 Procedure for processing errors that occurred during Simplified SPI (CSI00) communication

The procedure for processing errors that occurred during Simplified SPI (CSI00) communication is described in Figure 15-74.

Figure 15-74. Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Remark m: Unit number (m = 0), n: Channel number (n = 0)

15.6 Operation of UART (UART0, UART1) Communication

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex asynchronous communication UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel). The LIN-bus and DMX512 can be implemented by using timer array unit 0 (channel 7) with an external interrupt (INTP0).

[Data transmission/reception]

- Data length of 7, 8, or 9 bits[Note]
- Select the MSB/LSB first
- 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, UART0 reception (channel 1) supports the SNOOZE mode. When RxD0 pin input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible.

In addition, the LIN-bus and DMX512 are accepted in UART0 (channels 0 and 1).

[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 0 (channel 7)

[DMX512 functions]

- BREAK signal detection
- Pulse width detection

} Using the external interrupt (INTP0) and timer array unit 0 (channel 7)

Note Only UART0 can be specified for the 9-bit data length.

UART0 uses channels 0 and 1.
UART1 uses channels 2 and 3.

- 20-pin products

Channel	Used as Simplified SPI (CSI)	Used as UART
0	–	UART0 (supporting LIN-bus, DMX512)
1	–	
2	–	–
3	–	

- 30-pin products

Channel	Used as Simplified SPI (CSI)	Used as UART
0	–	UART0 (supporting LIN-bus, DMX512)
1	–	
2	–	UART1
3	–	

- 38-pin products

Channel	Used as Simplified SPI (CSI)	Used as UART
0	CSI00	UART0 (supporting LIN-bus, DMX512)
1	–	
2	–	UART1
3	–	

Select any function for each channel. Only the selected function is possible. If UART0 is selected for channels 0 and 1 of unit 0, for example, these channels cannot be used for CSI00. At this time, however, channel 2, 3, or other channels of the same unit can be used for a function other than UART0, such as CSI10, UART1, and IIC10.

Caution When using serial array unit as UARTs, the channels of both the transmitting side (even-number channel) and the receiving side (odd-number channel) can be used only as UARTs.

UART performs the following five types of communication operations.

- UART transmission (See **15.6.1**.)
- UART reception (See **15.6.2**.)
- LIN transmission (UART0 only) (See **15.7.1**.)
- LIN reception (UART0 only) (See **15.7.2**.)
- DMX512 reception (UART0 only) (See **15.8**.)

15.6.1 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
Target channel	Channel 0 of SAU0	Channel 2 of SAU0
Pins used	TxD0	TxD1
Interrupt	INTST0	INTST1
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	None	
Transfer data length	7, 8, or 9 bits Note 1	
Transfer rate 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 bits	
Data direction	MSB or LSB first	

Notes 1. Only UART0 can be specified for the 9-bit data length.

2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C)** or **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T_A = −40 to +125°C)**).

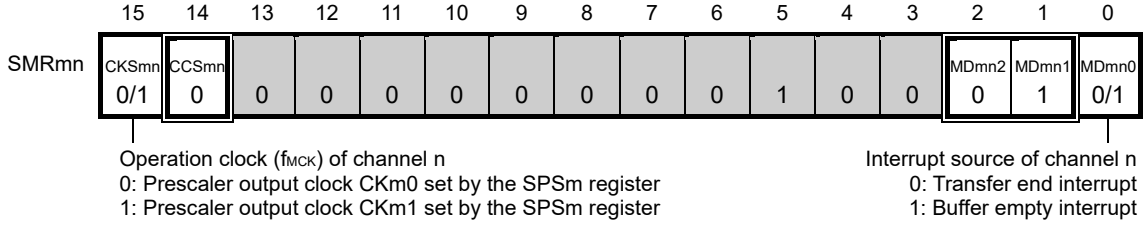
Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

2. m: Unit number (m = 0), n: Channel number (n = 0, 2), mn = 00, 02

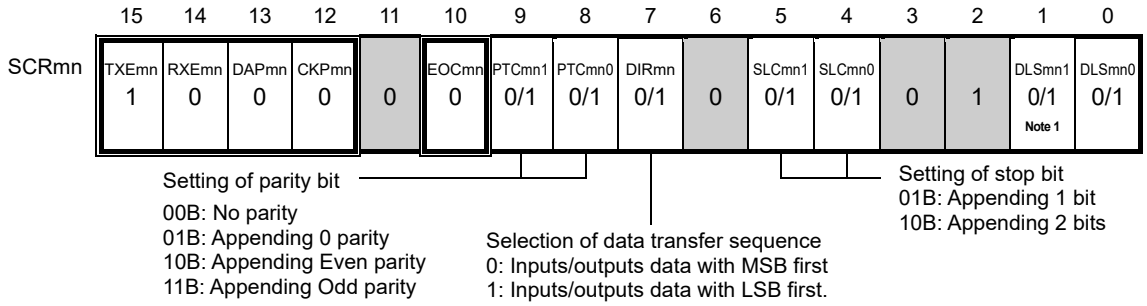
(1) Register setting

Figure 15-75. Example of Contents of Registers for UART Transmission of UART (UART0, UART1) (1/2)

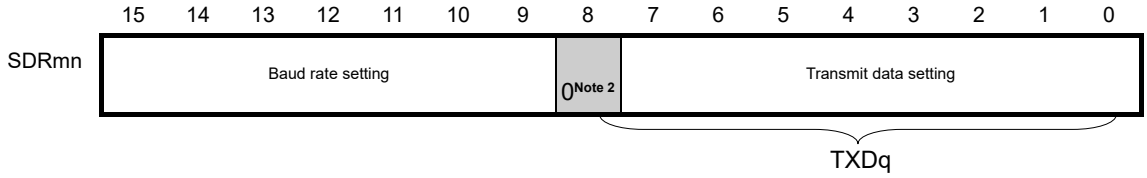
(a) Serial mode register mn (SMRmn)



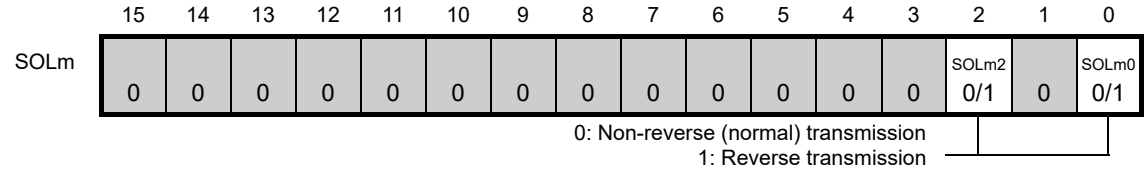
(b) Serial communication operation setting register mn (SCRmn)



(c) Serial data register mn (SDRmn) (lower 8 bits: TXDq)



(d) Serial output level register m (SOLm) … Sets only the bits of the target channel.



Notes 1. The SCR00 register only. This bit is fixed to 1 for the SCR02 register.

2. When UART0 performs 9-bit communication (by setting the DLS001 and DLS000 bits of the SCR00 register to 1), bits 0 to 8 of the SDR00 register are used as the transmission data specification area. Only UART0 can be specified for the 9-bit data length..

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 2), q: UART number (q = 0, 1), mn = 00, 02

2. ☐: Setting is fixed in the UART transmission mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 15-75. Example of Contents of Registers for UART Transmission of UART (UART0, UART1) (2/2)

(e) Serial output register m (SOm) … Sets only the bits of the target channel.

SOm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	CKOm2 ×	1	CKOm0 ×	0	0	0	0	1	SOm2 0/1 Note	1	SOm0 0/1 Note

0: Serial data output value is “0”
1: Serial data output value is “1”

(f) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

SOEm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	0	SOEm0 0/1

(g) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

SSm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

Note Before transmission is started, be sure to set to 1 when the SOLmn bit of the target channel is set to 0, and set to 0 when the SOLmn bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 2), q: UART number (q = 0, 1), mn = 00, 02

2. ☐ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 15-76. Initial Setting Procedure for UART Transmission

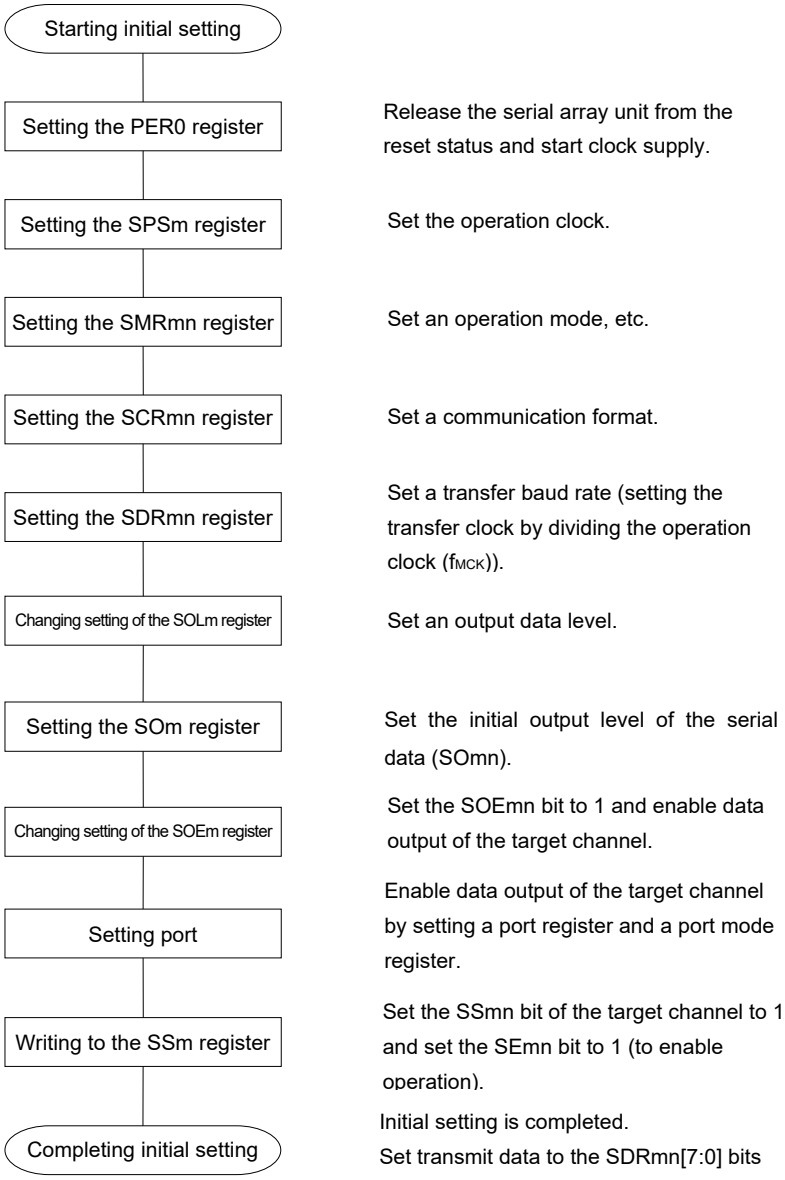


Figure 15-77. Procedure for Stopping UART Transmission

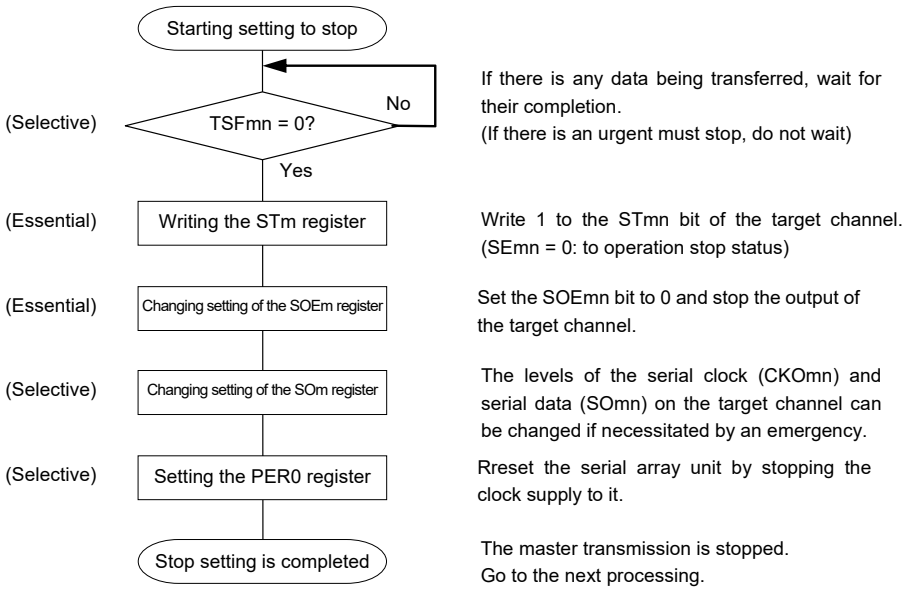
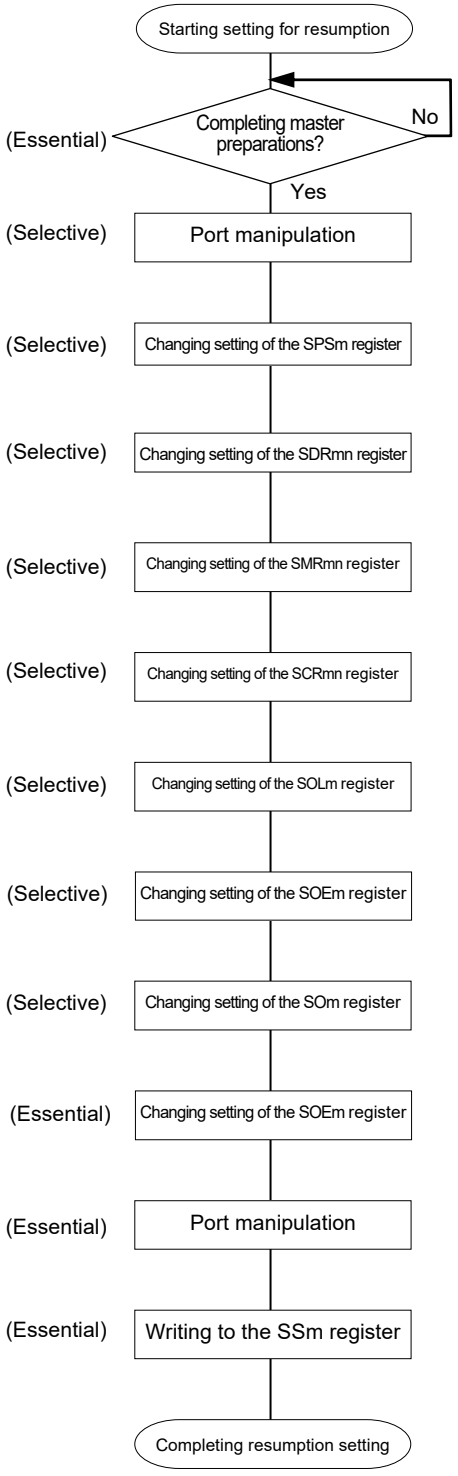


Figure 15-78. Procedure for Resuming UART Transmission



Wait until stop the communication target or communication operation completed

Disable data output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change the serial communication operation setting register mn (SCRmn) setting.

Re-set the register to change serial output level register m (SOLm) setting.

Clear the SOEmn bit to 0 and stop output.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to 1 and enable output.

Enable data output of the target channel by setting a port register and a port mode register.

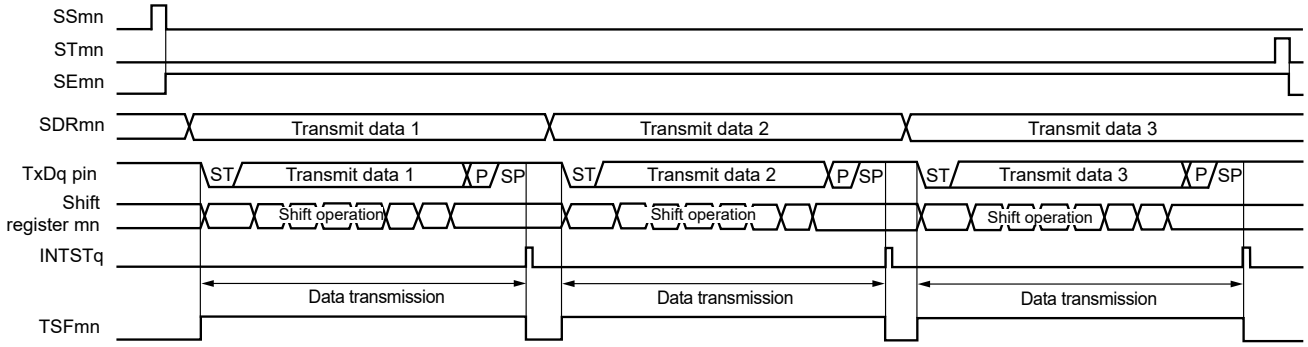
Set the SSmn bit of the target channel to 1 and set the SEmn bit to 1 (to enable operation).

Setting is completed.
Set transmit data to the SDRmn[7:0] bits (TXDq register) (8 bits) or SDRmn[8:0] bits (9 bits) and start communication.

Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target stops or transmission finishes, and then perform initialization instead of restarting the transmission.

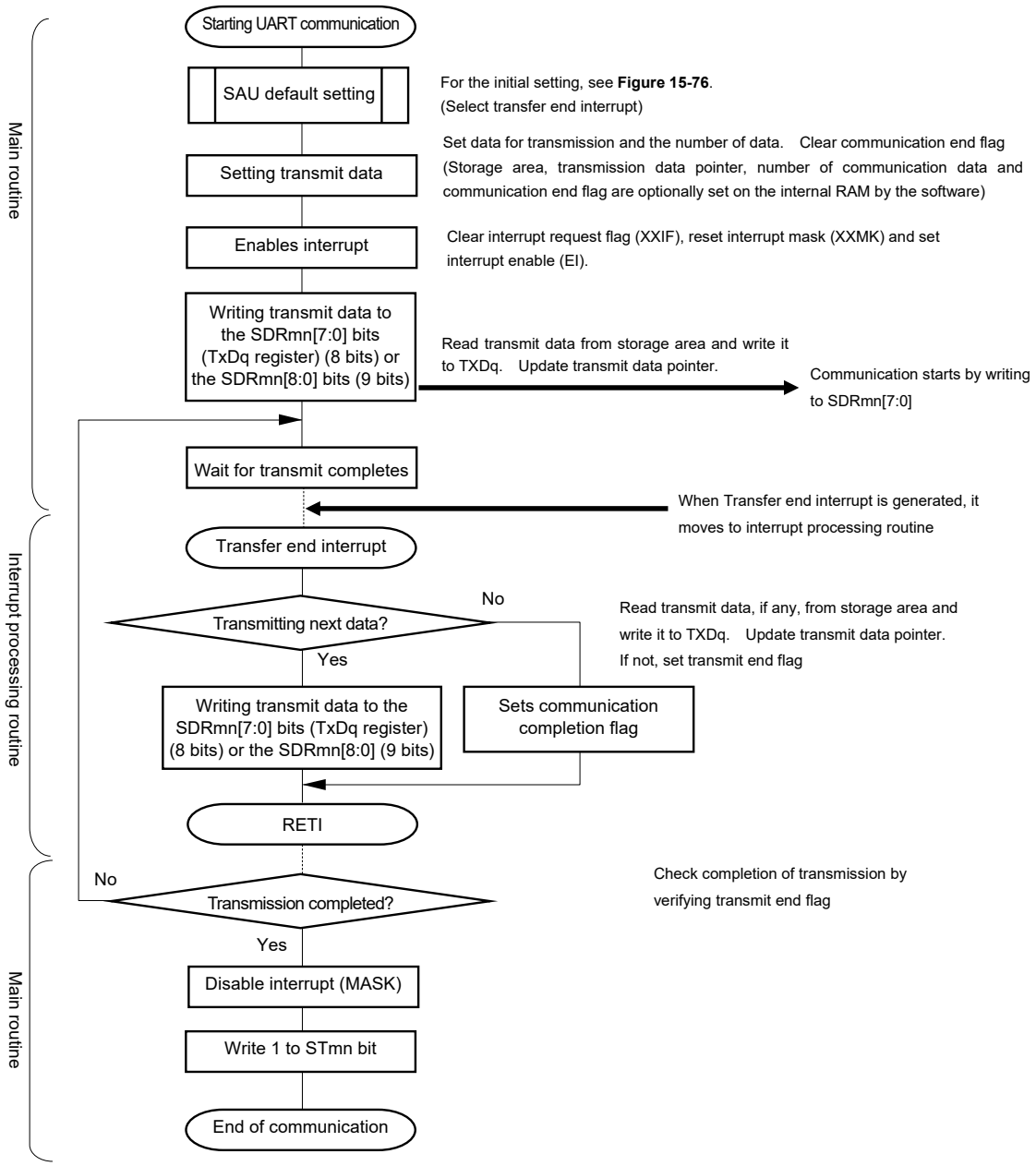
(3) Processing flow (in single-transmission mode)

Figure 15-79. Timing Chart of UART Transmission (in Single-Transmission Mode)



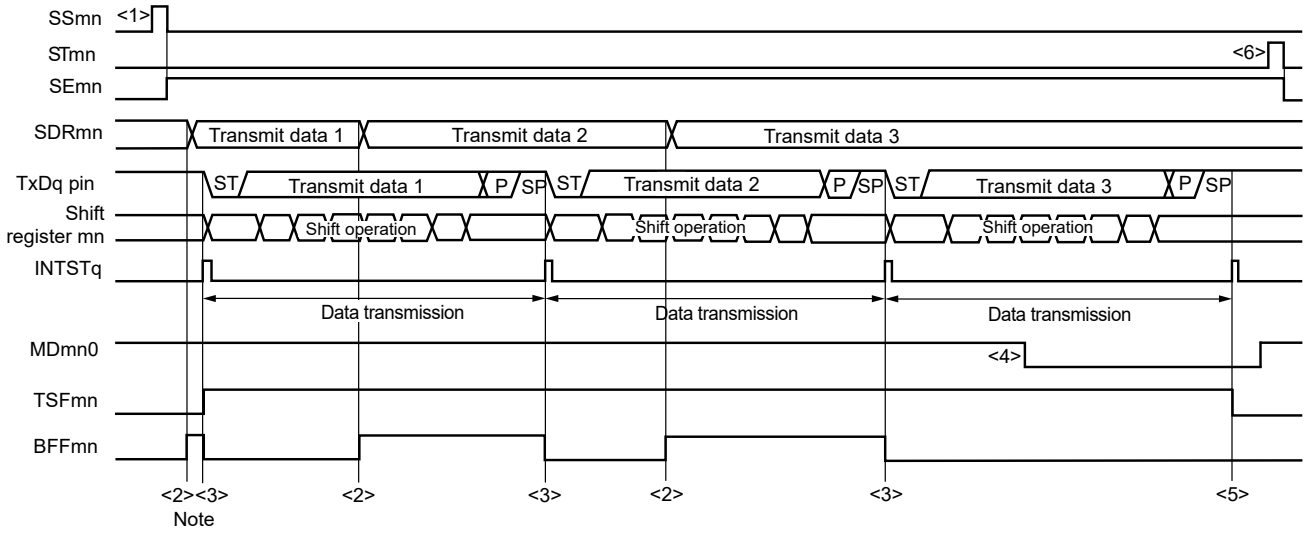
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2), q: UART number (q = 0, 1), mn = 00, 02

Figure 15-80. Flowchart of UART Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 15-81. Timing Chart of UART Transmission (in Continuous Transmission Mode)

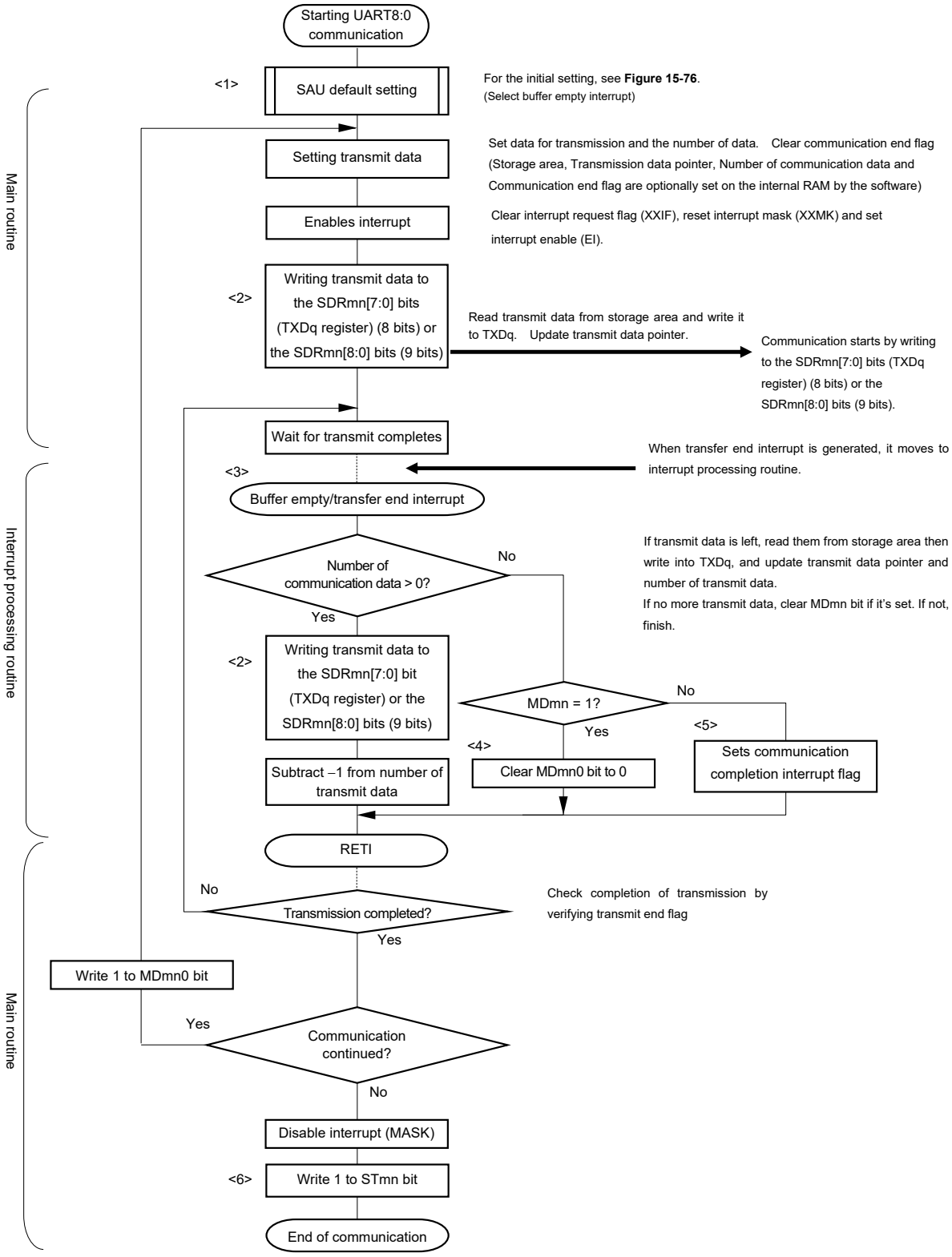


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SSRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2), q: UART number (q = 0, 1), mn = 00, 02

Figure 15-82. Flowchart of UART Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 15-81 Timing Chart of UART Transmission (in Continuous Transmission Mode)**.

15.6.2 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
Target channel	Channel 1 of SAU0	Channel 3 of SAU0
Pins used	RxD0	RxD1
Interrupt	INTSR0	INTSR1
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error interrupt	INTSRE0	INTSRE1
Error detection flag	• Framing error detection flag (FEFmn) • Parity error detection flag (PEFmn) • Overrun error detection flag (OVFmn)	
Transfer data length	7, 8 or 9 bits Note 1	
Transfer rate 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	

Notes 1. Only UART0 can be specified for the 8-bit data length.

2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C)** or **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T_A = −40 to +125°C)**).

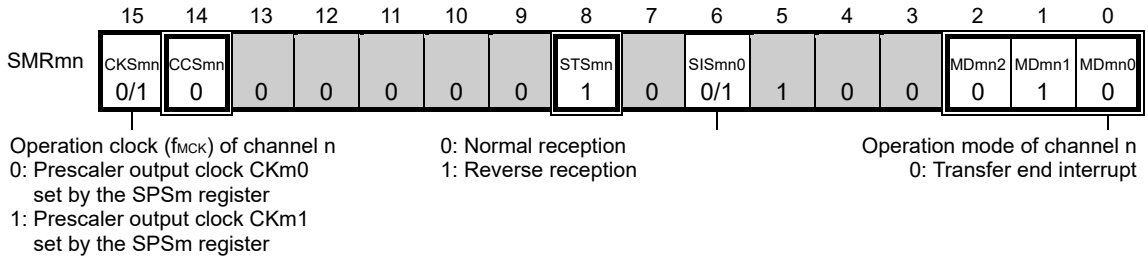
Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

2. m: Unit number (m = 0), n: Channel number (n = 1, 3), mn = 01, 03

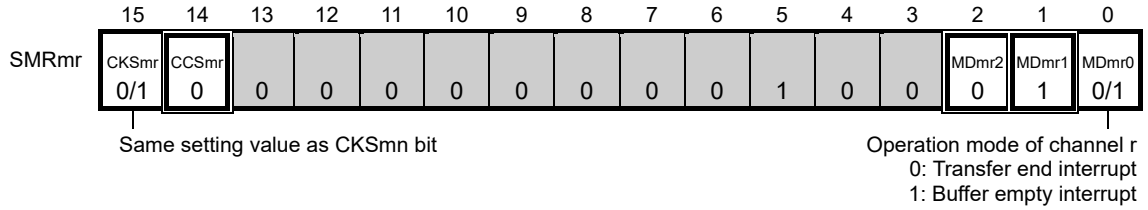
(1) Register setting

Figure 15-83. Example of Contents of Registers for UART Reception of UART (UART0, UART1) (1/2)

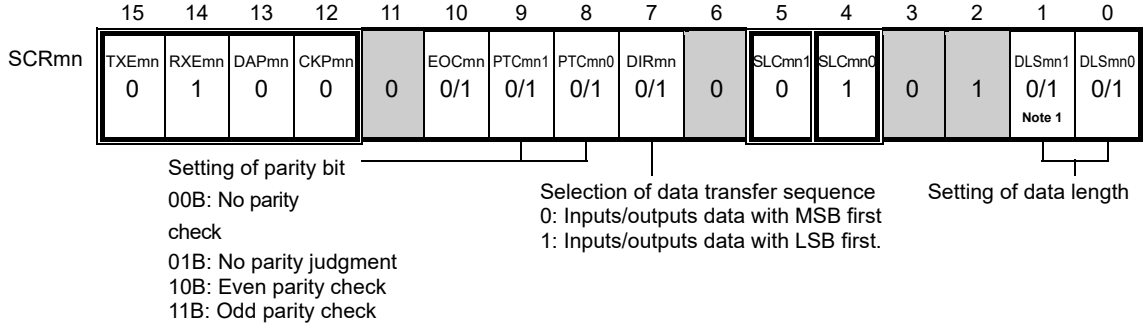
(a) Serial mode register mn (SMRmn)



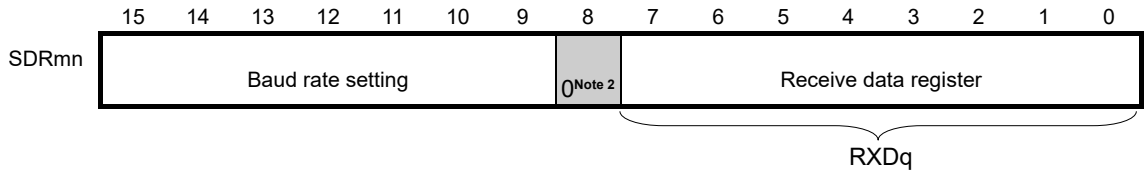
(b) Serial mode register mr (SMRmr)



(c) Serial communication operation setting register mn (SCRmn)



(d) Serial data register mn (SDRmn) (lower 8 bits: RXDq)



Notes 1. The SCR00 register (UART0) only. This is fixed to 1 for the SCR02 register.

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 can be specified for the 8-bit data length.

Caution For the UART reception, be sure to set the SMRmr register of channel r to UART transmission mode that is to be paired with channel n.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 1, 3), mn = 01, 03
r: Channel number (r = n – 1), q: UART number (q = 0, 1)

2. ☐: Setting is fixed in the UART reception mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 15-83. Example of Contents of Registers for UART Reception of UART (UART0, UART1) (2/2)

(e) Serial output register m (SOm) … The register that not used in this mode.

SOm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	CKOm2 ×	1	CKOm0 ×	0	0	0	0	1	SOm2 ×	1	SOm0 ×

(f) Serial output enable register m (SOEm) …The register that not used in this mode.

SOEm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	0	SOEm0 ×

(g) Serial channel start register m (SSm) … Sets only the bits of the target channel is 1.

SSm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 ×	SSm1 0/1	SSm0 ×

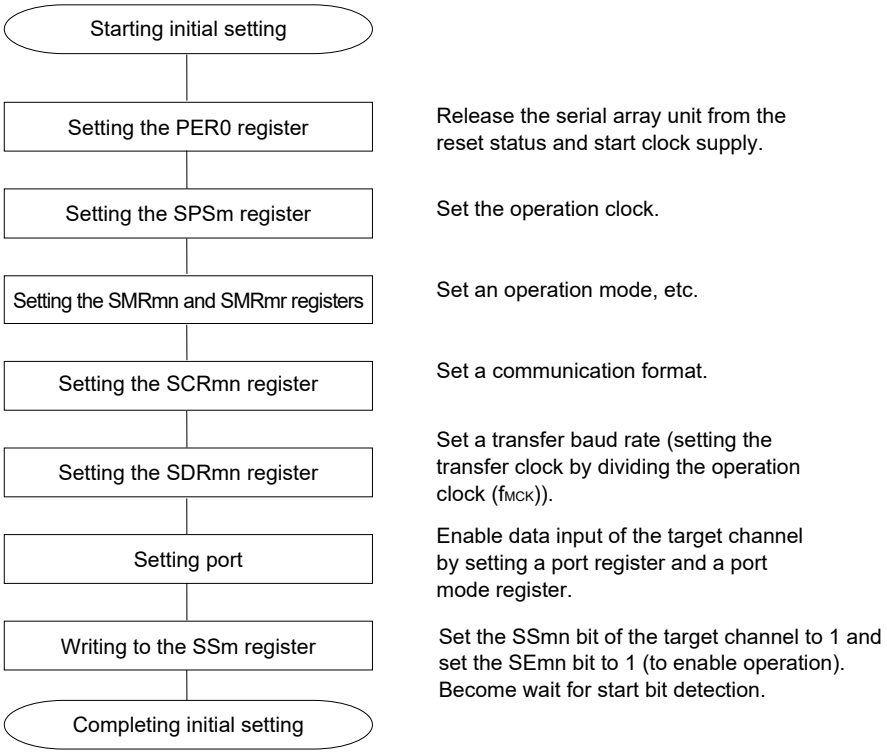
Caution For the UART reception, be sure to set the SMRmr register of channel r to UART Transmission mode that is to be paired with channel n.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 1, 3), mn = 01, 03
r: Channel number (r = n – 1), q: UART number (q = 0, 1)

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 15-84. 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 4 or more f_{MCK} clocks have elapsed.

Figure 15-85. Procedure for Stopping UART Reception

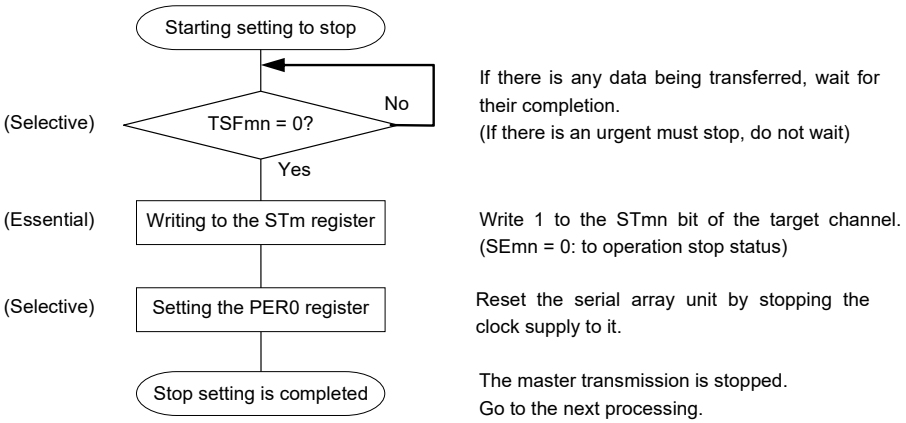
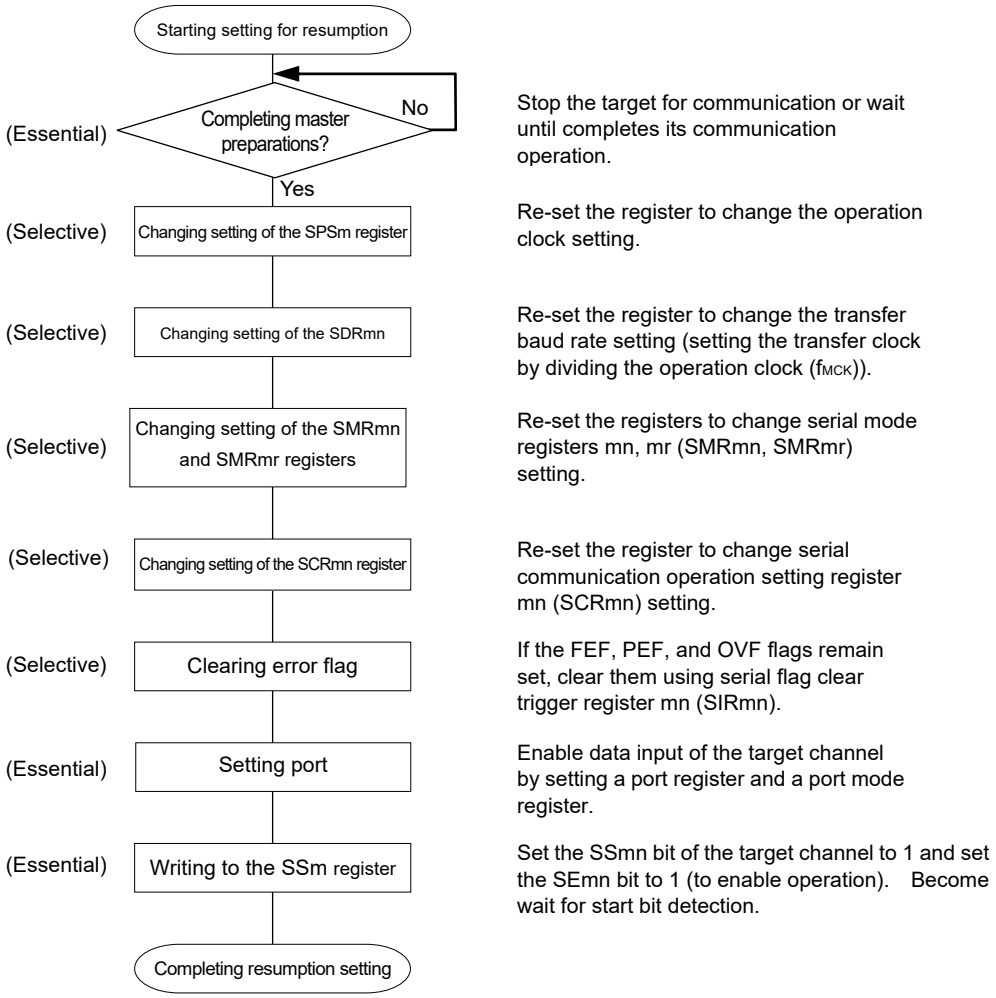


Figure 15-86. Procedure for Resuming UART Reception

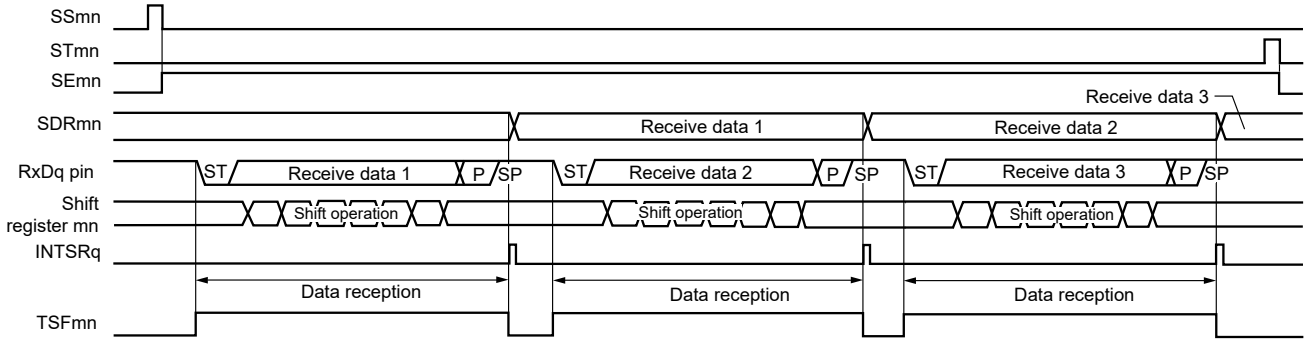


Caution After is set RXEmn bit to 1 of SCRmn register, set the SSmn = 1 from an interval of at least four clocks of f_{MCK}.

Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

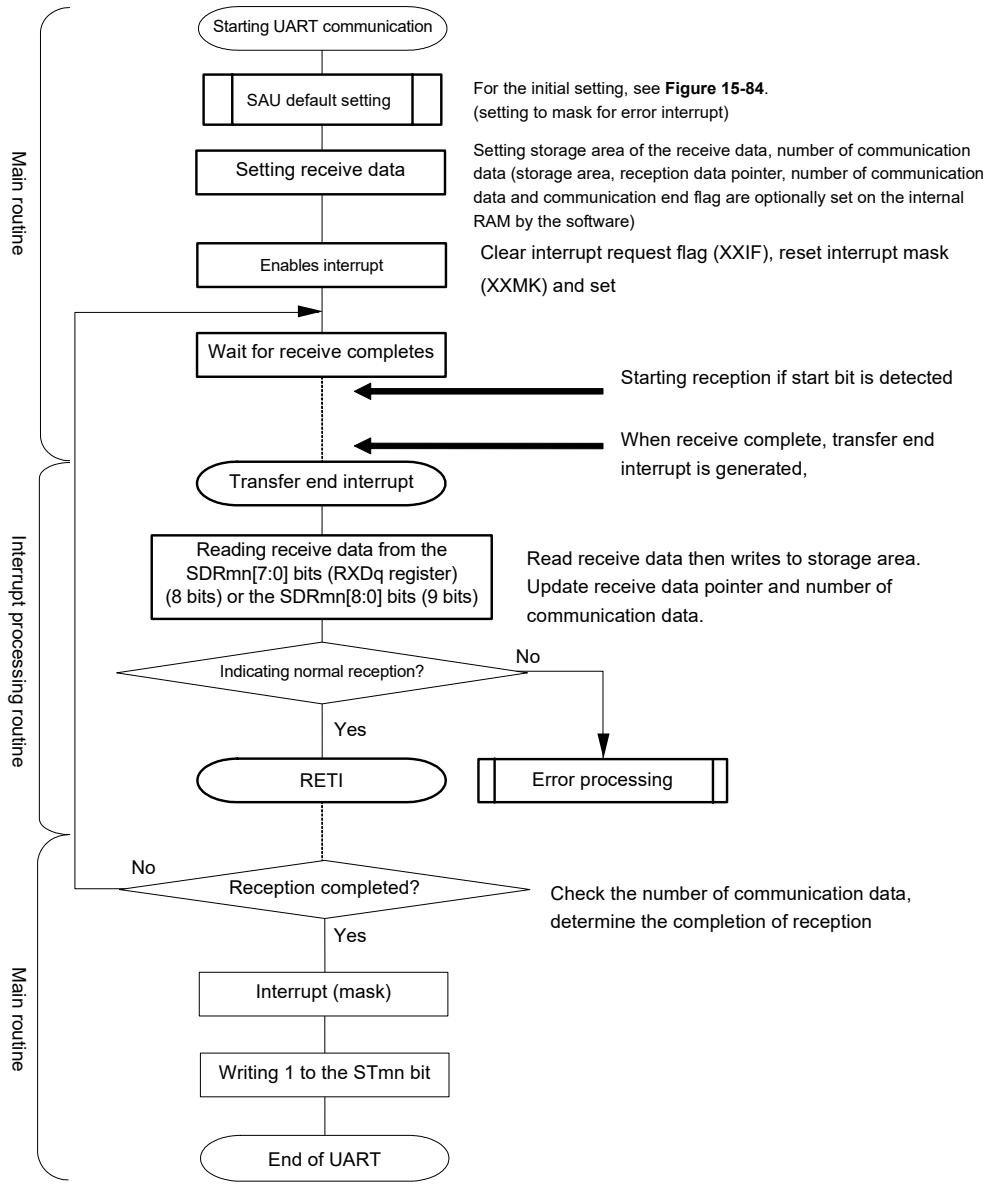
(3) Processing flow

Figure 15-87. Timing Chart of UART Reception



Remark m: Unit number (m = 0), n: Channel number (n = 1, 3), mn = 01, 03
r: Channel number (r = n – 1), q: UART number (q = 0, 1)

Figure 15-88. Flowchart of UART Reception



15.6.3 SNOOZE mode function

The SNOOZE mode makes the UART perform reception operations upon RxDq pin input detection while in the STOP mode. Normally the UART stops communication in the STOP mode. However, using the SNOOZE mode enables the UART to perform reception operations without CPU operation.

Only 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 15-91 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)** and **Figure 15-93 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 15-3.
- Set the EOCmn and SSECmn bits. This is for enabling or stopping generation of an error interrupt (INTSRE0) when a communication error occurs.
- When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 just before switching to the STOP mode. After the initial setting has completed, set the SSm1 bit of serial channel start register m (SSm) to 1.

Upon detecting the edge of RxDq (start bit input) after a transition was made to the STOP mode, UART reception is started.

Cautions 1. The SNOOZE mode can only be used when the high-speed on-chip oscillator clock (f_{IH}) is selected for f_{CLK}.

2. The transfer rate in the SNOOZE mode is only 4800 bps.

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**

4. If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFmn, FEFmn, or OVFmn flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFmn, FEFmn, or OVFmn flag before setting the SWC0 bit to 1 and read the value in bits 7 to 0 (RxDq register) of the SDRm1 register.

5. The CPU shifts from the STOP mode to the SNOOZE mode on detecting the valid edge of the RxDq 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.

Remark m = 0; n = 0; q = 0

Table 15-3. Baud Rate Setting for UART Reception in SNOOZE Mode

High-speed On-chip Oscillator (f_{IH})	Baud Rate for UART Reception in SNOOZE Mode			
	Baud Rate of 4800 bps			
	Operation Clock (f_{MCK})	SDRmn [15:9]	Maximum Permissible Value	Minimum Permissible Value
32 MHz ± 1.0% Note	$f_{CLK}/2^5$	105	2.27%	–1.53%
24 MHz ± 1.0% Note	$f_{CLK}/2^5$	79	1.60%	–2.18%
16 MHz ± 1.0% Note	$f_{CLK}/2^4$	105	2.27%	–1.53%
12 MHz ± 1.0% Note	$f_{CLK}/2^4$	79	1.60%	–2.19%
8 MHz ± 1.0% Note	$f_{CLK}/2^3$	105	2.27%	–1.53%
6 MHz ± 1.0% Note	$f_{CLK}/2^3$	79	1.60%	–2.19%
4 MHz ± 1.0% Note	$f_{CLK}/2^2$	105	2.27%	–1.53%
3 MHz ± 1.0% Note	$f_{CLK}/2^2$	79	1.60%	–2.19%
2 MHz ± 1.0% Note	$f_{CLK}/2$	105	2.27%	–1.54%
1 MHz ± 1.0% Note	f_{CLK}	105	2.27%	–1.57%

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.

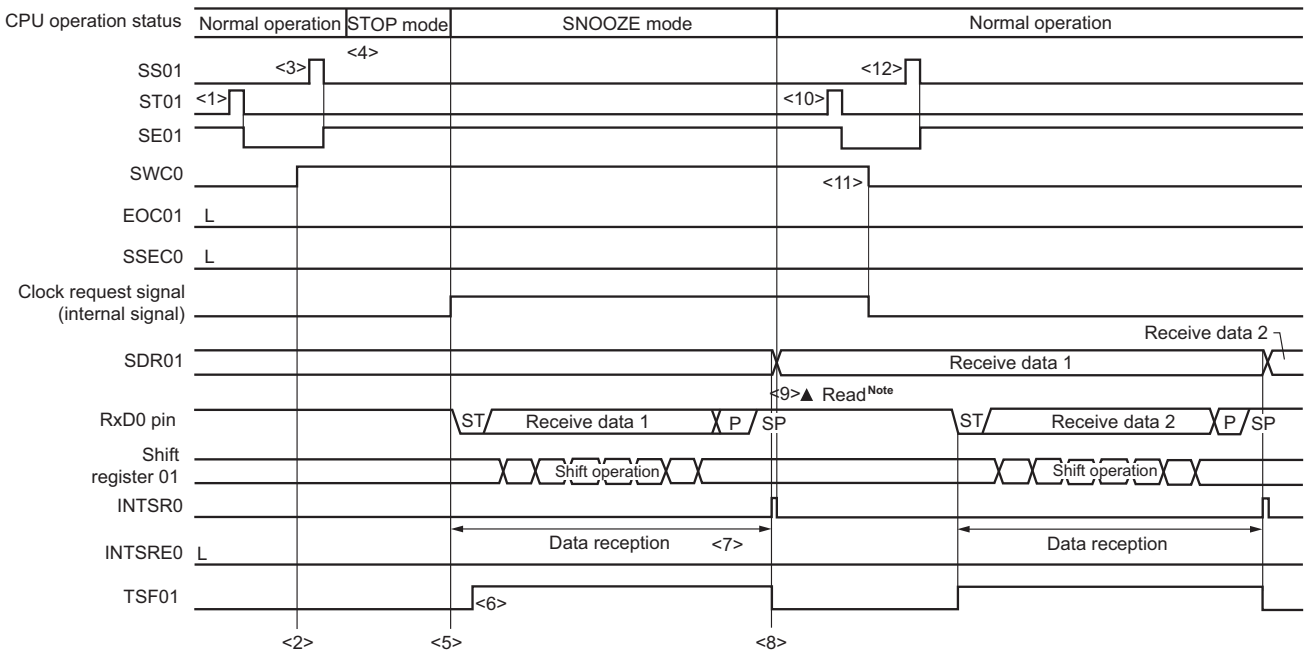
- In the case of f_{IH} ± 1.5%, perform (Maximum permissible value – 0.5%) and (Minimum permissible value + 0.5%) to the values in the above table.
- In the case of f_{IH} ± 2.0%, perform (Maximum permissible value – 1.0%) and (Minimum permissible value + 1.0%) to the values in the above table.

Remark The maximum permissible value and minimum permissible value are permissible values for the baud rate in UART reception. The baud rate on the transmitting side should be set to fall inside this range.

(1) SNOOZE mode operation (EOCm1 = 0, SSECm = 0/1)

Because of the setting of EOCm1 = 0, even though a communication error occurs, an error interrupt (INTSREq) is not generated, regardless of the setting of the SSECm bit. A transfer end interrupt (INTSRq) will be generated.

Figure 15-89. Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)



Note Read the received data when SWCm is 1

Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

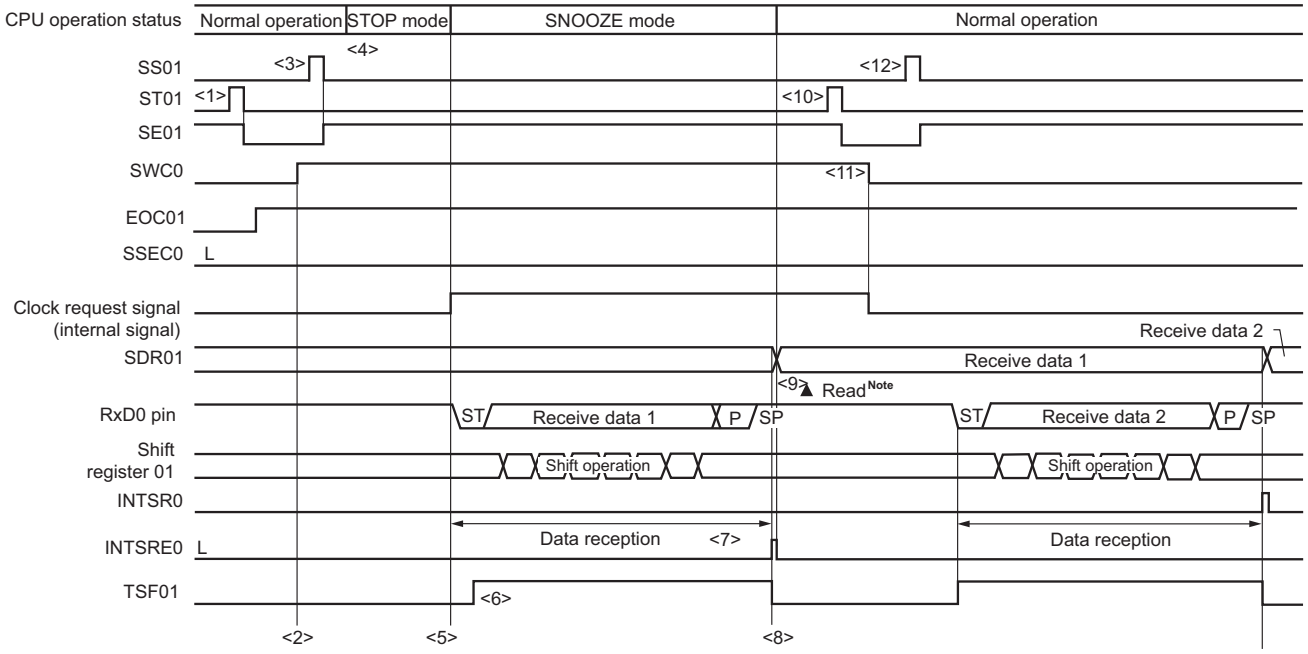
Remarks 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 15-91 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)**.

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 15-90. Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)



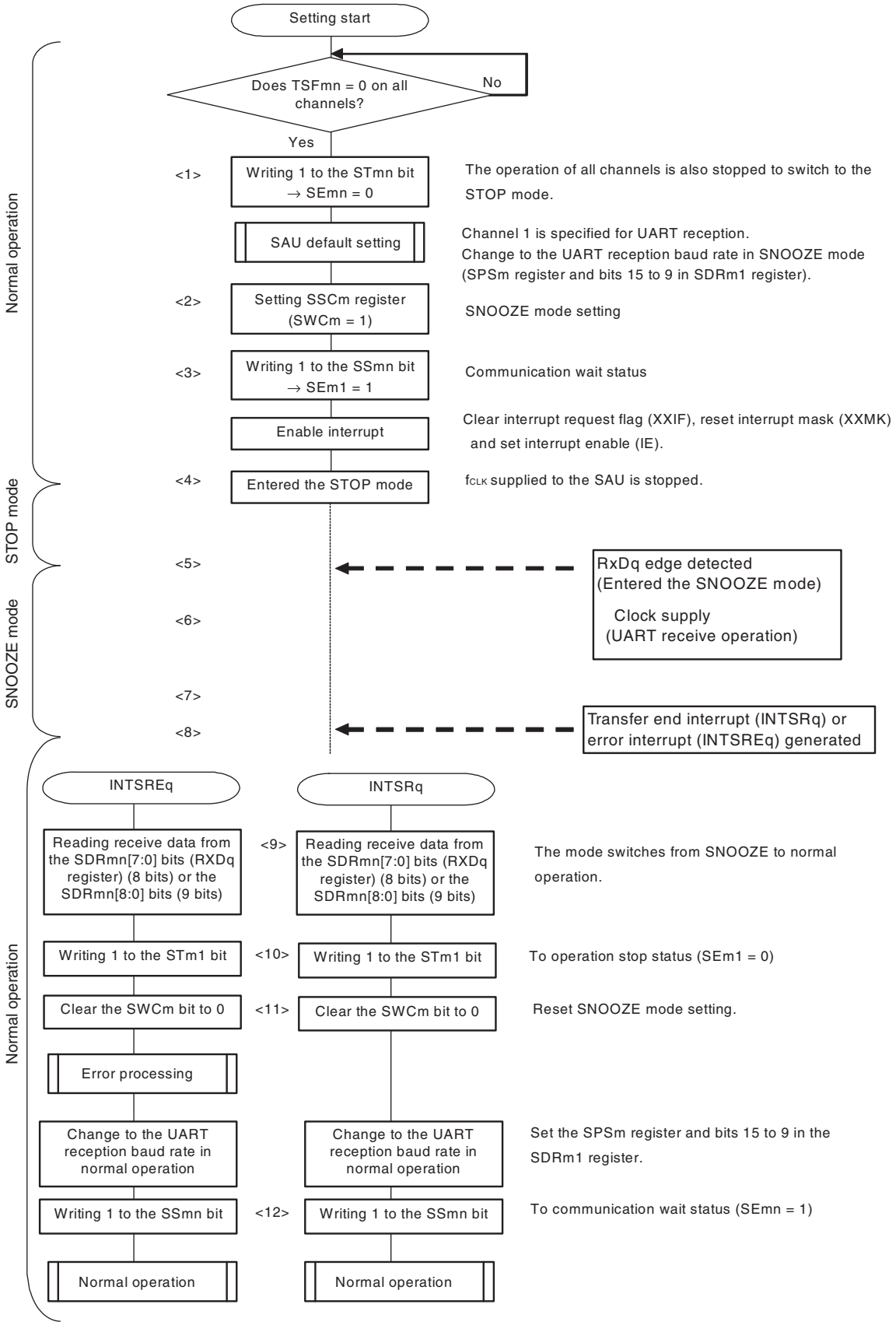
Note Read the received data when SWCm = 1.

Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

Remarks 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 15-91 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)**.

2. m = 0; q = 0

Figure 15-91. Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)



(**Remarks** are listed on the next page.)

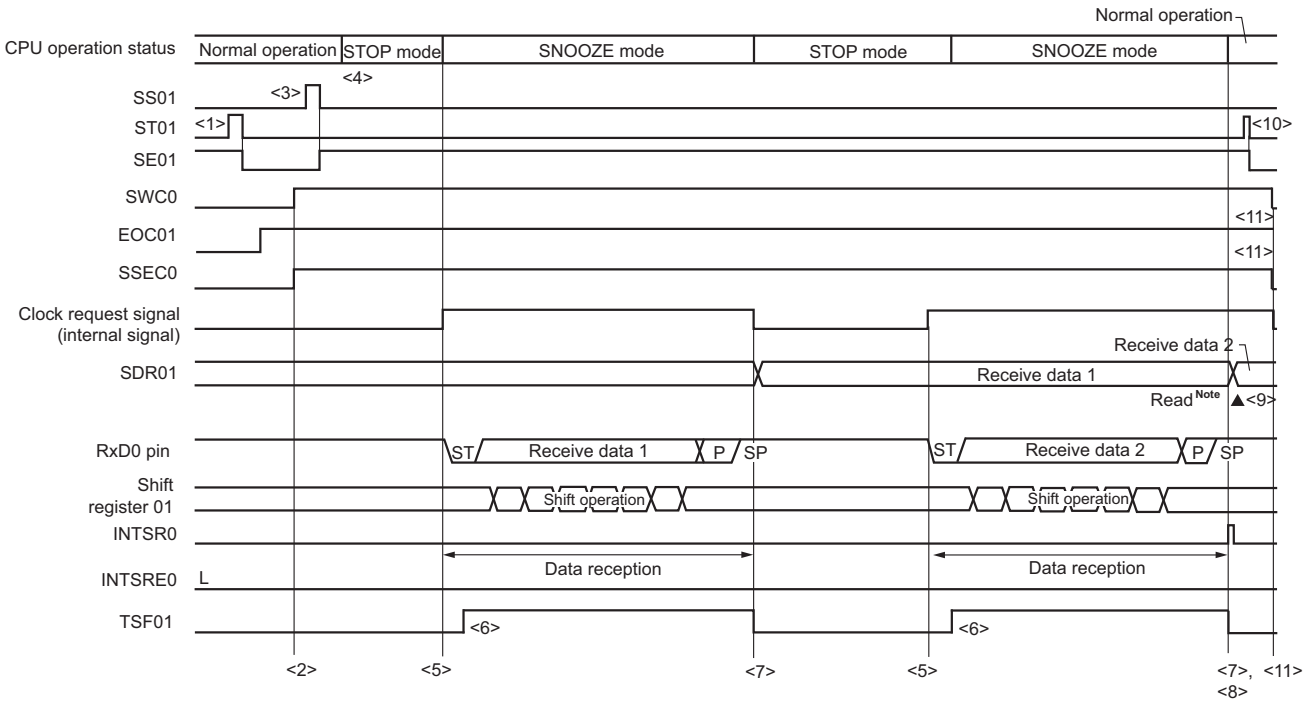
Remarks 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 15-89 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)** and **Figure 15-90 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)**.

2. m = 0; q = 0; n = 0, 1

(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 15-92. Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



Note Read the received data when SWCm = 1.

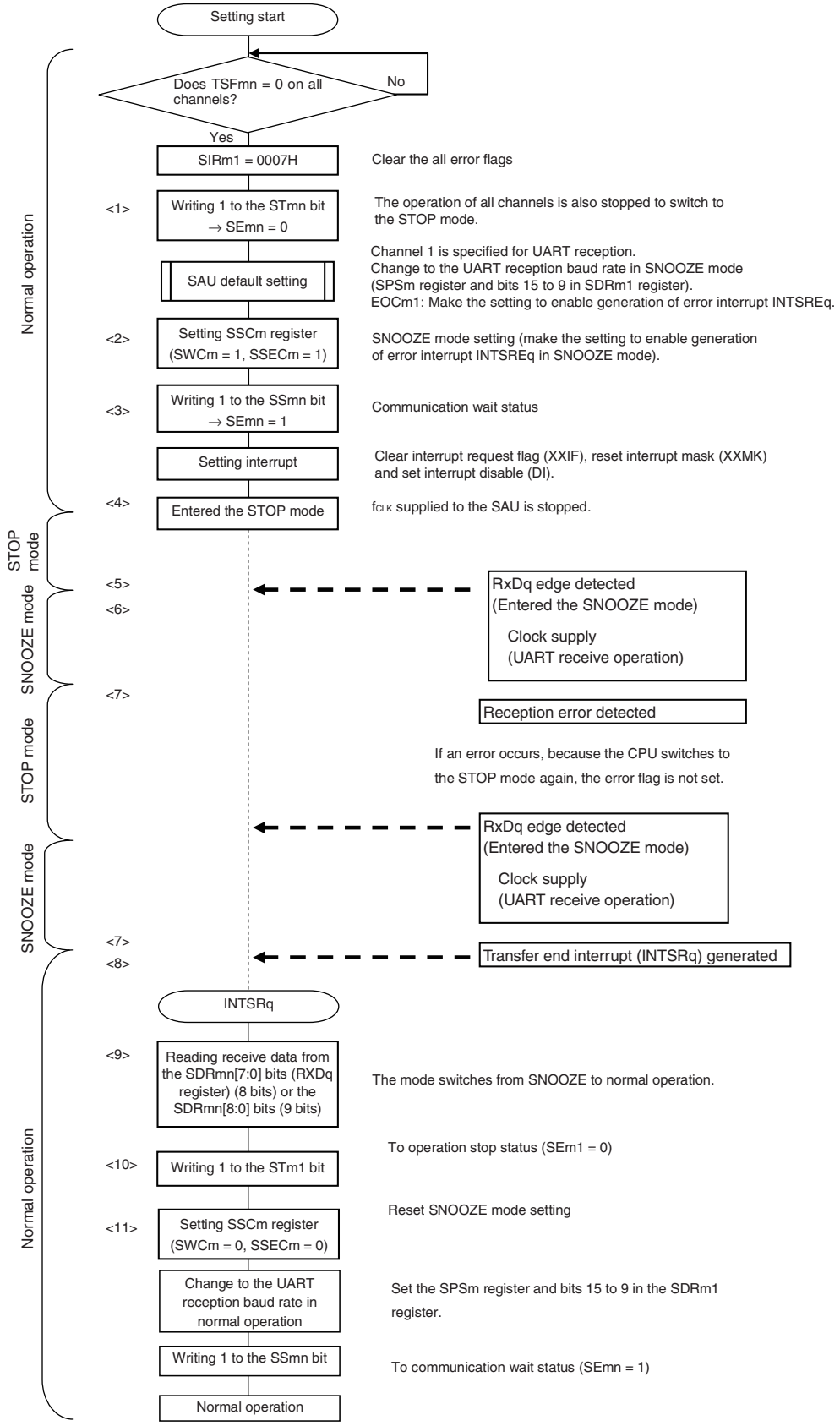
Cautions 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit and stop the operation).
After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).

2. If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFm1, FEFm1, or OVFm1 flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFm1, FEFm1, or OVFm1 flag before setting the SWCm bit to 1 and read the value in SDRm1[7:0] (RxDq register) (8 bits) or SDRm1[8:0] (9 bits).

Remarks 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 15-93 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)**.

2. m = 0; q = 0

Figure 15-93. Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



(**Caution** and **Remark** are listed on the next page.)

Caution If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFm1, FEFm1, or OVFm1 flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFm1, FEFm1, or OVFm1 flag before setting the SWCm bit to 1 and read the value in SDRm1[7:0] (RxDq register) (8 bits) or SDRm1[8:0] (9 bits).

Remarks 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 15-92 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)**.

2. m = 0; q = 0; n = 0, 1

15.6.4 Calculating baud rate

(1) Baud rate calculation expression

The baud rate for UART (UART0, UART1) communication can be calculated by the following expressions.

(Baud rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [bps]

Caution Setting serial data register mn (SDRmn) SDRmn[15:9] = (0000000B, 0000001B) is prohibited.

Remarks 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.

2. m: Unit number (m = 0), n: Channel number (n = 0), mn = 00 to 03

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 15-4. 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
0	X	X	X	X	0	0	0	0	f_{CLK}	32 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	16 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	8 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	4 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	2 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	1 MHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	500 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	250 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	125 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz
	X	X	X	X	1	1	0	0	$f_{CLK}/2^{12}$	7.81 kHz
	X	X	X	X	1	1	0	1	$f_{CLK}/2^{13}$	3.91 kHz
	X	X	X	X	1	1	1	0	$f_{CLK}/2^{14}$	1.95 kHz
	X	X	X	X	1	1	1	1	$f_{CLK}/2^{15}$	977 Hz
1	0	0	0	0	X	X	X	X	f_{CLK}	32 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	16 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	8 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	4 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	2 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	1 MHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	500 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	250 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	125 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	62.5 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	31.25 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	15.63 kHz
	1	1	0	0	X	X	X	X	$f_{CLK}/2^{12}$	7.81 kHz
	1	1	0	1	X	X	X	X	$f_{CLK}/2^{13}$	3.91 kHz
	1	1	1	0	X	X	X	X	$f_{CLK}/2^{14}$	1.95 kHz
	1	1	1	1	X	X	X	X	$f_{CLK}/2^{15}$	977 Hz
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).

Remarks 1. X: Don't care

2. m: Unit number (m = 0), n: Channel number (n = 0 to 3), mn = 00 to 03

(2) Baud rate error during transmission

The baud rate error of UART (UART0, UART1) communication during transmission can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

(Baud rate error) = (Calculated baud rate value) ÷ (Target baud rate) × 100 − 100 [%]

Here is an example of setting a UART baud rate at f_{CLK} = 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), n: Channel number (n = 0, 2), mn = 00, 02

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0, UART1) communication can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

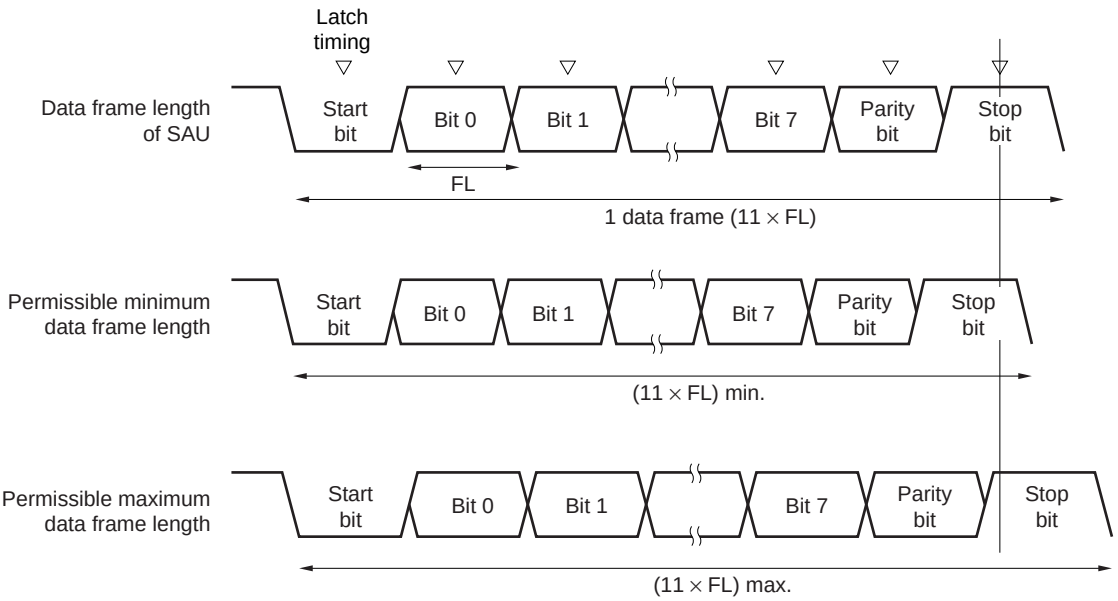
$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

Brate: Calculated baud rate value at the reception side (See **15.6.4 (1) Baud rate calculation expression**.)
k: SDRmn[15:9] + 1
Nfr: 1 data frame length [bits]
= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Remark m: Unit number (m = 0), n: Channel number (n = 1, 3), mn = 01, 03

Figure 15-94. Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in Figure 15-94, 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.

15.6.5 Procedure for processing errors that occurred during UART (UART0, UART1) communication

The procedure for processing errors that occurred during UART (UART0, UART1) communication is described in Figures 15-95 and 15-96.

Figure 15-95. Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Figure 15-96. Processing Procedure in Case of Framing Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STmn bit of serial channel stop register m (STm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operating.	
Synchronization with other party of communication		Synchronization with the other party of communication is re-established and communication is resumed because it is considered that a framing error has occurred because the start bit has been shifted.
Sets the SSmn bit of serial channel start register m (SSm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), mn = 00 to 03

15.7 LIN Communication Operation

15.7.1 LIN transmission

Of UART transmission, UART0 supports LIN communication.

For LIN transmission, channel 0 of unit 1 is used.

UART	UART0	UART1
Support of LIN communication	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] (SDR10 [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 in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T_A = –40 to +105°C)** or **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T_A = –40 to +125°C)**). In addition, LIN communication is usually 2.4/9.6/19.2 kbps is often used.

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.

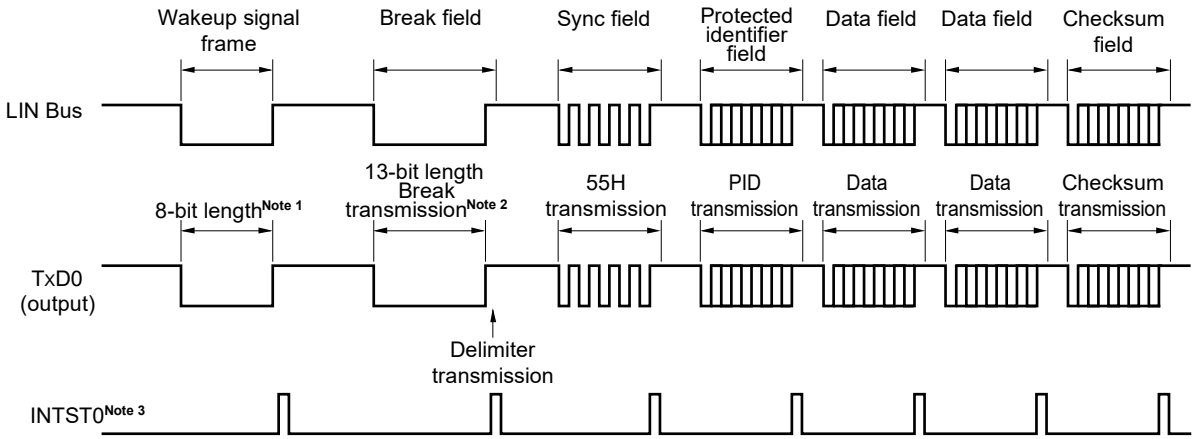
Usually, the master is connected to a network such as CAN (Controller Area Network).

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 15-97 outlines a master transmission operation of LIN.

Figure 15-97. Master Transmission Operation of LIN



Notes 1. Set the baud rate in accordance with the wakeup signal regulations and transmit data of 80H.

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.

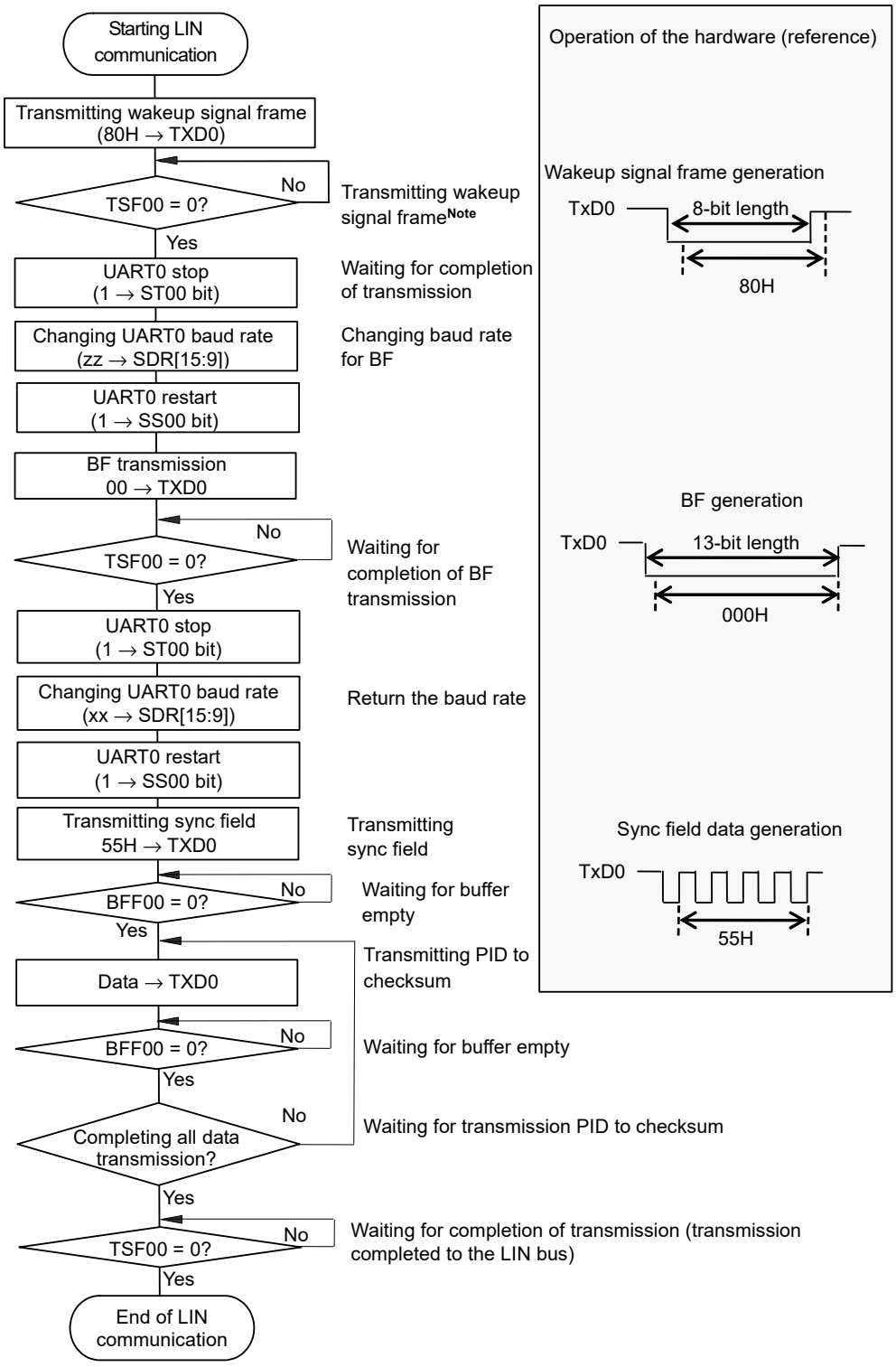
(Baud rate of break field) = 9/13 × N

By transmitting data of 00H at this baud rate, a break field is generated.

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 15-98. Flowchart for LIN Transmission



Note When LIN-bus start from sleep status only

Remark Default setting of the UART is complete, and the flow from the transmission enable status.

15.7.2 LIN reception

Of UART reception, UART0 supports LIN communication.

For LIN reception, channel 1 of unit 0 is used.

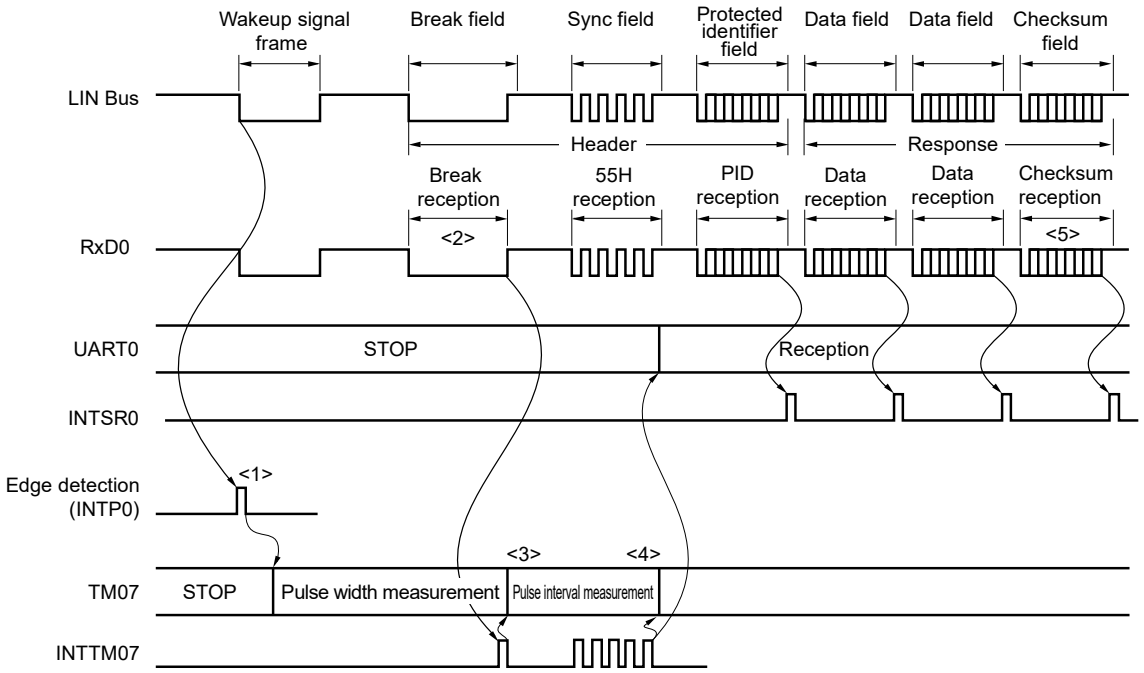
UART	UART0	UART1
Support of LIN communication	Supported	Not supported
Target channel	Channel 1	–
Pins used	RxD0	–
Interrupt	INTSR0	–
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error interrupt	INTSRE0	–
Error detection flag	• Framing error detection flag (FEF11) • Overrun error detection flag (OVF11)	
Transfer data length	8 bits	
Transfer rate[Note]	Max. $f_{MCK}/6$ [bps] (SDR11 [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 in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T_A = –40 to +105°C)** or **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T_A = –40 to +125°C)**).

Remark f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

Figure 15-99 outlines a reception operation of LIN.

Figure 15-99. Reception Operation of LIN



Here is the flow of signal processing.

<1> The wakeup signal is detected by detecting an interrupt edge (INTP0) on a pin. When the wakeup signal is detected, change TM07 to pulse width measurement upon detection of the wakeup signal to measure the low-level width of the BF signal. Then wait for BF signal reception.

<2> TM07 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.

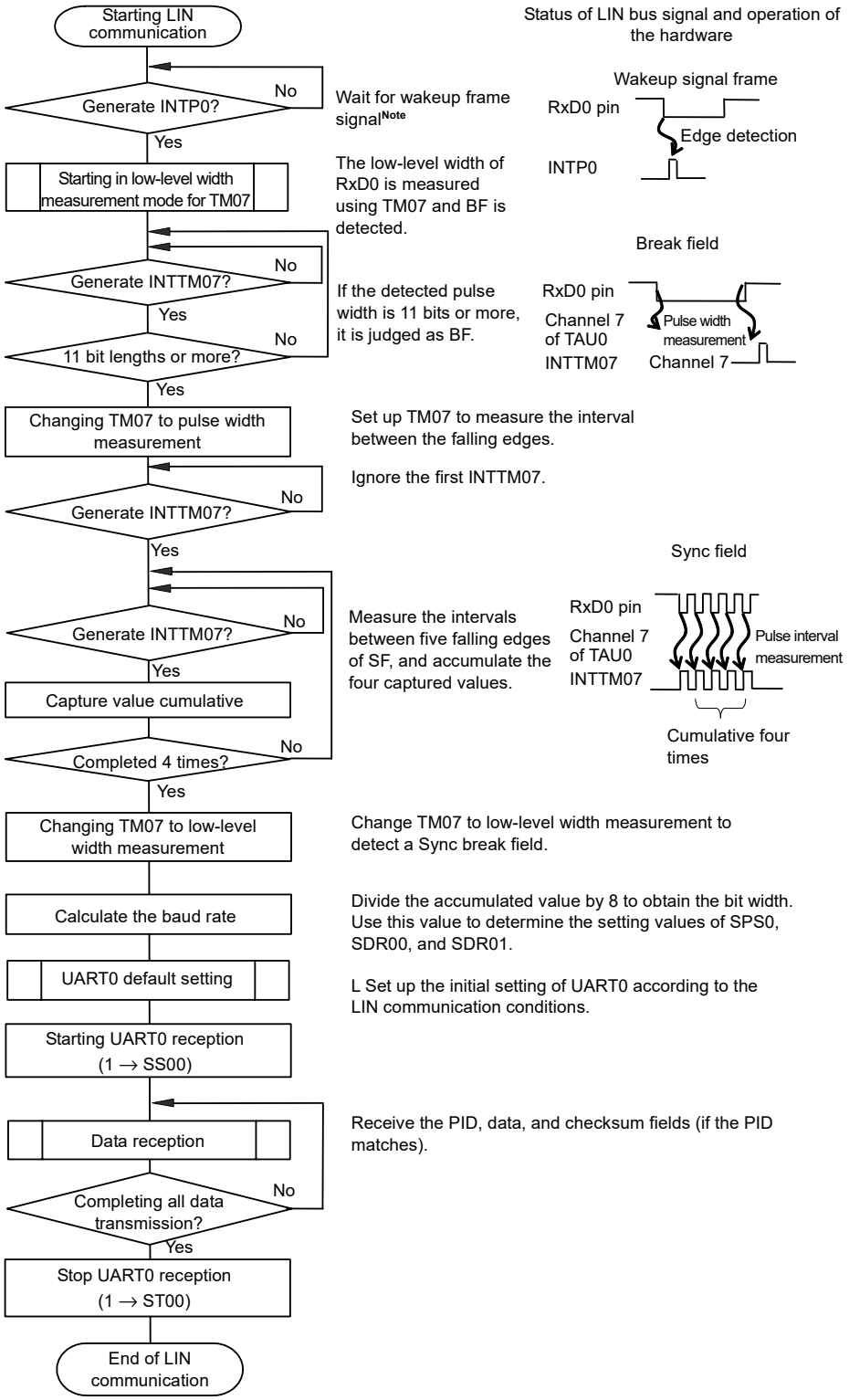
<3> When the BF signal has been received normally, change TM07 to pulse interval measurement and measure the interval between the falling edges of the RxD0 signal in the Sync field four times.

<4> When BF reception has been correctly completed, start channel 7 of the timer array unit and measure the bit interval (pulse width) of the sync field (see **6.8.3 Operation as input pulse interval measurement**).

<5> Calculate a baud rate error from the bit interval of sync field (SF). Stop UART0 once and adjust (re-set) the baud rate.

<6> 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 15-100. Flowchart for LIN Reception



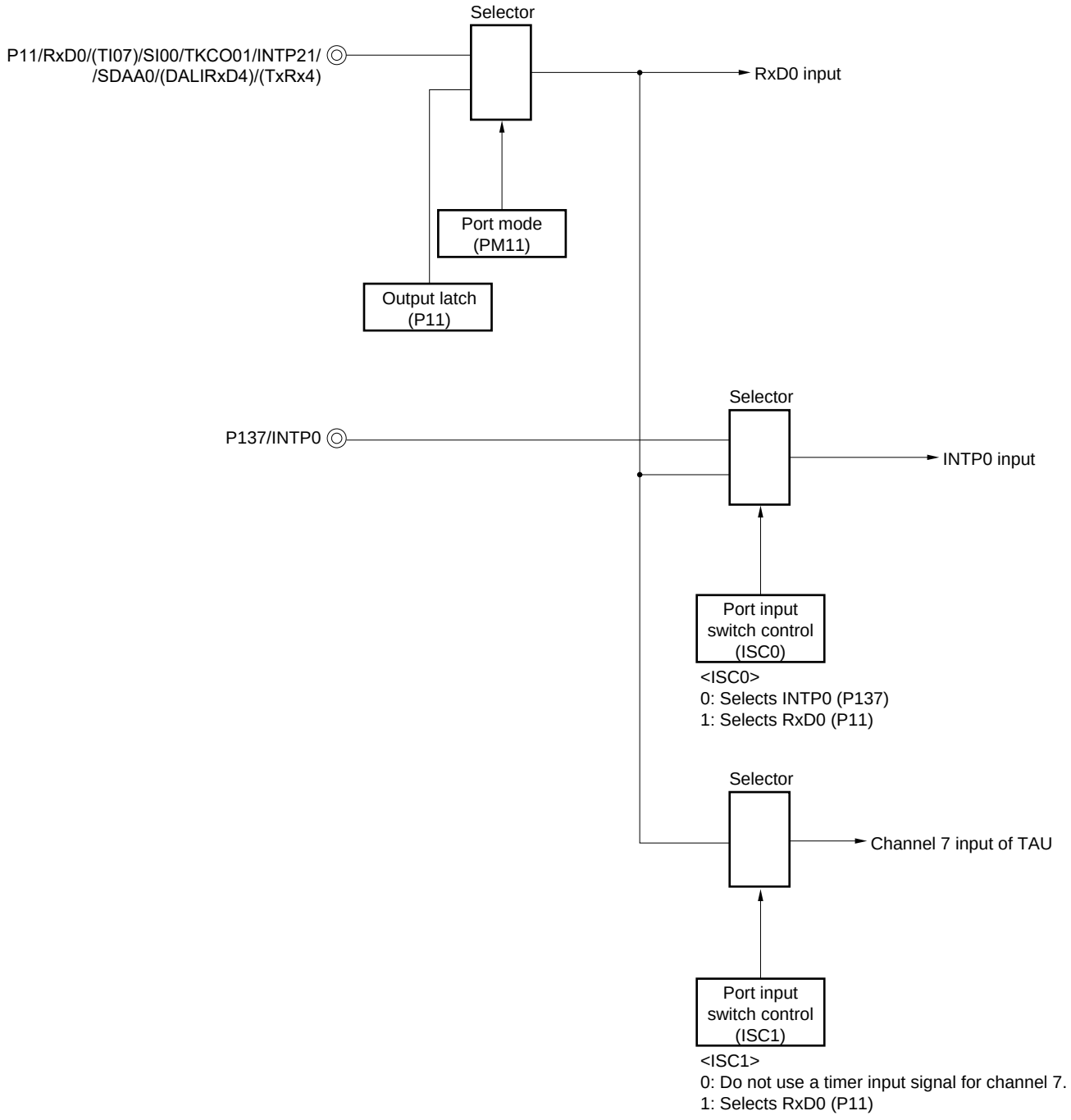
Note Required in the sleep status only.

Figure 15-101 shows the configuration of a port that manipulates reception of LIN.

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 controlling switch of port input (ISC0/ISC1), the input source of port input (RxD0) for reception can be input to the external interrupt pin (INTP0) and timer array unit

Figure 15-101. Port Configuration for Manipulating Reception of LIN



Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (See **Figure 15-20**.)

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 7 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 an error (The interval of the edge input to RxD0 is measured in the capture mode.)
  Measured the low-level width, determine whether break field (BF).
- Channels 0 and 1 (UART0) of serial array unit 1 (SAU0)

15.8 DMX512 Communication Operation

Of UART reception, UART0 supports DMX512 communication.
For DMX512 reception, channel 1 is used.

UART	UART0	UART1
Support of DMX512 communication	Supported	Not supported
Target channel	Channel 1	–
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	250 [kbps]	
Data phase	Forward output (default: high level)	
Parity bit	No parity bit (The parity bit is not checked.)	
Stop bit	Appending 2 bits	
Data direction	MSB first or LSB first	

About a reception operation of DMX512, see **16.11 DMX512 Communication Operation.**

Caution When the input signal from the RxD0 pin is taken to the INTP0 or TI07 pin during DMX512 communication using UART0, the input switch control register has to be set. For details, refer to 15. 3. 15 Input switch control register (ISC).

CHAPTER 16 SERIAL ARRAY UNIT 4 (DALI/UART4)

Channels 0 and 1 of serial array unit 4 function as a DALI/UART4 support to master and slave of DALI communications. DALI/UART4, like UART0 and UART1, can also be used for UART communications.

16.1 Functions of Serial Array Unit 4 (DALI/UART4)

(1) Asynchronous serial communication (UART) mode

This is a start-stop synchronization function using two lines: serial transmission data (DALITxD4) and serial reception data (DALIRxD4) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (start-stop synchronization communication) (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex start-stop synchronization communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel).

For details about the settings, see **16.5.1 UART transmission** or **16.5.2 UART reception.**

[Data transmission/reception]

- Available data of 7, 8, or 9 bit lengths
- Selectable the MSB/LSB first
- Available level setting of transmit/receive data (select of reverse)
- Parity bit appending and parity check functions
- Stop bit appending, stop bit check functions

[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, UARTs of following channels supports the SNOOZE mode. When RxD input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. TO supported for asynchronous reception.

Caution When UART4 is set to SNOOZE mode, CSI00, UART0, and the A/D converter cannot be set to SNOOZE mode. It is possible to set CSI00, UART0, and the A/D converter to SNOOZE mode at the same time.

(2) DALI mode

This mode is used to perform data transmission/reception as master and slave of DALI (Digital Addressable Lighting Interface). See **16.6.1 DALI transmission** or **16.6.2 DALI reception** for details.

DALI reception is not supported in the SNOOZE mode.

However, an operation equivalent to SNOOZE that uses an external interrupt to return from the STOP mode can be used.

See **16.7 Standby Function (Only DALI/UART4 Reception)** for details.

Remark DALI is an international open standard lighting control communication protocol, mainly used for light control of multiple fluorescent lamps or LED lights.

DALI is a network consisting of up to 64 short addresses and 16 group addresses, and performs half-duplex command communication between one master and one slave or multiple slaves.

DALI commands are used for purposes such as setting the light control level with 8-bit accuracy and saving or switching among up to 16 arbitrary light control levels.

[Data transmission/reception]

- Transmission of data of 8, 16, or 24 bit lengths
- Reception of data of 16, 17, or 24 bit lengths
- MSB first
- Available level setting of transmit/receive data (select of reverse)
- Stop bit appending 2 bits, stop bit check functions

[Interrupt function]

- Transfer end interrupt
- Reception interrupt
- Error interrupt in case of manchester framing error, framing error, or overrun error

[Error detection flag]

- Manchester framing error, Framing error, or overrun error

(3) SNOOZE mode

UART reception mode (channel 1) supports the SNOOZE mode. When DALIRxD4 pin input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible.

(4) Single-wire data mode

In this half-duplex communication mode, the same pin is shared as the data transmit pin and receive pin. The TxRx4 pin is used as the transmit/receive pin. For details, see **16.8 Single-wire Data Mode**.

16.2 Configuration of Serial Array Unit 4 (DALI/UART4)

The serial array unit 4 (DALI/UART4) includes the following hardware.

Table 16-1. Configuration of Serial Array Unit 4 (DALI/UART4)

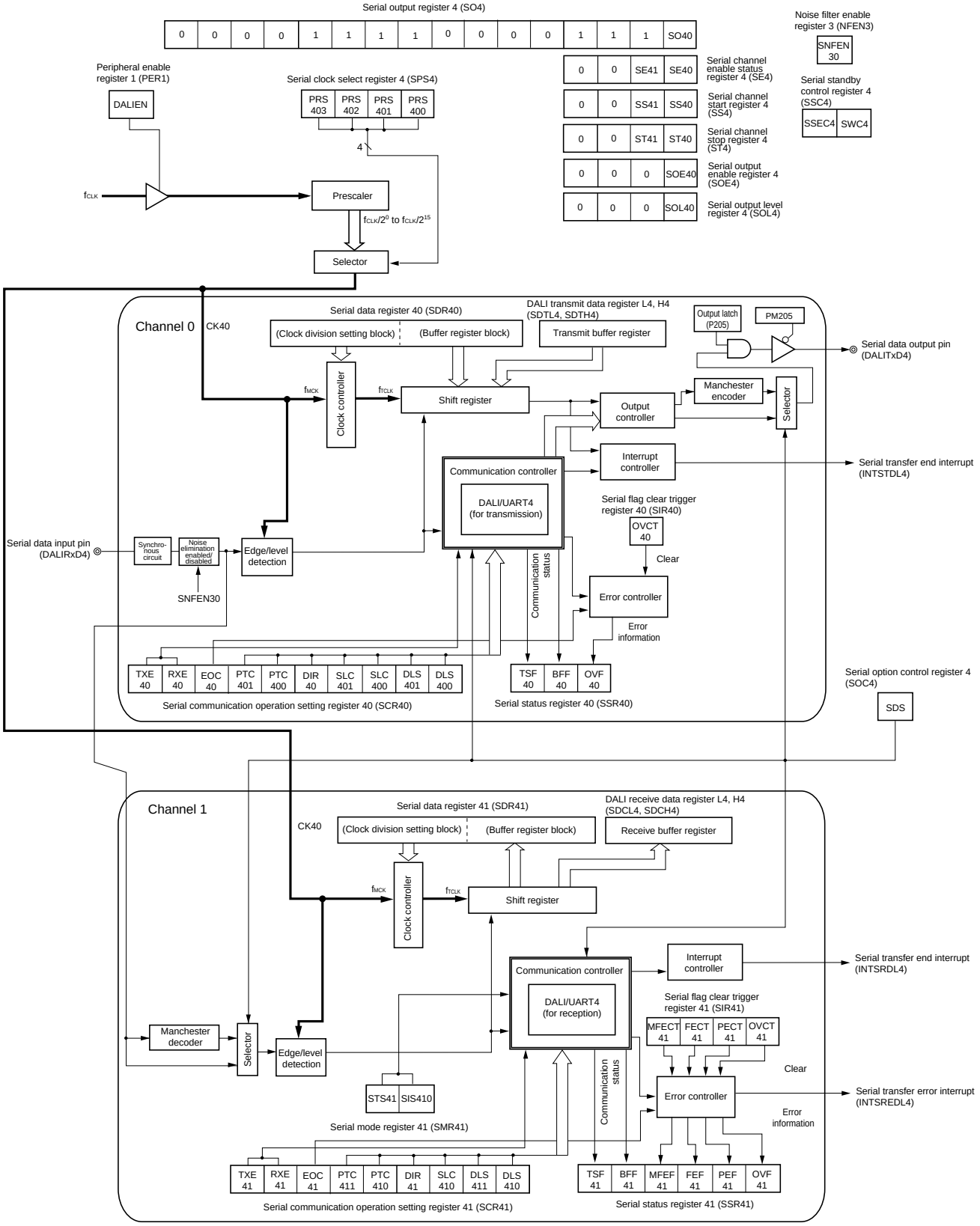
Item	Configuration
Shift register	9 bits
Buffer register	Lower 9 bits of serial data register 40 (SDR40)[Note] DALI transmit data register H4, L4 (SDTH4, SDTL4) DALI receive data register H4, L4 (SDCH4, SDCL4)
Serial data input	DALIRxD4 pin
Serial data output	DALITxD4 pin
Serial data I/O	TxRx4 pin
Control registers	<Registers of unit setting block> • Peripheral enable register 1 (PER1) • Serial clock select register 4 (SPS4) • Serial channel enable status register 4 (SE4) • Serial channel start register 4 (SS4) • Serial channel stop register 4 (ST4) • Serial output enable register 4 (SOE4) • Serial output register 4 (SO4) • Serial output level register 4 (SOL4) • Serial standby control register 4 (SSC4) • Single-wire UART control register (SUCTL) • Noise filter enable register 3 (NFEN3) • Serial option control register 4 (SOC4)
	<Registers of each channel> • Serial data registers 40, 41 (SDR40, SDR41) • Serial mode registers 40, 41 (SMR40, SMR41) • Serial communication operation setting registers 40, 41 (SCR40, SCR41) • Serial status registers 40, 41 (SSR40, SSR41) • Serial flag clear trigger registers 40, 41 (SIR40, SIR41)
	• Port output mode register 20 (POM20) • Port mode register 20 (PM20) • Port register 20 (P20) • Peripheral I/O redirection register (PIOR1)

Note The lower 8 bits of serial data register 41 (SDR41) can be read or written as the following SFR, depending on the communication mode.

- UART reception … RXD4 (UART4 receive data register)
- UART transmission … TXD4 (UART4 transmit data register)

Figure 16-1 shows the block diagram of the serial array unit 4 (DALI/UART4).

Figure 16-1. Block Diagram of Serial Array Unit 4 (DALI/UART4)



16.2.1 Shift register

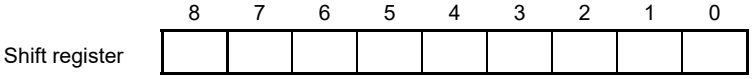
This is a 9-bit register that converts parallel data into serial data or vice versa.

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 9 bits of serial data register 4n (SDR4n), or DALI transmit data register H4, L4 (SDTH4, SDTL4), DALI receive data register H4, L4 (SDCH4, SDCL4).



16.2.2 Lower 9 bits of the serial data register 4n (SDR4n)

The SDR4n register is the transmit/receive data register (16 bits) of channel n. Bits 8 to 0 (lower 9 bits) of SDR4n 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 in UART mode, parallel data converted by the shift register is stored in the lower 9 bits. When data is to be transmitted in UART mode, set transmit to be transferred to the shift register to the lower 9 bits.

The data stored in the lower 9 bits of this register is as follows, depending on the setting of bits 0 and 1 (DLS4n0, DLS4n1) of serial communication operation setting register 4n (SCR4n), regardless of the output sequence of the data.

- 9-bit data length (stored in bits 0 to 8 of SDR4n register)
- 7-bit data length (stored in bits 0 to 6 of SDR4n register)
- 8-bit data length (stored in bits 0 to 7 of SDR4n register)

The SDR4n register can be read or written in 16-bit units.

The lower 9 bits of the SDR4n register can be read or written as the following SFR, depending on the communication mode.

- UART reception … RXD4 (UART4 receive data register)
- UART transmission … TXD4 (UART4 transmit data register)

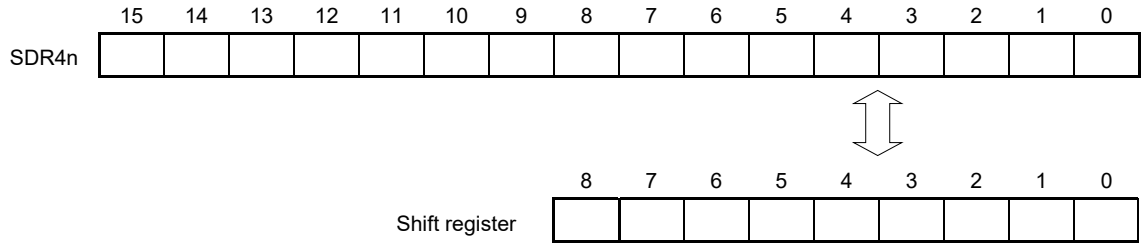
Reset signal generation clears the SDR4n register to 0000H.

Remarks 1. After data is received, “0” is stored in bits 0 to 8 in bit portions that exceed the data length.

2. n: Channel number (n = 0, 1), q: UART number (q = 4)

Figure 16-2. Format of Serial Data Register 4n (SDR4n)

Address: F0570H, F0571H (SDR40), F0572H, F0573H (SDR41) After reset: 0000H R/W



Remark For the function of the higher 7 bits of the SDR4n register, see **16.3 Registers Controlling Serial Array Unit 4 (DALI/UART4)**.

16.2.3 DALI transmit data registers H4, L4 (SDTH4, SDTL4)

The SDTH4, SDTL4 registers are the transmit data register (16 bits) of DALI.

Of the transmit data sent to the shift register during DALI communications, bits 15 to 0 are set in the SDTL4 register and bits 23 to 16 are set in the SDTH4 register.

The data stored this register is as follows, depending on the setting of bits 0 and 1 (DLS4n0, DLS4n1) of serial communication operation setting register 4n (SCR4n), regardless of the output sequence of the data.

- 24-bit data length (stored in bits 0 to 7 of SDTH4 register and stored in bits 0 to 15 of SDTL4 register)
- 17-bit data length (stored in bit 0 of SDTH4 register and stored in bits 0 to 15 of SDTL4 register)
- 16-bit data length (stored in bits 0 to 15 of SDTL4 register)
- 8-bit data length (stored in bits 0 to 7 of SDTL4 register)

The SDTH4 and SDTL4 registers can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears the SDTH4 and SDTL4 registers to 0000H.

Figure 16-3. Format of DALI Transmits Data Register H4, L4 (SDTH4, SDTL4)

Address: F0578H (SDTL4), F057AH (SDTH4) After reset: 0000H R/W

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDTH4	0	0	0	0	0	0	0	0								

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDTL4																

	7	6	5	4	3	2	1	0
Shift register								

Cautions 1. Be sure to clear bits 8 to 15 of the SDTH4 register to "0".

2. Data transmission starts when data is written to the SDTL4 register. If data longer than 16 bits is set, it should be written first to the SDTH4 register, and then to the SDTL4 register.

Remark n: Channel number (n = 0, 1)

16.2.4 DALI receive data registers H4, L4 (SDCH4, SDCL4)

The SDCH4, SDCL4 registers are the receive data register (16 bits) of DALI.

When DALI is received, parallel data that was converted in the shift register is written starting from LSB of the SDCL4 register.

The data stored this register is as follows, depending on the setting of bits 0 and 1 (DLS4n0, DLS4n1) of serial communication operation setting register 4n (SCR4n), regardless of the output sequence of the data.

- 24-bit data length (stored in bits 0 to 7 of SDCH4 register and stored in bits 0 to 15 of SDCL4 register)
- 17-bit data length (stored in bit 0 of SDCH4 register and stored in bits 0 to 15 of SDCL4 register)
- 16-bit data length (stored in bits 0 to 15 of SDCL4 register)
- 8-bit data length (stored in bits 0 to 7 of SDCL4 register)

The SDCH4 and SDCL4 registers can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears the SDCH4 and SDCL4 registers to 0000H.

Figure 16-4. Format of DALI Receive Data Registers H4, L4 (SDCH4, SDCL4)

Address: F057CH (SDCL4), F057EH (SDCH4) After reset: 0000H R

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDCH4	0	0	0	0	0	0	0	0								

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDCL4																

	7	6	5	4	3	2	1	0
Shift register								

Caution Bits 8 to 15 of the SDCH4 register are fixed to 0.

Remarks 1. After reception is completed, the part exceeding the data length is filled with zeros.

2. n: Channel number (n = 0, 1)

16.3 Registers Controlling Serial Array Unit 4 (DALI/UART4)

Serial array unit 4 (DALI/UART4) is controlled by the following registers.

- Peripheral enable register 1 (PER1)
- Serial clock select register 4 (SPS4)
- Serial mode registers 40, 41 (SMR40, SMR41)
- Serial communication operation setting registers 40, 41 (SCR40, SCR41)
- Serial data registers 40, 41 (SDR40, SDR41)
- Serial flag clear trigger registers 40, 41 (SIR40, SIR41)
- Serial status registers 40, 41 (SSR40, SSR41)
- Serial option control register 4 (SOC4)
- Serial channel start register 4 (SS4)
- Serial channel stop register 4 (ST4)
- Serial channel enable status register 4 (SE4)
- Serial output enable register 4 (SOE4)
- Serial output level register 4 (SOL4)
- Serial output register 4 (SO4)
- Serial standby control register 4 (SSC4)
- Single-wire UART control register (SUCTL)
- Noise filter enable register 3 (NFEN3)
- Port output mode register 1, 20 (POM1, POM20)
- Port mode register 1, 20 (PM1, PM20)
- Port register 1, 20 (P1, P20)
- Peripheral I/O redirection register (PIOR1)

16.3.1 Peripheral enable register 1 (PER1)

PER1 is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When serial array unit 4 (DALI/UART4) is used, 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.

Reset signal generation clears the PER1 register to 00H.

Figure 16-5. Format of Peripheral Enable Register 1 (PER1)

Address: F0508H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
PER1	0	0	0	0	0	0	0	DALIEN

DALIEN	Control of serial array unit 4 (DALI/UART4) input clock supply
0	Stops supply of input clock. • SFR used by serial array unit 4 (DALI/UART4) cannot be written. • Serial array unit 4 (DALI/UART4) is in the reset status.
1	Enables input clock supply. • SFR used by serial array unit 4 (DALI/UART4) can be read/written.

Cautions 1. When setting serial array unit 4 (DALI/UART4), be sure to set the DALIEN bit to 1 first. If DALIEN = 0, writing to a control register of serial array unit 4 (DALI/UART4) is ignored, and, even if the register is read, only the default value is read (except for serial standby control register 4 (SSC4), noise filter enable register 3 (NFEN3), port output mode register 20 (POM20), port mode register 20 (PM20), and port register 20 (P20)).

2. After setting the DALIEN bit to 1, be sure to set serial clock select register 4 (SPS4) after 4 or more fCLK clocks have elapsed.

3. Be sure to clear bits 1 to 7 to 0.

16.3.2 Serial clock select register 4 (SPS4)

The SPS4 register is a 16-bit register that is used to select operation clock (CK40) that are commonly supplied to each channel.

Rewriting the SPS4 register is prohibited when the register is in operation (when SE4n = 1).

When SPS4 = 0003H to 000FH is selected as the operating clock, the frequency of the selected clock remains the same even when the PLL function is changed from “Use” to “Do not use”. Therefore, when switching the CPU from normal operation mode to standby mode, the clock supplied to serial array unit 4 can be allowed to continue and remain receivable and the PLL function use setting can be changed.

The SPS4 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SPS4 register can be set with an 8-bit memory manipulation instruction with SPS4L.

Reset signal generation clears the SPS4 register to 0000H.

Figure 16-6. Format of Serial Clock Select Register 4 (SPS4)

Address: F05A6H, F05A7H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPS4	0	0	0	0	0	0	0	0	0	0	0	0	PRS 403	PRS 402	PRS 401	PRS 400

PRS 403	PRS 402	PRS 401	PRS 400	Section of operation clock (CK40)[Note]					
				f_{CLK} = 4 MHz (not using PLL)		f_{CLK} = 16 MHz (using PLL)		f_{CLK} = 32 MHz (using PLL)	
0	0	0	0	f_{CLK}	4 MHz	f_{CLK}	16 MHz	f_{CLK}	32 MHz
0	0	0	1	f_{CLK}	4 MHz	f_{CLK}	16 MHz	$f_{CLK}/2$	16 MHz
0	0	1	0	f_{CLK}	4 MHz	$f_{CLK}/2$	8 MHz	$f_{CLK}/2^2$	8 MHz
0	0	1	1	f_{CLK}	4 MHz	$f_{CLK}/2^2$	4 MHz	$f_{CLK}/2^3$	4 MHz
0	1	0	0	$f_{CLK}/2$	2 MHz	$f_{CLK}/2^3$	2 MHz	$f_{CLK}/2^4$	2 MHz
0	1	0	1	$f_{CLK}/2^2$	1 MHz	$f_{CLK}/2^4$	1 MHz	$f_{CLK}/2^5$	1 MHz
0	1	1	0	$f_{CLK}/2^3$	500 kHz	$f_{CLK}/2^5$	500 kHz	$f_{CLK}/2^6$	500 kHz
0	1	1	1	$f_{CLK}/2^4$	250 kHz	$f_{CLK}/2^6$	250 kHz	$f_{CLK}/2^7$	250 kHz
1	0	0	0	$f_{CLK}/2^5$	125 kHz	$f_{CLK}/2^7$	125 kHz	$f_{CLK}/2^8$	125 kHz
1	0	0	1	$f_{CLK}/2^6$	62.5 kHz	$f_{CLK}/2^8$	62.5 kHz	$f_{CLK}/2^9$	62.5 kHz
1	0	1	0	$f_{CLK}/2^7$	31.3 kHz	$f_{CLK}/2^9$	31.3 kHz	$f_{CLK}/2^{10}$	31.3 kHz
1	0	1	1	$f_{CLK}/2^8$	15.6 kHz	$f_{CLK}/2^{10}$	15.6 kHz	$f_{CLK}/2^{11}$	15.6 kHz
1	1	0	0	$f_{CLK}/2^9$	7.81 kHz	$f_{CLK}/2^{11}$	7.81 kHz	$f_{CLK}/2^{12}$	7.81 kHz
1	1	0	1	$f_{CLK}/2^{10}$	3.91 kHz	$f_{CLK}/2^{12}$	3.91 kHz	$f_{CLK}/2^{13}$	3.91 kHz
1	1	1	0	$f_{CLK}/2^{11}$	1.95 kHz	$f_{CLK}/2^{13}$	1.95 kHz	$f_{CLK}/2^{14}$	1.95 kHz
1	1	1	1	$f_{CLK}/2^{12}$	977 Hz	$f_{CLK}/2^{14}$	977 Hz	$f_{CLK}/2^{15}$	977 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 4 (ST4) = 000FH) the operation of the serial array unit 4 (DALI/UART4).

Cautions 1. Be sure to clear bits 15 to 4 to "0".

2. After setting bit 0 (DALIEN) of the PER1 register to 1, be sure to set serial clock select register 4 (SPS4) after 4 or more fCLK clocks have elapsed.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

f_{SUB}: Subsystem clock frequency

16.3.3 Serial mode register 4n (SMR4n)

The SMR4n register is a register that sets an operation mode of channel n. It is also used to set a start trigger and select an interrupt source. This register is also used to invert the level of the receive data.

Rewriting the SMR4n register is prohibited when the register is in operation (when SE4n = 1). However, the MD4n0 bit can be rewritten during operation.

The SMR4n register can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets the SMR4n register to 0020H.

Figure 16-7. Format of Serial Mode Register 4n (SMR4n)

Address: F0590H, F0591H (SMR40), F0592H, F0593H (SMR41) After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR4n	0	0	0	0	0	0	0	STS 4n[Note]	0	SIS 4n0[Note]	1	0	0	0	1	MD 4n0

STS 4n[Note]	Selection of start trigger source
0	Only software trigger is valid (selected for UART transmission).
1	Valid edge of the DALIRxD4 pin (selected for UART reception)
Transfer is started when the above source is satisfied after 1 is set to the SS4 register.	

SIS 4n0[Note]	Controls inversion of level of receive data of channel n in UART mode
0	Falling edge is detected as the start bit. The input communication data is captured as is.
1	Rising edge is detected as the start bit. The input communication data is inverted and captured.

MD 4n0	Selection of interrupt source of channel n
0	Transfer end interrupt
1	Buffer empty interrupt (Occurs when data is transferred from the SDR4n register to the shift register.)
For successive transmission, the next transmit data is written by setting the MD4n0 bit to 1 when SDR4n data has run out.	

Note The SMR41 register only.

Caution Be sure to clear bits 15 to 9, 7, 4 to 2 (or bits 15 to 6, 4 to 2 for the SMR40 register) to "0". Be sure to set bits 5 and 1 to "1".

Remark n: Channel number (n = 0, 1), q: UART number (q = 4)

16.3.4 Serial communication operation setting register 4n (SCR4n)

The SCR4n register is a communication operation setting register of channel n. It is used to set a data transmission/reception mode, whether an error signal is to be masked or not, parity bit, start bit, stop bit, and data length.

Rewriting the SCR4n register is prohibited when the register is in operation (when SE4n = 1).

The SCR4n register can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets the SCR4n register to 0087H.

Figure 16-8. Format of Serial Communication Operation Setting Register 4n (SCR4n) (1/2)

Address: F0598H, F0599H (SCR40), F059AH, F059BH (SCR41) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR4n	TXE 4n	RXE 4n	0	0	0	EOC 4n	PTC 4n1	PTC 4n0	DIR 4n	0	SLC 4n1 Note 1	SLC 4n0	0	1	DLS 4n1	DLS 4n0

TXE 4n	RXE 4n	Setting of operation mode of channel n
0	0	Disable communication.
0	1	Reception only (setting for SCR41)
1	0	Transmission only (setting for SCR40)
1	1	Transmission/reception

EOC 4n	Selection of masking of error interrupt signal (INTSREDL4)
0	Masks error interrupt INTSREDL4 (INTSRDL4 is not masked).
1	Enables generation of error interrupt INTSREDL4 (INTSRDL4 is masked if an error occurs).
Set EOC40 = 0 during UART transmission.	

PTC 4n1	PTC 4n0	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 2.	No parity judgment
1	0	Outputs even parity.	Judged as even parity.
1	1	Outputs odd parity.	Judges as odd parity.

Notes 1. The SCR40 register only.

2. 0 is always added regardless of the data contents.

Caution Be sure to clear bits 3, 6, 11 to 13 (or bits 3, 5, 6, 11 to 13 for the SCR41 register) to "0". Be sure to set bit 2 to "1".

Remark n: Channel number (n = 0, 1)

Figure 16-8. Format of Serial Communication Operation Setting Register 4n (SCR4n) (2/2)

Address: F0598H, F0599H (SCR40), F059AH, F059BH (SCR41) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR4n	TXE 4n	RXE 4n	0	0	0	EOC 4n	PTC 4n1	PTC 4n0	DIR 4n	0	SLC 4n1Note	SLC 4n0	0	1	DLS 4n1	DLS 4n0

DIR 4n	Selection of data transfer sequence
0	Inputs/outputs data with MSB first.
1	Inputs/outputs data with LSB first.

SLC 4n1Note	SLC 4n0	Setting of stop bit
0	0	No stop bit
0	1	Stop bit length = 1 bit
1	0	Stop bit length = 2 bits (SLC401, SLC400 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 (SLC4n1, SLC4n0 = 0, 1) during UART reception.		

DLS 4n1	DLS 4n0	Setting of data length	
		UART communication mode	DALI communication mode
0	0	Setting prohibited	24-bit data length (stored in bits 7 to 0 of SDCH4 and SDTH4, in bits 15 to 0 of SDCL4 and SDTL4)
0	1	9-bit data length (stored in bits 0 to 8 of SDR4n (mn = 40, 41))	17-bit data length (stored in bit 0 of SDCH4, in bits 15 to 0 of SDCL4)
1	0	7-bit data length (stored in bits 0 to 6 of SDR4n)	16-bit data length (stored in bits 15 to 0 of SDCL4 and SDTL4)
1	1	8-bit data length (stored in bits 0 to 7 of SDR4n)	8-bit data length (stored in bits 7 to 0 of SDTL4)

Note The SCR40 register only.

Caution Be sure to clear bits 3, 6, 11 to 13 (or bits 3, 5, 6, 11 to 13 for the SCR41 register) to "0". Be sure to set bit 2 to "1".

Remark n: Channel number (n = 0, 1)

16.3.5 Higher 7 bits of the serial data register 4n (SDR4n)

The SDR4n register is the transmit/receive data register (16 bits) of channel n. Bits 8 to 0 (lower 9 bits) of SDR4n register 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}).

The clock set by dividing the operating clock by the higher 7 bits of the SDR4n register is used as the transfer clock.

The lower 9 bits of the SDR4n register function as a transmit/receive buffer register. During reception, the parallel data converted by the shift register is stored in the lower 9 bits, and during transmission, the data to be transmitted to the shift register is set to the lower 9 bits.

The SDR4n register can be read or written in 16-bit units.

However, the higher 7 bits can be written or read only when the operation is stopped (SE4n = 0). During operation (SE4n = 1), a value is written only to the lower 9 bits of the SDR4n register. When the SDR4n register is read during operation, 0 is always read.

Reset signal generation clears the SDR4n register to 0000H.

Figure 16-9. Format of Serial Data Register 4n (SDR4n)

Address: F0570H, F0571H (SDR40), F0572H, F0573H (SDR41) After reset: 0000H R/W

	F0570H (SDR40)								F0571H (SDR40)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR4n																

SDR4n[15:9]							Transfer clock setting by dividing the operating clock (f_{MCK})
0	0	0	0	0	0	0	f_{MCK}
0	0	0	0	0	0	1	$f_{MCK}/2$
0	0	0	0	0	1	0	$f_{MCK}/3$
0	0	0	0	0	1	1	$f_{MCK}/4$
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	$f_{MCK}/127$
1	1	1	1	1	1	1	$f_{MCK}/128$

Cautions 1. Setting SDR4n[15:9] = (0000000B, 0000001B, 0000010B) is prohibited.

2. Do not write eight bits to the lower eight bits if operation is stopped (SE4n = 0). (If these bits are written to, the higher seven bits are cleared to 0.)

3. If data is written to the SDR4n register when BFF4n = 1, the transmit/receive data stored in the register is destroyed and an overrun error (OVE4n = 1) is detected.

Remarks 1. For the function of the lower 9 bits of the SDR4n register, see **16.2 Configuration of Serial Array Unit 4 (DALI/UART4)**.

2. n: Channel number (n = 0, 1)

16.3.6 Serial status register 4n (SSR4n)

The SSR4n register is a register that indicates the communication status and error occurrence status of channel n. The errors indicated by this register are a framing error, manchester framing error, parity error, and overrun error.

The SSR4n register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSR4n register can be set with an 8-bit memory manipulation instruction with SSR4nL.

Reset signal generation clears the SSR4n register to 0000H.

Figure 16-10. Format of Serial Status Register 4n (SSR4n) (1/3)

Address: F0580H, F0581H (SSR40), F0582H, F0583H (SSR41) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSR4n	0	0	0	0	0	0	0	0	MFEF 4n[Note]	TSF 4n	BFF 4n	0	0	FEF 4n[Note]	PEF 4n[Note]	OVF 4n

MFEF 4n[Note]	Manchester framing error detection flag of channel n
0	No error occurs.
1	An error occurs (during DALI reception)
<Condition of clear> 1 is written to MFECT4n bit of SIR4n register <Condition of set> A stop bit is not detected when DALI reception ends.	

TSF4n	Communication status indication flag of channel n
0	Communication is stopped or suspended.
1	Communication is in progress.
<Clear conditions> • The ST4n bit of the ST4 register is set to 1 (communication is stopped) or the SS4n bit of the SS4 register is set to 1 (communication is suspended). • Communication ends. <Set condition> • Communication starts.	

Note The SSR41 register only.

Caution If data is written to the SDR4n register when BFF4n = 1, the transmit/receive data stored in the register is destroyed and an overrun error (OVE4n = 1) is detected.

Remark n: Channel number (n = 0, 1)

Figure 16-10. Format of Serial Status Register 4n (SSR4n) (2/3)

Address: F0580H, F0581H (SSR40), F0582H, F0583H (SSR41) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSR4n	0	0	0	0	0	0	0	0	MFEF 4n[Note]	TSF 4n	BFF 4n	0	0	FEF 4n[Note]	PEF 4n[Note]	OVF 4n

BFF 4n	Buffer register status indication flag of channel n
0	Valid data is not stored in the SDR4n register.
1	Valid data is stored in the SDR4n register.

<Clear conditions>

- Transferring transmit data from the SDR4n register to the shift register ends during transmission.
- Reading receive data from the SDR4n register ends during reception.
- The ST4n bit of the ST4 register is set to 1 (communication is stopped) or the SS4n bit of the SS4 register is set to 1 (communication is enabled).

<Set conditions>

- Transmit data is written to the SDR4n register while the TXE4n bit of the SCR4n register is set to 1 (transmission or transmission and reception mode in each communication mode).
- Receive data is stored in the SDR4n register while the RXE4n bit of the SCR4n register is set to 1 (reception or transmission and reception mode in each communication mode).
- A reception error occurs.

FEF 4n[Note]	Framing error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception).

<Clear condition>

- 1 is written to the FECT4n bit of the SIR4n register.

<Set condition>

- A stop bit is not detected when UART reception ends.

PEF4 n	Parity error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception)

<Clear condition>

- 1 is written to the PECT4n bit of the SIR4n register.

<Set condition>

- The parity of the transmit data and the parity bit do not match when UART reception ends (parity error).

Note The SSR41 register only.

Remark n: Channel number (n = 0, 1)

Figure 16-10. Format of Serial Status Register 4n (SSR4n) (3/3)

Address: F0580H, F0581H (SSR40), F0582H, F0583H (SSR41) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSR4n	0	0	0	0	0	0	0	0	MFEF 4n[Note]	TSF 4n	BFF 4n	0	0	FEF 4n[Note]	PEF 4n[Note]	OVF 4n

OVF 4n	Overrun error detection flag of channel n
0	No error occurs.
1	An error occurs
<Clear condition> • 1 is written to the OVCT4n bit of the SIR4n register. <Set condition> • Even though receive data is stored in the SDR4n register, that data is not read and transmit data or the next receive data is written while the RXE4n bit of the SCR4n register is set to 1 (reception or transmission and reception mode in each communication mode).	

Note The SSR41 register only.

Remark n: Channel number (n = 0, 1)

16.3.7 Serial flag clear trigger register 4n (SIR4n)

The SIR4n register is a trigger register that is used to clear each error flag of channel n.

When each bit (MFECT4n, FECT4n, PECT4n, OVCT4n) of this register is set to 1, the corresponding bit (MFEF4n, FEF4n, PEF4n, OVF4n) of serial status register 4n is cleared to 0. Because the SIR4n register is a trigger register, it is cleared immediately when the corresponding bit of the SSR4n register is cleared.

The SIR4n register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SIR4n register can be set with an 8-bit memory manipulation instruction with SIR4nL.

Reset signal generation clears the SIR4n register to 0000H.

Figure 16-11. Format of Serial Flag Clear Trigger Register 4n (SIR4n)

Address: F0588H, F0589H (SIR40), F058AH, F058BH (SIR41) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIR4n	0	0	0	0	0	0	0	0	MFECT 4n[Note]	0	0	0	0	FECT 4n[Note]	PECT 4n[Note]	OVCT 4n

MFECT 4n[Note]	Clear trigger of manchester framing error of channel n
0	Not cleared
1	Clears the MFEF4n bit of the SSR4n register to 0.

FECT 4n[Note]	Clear trigger of framing error of channel n
0	Not cleared
1	Clears the FEF4n bit of the SSR4n register to 0.

PECT 4n[Note]	Clear trigger of parity error flag of channel n
0	Not cleared
1	Clears the PEF4n bit of the SSR4n register to 0.

OVCT 4n	Clear trigger of overrun error flag of channel n
0	Not cleared
1	Clears the OVF4n bit of the SSR4n register to 0.

Note The SIR41 register only.

Caution Be sure to clear bits 15 to 8, 6 to 3 (or bits 15 to 1 for the SIR40 register) to "0".

Remarks 1. n: Channel number (n = 0, 1)

2. When the SIR4n register is read, 0000H is always read.

16.3.8 Serial channel start register 4 (SS4)

The SS4 register is a trigger register that is used to enable starting communication/count by each channel.

When 1 is written a bit of this register (SS4n), the corresponding bit (SE4n) of serial channel enable status register 4 (SE4) is set to 1 (Operation is enabled). Because the SS4n bit is a trigger bit, it is cleared immediately when SE4n = 1.

The SS4 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SS4 register can be set with an 1-bit or 8-bit memory manipulation instruction with SS4L.

Reset signal generation clears the SS4 register to 0000H.

Figure 16-12. Format of Serial Channel Start Register 4 (SS4)

Address: F05A2H, F05A3H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS41	SS40

SS4n	Operation start trigger of channel n
0	No trigger operation
1	Sets the SE4n bit to 1 and enters the communication wait status[Note].

Note If set the SS4n = 1 to during a communication operation, will wait status to stop the communication. At this time, holding status value of control register and shift register, DALITxD4 and TxRx4 pins, and FEF4n, PEF4n, OVF4n flags.

Cautions 1. Be sure to clear bits 15 to 2 to "0".

2. For the UART reception, set the RXE4n bit of SCR4n register to 1, and then be sure to set SS4n to 1 after 4 or more f_{MCK} clocks have elapsed.

Remarks 1. n: Channel number (n = 0, 1)

2. When the SS4 register is read, 0000H is always read.

16.3.9 Serial channel stop register 4 (ST4)

The ST4 register is a trigger register that is used to enable stopping communication/count by each channel.

When 1 is written a bit of this register (ST4n), the corresponding bit (SE4n) of serial channel enable status register 4 (SE4) is cleared to 0 (operation is stopped). Because the ST4n bit is a trigger bit, it is cleared immediately when SE4n = 0.

The ST4 register can set written by a 16-bit memory manipulation instruction.

The lower 8 bits of the ST4 register can be set with a 1-bit or 8-bit memory manipulation instruction with ST4L.

Reset signal generation clears the ST4 register to 0000H.

Figure 16-13. Format of Serial Channel Stop Register 4 (ST4)

Address: F05A4H, F05A5H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ST41	ST40

ST4n	Operation stop trigger of channel n
0	No trigger operation
1	Clears the SE4n bit to 0 and stops the communication operation[Note].

Note Holding status value of the control register and shift register, the DALITxD4 and TxRx4 pins, and FEF4n, PEF4n, OVF4n flags.

Caution Be sure to clear bits 15 to 2 to "0".

Remarks 1. n: Channel number (n = 0, 1)

2. When the ST4 register is read, 0000H is always read.

16.3.10 Serial channel enable status register 4 (SE4)

The SE4 register indicates whether data transmission/reception operation of each channel is enabled or stopped.

When 1 is written a bit of serial channel start register 4 (SS4), the corresponding bit of this register is set to 1. When 1 is written a bit of serial channel stop register 4 (ST4), the corresponding bit is cleared to 0.

The SE4 register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the SE4 register can be set with a 1-bit or 8-bit memory manipulation instruction with SE4L.

Reset signal generation clears the SE4 register to 0000H.

Figure 16-14. Format of Serial Channel Enable Status Register 4 (SE4)

Address: F05A0H, F05A1H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SE41	SE40

SE4n	Indication of operation enable/stop status of channel n
0	Operation stops
1	Operation is enabled.

Remark n: Channel number (n = 0, 1)

16.3.11 Serial output enable register 4 (SOE4)

The SOE4 register is a register that is used to enable or stop output of the serial communication operation of channel 0.

Channel 0 that enables serial output cannot rewrite by software the value of the SO40 bit of serial output register 4 (SO4) to be described below, and a value reflected by a communication operation is output from the serial data output pin.

For channel 0, whose serial output is stopped, the SO40 bit value of the SO4 register can be set by software, and that value can be output from the serial data output pin.

The SOE4 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SOE4 register can be set with a 1-bit or 8-bit memory manipulation instruction with SOE4L.

Reset signal generation clears the SOE4 register to 0000H.

Figure 16-15. Format of Serial Output Enable Register 4 (SOE4)

Address: F05AAH, F05ABH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE40

SOE40	Serial output enable/stop of channel 0
0	Stops output by serial communication operation.
1	Enables output by serial communication operation.

Caution Be sure to clear bits 15 to 1 to “0”.

16.3.12 Serial output register 4 (SO4)

The SO4 register is a buffer register for serial output of channel 0.

The value of the SO40 bit of this register is output from the serial data output pin of channel 0.

The SO40 bit of this register can be rewritten by software only when serial output is disabled (SOE40 = 0). When serial output is enabled (SOE40 = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.

To use the pin for serial interface 4 as a port function pin, set the corresponding SO40 bit to “1”.

The SO4 register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears the SO4 register to 0303H.

Figure 16-16. Format of Serial Output Register 4 (SO4)

Address: F05A8H, F05A9H After reset: 0303H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO4	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	SO40

SO40	Serial data output of channel 0
0	Serial data output value is “0”.
1	Serial data output value is “1”.

Caution Be sure to set bits 9, 8, 1 to “1”. And be sure to clear bits 15 to 12, 7 to 4 to “0”.

16.3.13 Serial output level register 4 (SOL4)

The SOL4 register is a register that is used to set inversion of the data output level of channel 0.

Inverting channel n by using this register is reflected on pin output only when serial output is enabled (SOE40 = 1). When serial output is disabled (SOE40 = 0), the value of the SO40 bit is output as is.

Rewriting the SOL4 register is prohibited when the register is in operation (when SE40 = 1).

The SOL4 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SOL4 register can be set with an 8-bit memory manipulation instruction with SOL4L.

Reset signal generation clears the SOL4 register to 0000H.

Figure 16-17. Format of Serial Output Level Register 4 (SOL4)

Address: F05B4H, F05B5H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL40

SOL40	Selects inversion of the level of the transmit data of channel 0
0	Communication data is output as is.
1	Communication data is inverted and output.

Caution Be sure to clear bits 15 to 1 to “0”.

16.3.14 Serial standby control register 4 (SSC4)

The SSC4 register is used to control the startup of reception (the SNOOZE mode) while in the STOP mode when receiving UART4 serial data.

The SSC4 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSC4 register can be set with an 8-bit memory manipulation instruction with SSC4L.

Reset signal generation clears the SSC4 register to 0000H.

Cautions 1. SNOOZE mode can be set only when the internal high-speed oscillation clock has been selected for f_{CLK}. Also, when in SNOOZE mode, PLL output cannot be used.
After returning from SNOOZE mode to normal operation mode, PLL output can be used. In such cases, an error ranging from +1.125 μsec (max.) to –0.406 μsec may occur (f_{CLK} = 32 MHz when using PLL ↔ 4 MHz).

2. Max. transfer rate is 4800 bps when UART is used in SNOOZE mode.

Figure 16-18. Format of Serial Standby Control Register 4 (SSC4)

Address: F05B8H, F05B9H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSC4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSEC4	SWC4

SSEC4	Selection of whether to enable or stop the generation of transfer end interrupts
0	Enable the generation of error interrupts (INTSREDL4). In the following cases, the clock request signal (an internal signal) to the clock generator is also cleared: • When the SWC4 bit is cleared to 0 • When the DALI/UART4 reception start bit is mistakenly detected
1	Stop the generation of error interrupts (INTSREDL4). In the following cases, the clock request signal (an internal signal) to the clock generator is also cleared: • When the SWC4 bit is cleared to 0 • When the DALI/UART4 reception start bit is mistakenly detected • When the transfer end interrupt generation timing is based on a parity error or framing error

SWC4	Selection of whether to enable or stop the startup of DALI/UART4 reception while in the STOP mode
0	Do not use the SNOOZE mode function.
1	Use the SNOOZE mode function.
• When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and A/D conversion is performed without operating the CPU (the SNOOZE mode). • The SNOOZE mode function can only be specified when the high-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (f_{CLK}). If any other clock is selected, specifying this mode is prohibited. • Even when using SNOOZE mode, be sure to set the SWC4 bit to 0 in normal operation mode and change it to 1 just before shifting to STOP mode. Also, be sure to change the SWC4 bit to 0 after returning from STOP mode to normal operation mode.	

16.3.15 Serial option control register 4 (SOC4)

The SOC4 register is used to control the DALI/UART4 communication mode.

The SOC4 register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears the SOC4 register to 0000H.

Figure 16-19. Format of Serial Option Control Register 4 (SOC4)

Address: F05BAH, F05BBH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOC4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SDS

SDS	Selection of DALI/UART mode
0	UART mode
1	DALI mode

16.3.16 Single-wire UART control register (SUCTL)

This register is used to set the DALI/UART4 communication method.

When the SUCTL bit is set to 1, single-wire DALI/UART4 communications can be executed.

When the single-wire method is selected, the TxRx4/TKCO05/DALIRxD4/INTP23/P206 pin is shared for transmission and reception.

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

Reset signal generation clears the SUCTL register to 00H.

Figure 16-20. Format of Single-Wire UART Control Register (SUCTL)

Address: F05C7H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SUCTL	0	0	0	0	0	0	0	SUCTL0

SUCTL0	DALI/UART4 communication method switching
0	2-wire data communication (using DALIRxD4 and DALITxD4 pins)
1	Single-wire data communication (using TxRx4 pin)

16.3.17 Noise filter enable register 3 (NFEN3)

The NFEN3 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.

When the noise filter is enabled, CPU/peripheral hardware clock (f_{CLK}) is synchronized with 2-clock match detection.

The NFEN3 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the NFEN3 register to 00H.

Figure 16-21. Format of Noise Filter Enable Register 3 (NFEN3)

Address: F05C1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN3	0	0	0	0	0	0	0	SNFEN30

SNFEN30	Use of noise filter of DALIRxD4 pin (DALIRxD4/TxRx4/TKCO05/INTP23/P206)
0	Noise filter OFF
1	Noise filter ON
Set SNFEN30 to 1 to use the DALIRxD4 pin. Clear SNFEN30 to 0 to use the other than DALIRxD4 pin.	

Caution Be sure to clear bits 7 to 1 to "0".

16.3.18 Port output mode registers 1, 20 (POM1, POM20)

This register set the output mode of port 1, 20 in 1-bit units.

The POM1 and POM20 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

In addition, POM1, POM20 register is set with PUxx register, whether or not to use the on-chip pull-up resistor.

Reset signal generation clears the POM1 and POM20 registers to 00H.

Figure 16-22. Format of Port Output Mode Registers 1 and 20 (POM1 and POM20)

Address F0051H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
POM1	0	0	0	0	0	POM12	POM11	POM10

Address F0530H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
POM20	0	POM206	POM205	POM204	POM203	POM202	POM201	POM200

POMmn	Pmn pin output buffer selection (m = 1, 20, n = 0 to 6)
0	Normal output mode When the input, enable to the PUmn bit
1	N-ch open-drain output (V_{DD} tolerance) mode When the input, disable to the PUmn bit

16.3.19 Port mode registers 1, 20 (PM1, PM20)

This register sets input/output of port 1, 20 in 1-bit units.

When using the port (PM10/SO00/TxD0/TKCO00/INTP20/SCLA0/(DALITxD4), P205/DALITxD4/TKBO21/TKCO04) to be shared with the serial data output pin for serial data output, set the PM10 and PM205 bits of the port mode registers (PM1, PM20) to 0. And set the P10 and P205 bits of the port register (P1, P20) to 1

When using the port (P11/SI00/RxD0/TKCO01/INTP21/SDAA0/(TI07)/(DALIRxD4)/(TxRx4), P206/DALIRxD4/TxRx4/TKCO05/INTP23) to be shared with the serial data input pin for serial data input, set the PM11 and PM 206 bits of the port mode register (PM1, PM20) bit corresponding to each port to 1. At this time, the P11 and P206 bits of the port register (P1, P20) bit may be 0 or 1.

The PM1, PM20 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets the PM1, PM20 register to FFH.

Figure 16-23. Format of Port Mode Register 1, 20 (PM1, PM20) (38-pin Products)

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	1	1	1	1	1	PM12	PM11	PM10

Address: F0510H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM20	1	PM206	PM205	PM204	PM203	PM202	PM201	PM200

PMmn	Pmn pin I/O mode selection (m = 1, 20, n = 0 to 6)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bits 3 to 7 of the PM1 register, and bit 7 of the PM20 register to “1”.
For 30- and 20-pin products, the following bits must be set to output mode (specified by setting the port registers and port mode registers to 0) by software after reset is cancelled.
30-pin products: Bit 2 of the PM1 register
20-pin products: Bit 2 of the PM1 register, and bits 4 to 6 of the PM20 register

16.3.20 Peripheral I/O redirection register (PIOR1)

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

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

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

Reset signal generation sets this register to 00H.

Figure 16-24. Format of Peripheral I/O Redirection Register (PIOR1)

Address: F05C0H After reset: 00H R/W

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

Bit	Function	20-pin		30-pin		38-pin	
		Setting value		Setting value		Setting value	
		0	1	0	1	0	1
PIOR11	DALITxD4/ DALIRxD4	–	P10/P11	P205/ P206	P10/P11	P205/ P206	P10/P11

16.4 Operation Stop Mode

Serial array unit 4 (DALI/UART4) 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 4 can be used as port function pins in this mode.

16.4.1 Stopping the operation by units

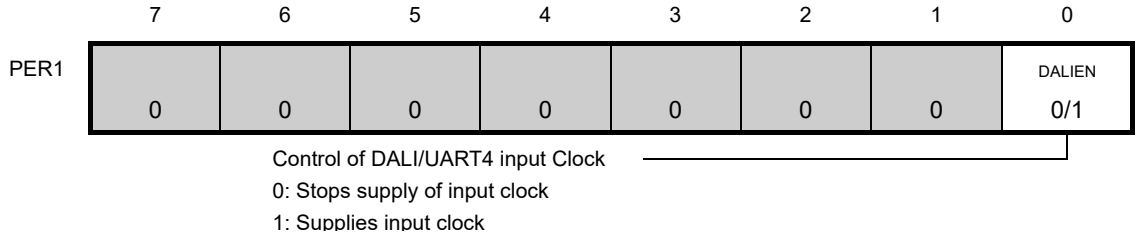
The stopping of the operation by units is set by using peripheral enable register 1 (PER1).

The PER1 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

To stop the operation of serial array unit 4, set bit 0 (DALIEN) to 0.

Figure 16-25. Peripheral Enable Register 1 (PER1) Setting When Stopping the Operation by Units

(a) Peripheral enable register 1 (PER1)



Cautions 1. If DALIEN = 0, writing to a control register of serial array unit 4 (DALI/UART4) is ignored, and, even if the register is read, only the default value is read
Note that this does not apply to the following registers.
- **Noise filter enable register 3 (NFEN3)**
- **Serial standby control register 4 (SSC4)**
- **Port output mode register 20 (POM20)**
- **Port mode register 20 (PM20)**
- **Port register 20 (P20)**

2. Be sure to clear bits 1 to 7 to "0".

Remark ☐: Setting disabled (fixed by hardware)
0/1: Set to 0 or 1 depending on the usage of the user

16.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

Figure 16-26. Each Register Setting When Stopping the Operation by Channels

(a) Serial channel stop register 4 (ST4) … This register is a trigger register that is used to enable stopping communication/count by each channel.

ST4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	ST41 0/1	ST40 0/1

1: Clears the SE4n bit to 0 and stops the communication operation

* Because the ST4n bit is a trigger bit, it is cleared immediately when SE4n = 0.

(b) Serial Channel Enable Status Register 4 (SE4) … This register indicates whether data transmission/ reception operation of each channel is enabled or stopped.

SE4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	SE41 0/1	SE40 0/1

0: Operation stops

* The SE4 register is a read-only status register, whose operation is stopped by using the ST4 register.

(c) Serial output enable register 4 (SOE4) … This register is a register that is used to enable or stop output of the serial communication operation of each channel.

SOE4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE40 0/1

0: Stops output by serial communication operation

* For channel n, whose serial output is stopped, the SO4n bit value of the SO4 register can be set by software.

(d) Serial output register 4 (SO4) …This register is a buffer register for serial output of each channel.

SO4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	SO40 0/1

“ 1: Serial data output value is “1”

* When using pins corresponding to each channel as port function pins, set the corresponding SO4n bits to “1”.

Remarks 1. n: Channel number (n = 0, 1)

2. ☐ : Setting disabled (fixed by hardware), 0/1: Set to 0 or 1 depending on the usage of the user

16.5 Communication Operation of UART

16.5.1 UART transmission

UART transmission is an operation to transmit data from the RL78/I1A to another device asynchronously (start-stop synchronization).

Of two channels used for UART, the even channel is used for UART transmission.

UART transmission	
Target channel	Channel 0
Pins used	DALITxD4
Transfer data length	7, 8, or 9 bits
Transfer rate	Max. f_{MCK}/6 [bps] (SDR4n [15:9] = 3 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps]**Note**
Data direction	MSB or LSB first
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)
Parity bit	The following selectable • No parity bit • Appending 0 parity • Appending even parity • Appending odd parity
Stop bit	The following selectable • Appending 1 bit • Appending 2 bits
Error detection flag	None
Interrupt	INTSTDL4
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.

Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C)** or **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T_A = −40 to +125°C)**).

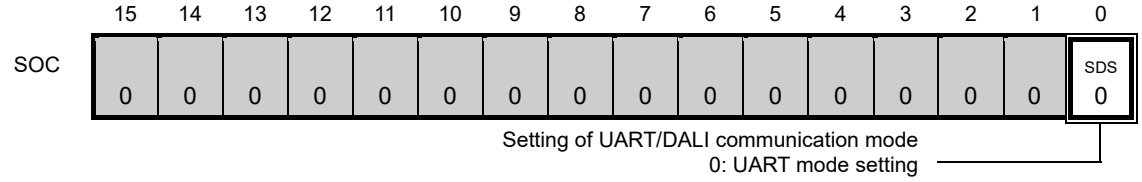
Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

2. n: Channel number (n = 0)

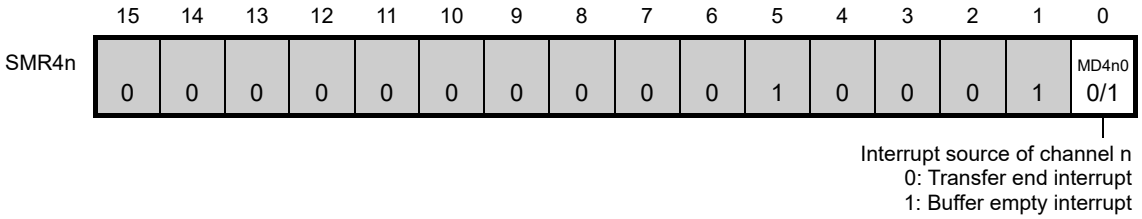
(1) Register setting

Figure 16-27. Example of Contents of Registers for UART Transmission (1/2)

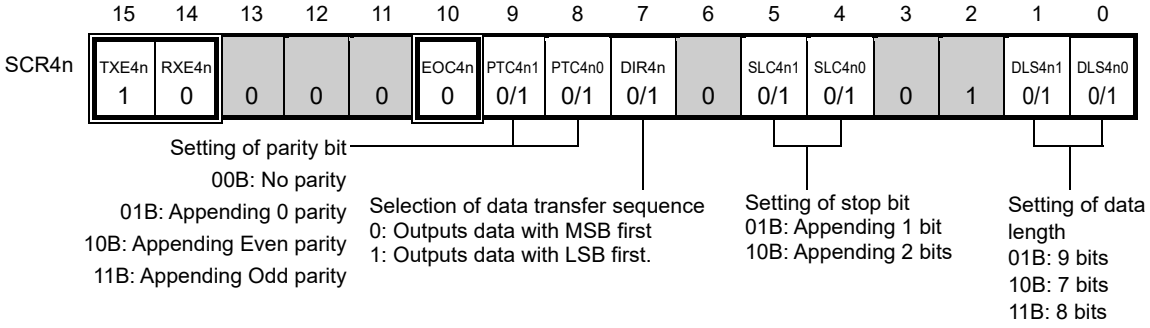
(a) Serial option control register 4 (SOC4)



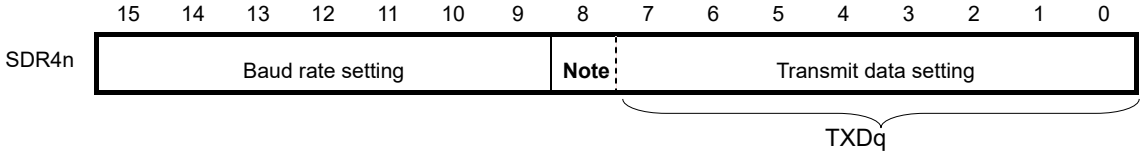
(b) Serial mode register 4n (SMR4n)



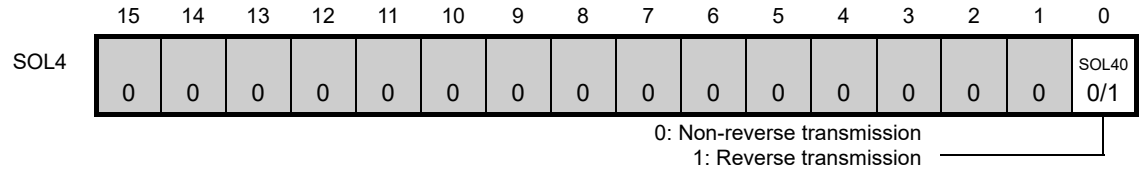
(c) Serial communication operation setting register 4n (SCR4n)



(d) Serial data register 4n (SDR4n) (lower 8 bits: TXDq)



(e) Serial output level register 4 (SOL4) … Sets only the bits of the target channel.



Note When performing 9-bit data length communication (DLS401, DLS400 = 0, 1), bits 0 to 8 of the SDR40 register are used as the reception data specification area.

Remarks 1. n: Channel number (n = 0), q: UART number (q = 4),
4n = 00, 02, 10

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

Figure 16-27. Example of Contents of Registers for UART Transmission (2/2)

(f) Serial output register 4 (SO4) … Sets only the bits of the target channel.

SO4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	SO40 0/1 Note

0: Serial data output value is “0”
1: Serial data output value is “1”

(g) Serial output enable register 4 (SOE4) … Sets only the bits of the target channel to 1.

SOE4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE40 0/1

(h) Serial channel start register 4 (SS4) … Sets only the bits of the target channel to 1.

SS4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS41 ×	SS40 0/1

Note Before transmission is started, be sure to set to 1 when the SOL4n bit of the target channel is set to 0, and set to 0 when the SOL4n bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

Remarks 1. n: Channel number (n = 0)

2. ☐: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 16-28. Initial Setting Procedure for UART Transmission

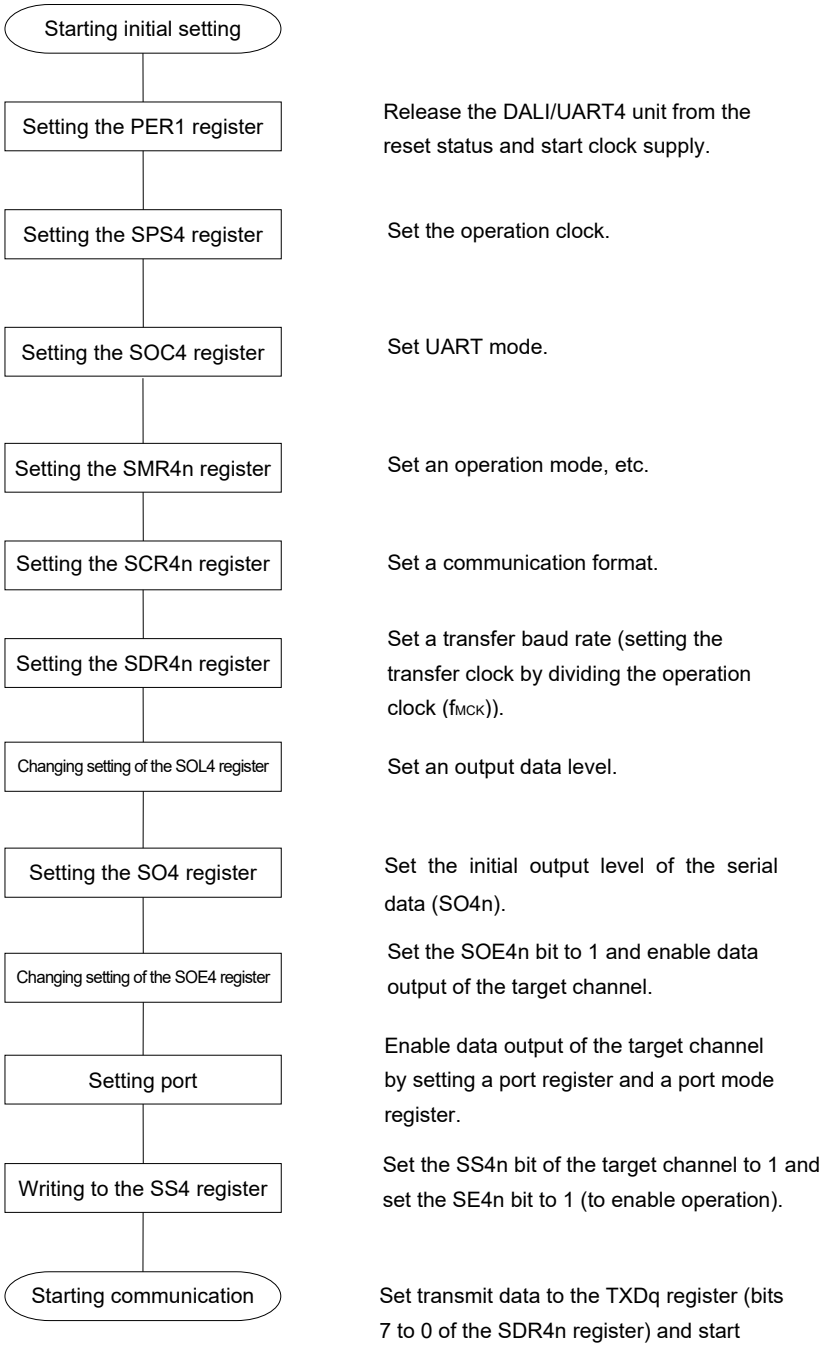


Figure 16-29. Procedure for Stopping UART Transmission

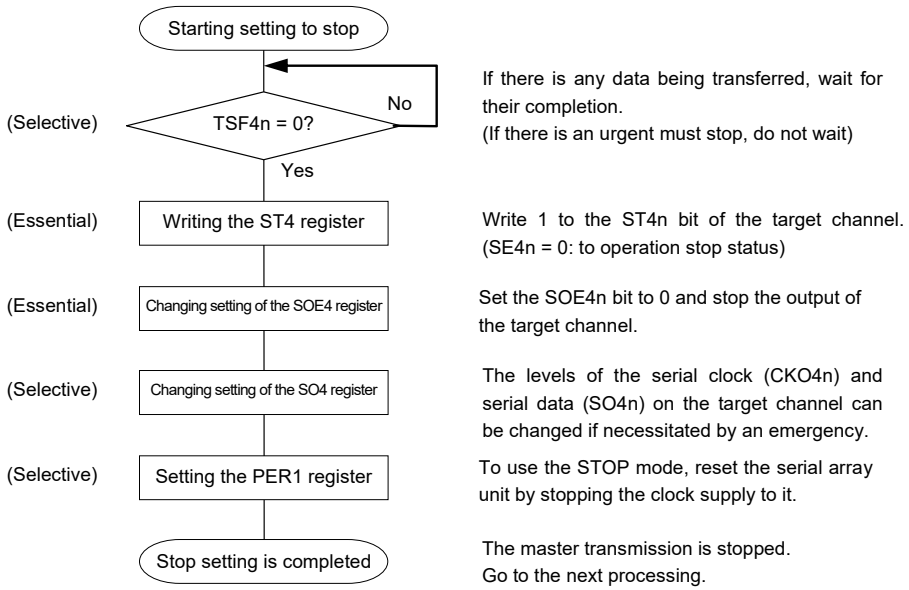
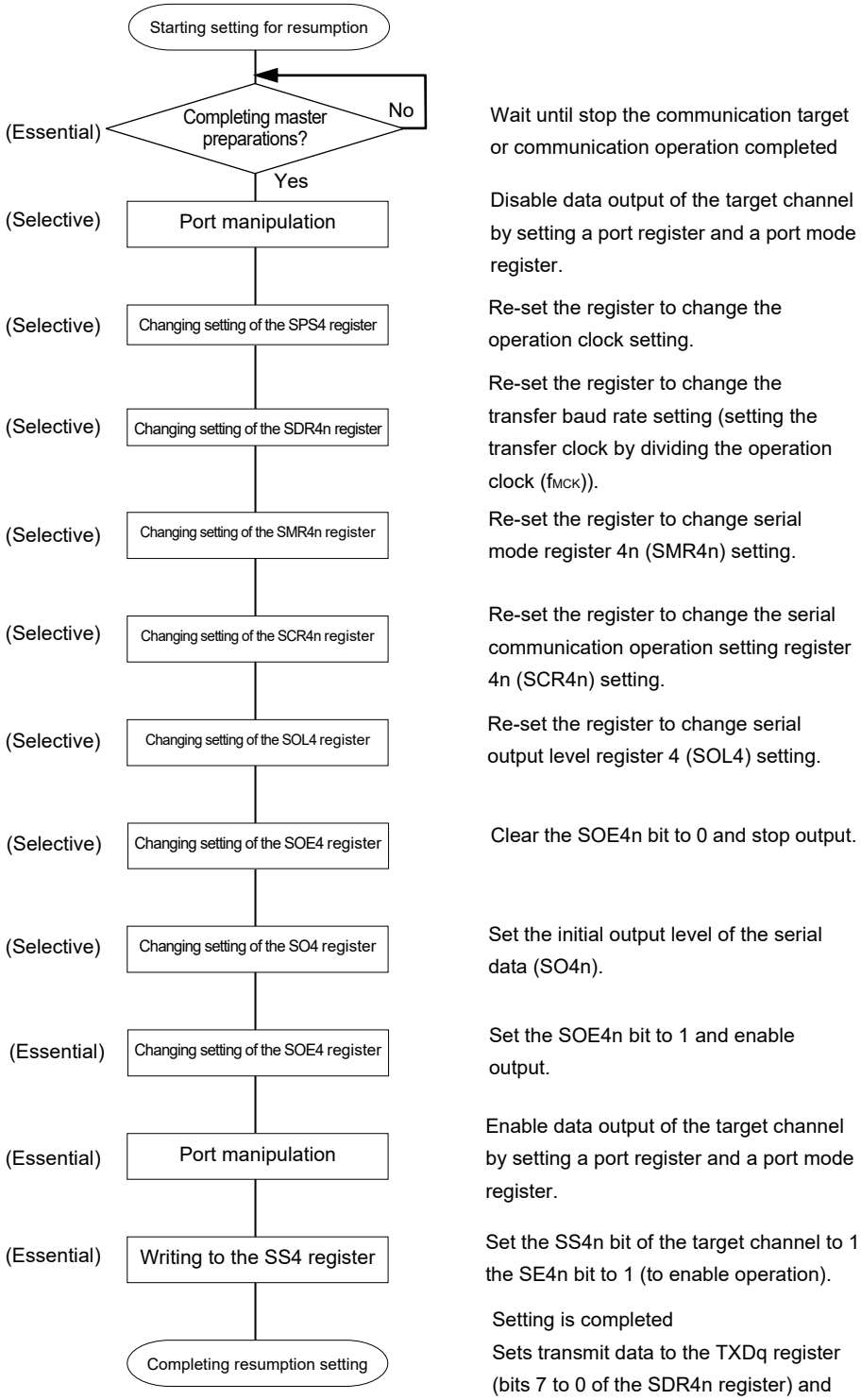


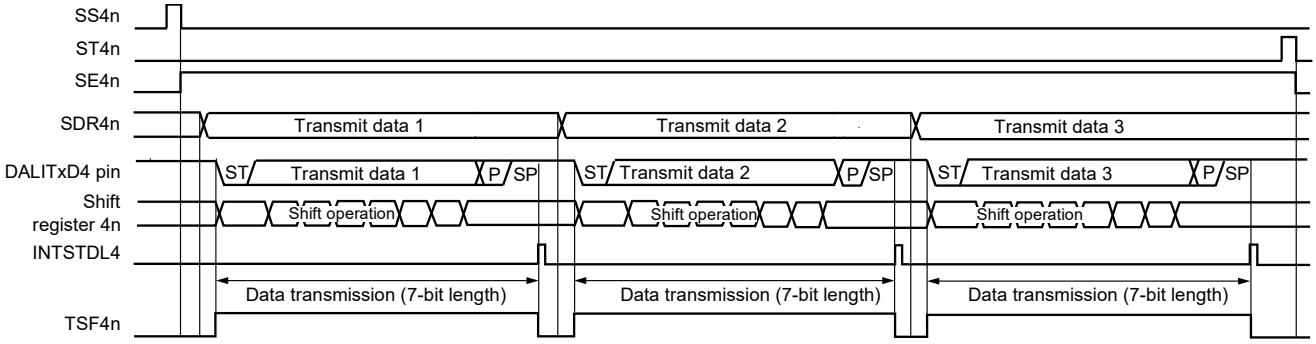
Figure 16-30. Procedure for Resuming UART Transmission



Remark If PER1 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target stops or transmission finishes, and then perform initialization instead of restarting the transmission.

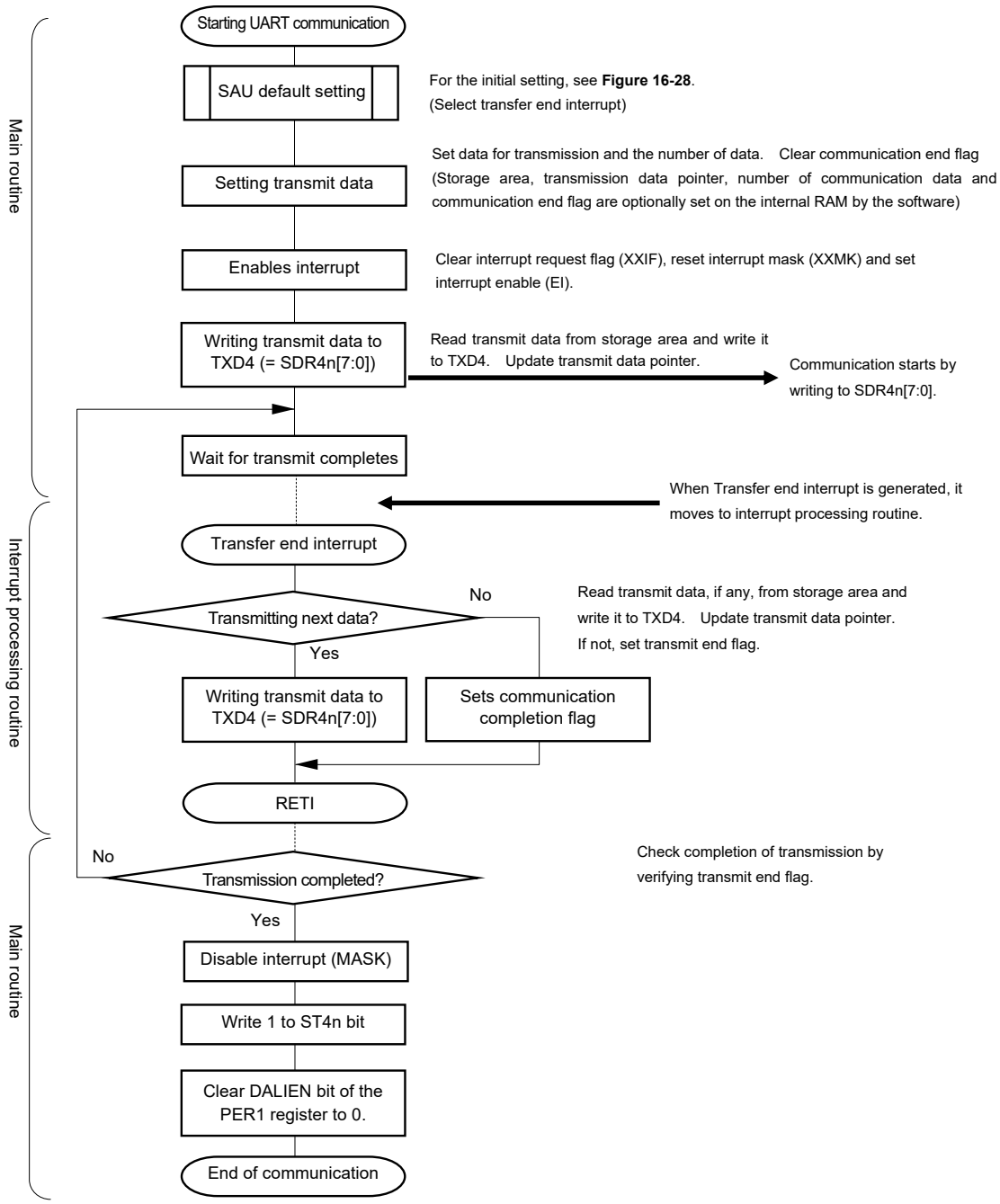
(3) Processing flow (in single-transmission mode)

Figure 16-31. Timing Chart of UART Transmission (in Single-Transmission Mode)



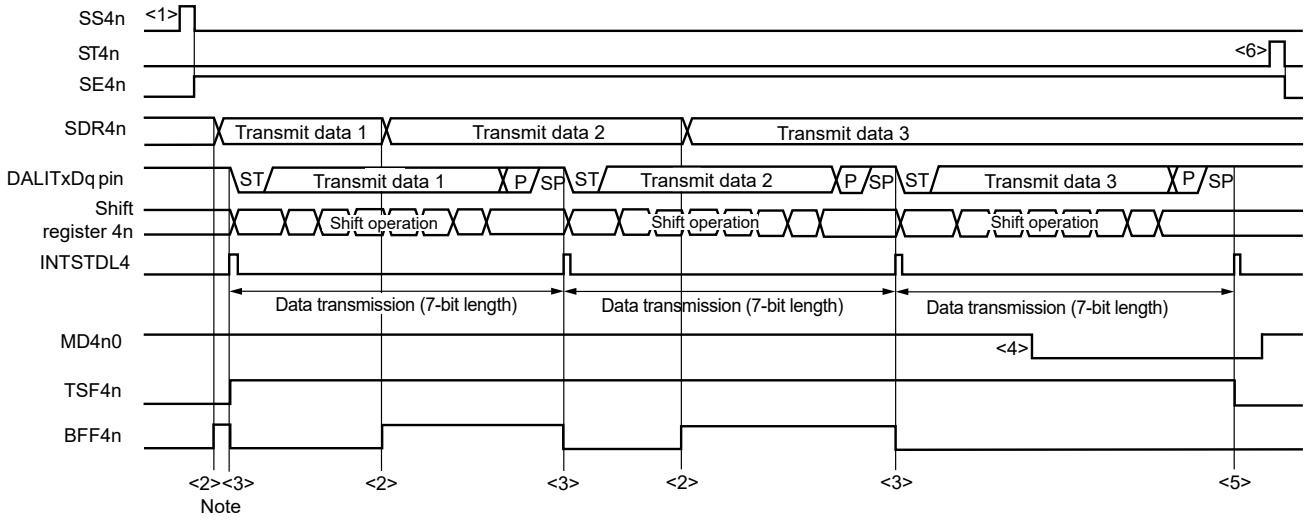
Remark n: Channel number (n = 0)

Figure 16-32. Flowchart of UART Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 16-33. Timing Chart of UART Transmission (in Continuous Transmission Mode)

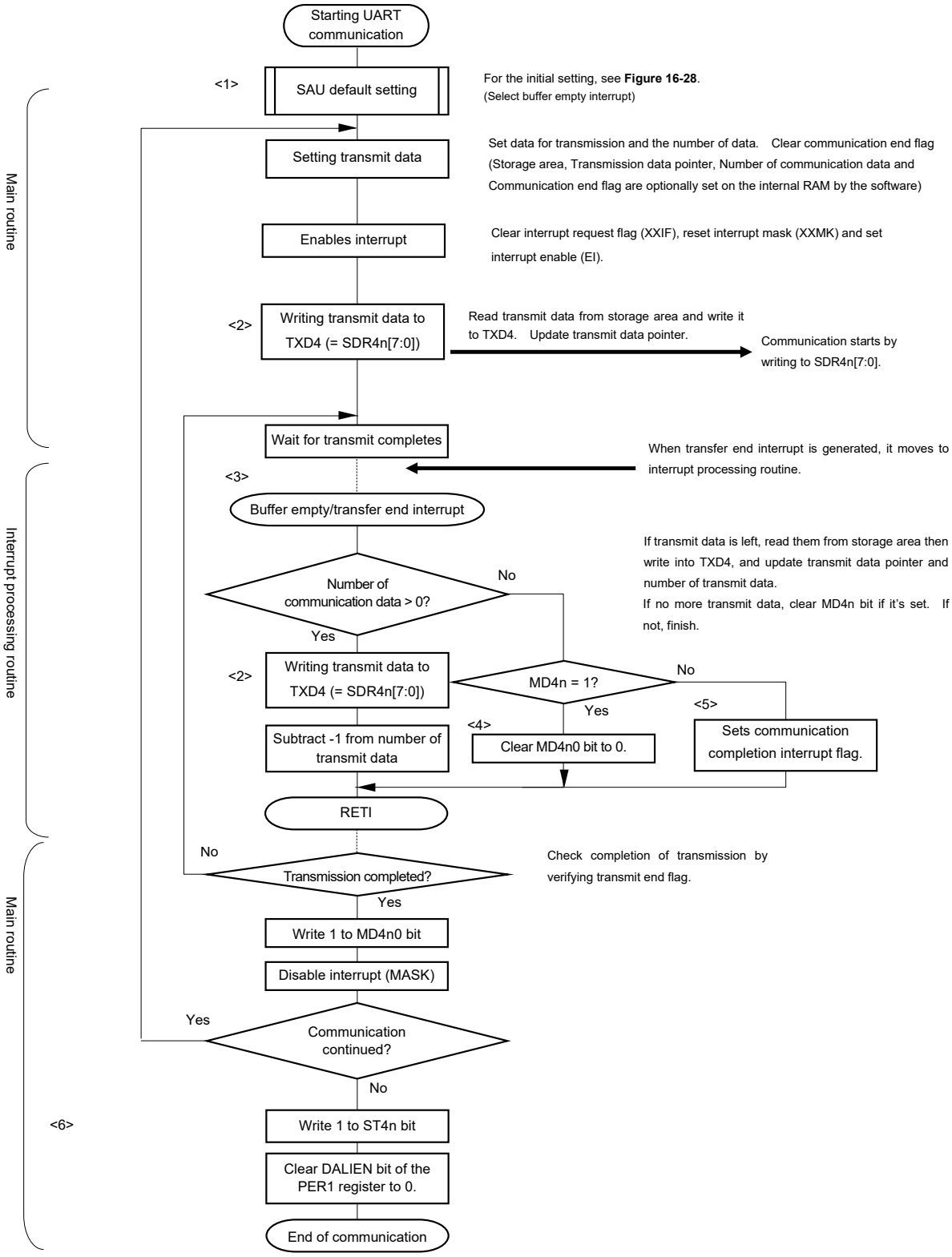


Note If transmit data is written to the SDR4n register while the BFF4n bit of serial status register 4n (SSR4n) is 1 (valid data is stored in serial data register 4n (SDR4n)), the transmit data is overwritten.

Caution The MD4n0 bit of serial mode register 4n (SSR4n) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.

Remark n: Channel number (n = 0)

Figure 16-34. Flowchart of UART Transmission (in Continuous Transmission Mode)



Caution After setting the DALIEN bit of peripheral enable register 1 (PER1) to 1, be sure to set serial clock select register 4 (SPS4) after 4 or more f_{CLK} clocks have elapsed.

Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 16-33. Timing Chart of UART Transmission (in Continuous Transmission Mode))**.

16.5.2 UART reception

UART reception is an operation wherein the RL78/I1A 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 reception	
Target channel	Channel 1
Pins used	DALIRxD4
Transfer data length	7, 8, or 9 bits
Transfer rate	Max. f_{MCK}/6 [bps] (SDR4n [15:9] = 3 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps]**Note**
Data direction	MSB or LSB first
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)
Parity bit	The following selectable • No parity bit (no parity check) • Appending 0 parity (no parity check) • Appending even parity • Appending odd parity
Stop bit	Appending 1 bit
Error detection flag	• Framing error detection flag (FEF4n) • Parity error detection flag (PEF4n) • Overrun error detection flag (OVF4n)
Interrupt	INTSRDL4
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error interrupt	INTSREDL4

Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C)** or **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T_A = −40 to +125°C)**).

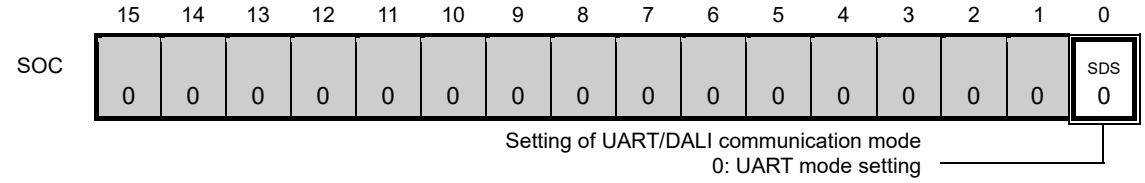
Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

2. n: Channel number (n = 1)

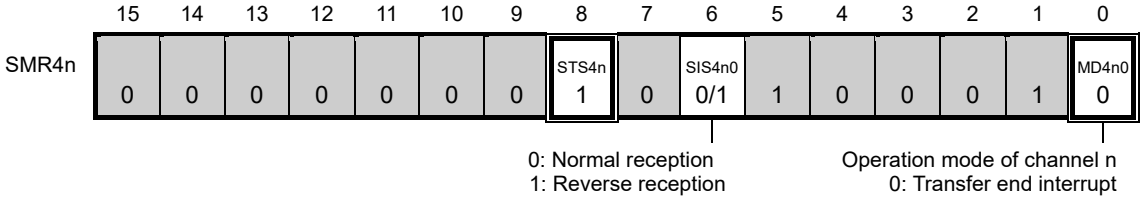
(1) Register setting

Figure 16-35. Example of Contents of Registers for UART Reception (1/2)

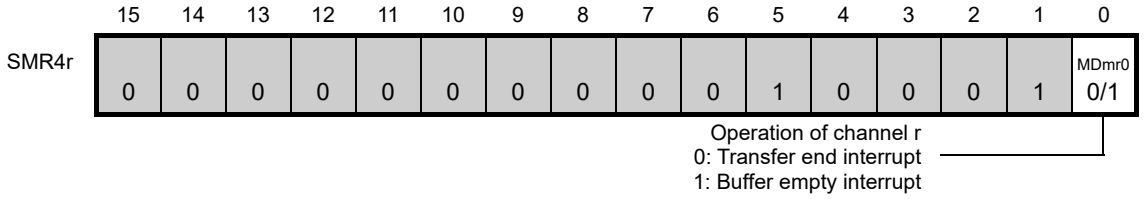
(a) Serial option control register 4 (SOC4)



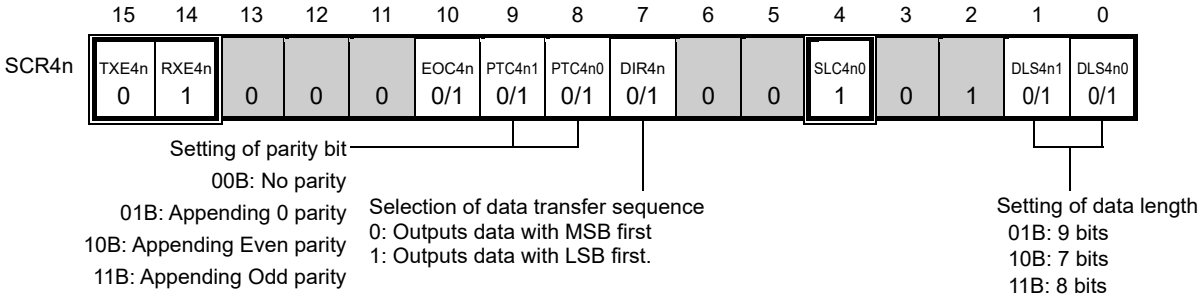
(b) Serial mode register 4n (SMR4n)



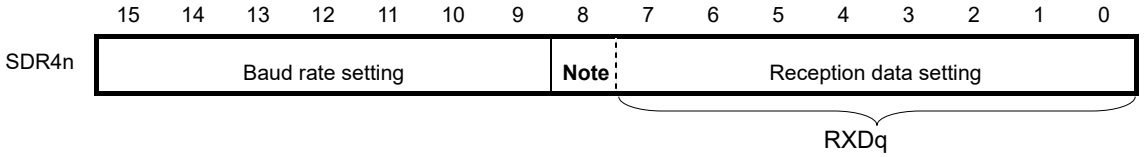
(c) Serial mode register 4r (SMR4r)



(d) Serial communication operation setting register 4n (SCR4n)



(e) Serial data register 4n (SDR4n) (lower 8 bits: RXDq)



Note When performing 9-bit data length communication (DLS411, DLS410 = 0, 1), bits 0 to 8 of the SDR41 register are used as the transmission data specification area.

Caution For the UART reception, be sure to set the SMR4r register of channel r that is to be paired with channel n.

Remarks 1. n: Channel number (n = 1), r: Channel number (r = 0), q: UART number (q = 4)

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

Figure 16-35. Example of Contents of Registers for UART Reception (2/2)

(f) Serial output register 4 (SO4) … The register that not used in this mode.

SO4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	SO40 ×

(g) Serial output enable register 4 (SOE4) …The register that not used in this mode.

SOE4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE40 ×

(h) Serial channel start register 4 (SS4) … Sets only the bits of the target channel is 1.

SS4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS41 0/1	SS40 ×

Caution For the UART reception, be sure to set the SMR4r register of channel r that is to be paired with channel n.

Remarks 1. n: Channel number (n = 1), r: Channel number (r = 0), q: UART number (q = 4)

2. ☐: Setting is fixed in the Simplified SPI (CSI) master transmission mode,

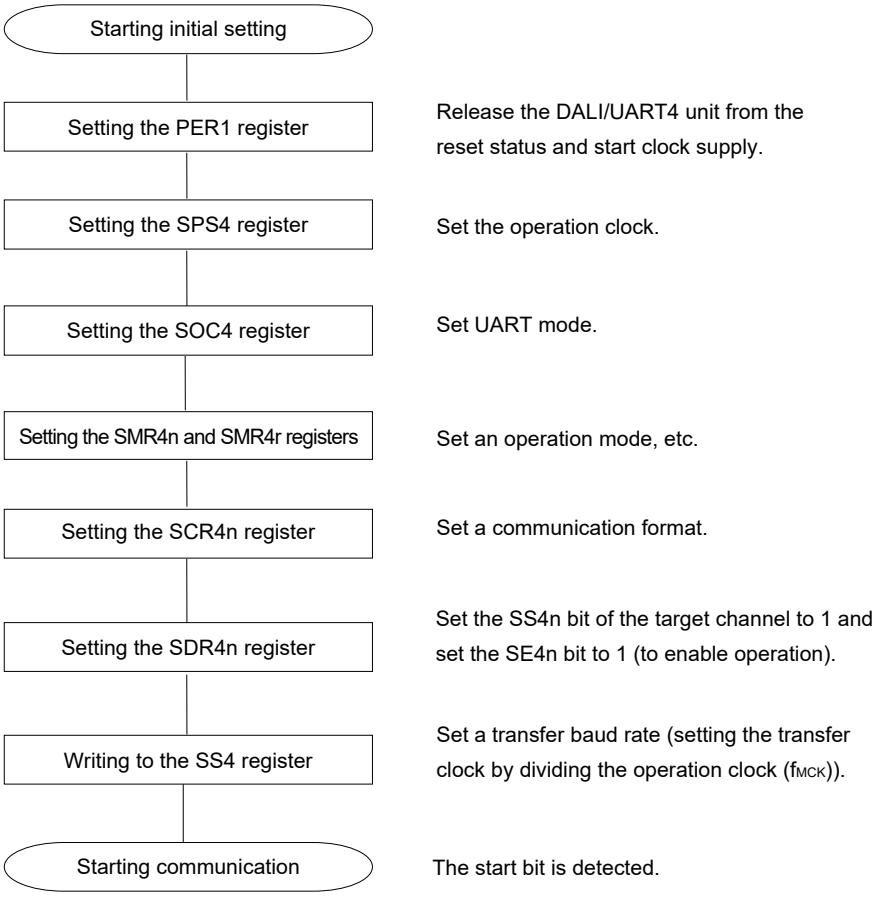
▒: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 16-36. Initial Setting Procedure for UART Reception



Caution Set the RXE4n bit of SCR4n register to 1, and then be sure to set SS4n to 1 after 4 or more f_{MCK} clocks have elapsed.

Figure 16-37. Procedure for Stopping UART Reception

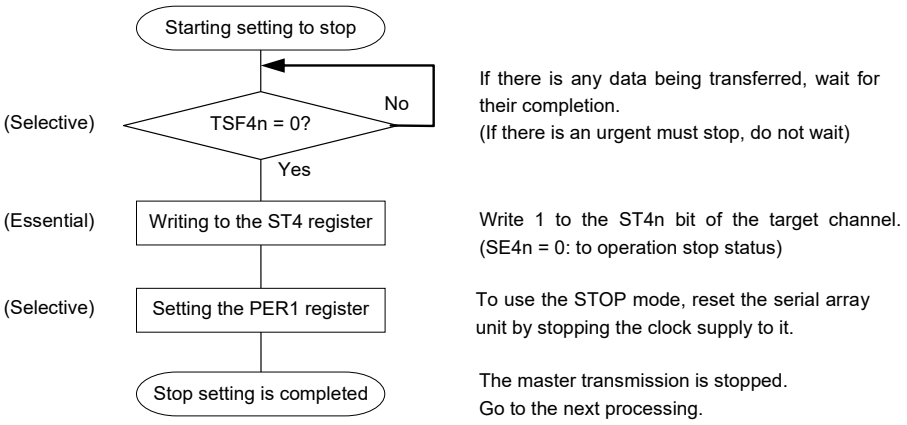
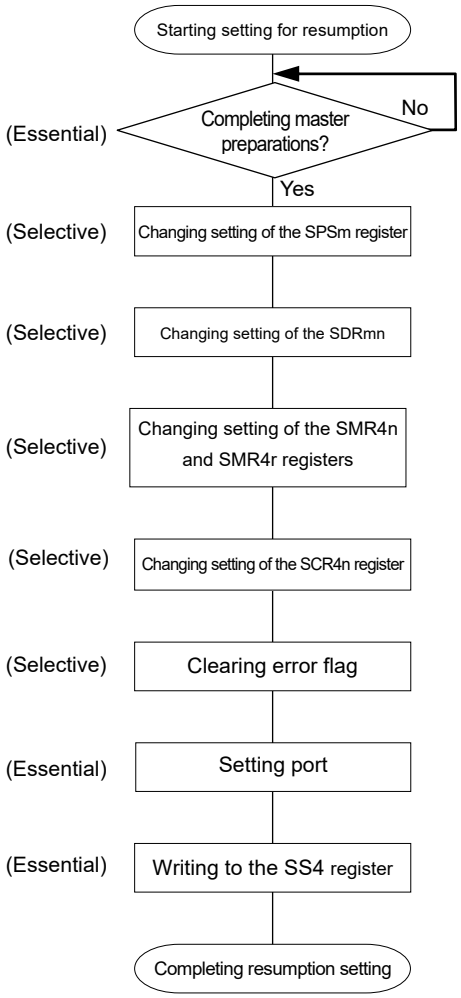


Figure 16-38. Procedure for Resuming UART Reception



Stop the target for communication or wait until completes its communication operation.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the registers to change serial mode registers 4n, 4r (SMR4n, SMR4r) setting.

Re-set the register to change serial communication operation setting register 4n (SCR4n) setting.

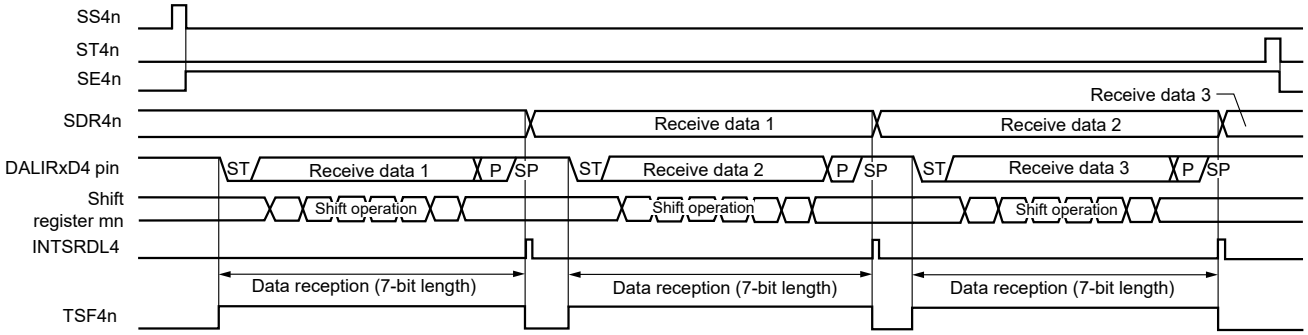
If the FEF, PEF, and OVF flags remain set, clear them using serial flag clear trigger register 4n (SIR4n).

Enable data input of the target channel by setting a port register and a port mode register.

Set the SS4n bit of the target channel to 1 and set the SE4n bit to 1 (to enable operation). Become wait for start bit detection.

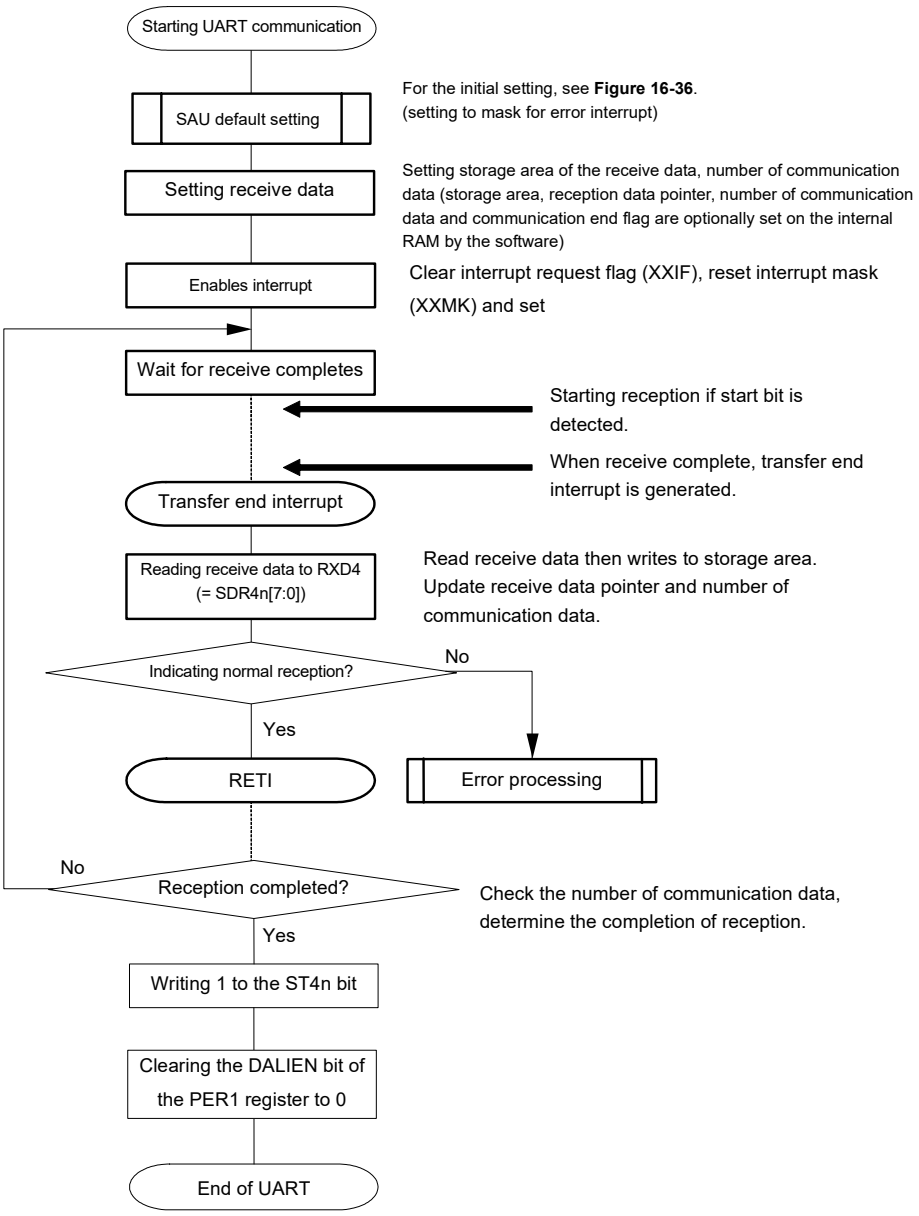
(3) Processing flow

Figure 16-39. Timing Chart of UART Reception



Remark n: Channel number (n = 1), r: Channel number (r = 0)

Figure 16-40. Flowchart of UART Reception



Caution After setting the DALIEN bit of peripheral enable register 1 (PER1) to 1, be sure to set serial clock select register 4 (SPS4) after 4 or more f_{CLK} clocks have elapsed.

16.5.3 SNOOZE mode function

The SNOOZE mode makes the UART perform reception operations upon RxDq pin input detection while in the STOP mode. Normally the UART stops communication in the STOP mode. However, using the SNOOZE mode enables the UART to perform reception operations without CPU operation.

Only UART4 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 16-43 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)** and **Figure 16-45 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 16-2.
- Set the EOCmn and SSECmn bits. This is for enabling or stopping generation of an error interrupt (INTSRE0) when a communication error occurs.
- When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 just before switching to the STOP mode. After the initial setting has completed, set the SSm1 bit of serial channel start register m (SSm) to 1.

Upon detecting the edge of RxDq (start bit input) after a transition was made to the STOP mode, UART reception is started.

Cautions 1. The SNOOZE mode can only be used when the high-speed on-chip oscillator clock (f_{IH}) is selected for f_{CLK}.

2. The transfer rate in the SNOOZE mode is only 4800 bps.

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**

4. If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFmn, FEFmn, or OVFmn flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFmn, FEFmn, or OVFmn flag before setting the SWC0 bit to 1 and read the value in bits 7 to 0 (RxDq register) of the SDRm1 register.

5. The CPU shifts from the STOP mode to the SNOOZE mode on detecting the valid edge of the RxDq 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.

Remark m = 4; n = 0; q = 4

Table 16-2. Baud Rate Setting for UART Reception in SNOOZE Mode

High-speed On-chip Oscillator (f_{IH})	Baud Rate for UART Reception in SNOOZE Mode			
	Baud Rate of 4800 bps			
	Operation Clock (f_{MCK})	SDRmn [15:9]	Maximum Permissible Value	Minimum Permissible Value
32 MHz ± 1.0%Note	$f_{CLK}/2^5$	105	2.27%	–1.53%
24 MHz ± 1.0%Note	$f_{CLK}/2^5$	79	1.60%	–2.18%
16 MHz ± 1.0%Note	$f_{CLK}/2^4$	105	2.27%	–1.53%
12 MHz ± 1.0%Note	$f_{CLK}/2^4$	79	1.60%	–2.19%
8 MHz ± 1.0%Note	$f_{CLK}/2^3$	105	2.27%	–1.53%
6 MHz ± 1.0%Note	$f_{CLK}/2^3$	79	1.60%	–2.19%
4 MHz ± 1.0%Note	$f_{CLK}/2^2$	105	2.27%	–1.53%
3 MHz ± 1.0%Note	$f_{CLK}/2^2$	79	1.60%	–2.19%
2 MHz ± 1.0%Note	$f_{CLK}/2$	105	2.27%	–1.54%
1 MHz ± 1.0%Note	f_{CLK}	105	2.27%	–1.57%

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.

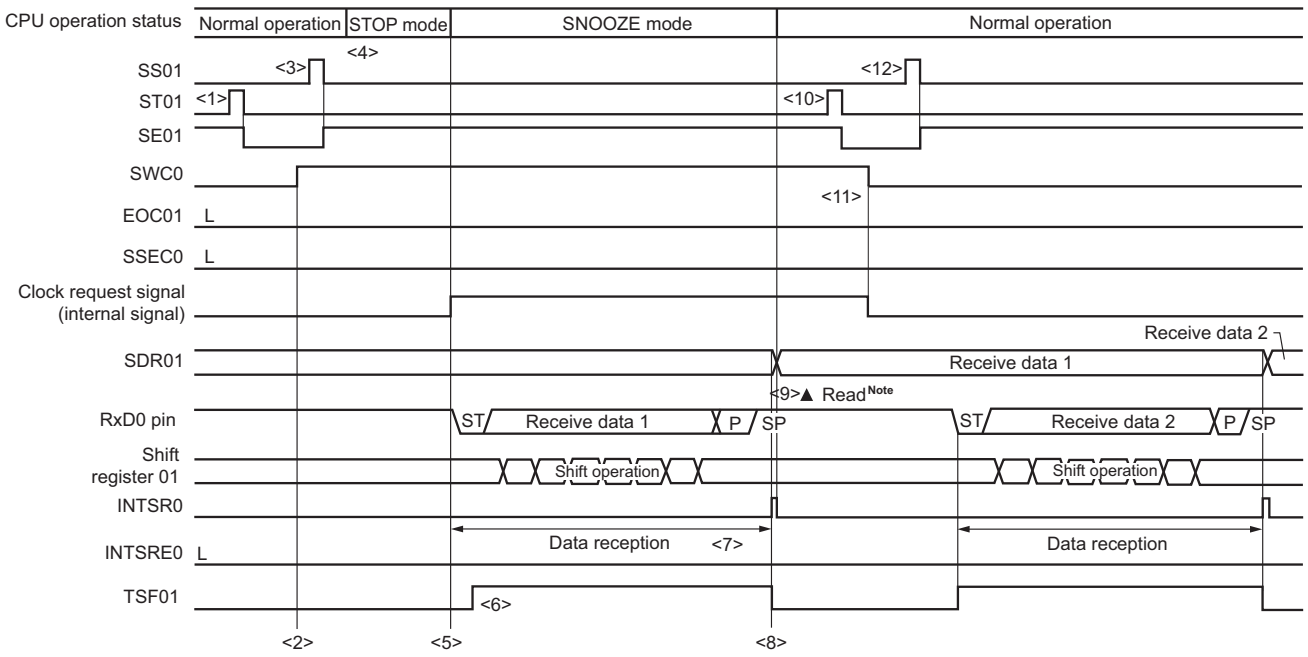
- In the case of f_{IH} ± 1.5%, perform (Maximum permissible value – 0.5%) and (Minimum permissible value + 0.5%) to the values in the above table.
- In the case of f_{IH} ± 2.0%, perform (Maximum permissible value – 1.0%) and (Minimum permissible value + 1.0%) to the values in the above table.

Remark The maximum permissible value and minimum permissible value are permissible values for the baud rate in UART reception. The baud rate on the transmitting side should be set to fall inside this range.

(1) SNOOZE mode operation (EOCm1 = 0, SSECm = 0/1)

Because of the setting of EOCm1 = 0, even though a communication error occurs, an error interrupt (INTSREq) is not generated, regardless of the setting of the SSECm bit. A transfer end interrupt (INTSRq) will be generated.

Figure 16-41. Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)



Note Read the received data when SWCm is 1

Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

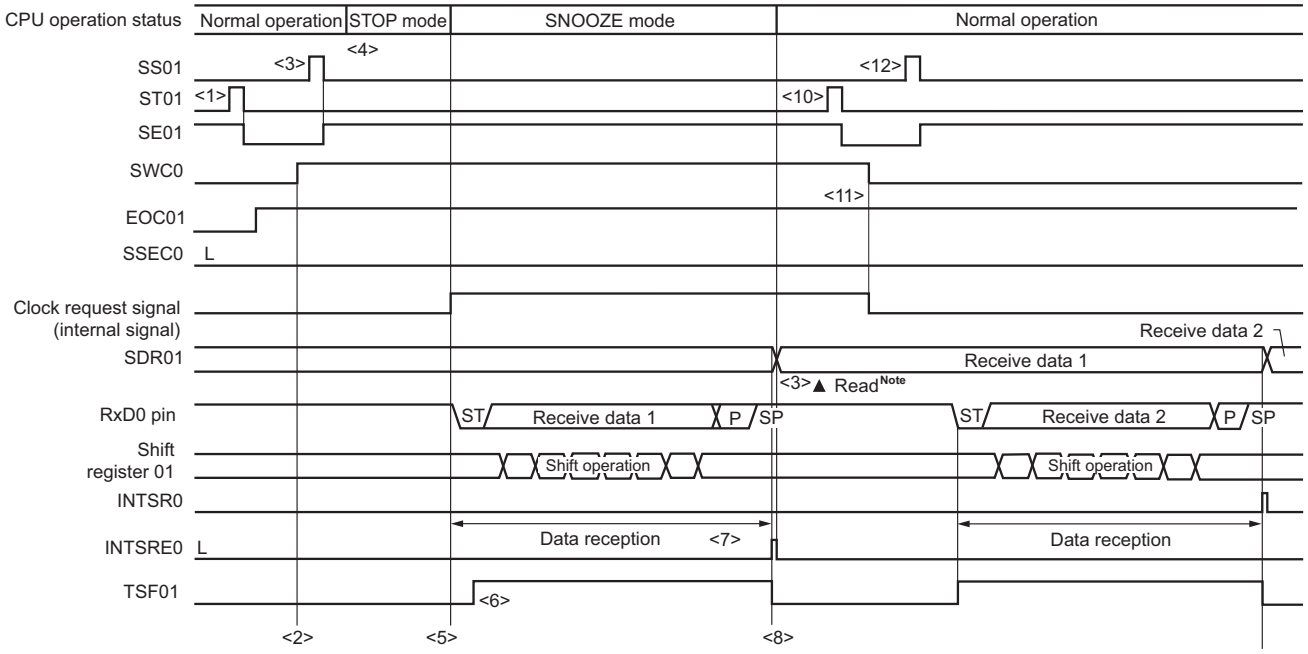
Remarks 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 16-43 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)**.

2. m = 4; q = 4

(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 16-42. Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)



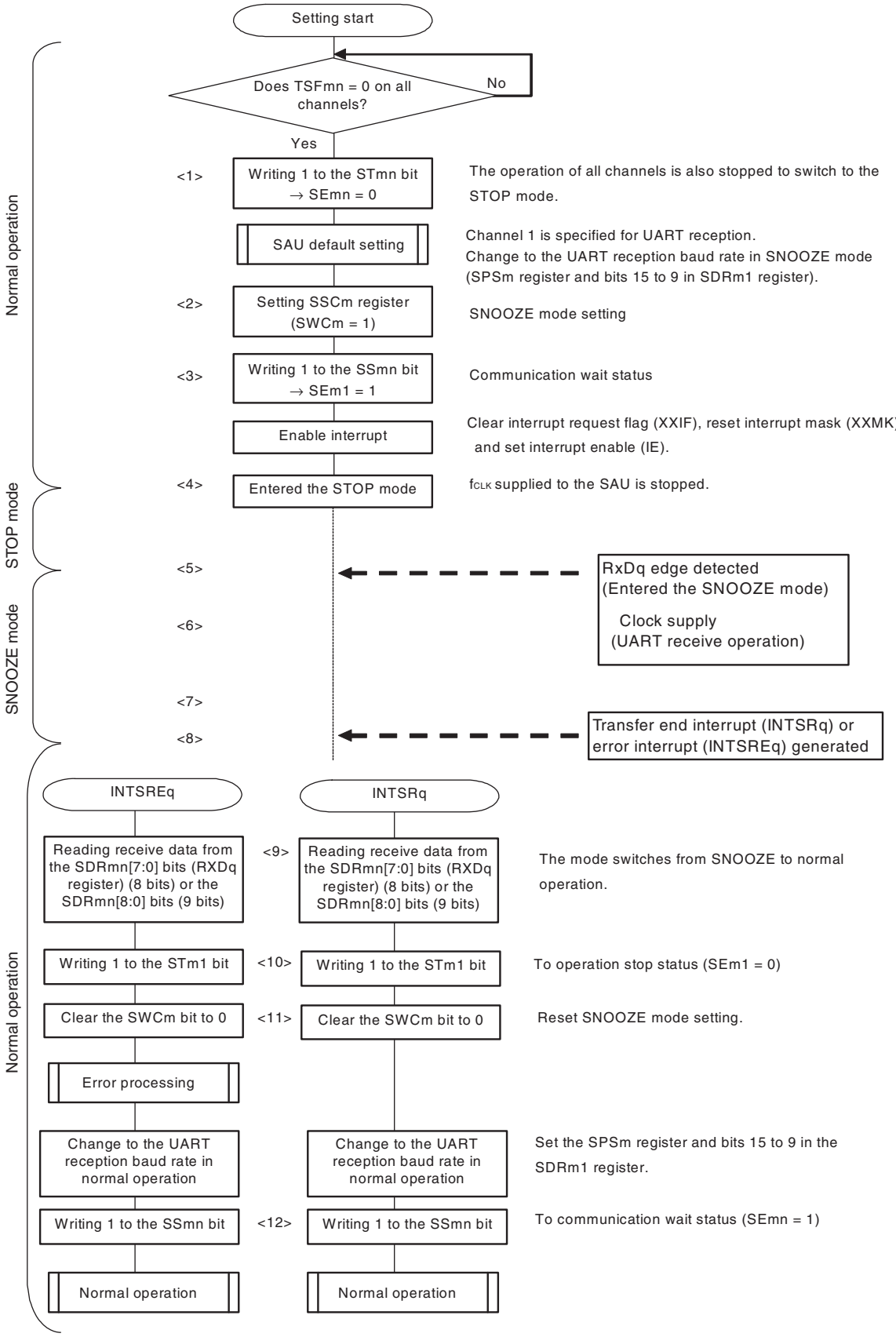
Note Read the received data when SWCm = 1.

Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

Remarks 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 16-43 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)**.

2. m = 4; q = 4

Figure 16-43. Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)



(**Remarks** are listed on the next page.)

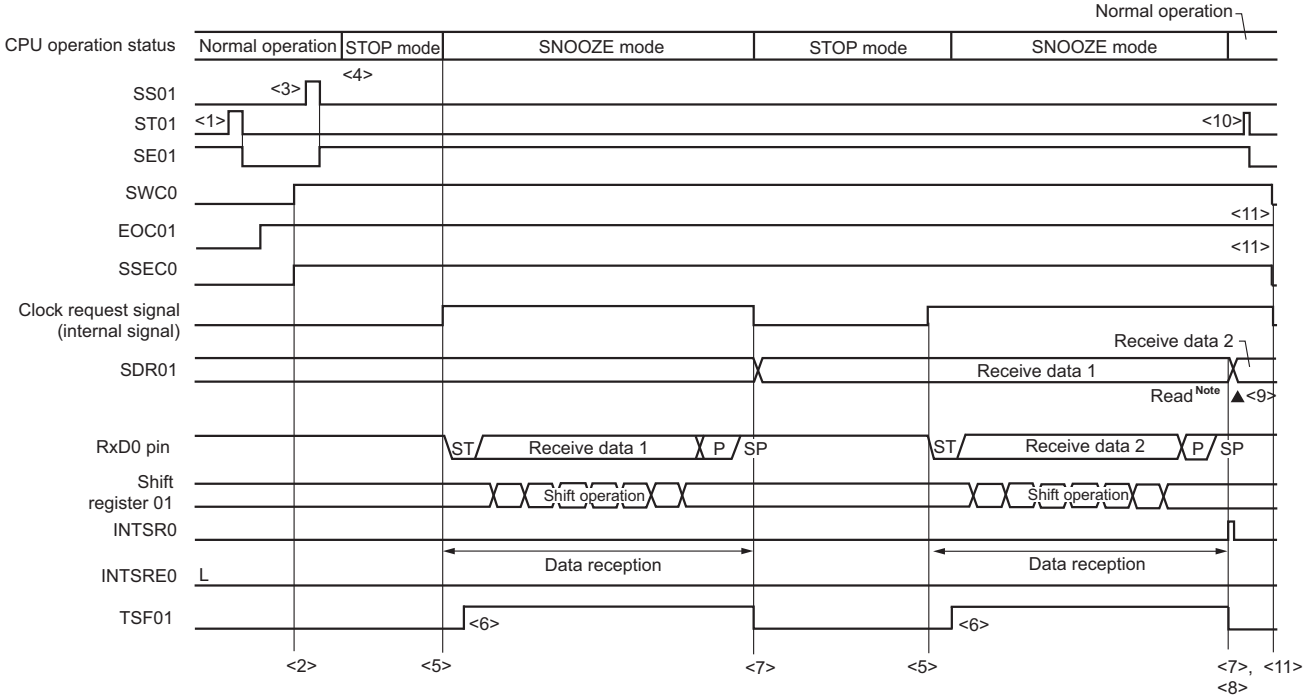
Remarks 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 16-41 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)** and **Figure 16-42 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)**.

2. m = 4; q = 4; n = 0, 1

(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 16-44. Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



Note Read the received data when SWCm = 1.

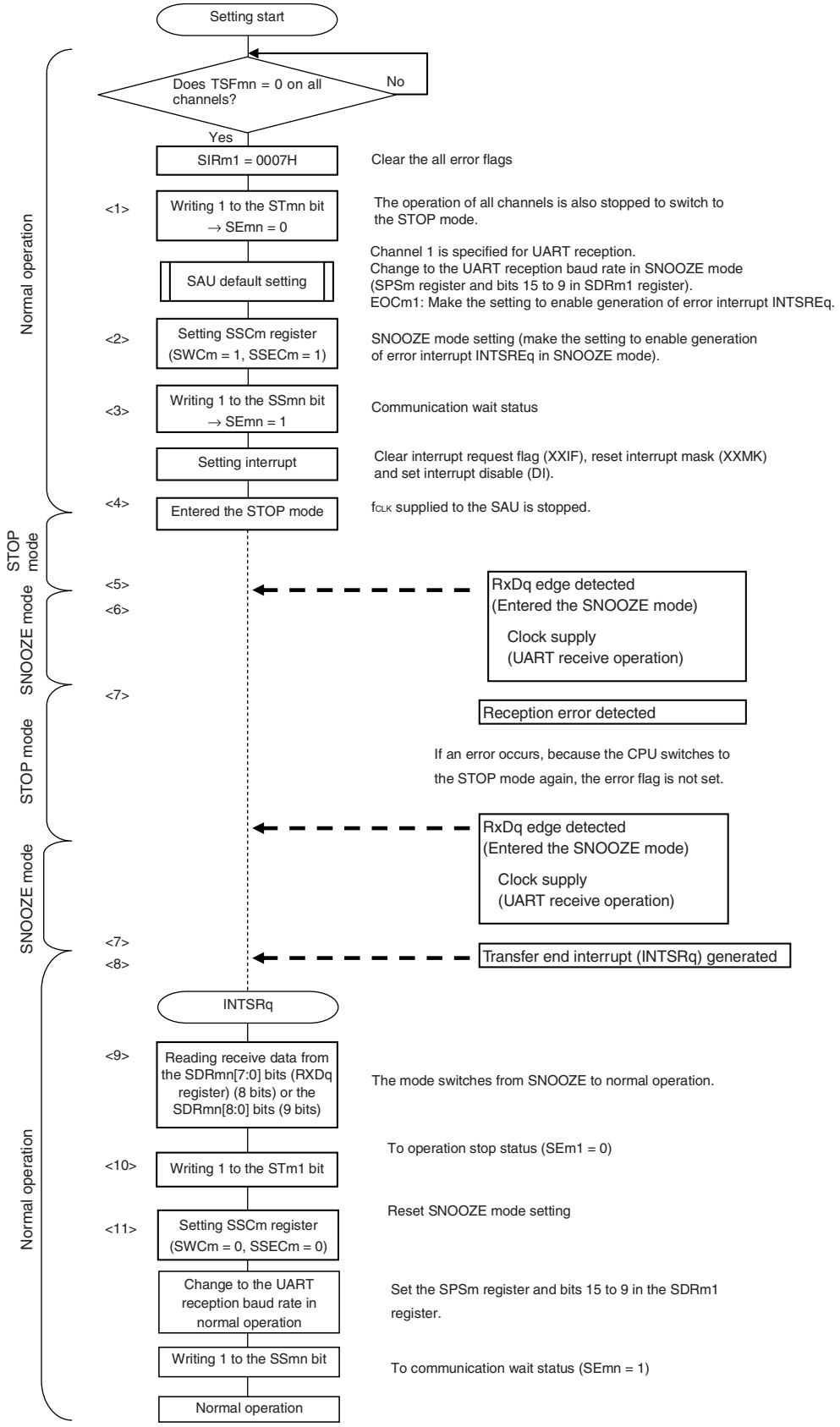
Cautions 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit and stop the operation).
After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).

2. If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFm1, FEFm1, or OVFm1 flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFm1, FEFm1, or OVFm1 flag before setting the SWCm bit to 1 and read the value in SDRm1[7:0] (RxDq register) (8 bits) or SDRm1[8:0] (9 bits).

Remarks 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 16-45 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)**.

2. m = 4; q = 4

Figure 16-45. Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



(**Caution** and **Remark** are listed on the next page.)

Caution If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFm1, FEFm1, or OVFm1 flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFm1, FEFm1, or OVFm1 flag before setting the SWCm bit to 1 and read the value in SDRm1[7:0] (RxDq register) (8 bits) or SDRm1[8:0] (9 bits).

Remarks 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 16-44 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)**.

2. m = 4; q = 4; n = 0, 1

16.6 DALI Mode

This mode is used to perform data transmission/reception as master and slave of DALI (Digital Addressable Lighting Interface).

DALI performs communication using the following protocol.

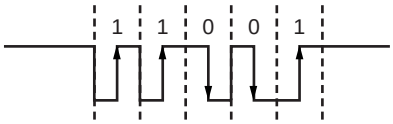
Caution See the latest standard specifications with regard to DALI communications standards.

(1) Data structure

<1> Bit definition

A falling edge is bit-defined as “0” and a rising edge as “1”, because DALI communication uses Manchester code. If no communication is performed, DALI communication is fixed to the high level.

Figure 16-46. Bit Definition

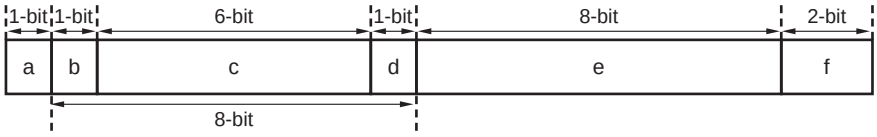


<2> Frame

- Forward frame

This is a frame used when transmitting from the master to a slave. A frame consists of 19, 20 or 27 bits.

Figure 16-47. Forward-frame Structure (19 Bits)



a: Start bit

This indicates the start of the frame. It is always the same waveform as “1”.

b-d: Address byte

This specifies the transmission destination of the frame.

e: Data byte

This specifies a command.

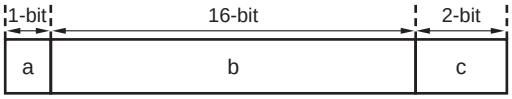
f: Stop bit

This indicates the end of the frame. It is fixed to the high level.

- Backward frame

This is a frame used when transmitting from the slave to a master. A frame consists of 11, 19 or 27 bits.

Figure 16-48. Backward-frame Structure (19 Bits)



a: Start bit

This indicates the start of the frame. It is always the same waveform as “1”.

b: Data byte

This replies to the master.

c: Stop bit

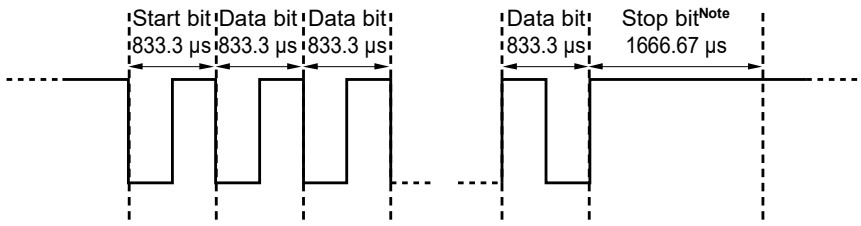
This indicates the end of the frame. It is fixed to the high level.

(2) Transmission/reception timing rules

<1> Timing in the frame

The DALI bit width is one bit (= 833.3 μs ±10%) for both Forward and Backward frames.

Figure 16-49. Timing in the Frame



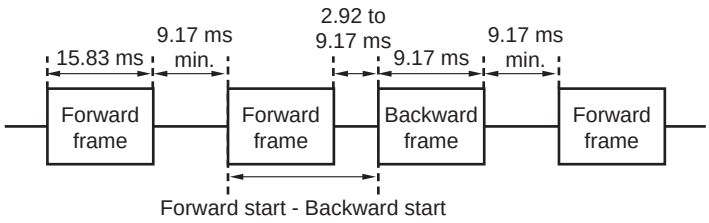
Note The stop bit width is 1666.67 μs for 2 stop bits.

<2> Timing among frames

With DALI, the following timing must be controlled in frame units.

- Forward frame width: 15.83 ms ±10% (19 bits), 16.67 ms ±10% (20 bits), 22.5 ms ±10% (27 bits)
- Backward frame width: 9.17 ms ±10% (11 bits), 16.67 ms ±10% (20 bits), 22.5 ms ±10% (27 bits)
- Communication interval between one Forward frame and the Backward frame: 2.92 to 9.17 ms
- Communication interval between one Forward frame and the next Forward frame: 9.17 ms min.
- Communication interval between one Backward frame and the next Forward frame: 9.17 ms min.

Figure 16-50. Timing Among Frames



Remark This is an example of a case of 1200 bps

16.6.1 DALI transmission

DALI transmission is an operation to transmit data from the RL78/I1A to another device asynchronously (start-stop synchronization).

Of two channels used for DALI, the even channel is used for DALI transmission.

DALI transmission	
Target channel	Channel 0
Pins used	DALITxD4
Transfer data length	8, 16, 17, or 24 bits
Transfer rate	Max. f_{MCK}/12 [bps] (SDR4n [15:9] = 3 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 256)$ [bps]**Note**
Data direction	MSB first
Data phase	Non-reverse output (default: high level), reverse output (default: low level)
Parity bit	No parity bit
Stop bit	Appending 2 bits
Error detection flag	None
Interrupt	INTSTDL4
	Transfer end interrupt

Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T_A = –40 to +105°C)** or **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T_A = –40 to +125°C)**).

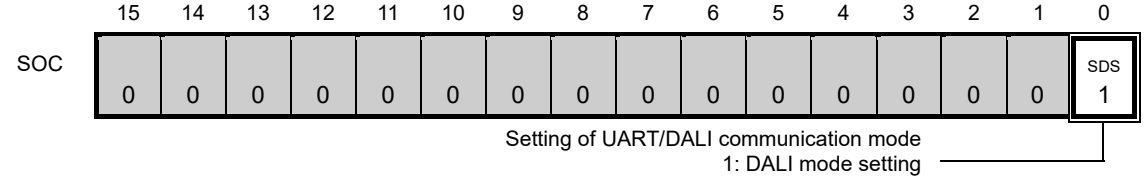
Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

2. n: Channel number (n = 0)

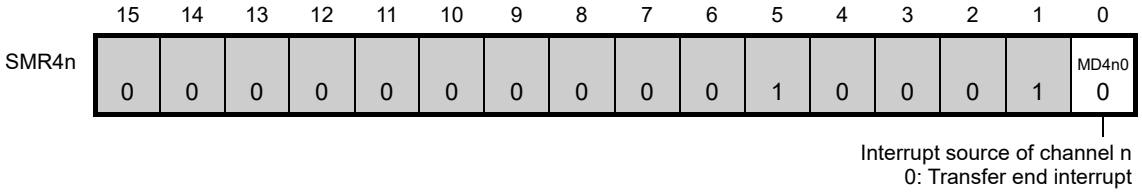
(1) Register setting

Figure 16-51. Example of Contents of Registers for DALI Transmission (1/2)

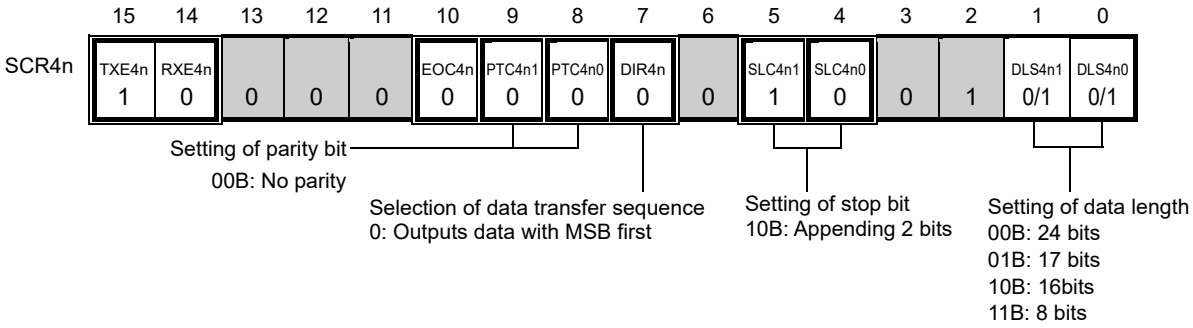
(a) Serial option control register 4 (SOC4)



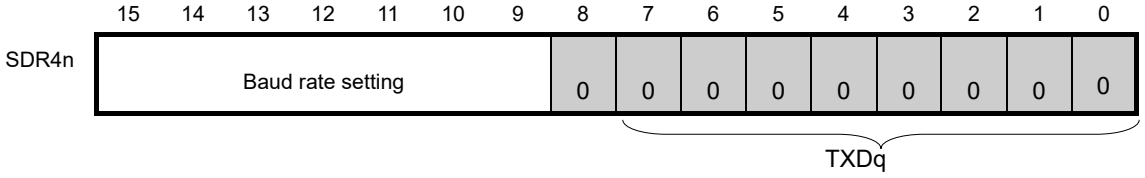
(b) Serial mode register 4n (SMR4n)



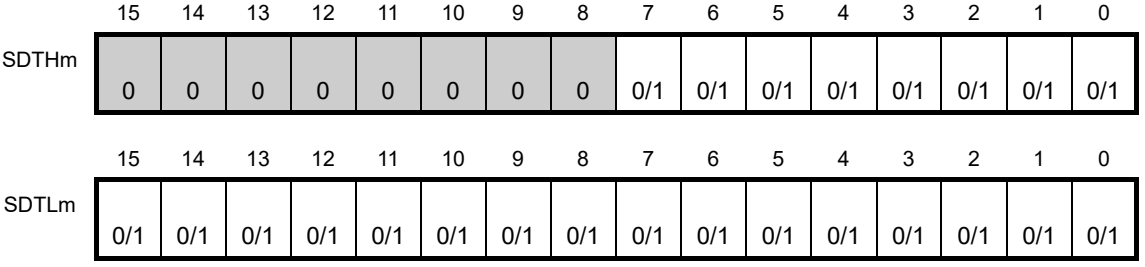
(c) Serial communication operation setting register 4n (SCR4n)



(d) Serial data register 4n (SDR4n) (lower 8 bits: TXDq)



(e) DALI transmit data register Hm, Lm (SDTHm, SDTLm)



Remarks 1. n: Channel number (n = 0), q: DALI/UART number (q = 4)

2. ☐: Setting is fixed in the DALI transmission mode, ▒: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 16-51. Example of Contents of Registers for DALI Transmission (2/2)

(f) Serial output level register 4 (SOL4) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL40 0

0: Non-reverse transmission
1: Reverse transmission

(g) Serial output register 4 (SO4) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO4	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	SO40 1

1: Serial data output value is “1”

(h) Serial output enable register 4 (SOE4) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE40 1

(i) Serial channel start register 4 (SS4) … Sets only the bits of the target channel to 1.

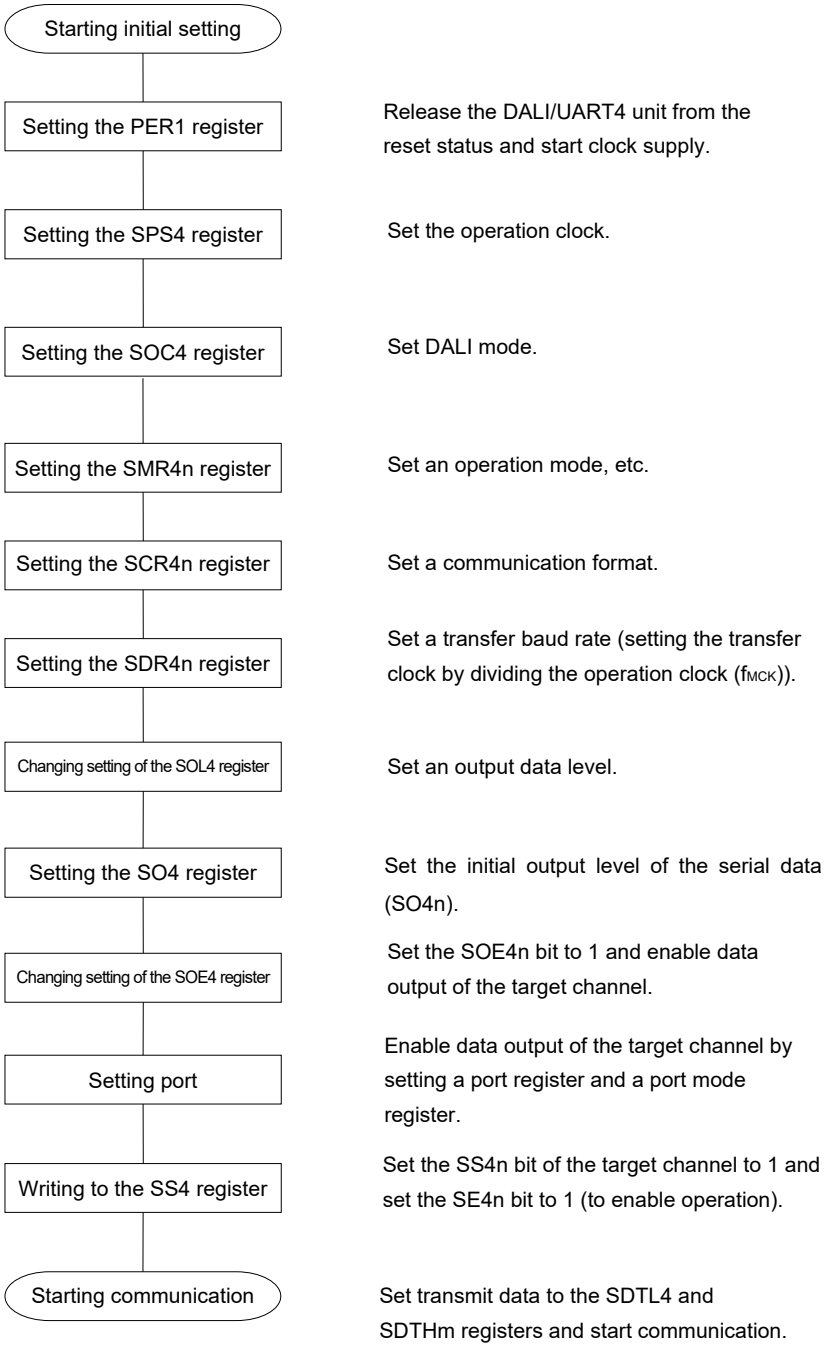
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS41 ×	SS40 0/1

Remarks 1. n: Channel number (n = 0)

2. ☐: Setting is fixed in the DALI 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

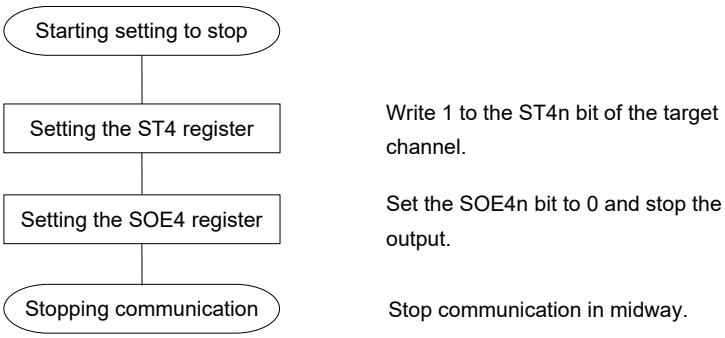
(2) Operation procedure

Figure 16-52. Initial Setting Procedure for DALI Transmission



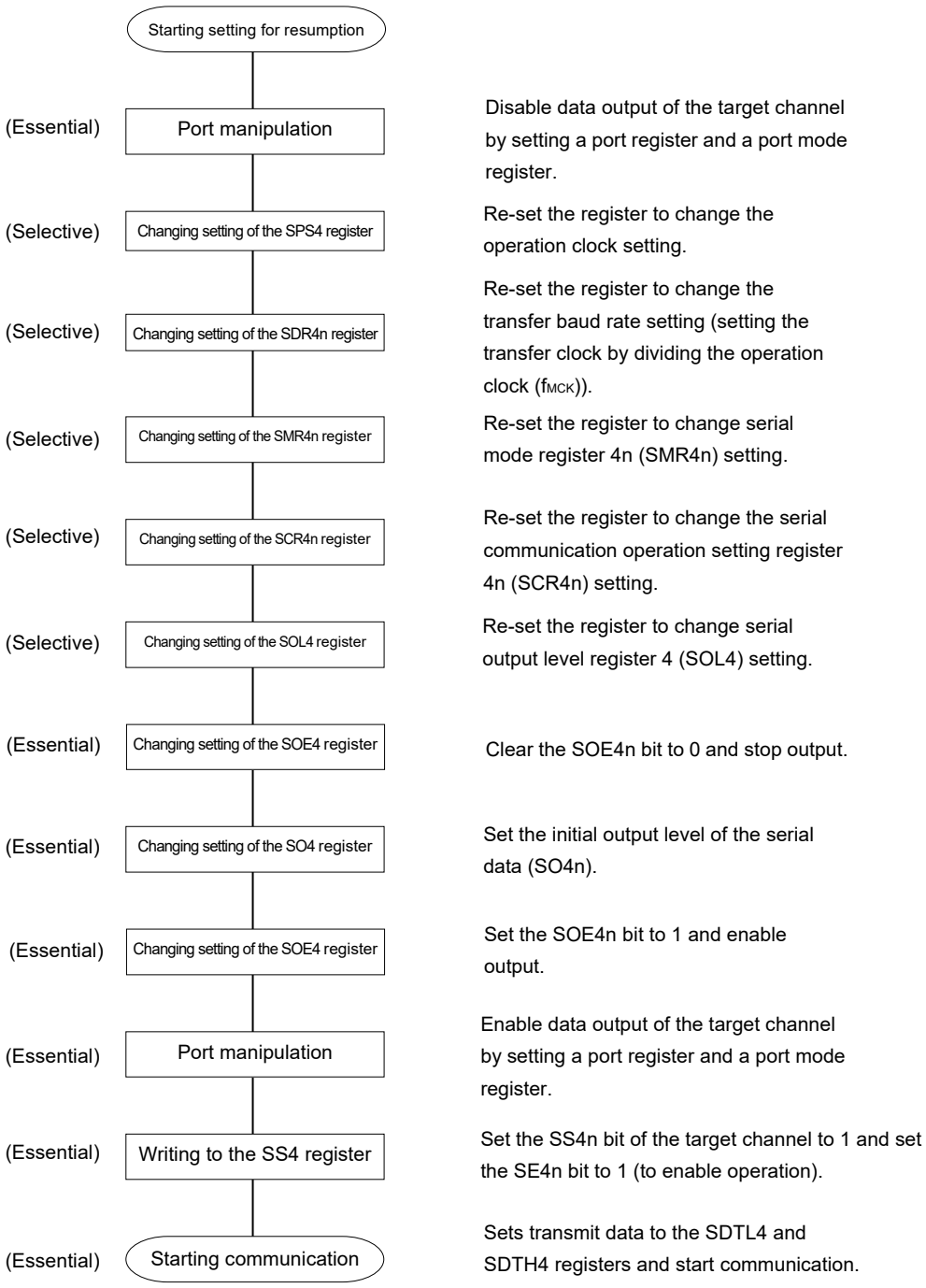
Caution After setting the DALIEN bit of peripheral enable register 1 (PER1) to 1, be sure to set serial clock select register 4 (SPS4) after 4 or more f_{CLK} clocks have elapsed.

Figure 16-53. Procedure for Stopping DALI Transmission



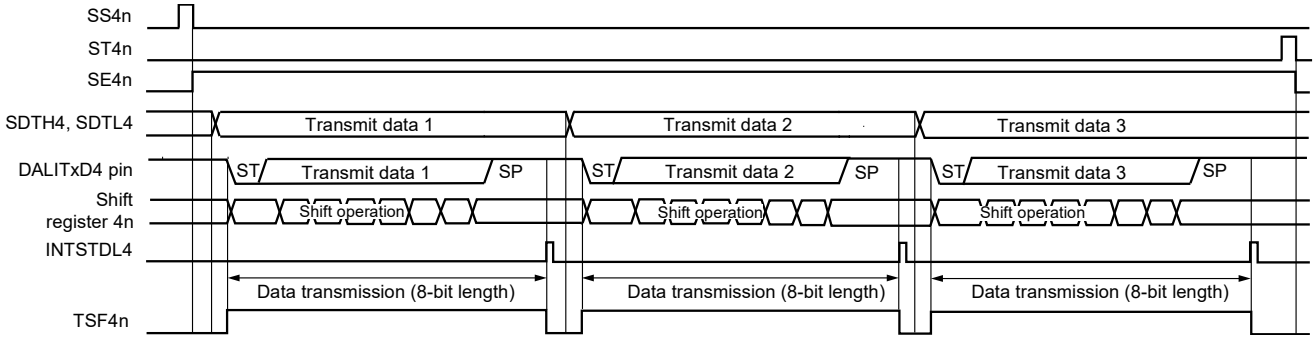
Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set serial output register 4 (SO4) (see **Figure 16-54 Procedure for Resuming DALI Transmission**).

Figure 16-54. Procedure for Resuming DALI Transmission



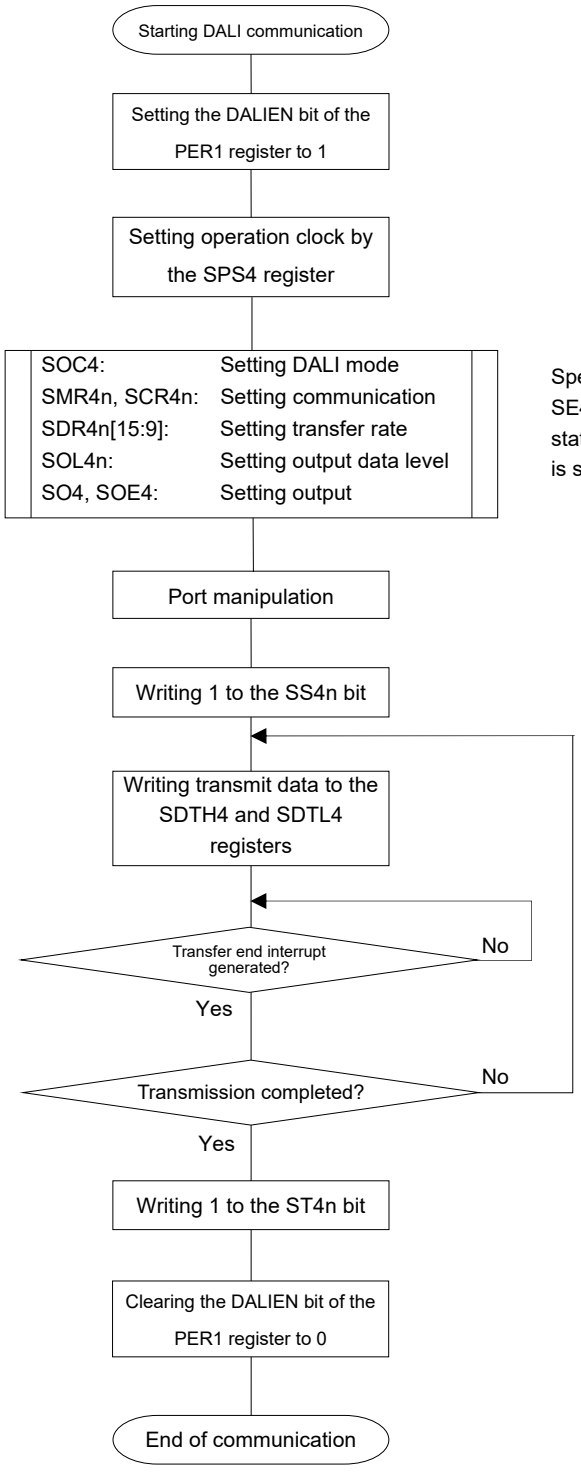
(3) Processing flow

Figure 16-55. Timing Chart of DALI Transmission



Remark n: Channel number (n = 0)

Figure 16-56. Flowchart of DALI Transmission



Specify the initial settings while the SE4n bit of serial channel enable status register m (SE4) is 0 (operation is stopped).

Caution After setting the DALIEN bit of peripheral enable register 1 (PER1) to 1, be sure to set serial clock select register 4 (SPS4) after 4 or more f_{CLK} clocks have elapsed.

16.6.2 DALI reception

DALI reception is an operation wherein the RL78/I1A asynchronously receives data from another device (start-stop synchronization).

For DALI reception, the odd-number channel of the two channels used for DALI is used. The SMR register of both the odd- and even-numbered channels must be set.

DALI Reception	
Target channel	Channel 1
Pins used	DALIRxD4
Transfer data length	8, 16, 17, or 24 bits
Transfer rate	Max. $f_{MCK}/12$ [bps] (SDR4n[15:9] = 3 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 256)$ [bps]**Note**
Data direction	MSB first
Data phase	Non-reverse output (default: high level), reverse output (default: low level)
Parity bit	No parity bit
Stop bit	Appending 2 bits
Error detection flag	• Framing error detection flag (FEF4n) • Manchester framing error detection flag (MFEF4n) • Overrun error detection flag (OVF4n)
Interrupt	INTSRDL4
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error interrupt	INTSREDL4

Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 32 ELECTRICAL SPECIFICATIONS (T_A = –40 to +105°C)** or **CHAPTER 33 ELECTRICAL SPECIFICATIONS (T_A = –40 to +125°C)**).

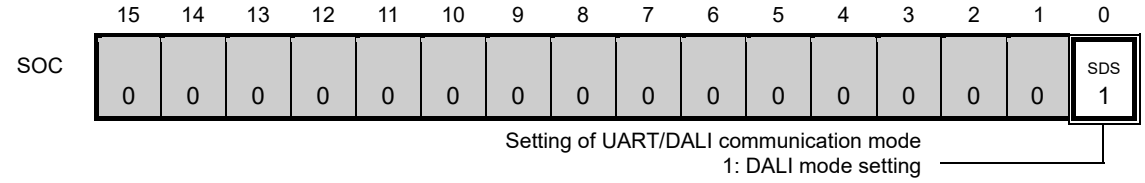
Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

2. n: Channel number (n = 1)

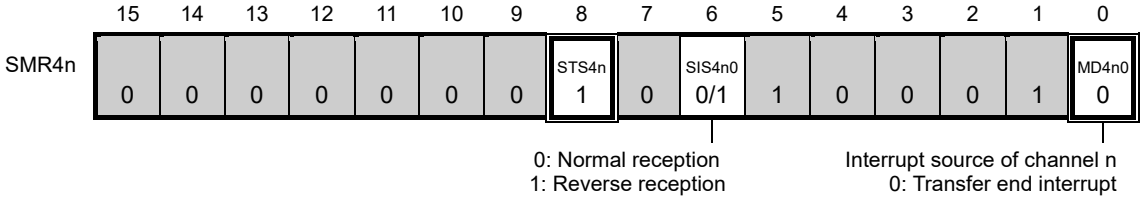
(1) Register setting

Figure 16-57. Example of Contents of Registers for DALI Reception (1/2)

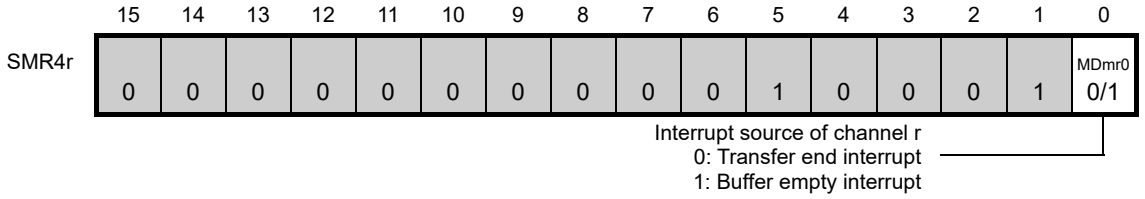
(a) Serial option control register 4 (SOC4)



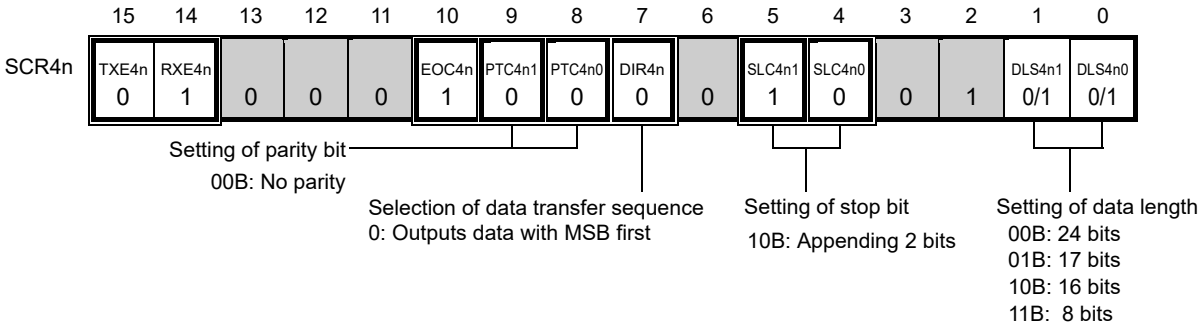
(b) Serial mode register 4n (SMR4n)



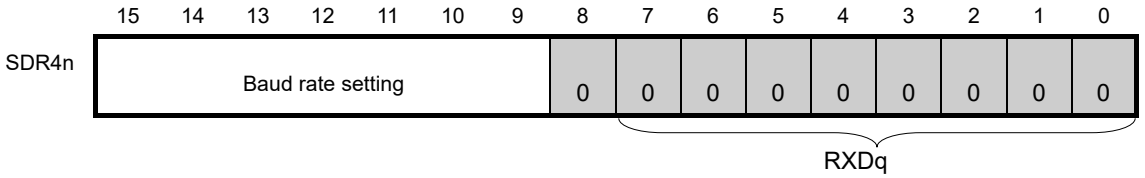
(c) Serial mode register 4r (SMR4r)



(d) Serial communication operation setting register 4n (SCR4n)



(e) Serial data register 4n (SDR4n) (lower 8 bits: TXDq)



Caution For the DALI reception, be sure to set the SMR4r register of channel r that is to be paired with channel n.

Remarks 1. n: Channel number (n = 1), r: Channel number (r = 0), q: DALI/UART number (q = 4)

2. ☐: Setting is fixed in the DALI reception mode, ▒: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 16-57. Example of Contents of Registers for DALI Reception (2/2)

(f) DALI receive data register (SDCHm, SDCLm)

SDCHm

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

SDCLm

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

(g) Serial output register 4 (SO4) …The register that not used in this mode.

SO4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	SO40 ×

(h) Serial output enable register 4 (SOE4) …The register that not used in this mode.

SOE4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE40 ×

(i) Serial channel start register 4 (SS4) … Sets only the bits of the target channel is 1.

SS4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS41 0/1	SS40 ×

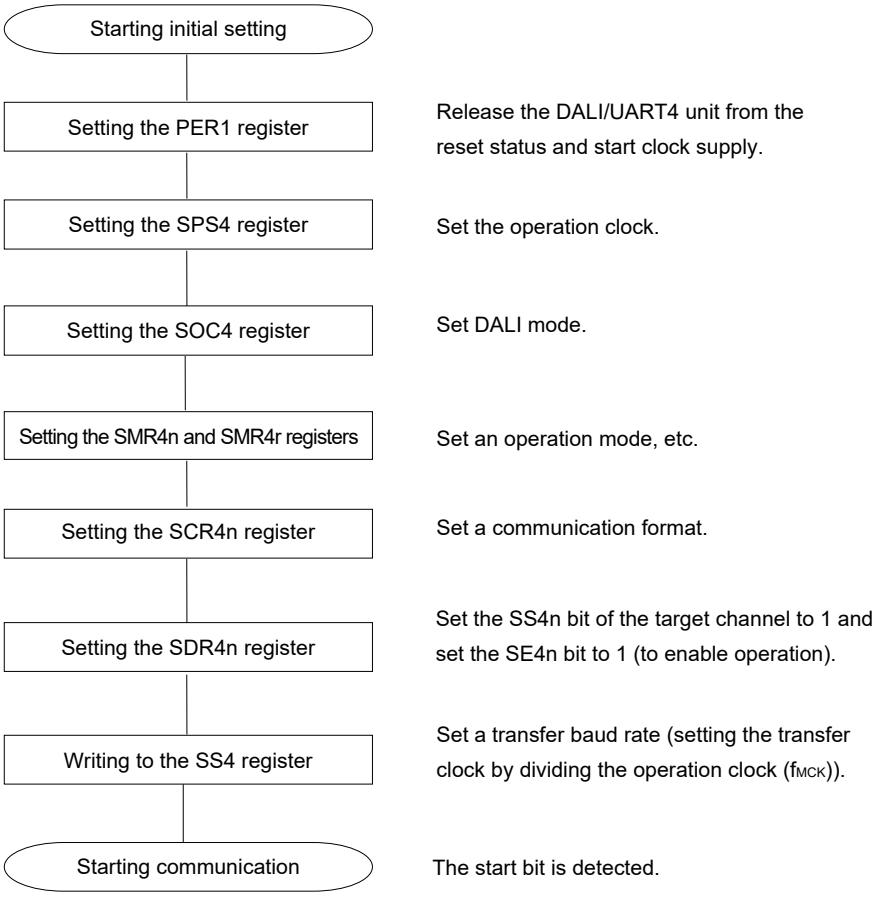
Caution For the DALI reception, be sure to set the SMR4r register of channel r that is to be paired with channel n.

Remarks 1. n: Channel number (n = 1), r: Channel number (r = 0), q: DALI/UART number (q = 4)

2. ☐: Setting is fixed in the DALI 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 16-58. Initial Setting Procedure for DALI Reception



Caution After setting the DALIEN bit of peripheral enable register 1 (PER1) to 1, be sure to set serial clock select register 4 (SPS4) after 4 or more f_{CLK} clocks have elapsed.

Figure 16-59. Procedure for Stopping DALI Reception

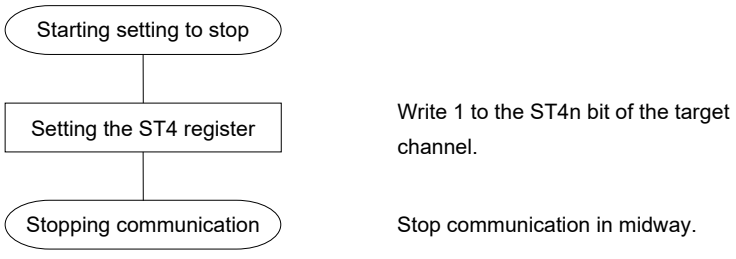
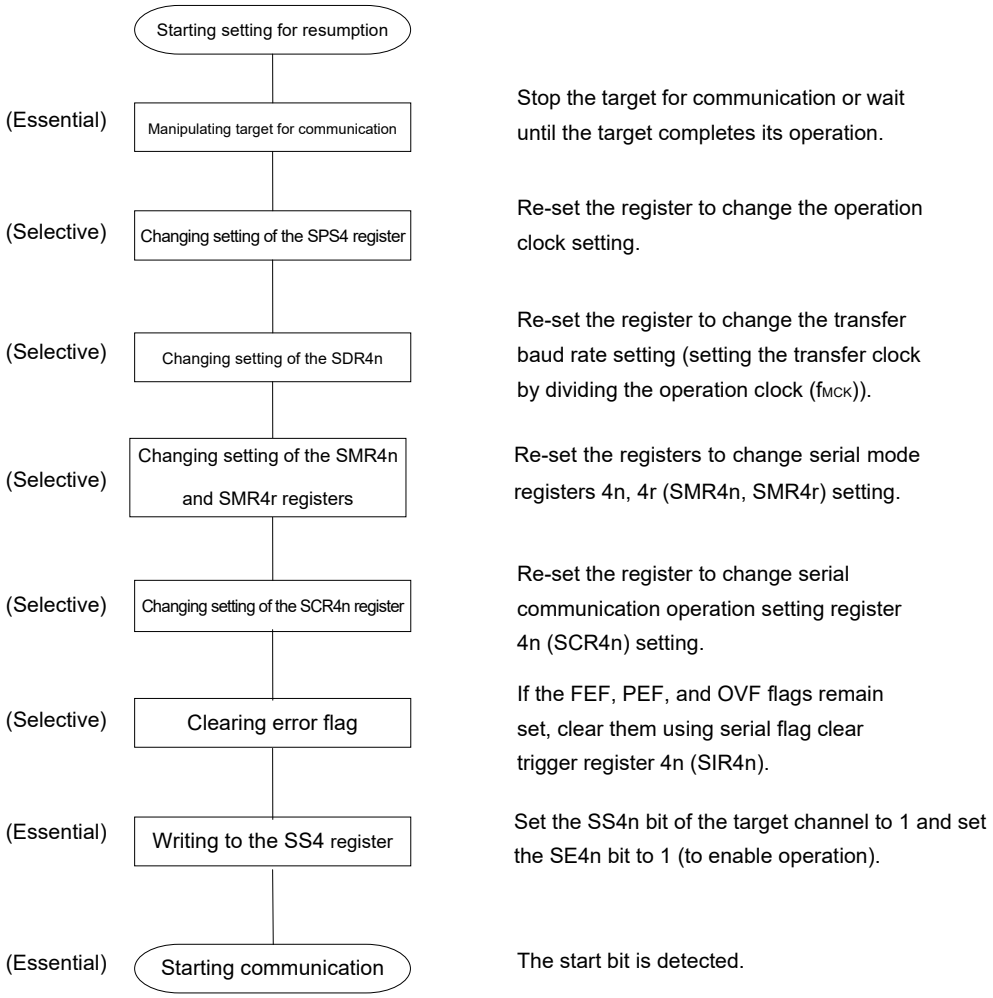
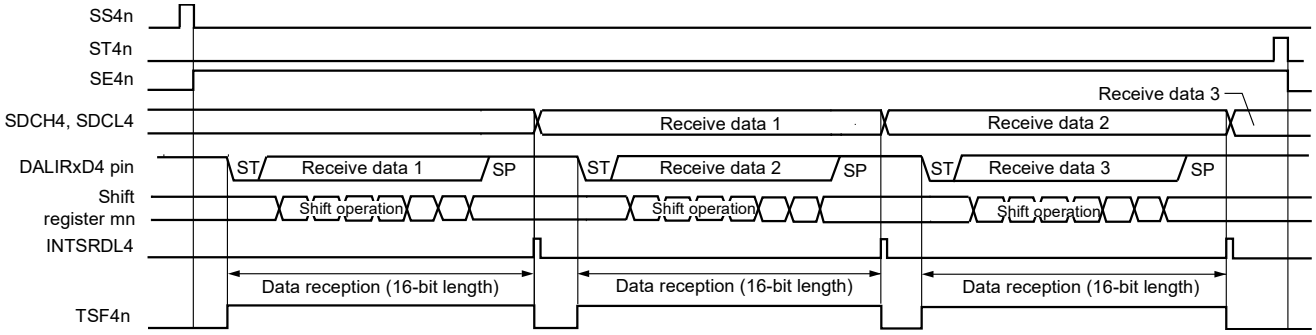


Figure 16-60. Procedure for Resuming DALI Reception



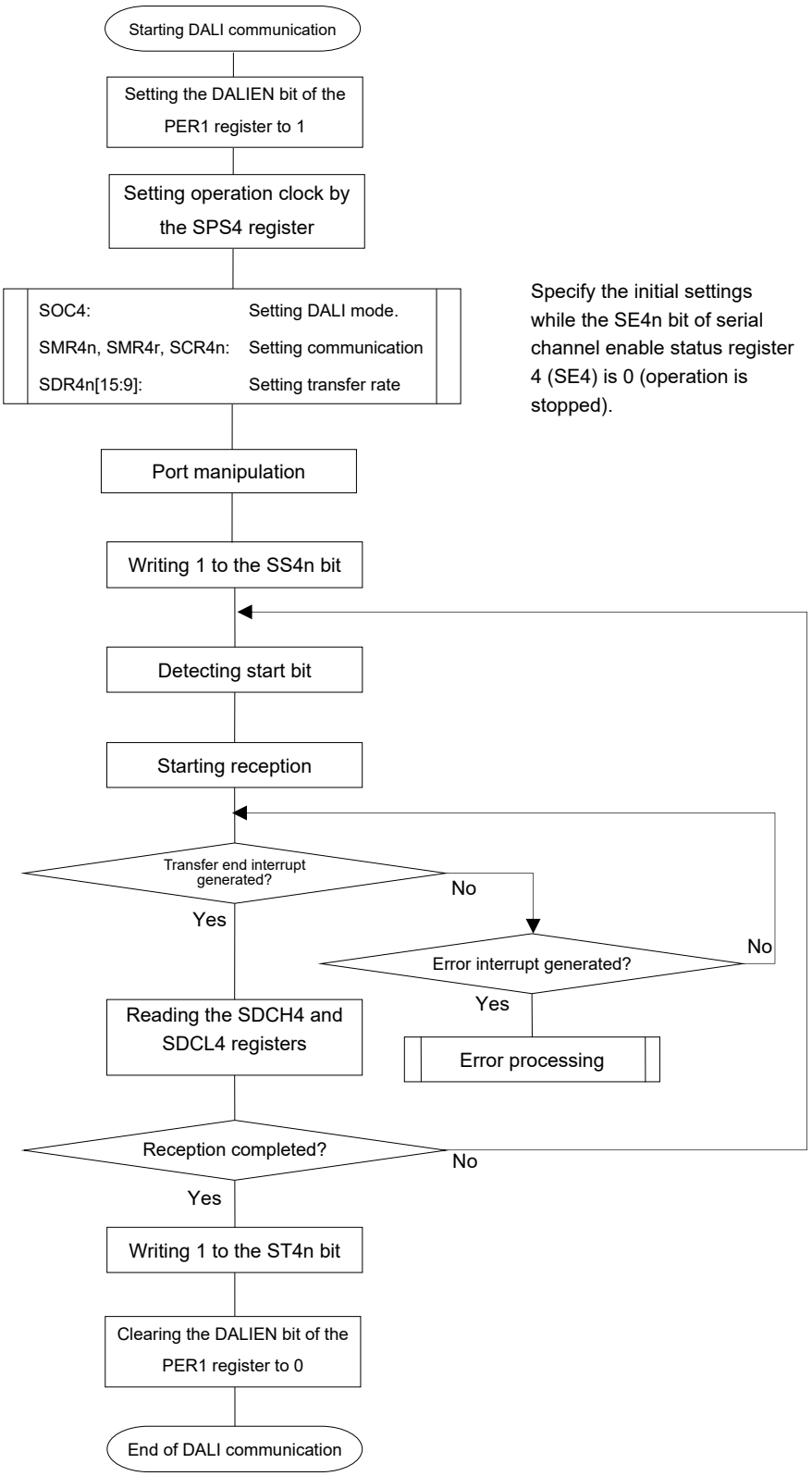
(3) Processing flow

Figure 16-61. Timing Chart of DALI Reception



Remark n: Channel number (n = 1), r: Channel number (r = 0)

Figure 16-62. Flowchart of DALI Reception



Caution After setting the DALIEN bit of peripheral enable register 1 (PER1) to 1, be sure to set serial clock select register 4 (SPS4) after 4 or more f_{CLK} clocks have elapsed.

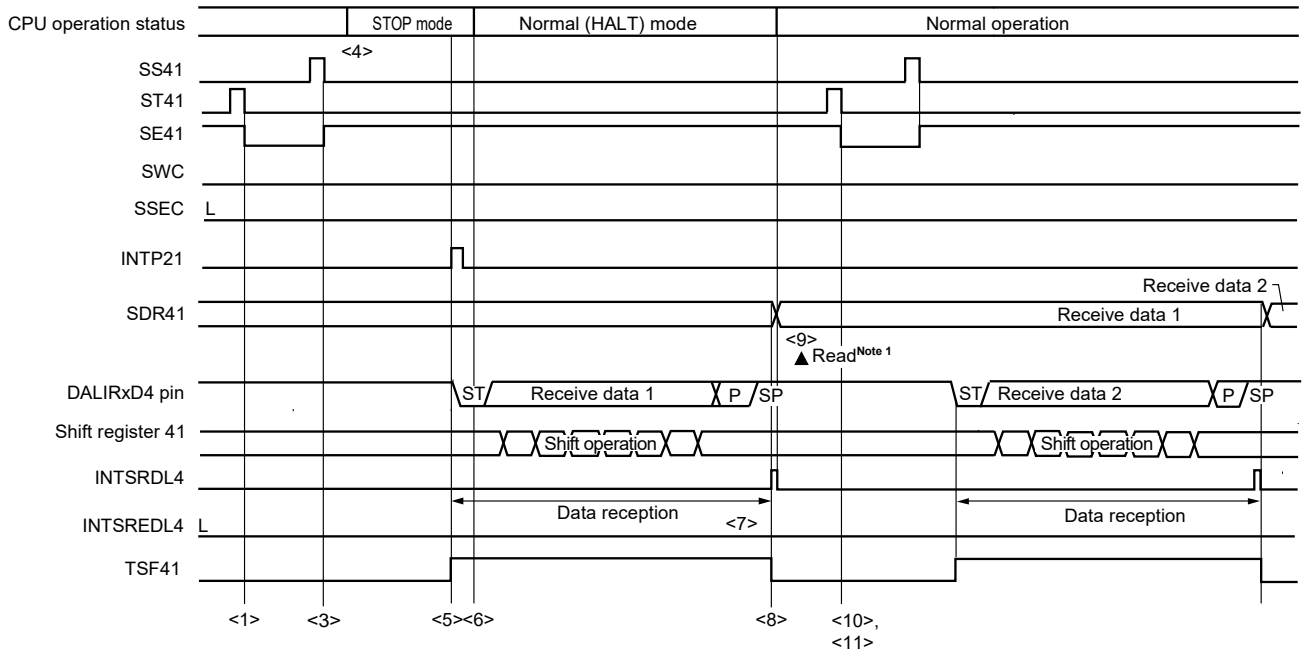
16.7 Standby Function (Only DALI/UART4 Reception)

(1) Returning from STOP & HALT modes (when DALI is received)

DALI can be received after wakeup from a STOP mode by using the interrupt function of the INTPx input. As a result, power-saving reception waiting can be realized. INTP21 or INTP23 which alternates the DALIRXD4 pin can be used as the INTPx interrupt.

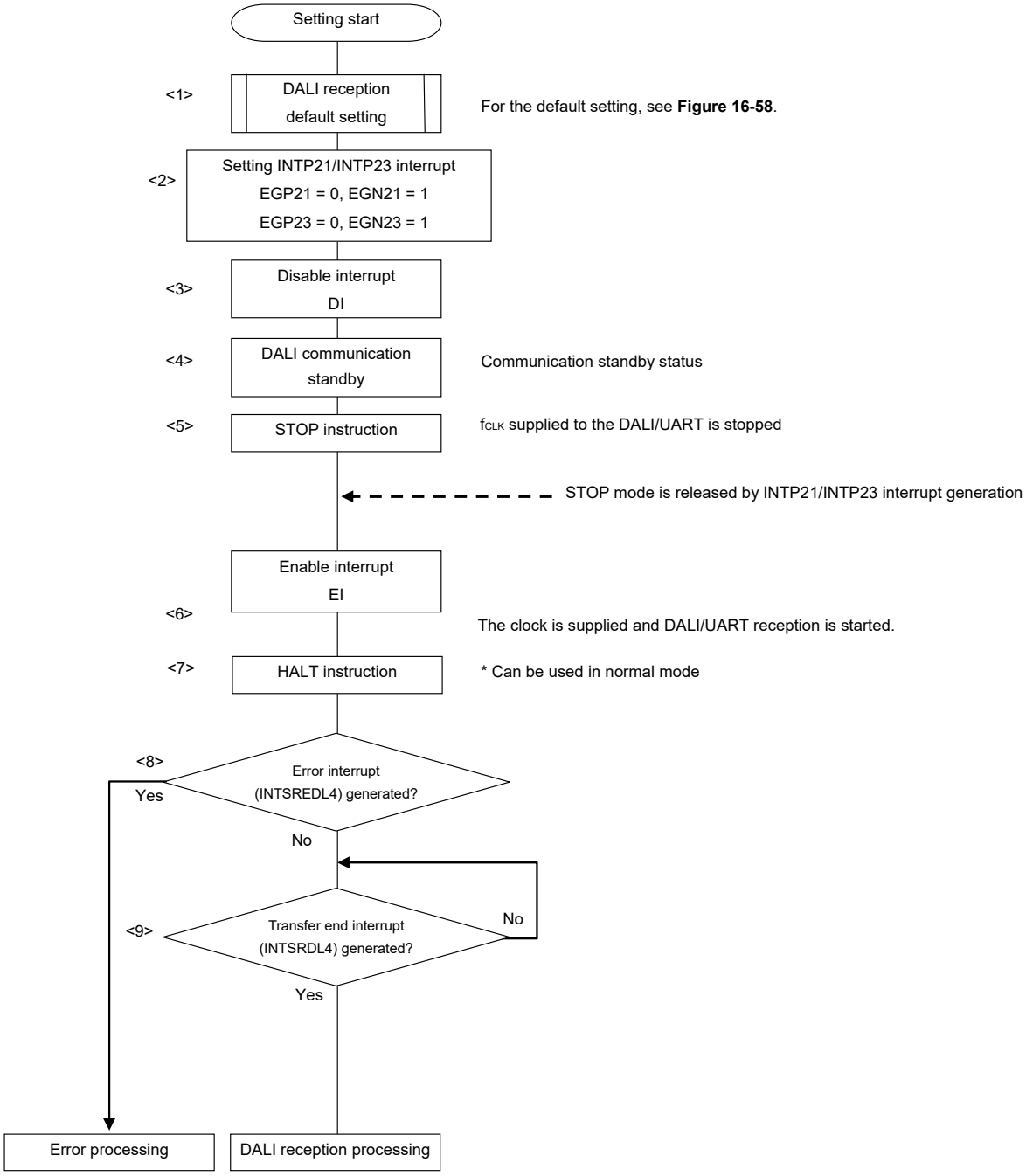
Furthermore, when combined with a HALT mode, power consumption during data reception can be reduced.

Figure 16-63. Timing Chart of Returning from STOP Mode



Remark <1> to <9> in the figure correspond to <1> to <9> in **Figure 16-64. Flowchart of Returning from STOP Mode for Reception**.

Figure 16-64. Flowchart of Returning from STOP Mode for Reception



Remarks 1. <1> to <9> in the figure correspond to <1> to <9> in **Figure 16-63. Timing Chart of Returning from STOP Mode**.

2. Keep SWC4 bit at 0 during use.

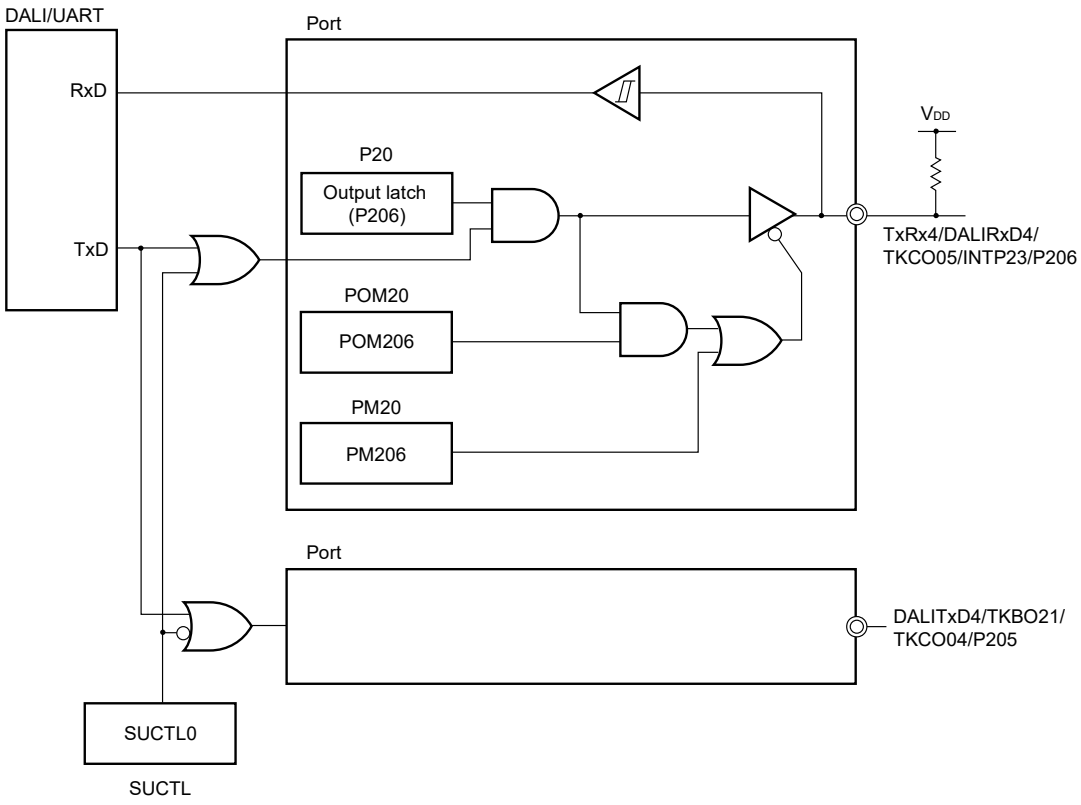
16.8 Single-wire Data Mode

When the SUCTL0 bit in the single-wire data control register (SUCTL) is set to 1, single-wire DALI/UART4 communications can be executed.

When the single-wire method is selected, the P206/TxRx4/TKCO05/DALIRxD4/INTP23 pin is shared for transmission and reception.

Figure 16-65 shows the block diagram of single-wire data mode.

Figure 16-65. Block Diagram of Single-wire Data Mode



Cautions 1. When DALI/UART4 is used in single-wire data mode, transmit data is received during a transmit operation.

2. When not used, with POM206 = 1 in the POM20 register (N-ch open drain output mode), a signal conflict may occur externally.

Remark P20: Port register 20
PM20: Port mode register 20
POM20: Port output mode register 20

16.9 Calculating Baud Rate

(1) Baud rate calculation expression

The baud rate for DALI/UART4 communication can be calculated by the following expressions.

UART communication

(Baud rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDR4n[15:9] + 1) ÷ 2 [bps]

DALI communication

(Baud rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDR4n[15:9] + 1) ÷ 4 [bps]

Caution Setting serial data register 4n (SDR4n) SDR4n[15:9] = (0000000B, 0000001B) is prohibited.

Remarks 1. When DALI/UART4 is used, the value of SDR4n[15:9] is the value of bits 15 to 9 of the SDR4n register (0000010B to 1111111B) and therefore is 2 to 127.

2. Because two bit in the UART communication corresponds to one bit in the DALI communication, DALI's baud-rate is a half of UART's baud-rate when f_{MCK} and SDR4n are set in the same way.

3. n: Channel number (n = 0, 1)

The operation clock (f_{MCK}) is determined by serial clock select register 4 (SPS4).

Table 16-3. Selection of Operation Clock For DALI/UART4

SPS4 Register				Operation Clock (f_{MCK})[Note]					
PRSm03	PRSm02	PRSm01	PRSm00	f_{CLK} = 4MHz (not using PLL)		f_{CLK} = 16MHz (using PLL)		f_{CLK} = 32 MHz (using PLL)	
0	0	0	0	f_{CLK}	4 MHz	f_{CLK}	16 MHz	f_{CLK}	32 MHz
0	0	0	1	f_{CLK}	4 MHz	f_{CLK}	16 MHz	$f_{CLK}/2$	16 MHz
0	0	1	0	f_{CLK}	4 MHz	$f_{CLK}/2$	8 MHz	$f_{CLK}/2^2$	8 MHz
0	0	1	1	f_{CLK}	4 MHz	$f_{CLK}/2^2$	4 MHz	$f_{CLK}/2^3$	4 MHz
0	1	0	0	$f_{CLK}/2$	2 MHz	$f_{CLK}/2^3$	2 MHz	$f_{CLK}/2^4$	2 MHz
0	1	0	1	$f_{CLK}/2^2$	1 kHz	$f_{CLK}/2^4$	1 kHz	$f_{CLK}/2^5$	1 kHz
0	1	1	0	$f_{CLK}/2^3$	500 kHz	$f_{CLK}/2^5$	500 kHz	$f_{CLK}/2^6$	500 kHz
0	1	1	1	$f_{CLK}/2^4$	250 kHz	$f_{CLK}/2^6$	250 kHz	$f_{CLK}/2^7$	250 kHz
1	0	0	0	$f_{CLK}/2^5$	125 kHz	$f_{CLK}/2^7$	125 kHz	$f_{CLK}/2^8$	125 kHz
1	0	0	1	$f_{CLK}/2^6$	62.5 kHz	$f_{CLK}/2^8$	62.5 kHz	$f_{CLK}/2^9$	62.5 kHz
1	0	1	0	$f_{CLK}/2^7$	31.25 kHz	$f_{CLK}/2^9$	31.25 kHz	$f_{CLK}/2^{10}$	31.25 kHz
1	0	1	1	$f_{CLK}/2^8$	15.63 kHz	$f_{CLK}/2^{10}$	15.63 kHz	$f_{CLK}/2^{11}$	15.63 kHz
1	1	0	0	$f_{CLK}/2^9$	7.81 kHz	$f_{CLK}/2^{11}$	7.81 kHz	$f_{CLK}/2^{12}$	7.81 kHz
1	1	0	1	$f_{CLK}/2^{10}$	3.91 kHz	$f_{CLK}/2^{12}$	3.91 kHz	$f_{CLK}/2^{13}$	3.91 kHz
1	1	1	0	$f_{CLK}/2^{11}$	1.95 kHz	$f_{CLK}/2^{13}$	1.95 kHz	$f_{CLK}/2^{14}$	1.95 kHz
1	1	1	1	$f_{CLK}/2^{12}$	977Hz	$f_{CLK}/2^{14}$	977Hz	$f_{CLK}/2^{15}$	977Hz

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 4 (STm) = 000FH) the operation of the serial array unit 4 (DALI/UART4).

Remark n: Channel number (n = 0, 1)

(2) Baud rate error during transmission

The baud rate error of DALI/UART4 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 baud rate at f_{CLK} = 32 MHz.

- UART communication

UART Baud Rate (Target Baud Rate)	f_{CLK} = 32 MHz			
	Operation Clock (f_{MCK})	SDR4n[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	312500 bps	+0.39 %

- DALI communication

UART Baud Rate (Target Baud Rate)	Operation Frequency (f_{CLK})	Operation Clock (f_{MCK})	SDR4n [15:9]	Calculated Baud Rate	Error from Target Baud Rate
1200 bps	32 MHz	$f_{CLK}/2^6$	103	1202 bps	+0.16 %
		$f_{CLK}/2^7$	51	1202 bps	+0.16 %
		$f_{CLK}/2^8$	25	1202 bps	+0.16 %
		$f_{CLK}/2^9$	12	1202 bps	+0.16 %
	16 MHz	$f_{CLK}/2^5$	103	1202 bps	+0.16 %
		$f_{CLK}/2^6$	51	1202 bps	+0.16 %
		$f_{CLK}/2^7$	25	1202 bps	+0.16 %
		$f_{CLK}/2^8$	12	1202 bps	+0.16 %
	4 MHz	$f_{CLK}/2^3$	103	1202 bps	+0.16 %
		$f_{CLK}/2^4$	51	1202 bps	+0.16 %
		$f_{CLK}/2^5$	25	1202 bps	+0.16 %
		$f_{CLK}/2^6$	12	1202 bps	+0.16 %
	24 MHz	$f_{CLK}/2^6$	77	1202 bps	+0.16 %
		$f_{CLK}/2^7$	38	1202 bps	+0.16 %
		$f_{CLK}/2^8$	19	1172 bps	+2.34 %
		$f_{CLK}/2^8$	18	1234 bps	+2.8 %
		$f_{CLK}/2^9$	9	1172 bps	+2.34 %
		$f_{CLK}/2^{10}$	4	1172 bps	+2.34 %
	20 MHz	$f_{CLK}/2^6$	64	1202 bps	+0.16 %
		$f_{CLK}/2^7$	32	1184 bps	+1.73 %
		$f_{CLK}/2^7$	31	1221 bps	+1.36 %
		$f_{CLK}/2^8$	15	1221 bps	+1.36 %
		$f_{CLK}/2^9$	7	1221 bps	+1.36 %
		$f_{CLK}/2^{10}$	3	1221 bps	+1.36 %

Remark n: Channel number (n = 0)

(3) Permissible baud rate range for reception (in the UART communication)

The permissible baud rate range for reception during DALI/UART4 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.

$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

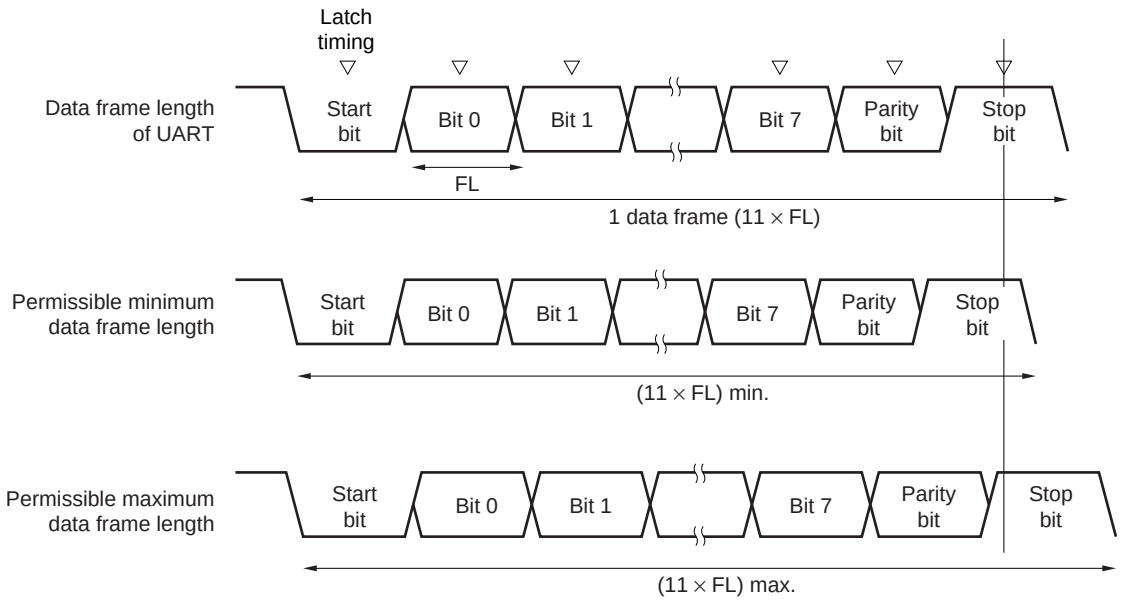
Brate: Calculated baud rate value at the reception side (See **16.9 (1) Baud rate calculation expression**.)

k: SDR4n[15:9] + 1

Nfr: 1 data frame length [bits]
= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Remark n: Channel number (n = 1)

Figure 16-66. Permissible Baud Rate Range for Reception (UART Communication, 1 Data Frame Length = 11 Bits)



As shown in Figure 16-66, the timing of latching receive data is determined by the division ratio set by bits 15 to 9 of serial data register 4n (SDR4n) after the start bit is detected. If the last data (stop bit) is received before this latch timing, the data can be correctly received.

16.10 Procedure for Processing Errors That Occurred During DALI/UART4 Communication

The procedure for processing errors that occurred during DALI/UART4 communication is described in **Figure 16-67** and **Figure 16-68**.

Figure 16-67. Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register 4n (SDR4n). →	The BFF4n bit of the SSR4n 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 4n (SSR4n).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register 4n (SIR4n). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSR4n register to the SIR4n register without modification.

Figure 16-68. Processing Procedure in Case of Framing Error and Manchester Framing Error

Software Manipulation	Hardware Status	Remark
Reads serial data register 4n (SDR4n)[Note]. →	The BFF4n bit of the SSR4n 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 4n (SSR4n).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register 4n (SIR4n). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSR4n register to the SIR4n register without modification.
Sets the ST4n bit of serial channel stop register 4 (STm) to 1. →	The SE4n bit of serial channel enable status register 4 (SEm) is set to 0 and channel n stops operating.	
Synchronization with other party of communication		Synchronization with the other party of communication is re-established and communication is resumed because it is considered that a framing error has occurred because the start bit has been shifted.
Sets the SS4n bit of serial channel start register 4 (SS4) to 1. →	The SE4n bit of serial channel enable status register 4 (SEm) is set to 1 and channel n is enabled to operate.	

Note DALI receive data register (SDCHm, SDCLm) in case of DALI reception or manchester framing error

Remark n: Channel number (n = 0, 1)

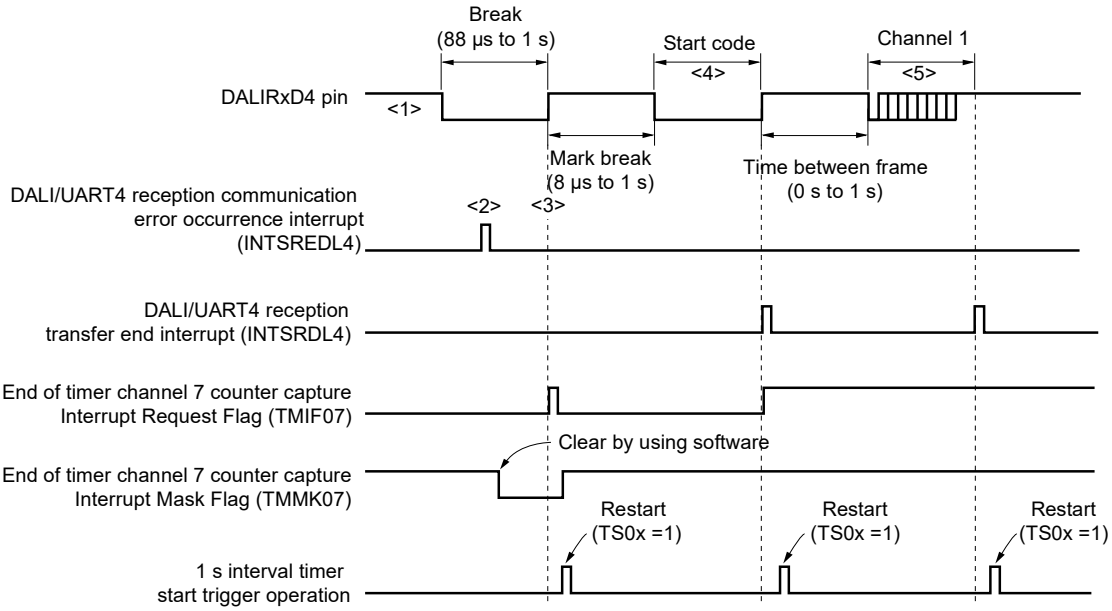
16.11 DMX512 Communication Operation

UART4 supports DMX512 communication.

UART	UART4
Support of DMX512 communication	Supported
Target channel	Channel 1
Pins used	DALIRxD4
Interrupt	INTSRDL4
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error interrupt	INTSREDL4
Error detection flag	• Framing error detection flag (FEF41) • Overrun error detection flag (OVF41)
Transfer data length	8 bits
Transfer rate	250 [kbps]
Data phase	Non-reverse output (default: high level)
Parity bit	No parity bit (The parity bit is not checked.)
Stop bit	Appending 2 bits
Data direction	MSB first or LSB first

Figure 16-69 outlines a reception operation of DMX512.

Figure 16-69. Reception Operation of DMX512



In order to receive DMX512 with DALI/UART4, the interval timer/pulse width measurement functions of TAU are used in combination.

Specifically, the timer input (TI07) of TAU is used to measure the Break width, a 1 second interval timer is created on a separate channel of TAU, and MarkBreak interval is measured to determine that it is within standard.

<Setting before reception>

- Set to initial setting 250 kbps in DALI/UART4 UART mode.
- Set TI07 as low width pulse width measurement function
- Set TAU (any channel) as 1s interval timer

<Receive procedure>

State <1> (standby state)

- Low level width measurement timer interrupt mask (TMMK07 = 1)
- DALI/UART4 = receive enabled, no interrupt mask (SREDLMK4 = 0, SRDLMK4 = 0)
- 1s interval timer = stop

State <2> (Break)

- INTSREDL4 interrupt occurs with Break signal

→Clear DALI/UART4 framing error

→Cancel low level width measurement timer interrupt mask by software (TMMK07 = 0)

State <3> (end of Break)

- Low level width measurement timer INTTM07 interrupt generated by rising signal

→ Calculate captured Break length.

If not within standard, go to state (1)

If Break width is within standard, activate 1s interval timer (TS0x = 1) and go to state <4>.

State <4> (StartCode received)

- INTSRDL4 interrupt generated by start code reception

→Check data; if 1s data is not 0, go to state <1>. If there is receive error (INTSREDL4), clear error and go to state (1).

- Clear and start 1s interval timer (TS0x = 1)

State <5> (Slotx received)

- INTSRDL4 interrupt is generated by slot reception -> Check data
- If receive error (INTSREDL4), clear error and go to state <1>.
- Clear and start 1s interval timer (TS0x = 1)

* Repeat state <5> until there is no more receive data

State <6> (if MarkBreak, TimeBetweenFrame error)

- 1s interval timer interrupt generated (INTTM0x)

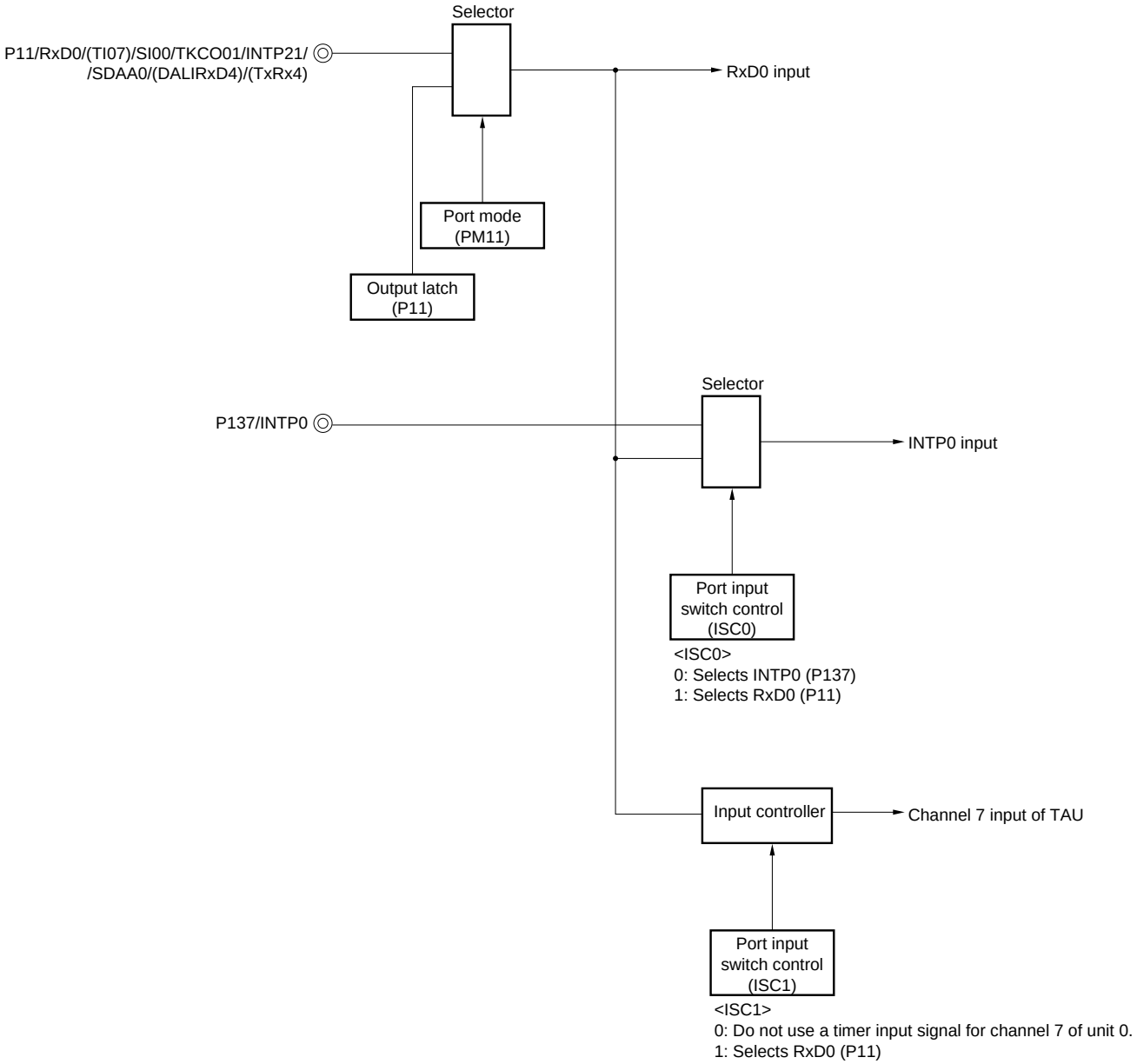
→Go to state <1>

Figure 16-70 shows the configuration of a port that manipulates reception of DMX512.

The BREAK signal transmitted from the master of DMX512 is received by detecting an edge of an external interrupt (INTP0). The input signal width transmitted from the master can be measured by using the external event capture operation of the timer array unit.

The input source of port input (DALIRxD0) for reception can be input to the external interrupt pin (INTP0) and timer array unit

Figure 16-70. Port Configuration for Manipulating Reception of DMX512



Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (See **Figure 15-20**.)

The peripheral functions used for the DMX512 communication operation are as follows.

<Peripheral functions used>

- External interrupt (INTP0); BREAK signal detection
  Usage: To detect an edge of the BREAK signal and the start of communication
- Channel 7 of timer array unit; input signal width detection
  Usage: To measure the input signal width (The interval of the edge input to DALIRxD4 is measured in the capture mode.)
- Channels 0 and 1 (UART4) of serial array unit 04(SAU)

CHAPTER 17 SERIAL INTERFACE IICA

Caution Most of the following descriptions in this chapter use the 38-pin products as an example.

17.1 Functions of Serial Interface IICA

Serial interface IICA has the following three modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

(2) I^2C bus mode (multimaster supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCLA0) line and a serial data bus (SDAA0) line.

This mode complies with the I^2C bus format and the master device can generated “start condition”, “address”, “transfer direction specification”, “data”, and “stop condition” data to the slave device, via the serial data bus. The slave device automatically detects these received status and data by hardware. This function can simplify the part of application program that controls the I^2C bus.

Since the SCLA0 and SDAA0 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 (INTIICA0) when an extension code from the master device or a local address has been received while in STOP mode. This can be set by using the WUP0 bit of IICA control register 01 (IICCTL01).

Figure 17-1 shows a block diagram of serial interface IICA.

Figure 17-1. Block Diagram of Serial Interface IICA

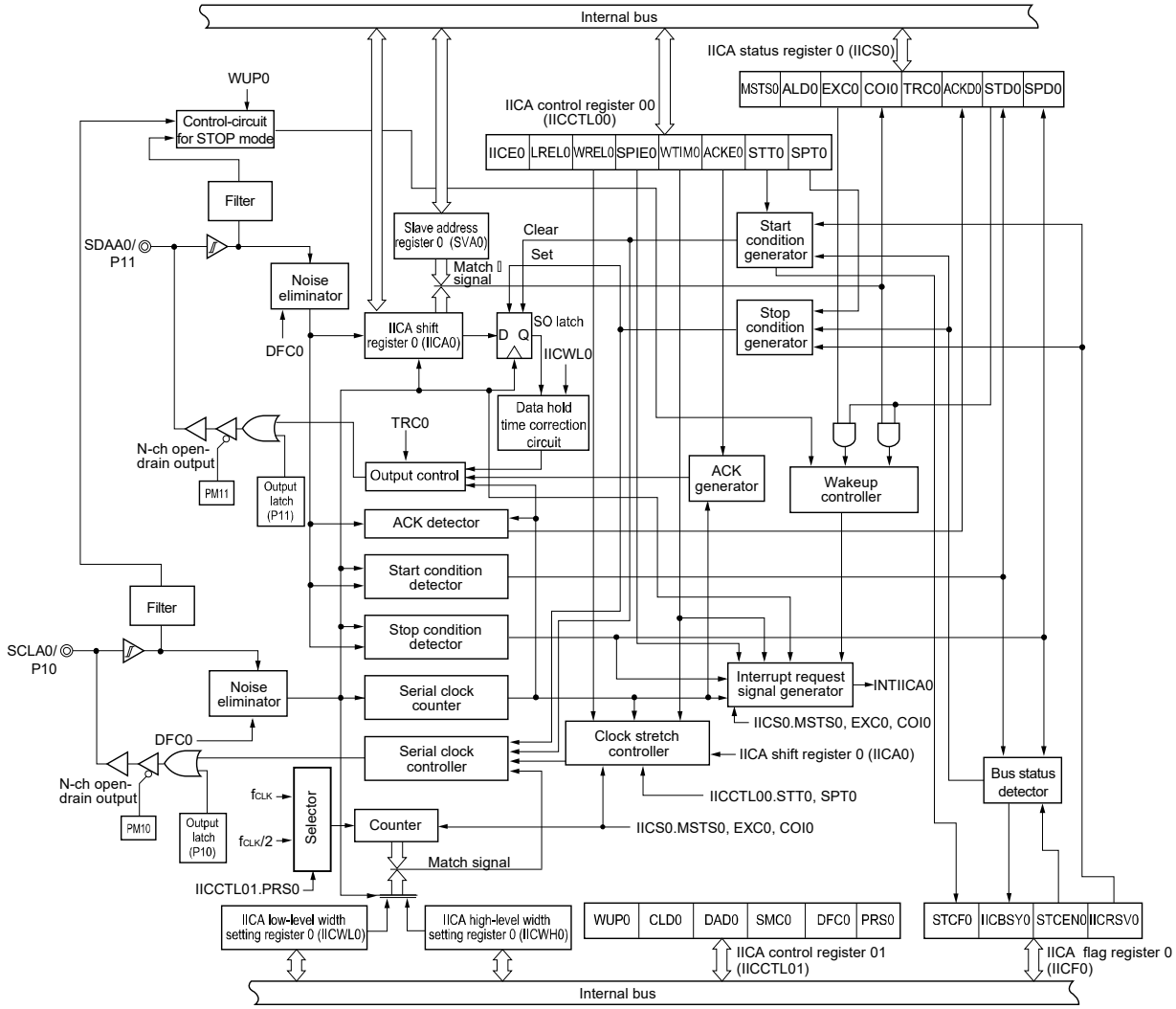
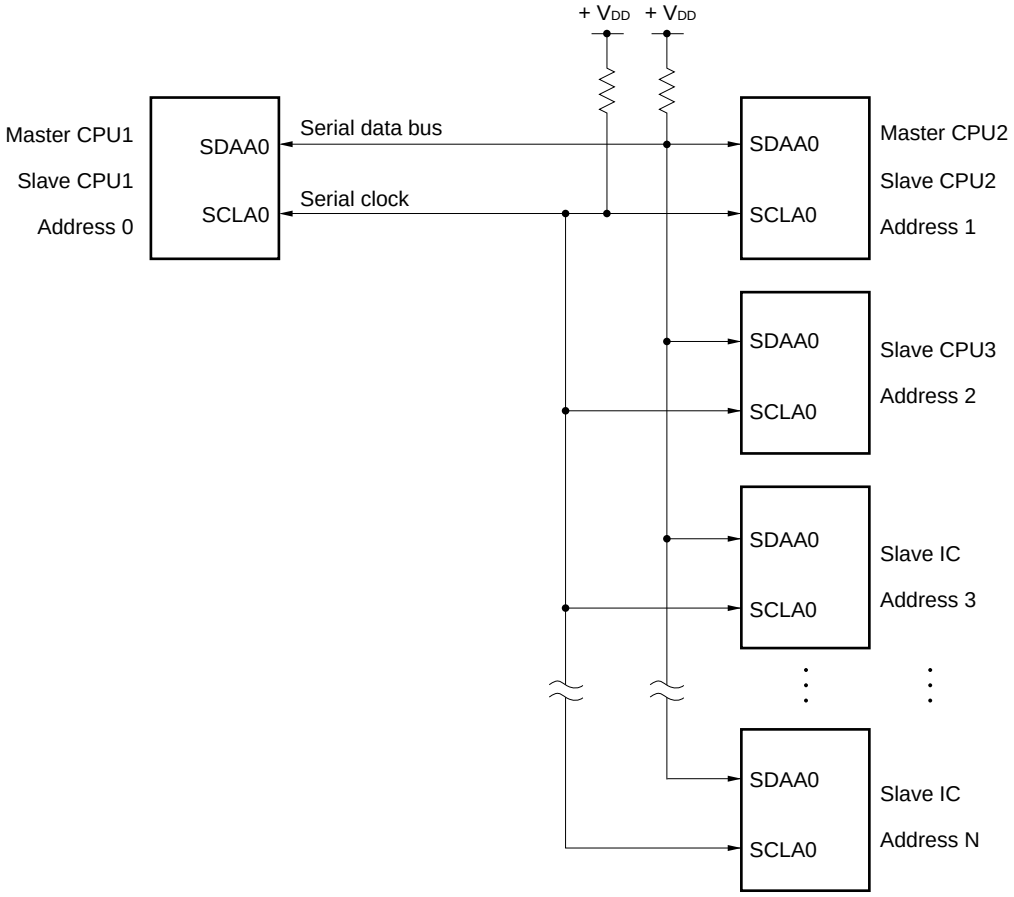


Figure 17-2 shows a serial bus configuration example.

Figure 17-2. Serial Bus Configuration Example Using I²C Bus



17.2 Configuration of Serial Interface IICA

Serial interface IICA includes the following hardware.

Table 17-1. Configuration of Serial Interface IICA

Item	Configuration
Registers	IICA shift register 0 (IICA0) Slave address register 0 (SVA0)
Control registers	Peripheral enable register 0 (PER0) IICA control register 00 (IICCTL00) IICA status register 0 (IICS0) IICA flag register 0 (IICF0) IICA control register 01 (IICCTL01) IICA low-level width setting register 0 (IICWL0) IICA high-level width setting register 0 (IICWH0) Port mode register 1 (PM1) Port output mode register 1 (POM1) Port register 1 (P1)

(1) IICA shift register 0 (IICA0)

The IICA0 register is used to convert 8-bit serial data to 8-bit parallel data and vice versa in synchronization with the serial clock. The IICA0 register can be used for both transmission and reception.

The actual transmit and receive operations can be controlled by writing and reading operations to the IICA0 register.

Cancel the clock stretch state and start data transfer by writing data to the IICA0 register during the clock stretch period.

The IICA0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears IICA0 to 00H.

Figure 17-3. Format of IICA Shift Register 0 (IICA0)

Address: FFF50H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IICA0								

Cautions 1. Do not write data to the IICA0 register during data transfer.

2. Write or read the IICA0 register only during the clock stretch period. Accessing the IICA0 register in a communication state other than during the clock stretch period is prohibited. When the device serves as the master, however, the IICA0 register can be written only once after the communication trigger bit (STT0) is set to 1.

3. When communication is reserved, write data to the IICA0 register after the interrupt triggered by a stop condition is detected.

(2) Slave address register 0 (SVA0)

This register stores seven bits of local addresses {A6, A5, A4, A3, A2, A1, A0} when in slave mode.

The SVA0 register can be set by an 8-bit memory manipulation instruction.

However, rewriting to this register is prohibited while STD0 = 1 (while the start condition is detected).

Reset signal generation clears the SVA0 register to 00H.

Figure 17-4. Format of Slave Address Register 0 (SVA0)

Address: F0234H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SVA0	A6	A5	A4	A3	A2	A1	A0	0Note

Note Bit 0 is fixed to 0.

(3) SO latch

The SO latch is used to retain the SDAA0 pin's output level.

(4) Wakeup controller

This circuit generates an interrupt request (INTIICA0) when the address received by this register matches the address value set to the slave address register 0 (SVA0) or when an extension code is received.

(5) Serial clock counter

This counter counts the serial clocks 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 (INTIICA0).

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 WTIM0 bit)
- Interrupt request generated when a stop condition is detected (set by the SPIE0 bit)

Remark WTIM0 bit: Bit 3 of IICA control register 00 (IICCTL00)
SPIE0 bit: Bit 4 of IICA control register 00 (IICCTL00)

(7) Serial clock controller

In master mode, this circuit generates the clock output via the SCLA0 pin from a sampling clock.

(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 and detect each status.

(10) Data hold time correction circuit

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

(11) Start condition generator

This circuit generates a start condition when the STT0 bit is set to 1.

However, in the communication reservation disabled status (IICRSV bit = 1), when the bus is not released (IICBSY bit = 1), start condition requests are ignored and the STCF bit is set to 1.

(12) Stop condition generator

This circuit generates a stop condition when the SPT0 bit is set to 1.

(13) Bus status detector

This circuit detects whether or not the bus is released by detecting start conditions and stop conditions.

However, as the bus status cannot be detected immediately following operation, the initial status is set by the STCEN bit.

Remark STT0 bit: Bit 1 of IICA control register 00 (IICCTL00)
SPT0 bit: Bit 0 of IICA control register 00 (IICCTL00)
IICRSV0 bit: Bit 0 of IICA flag register 0 (IICF0)
IICBSY0 bit: Bit 6 of IICA flag register 0 (IICF0)
STCF0 bit: Bit 7 of IICA flag register 0 (IICF0)
STCEN0 bit: Bit 1 of IICA flag register 0 (IICF0)

17.3 Registers Controlling Serial Interface IICA

Serial interface IICA is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- IICA control register 00 (IICCTL00)
- IICA flag register 0 (IICF0)
- IICA status register 0 (IICS0)
- IICA control register 01 (IICCTL01)
- IICA low-level width setting register 0 (IICWL0)
- IICA high-level width setting register 0 (IICWH0)
- Port mode register 1 (PM1)
- Port output mode register 1 (POM1)
- Port register 1 (P1)

17.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When serial interface IICA is used, 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.

Reset signal generation clears this register to 00H.

Figure 17-5. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	3	<2>	1	<0>
PER0	RTCEN	0	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

IICA0EN	Control of serial interface IICA input clock supply
0	Stops input clock supply. • SFR used by serial interface IICA cannot be written. • Serial interface IICA is in the reset status.
1	Enables input clock supply. • SFR used by serial interface IICA can be read/written.

Cautions 1. When setting serial interface IICA, be sure to set the following registers first while the IICA0EN bit is set to 1. If IICA0EN = 0, the control registers of serial interface IICA are set to their initial values, and writing to them is ignored (except for port mode register 1 (PM1), port output mode register 1 (POM1) and port register 1 (P1)).

- **IICA control register 00 (IICCTL00)**
- **IICA flag register 0 (IICF0)**
- **IICA status register 0 (IICS0)**
- **IICA control register 01 (IICCTL01)**
- **IICA low-level width setting register 0 (IICWL0)**
- **IICA high-level width setting register 0 (IICWH0)**

2. Be sure to clear the bits 1, 3, and 6 to “0”.

17.3.2 IICA control register 00 (IICCTL00)

This register is used to enable/stop I^2C operations, set clock stretching timing, and set other I^2C operations.

The IICCTL00 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, set the SPIE0, WTIM0, and ACKE0 bits while IICE0 = 0 or during the clock stretch period. These bits can be set at the same time when the IICE0 bit is set from “0” to “1”.

Reset signal generation clears this register to 00H.

Figure 17-6. Format of IICA Control Register 00 (IICCTL00) (1/4)

Address: F0230H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICCTL00	IICE0	LREL0	WREL0	SPIE0	WTIM0	ACKE0	STT0	SPT0

IICE0	I^2C operation enable
0	Stop operation. Reset the IICA status register 0 (IICS0)[Note 1]. Stop internal operation.
1	Enable operation.
Be sure to set this bit (1) while the SCLA0 and SDAA0 lines are at high level.	
Condition for clearing (IICE0 = 0)	Condition for setting (IICE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

LREL0 [Notes 2, 3]	Exit from communications
0	Normal operation
1	This 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 SCLA0 and SDAA0 lines are set to high impedance. The following flags of IICA control register 00 (IICCTL00) and the IICA status register 0 (IICS0) are cleared to 0. • STT0 • SPT0 • MSTS0 • EXC0 • COI0 • TRC0 • ACKD0 • STD0
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 or extension code reception occurs after the start condition.	
Condition for clearing (LREL0 = 0)	Condition for setting (LREL0 = 1)
• Automatically cleared after execution • Reset	• Set by instruction

WREL0 [Notes 2, 3]	Clock stretching cancellation
0	Do not cancel clock stretching
1	Cancel clock stretching. This setting is automatically cleared after clock stretching is canceled.
When the WREL0 bit is set (clock stretching canceled) during the clock stretch period at the ninth clock pulse in the transmission status (TRC0 = 1), the SDAA0 line goes into the high impedance state (TRC0 = 0).	
Condition for clearing (WREL0 = 0)	Condition for setting (WREL0 = 1)
• Automatically cleared after execution • Reset	• Set by instruction

Notes 1. The IICA status register 0 (IICS0), the STCF and IICBSY bits of the IICA flag register 0 (IICF0), and the CLD0 and DAD0 bits of IICA control register 01 (IICCTL01) are reset.

2. The signal of this bit is invalid while IICE0 is 0.

3. When the LREL0 and WREL0 bits are read, 0 is always read.

Caution If the operation of I^2C is enabled (IICE0 = 1) when the SCLA0 line is high level, the SDAA0 line is low level, and the digital filter is turned on (DFC0 bit of IICCTL01 register = 1), a start condition will be inadvertently detected immediately. In this case, set (1) the LREL0 bit by using a 1-bit memory manipulation instruction immediately after enabling operation of I^2C (IICE0 = 1).

Figure 17-6. Format of IICA Control Register 00 (IICCTL00) (2/4)

SPIE0[Note 1]	Enable/disable generation of interrupt request when stop condition is detected
0	Disable
1	Enable
If the WUP0 bit of IICA control register 01 (IICCTL01) is 1, no stop condition interrupt will be generated even if SPIE0 = 1.	
Condition for clearing (SPIE0 = 0)	Condition for setting (SPIE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

WTIM0[Note 1]	Control of clock stretching and interrupt request generation
0	Interrupt request is generated at the eighth clock's falling edge. Master mode: After output of eight clocks, clock output is set to low level and clock stretching is set. Slave mode: After input of eight clocks, the clock is set to low level and clock stretching is set for master device.
1	Interrupt request is generated at the ninth clock's falling edge. Master mode: After output of nine clocks, clock output is set to low level and clock stretching is set. Slave mode: After input of nine clocks, the clock is set to low level and clock stretching is set for master device.
An interrupt is generated at the falling edge of the ninth clock during address transfer independently of the setting of this bit. The setting of this bit is valid when the address transfer is completed. When in master mode, a clock stretching is inserted at the falling edge of the ninth clock during address transfers. For a slave device that has received a local address, a clock stretching is inserted at the falling edge of the ninth clock after an acknowledge (ACK) is issued. However, when the slave device has received an extension code, a clock stretching is inserted at the falling edge of the eighth clock.	
Condition for clearing (WTIM0 = 0)	Condition for setting (WTIM0 = 1)
• Cleared by instruction • Reset	• Set by instruction

ACKE0[Notes 1, 2]	Acknowledgment control
0	Disable acknowledgment.
1	Enable acknowledgment. During the ninth clock period, the SDAA0 line is set to low level.
Condition for clearing (ACKE0 = 0)	Condition for setting (ACKE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

Notes 1. The signal of this bit is invalid while IICE0 is 0. Set this bit during that period.

2. The set value is invalid during address transfer and if the code is not an extension code.
When the device serves as a slave and the addresses match, an acknowledgment is generated regardless of the set value.

Figure 17-6. Format of IICA Control Register 00 (IICCTL00) (3/4)

STT0[Notes 1, 2]	Start condition trigger
0	Do not generate a start condition.
1	When bus is released (in standby state, when IICBSY = 0): If this bit is set (1), a start condition is generated (startup as the master). When a third party is communicating: • When communication reservation function is enabled (IICRSV = 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 (IICRSV = 1) Even if this bit is set (1), the STT0 bit is cleared and the STT0 clear flag (STCF) is set (1). No start condition is generated. In the clock stretch state (when master device): Generates a restart condition after releasing the clock stretching.

Cautions concerning set timing
- For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the clock stretch period when the ACKE0 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 (SPT0).
- Once STT0 is set (1), setting it again (1) before the clear condition is met is not allowed.

Condition for clearing (STT0 = 0)	Condition for setting (STT0 = 1)
• Cleared by setting the STT0 bit to 1 while communication reservation is prohibited. • Cleared by loss in arbitration • Cleared after start condition is generated by master device • Cleared by LREL0 = 1 (exit from communications) • When IICE0 = 0 (operation stop) • Reset	• Set by instruction

Notes 1. The signal of this bit is invalid while IICE0 is 0.

2. The STT0 bit is always read as 0.

Remark IICRSV0: Bit 0 of IIC flag register 0 (IICF0)
STCF0: Bit 7 of IIC flag register 0 (IICF0)

Figure 17-6. Format of IICA Control Register 00 (IICCTL00) (4/4)

SPT0[Note]	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 in the clock stretch period when the ACKE0 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 (STT0).
- The SPT0 bit can be set to 1 only when in master mode.
- When the WTIM0 bit has been cleared to 0, if the SPT0 bit is set to 1 during the clock stretch period that follows output of eight clocks, note that a stop condition will be generated during the high-level period of the ninth clock. The WTIM0 bit should be changed from 0 to 1 during the clock stretch period following the output of eight clocks, and the SPT0 bit should be set to 1 during the clock stretch period that follows the output of the ninth clock.
- Once SPT0 is set (1), setting it again (1) before the clear condition is met is not allowed.

Condition for clearing (SPT0 = 0)	Condition for setting (SPT0 = 1)
• Cleared by loss in arbitration • Automatically cleared after stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When IICE0 = 0 (operation stop) • Reset	• Set by instruction

Note The SPT0 bit is always read as 0.

Caution When bit 3 (TRC0) of the IICA status register 0 (IICS0) is set to 1 (transmission status), bit 5 (WREL0) of IICA control register 00 (IICCTL00) is set to 1 during the ninth clock and clock stretching is canceled, after which the TRC0 bit is cleared (reception status) and the SDAA0 line is set to high impedance. Release the clock stretch performed while the TRC0 bit is 1 (transmission status) by writing to the IICA shift register 0.

Remark Bit 0 (SPT0) becomes 0 when it is read after data setting.

17.3.3 IICA status register 0 (IICS0)

This register indicates the status of I^2C.

The IICS0 register is read by a 1-bit or 8-bit memory manipulation instruction only when STT0 = 1 and during the clock stretch period.

Reset signal generation clears this register to 00H.

Caution Reading the IICS0 register while the address match wakeup function is enabled (WUP0 = 1) in STOP mode is prohibited. When the WUP0 bit is changed from 1 to 0 (wakeup operation is stopped), regardless of the INTIICA0 interrupt request, the change in status is not reflected until the next start condition or stop condition is detected. To use the wakeup function, therefore, enable (SPIE0 = 1) the interrupt generated by detecting a stop condition and read the IICS0 register after the interrupt has been detected.

Remark STT0: bit 1 of IICA control register 00 (IICCTL00)
WUP0: bit 7 of IICA control register 01 (IICCTL01)

Figure 17-7. Format of IICA Status Register 0 (IICS0) (1/3)

Address: FFF51H After reset: 00H R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICS0	MSTS0	ALD0	EXC0	COI0	TRC0	ACKD0	STD0	SPD0

MSTS0	Master status check flag
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTS0 = 0)	Condition for setting (MSTS0 = 1)
• When a stop condition is detected • When ALD0 = 1 (arbitration loss) • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is generated

ALD0	Detection of arbitration loss
0	This status means either that there was no arbitration or that the arbitration result was a “win”.
1	This status indicates the arbitration result was a “loss”. The MSTS0 bit is cleared.

Condition for clearing (ALD0 = 0)	Condition for setting (ALD0 = 1)
• Automatically cleared after the IICS0 register is read[Note] • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When the arbitration result is a “loss”.

Note This register is also cleared when a 1-bit memory manipulation instruction is executed for bits other than the IICS0 register. Therefore, when using the ALD0 bit, read the data of this bit before the data of the other bits.

Remark LREL0: Bit 6 of IICA control register 00 (IICCTL00)
IICE0: Bit 7 of IICA control register 00 (IICCTL00)

Figure 17-7. Format of IICA Status Register 0 (IICS0) (2/3)

EXC0	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXC0 = 0)	Condition for setting (EXC0 = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 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).

COI0	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COI0 = 0)	Condition for setting (COI0 = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When the received address matches the local address (slave address register 0 (SVA0)) (set at the rising edge of the eighth clock).

TRC0	Detection of transmit/receive status
0	Receive status (other than transmit status). The SDAA0 line is set for high impedance.
1	Transmit status. The value in the SO0 latch is enabled for output to the SDAA0 line (valid starting at the falling edge of the first byte’s ninth clock).

Condition for clearing (TRC0 = 0)	Condition for setting (TRC0 = 1)
<Both master and slave> • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Cleared by WREL0 = 1[Note] (clock stretching cancel) • When the ALD0 bit changes from 0 to 1 (arbitration loss) • Reset • When not used for communication (MSTS0, EXC0, COI0 = 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)

Note When bit 3 (TRC0) of the IICA status register 0 (IICS0) is set to 1 (transmission status), bit 5 (WREL0) of IICA control register 00 (IICCTL00) is set to 1 during the ninth clock and clock stretching is canceled, after which the TRC0 bit is cleared (reception status) and the SDAA0 line is set to high impedance. Release the clock stretching performed while the TRC0 bit is 1 (transmission status) by writing to the IICA shift register 0.

Remark LREL0: Bit 6 of IICA control register 00 (IICCTL00)
IICE0: Bit 7 of IICA control register 00 (IICCTL00)

Figure 17-7. Format of IICA Status Register 0 (IICS0) (3/3)

ACKD0	Detection of acknowledge (ACK)
0	Acknowledge was not detected.
1	Acknowledge was detected.

Condition for clearing (ACKD0 = 0)	Condition for setting (ACKD0 = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• After the SDAA0 line is set to low level at the rising edge of SCLA0 line's ninth clock

STD0	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 (STD0 = 0)	Condition for setting (STD0 = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is detected

SPD0	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 (SPD0 = 0)	Condition for setting (SPD0 = 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 WUP0 bit changes from 1 to 0 • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a stop condition is detected

Remark LREL0: Bit 6 of IICA control register 00 (IICCTL00)
IICE0: Bit 7 of IICA control register 00 (IICCTL00)

17.3.4 IICA flag register 0 (IICF0)

This register sets the operation mode of I^2C and indicates the status of the I^2C bus.

The IICF0 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the STT0 clear flag 0 (STCF0) and I^2C bus status flag 0 (IICBSY0) bits are read-only.

The IICRSV0 bit can be used to enable/disable the communication reservation function.

The STCEN0 bit can be used to set the initial value of the IICBSY0 bit.

The IICRSV0 and STCEN0 bits can be written only when the operation of I^2C is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) = 0). When operation is enabled, the IICF0 register can be read.

Reset signal generation clears this register to 00H.

Figure 17-8. Format of IICA Flag Register 0 (IICF0)

Address: FFF52H After reset: 00H R/W[Note]

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
IICF0	STCF0	IICBSY0	0	0	0	0	STCEN0	IICRSV0

STCF0	STT0 clear flag
0	Generate start condition
1	Start condition generation unsuccessful: clear the STT0 flag

Condition for clearing (STCF0 = 0)	Condition for setting (STCF0 = 1)
• Cleared by STT0 = 1 • When IICE0 = 0 (operation stop) • Reset	• Generating start condition unsuccessful and the STT0 bit cleared to 0 when communication reservation is disabled (IICRSV0 = 1).

IICBSY0	I^2C bus status flag
0	Bus release status (communication initial status when STCEN0 = 1)
1	Bus communication status (communication initial status when STCEN0 = 0)

Condition for clearing (IICBSY0 = 0)	Condition for setting (IICBSY0 = 1)
• Detection of stop condition • When IICE0 = 0 (operation stop) • Reset	• Detection of start condition • Setting of the IICE0 bit when STCEN0 = 0

STCEN0	Initial start enable trigger
0	After operation is enabled (IICE0 = 1), enable generation of a start condition upon detection of a stop condition.
1	After operation is enabled (IICE0 = 1), enable generation of a start condition without detecting a stop condition.

Condition for clearing (STCEN0 = 0)	Condition for setting (STCEN0 = 1)
• Cleared by instruction • Detection of start condition • Reset	• Set by instruction

IICRSV0	Communication reservation function disable bit
0	Enable communication reservation
1	Disable communication reservation

Condition for clearing (IICRSV0 = 0)	Condition for setting (IICRSV0 = 1)
• Cleared by instruction • Reset	• Set by instruction

Note Bits 6 and 7 are read-only.

Cautions 1. Write to the STCEN0 bit only when the operation is stopped (IICE0 = 0).

2. As the bus release status (IICBSY0 = 0) is recognized regardless of the actual bus status when STCEN0 = 1, when generating the first start condition (STT0 = 1), it is necessary to verify that no third party communications are in progress in order to prevent such communications from being destroyed.

3. Write to IICRSV0 only when the operation is stopped (IICE0 = 0).

Remark STT0: Bit 1 of IICA control register 00 (IICCTL00)
IICE0: Bit 7 of IICA control register 00 (IICCTL00)

17.3.5 IICA control register 01 (IICCTL01)

This register is used to set the operation mode of I^2C and detect the statuses of the SCLA0 and SDAA0 pins.

The IICCTL01 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the CLD0 and DAD0 bits are read-only.

Set the IICCTL01 register, except the WUP0 bit, while operation of I^2C is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) is 0).

Reset signal generation clears this register to 00H.

Figure 17-9. Format of IICA Control Register 01 (IICCTL01) (1/2)

Address: F0231H After reset: 00H R/W[Note 1]

Symbol	7	6	<5>	<4>	<3>	<2>	1	<0>
IICCTL01	WUP0	0	CLD0	DAD0	SMC0	DFC0	0	PRS0

WUP0	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 WUP0 = 1, execute the STOP instruction at least three cycles of f_{MCK} after setting (1) the WUP0 bit (see **Figure 17-23 Flow When Setting WUP0 = 1**).
Clear (0) the WUP0 bit after the address has matched or an extension code has been received. The subsequent communication can be entered by the clearing (0) WUP0 bit. (The clock stretching must be released and transmit data must be written after the WUP0 bit has been cleared (0).)
The interrupt timing when the address has matched or when an extension code has been received, while WUP0 = 1, is identical to the interrupt timing when WUP0 = 0. (A delay of the difference of sampling by the clock will occur.) Furthermore, when WUP0 = 1, a stop condition interrupt is not generated even if the SPIE0 bit is set to 1.
When WUP0 = 0 is set by a source other than an interrupt from serial interface IICA, operation as the master device cannot be performed until the subsequent start condition or stop condition is detected. Do not output a start condition by setting (1) the STT0 bit, without waiting for the detection of the subsequent start condition or stop condition.

Condition for clearing (WUP0 = 0)	Condition for setting (WUP0 = 1)
• Cleared by instruction (after address match or extension code reception)	• Set by instruction (when the MSTS0, EXC0, and COI0 bits are “0”, and the STD0 bit also “0” (communication not entered))[Note 2]

Notes 1. Bits 4 and 5 are read-only.

2. The status of the IICA status register 0 (IICS0) must be checked and the WUP0 bit must be set during the period shown below.

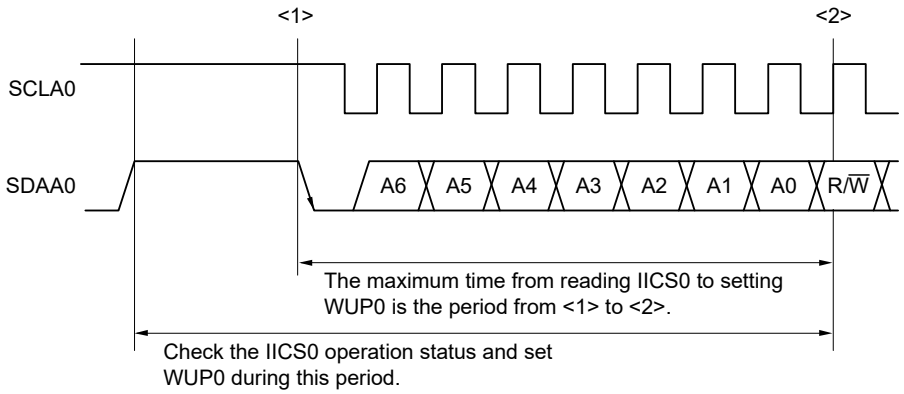


Figure 17-9. Format of IICA Control Register 01 (IICCTL01) (2/2)

CLD0	Detection of SCLA0 pin level (valid only when IICE0 = 1)
0	The SCLA0 pin was detected at low level.
1	The SCLA0 pin was detected at high level.

Condition for clearing (CLD0 = 0)	Condition for setting (CLD0 = 1)
• When the SCLA0 pin is at low level • When IICE0 = 0 (operation stop) • Reset	• When the SCLA0 pin is at high level

DAD0	Detection of SDAA0 pin level (valid only when IICE0 = 1)
0	The SDAA0 pin was detected at low level.
1	The SDAA0 pin was detected at high level.

Condition for clearing (DAD0 = 0)	Condition for setting (DAD0 = 1)
• When the SDAA0 pin is at low level • When IICE0 = 0 (operation stop) • Reset	• When the SDAA0 pin is at high level

SMC0	Operation mode switching
0	Operates in normal mode (fastest transfer rate: 100 kbps).
1	Operates in fast mode (fastest transfer rate: 400 kbps).

DFC0	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
Use the digital filter only in fast mode. The digital filter is used for noise elimination. In fast mode, the transfer clock does not vary, regardless of the DFC0 bit being set (1) or cleared (0). The digital filter is used for noise elimination in fast mode.	

PRS0	IICA operation clock (f_{MCK}) control
0	Selects f_{CLK} (1 MHz ≤ f_{CLK} ≤ 20 MHz)
1	Selects $f_{CLK}/2$ (20 MHz < f_{CLK})

Cautions 1. The fastest operation frequency of the IICA operation clock (f_{MCK}) is 20 MHz (max.). Set bit 0 (PRS0) of the IICA control register 01 (IICCTL01) to “1” only when the f_{CLK} exceeds 20 MHz.

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.)

Normal mode: f_{CLK} = 1 MHz (min.)

Remark IICE0: Bit 7 of IICA control register 00 (IICCTL00)

17.3.6 IICA low-level width setting register 0 (IICWL0)

This register is used to set the low-level width (t_{LOW}) of the SCLA0 pin signal that is output by serial interface IICA and to control the SDAA0 pin signal.

The IICWL0 register can be set by an 8-bit memory manipulation instruction.

Set the IICWL0 register while operation of I^2C is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) is 0).

Reset signal generation sets this register to FFH.

For details about setting the IICWL0 register, see **17.4.2 Setting transfer clock by using IICWL0 and IICWH0 registers**.

The data hold time is one-quarter of the time set by the IICWL0 register.

Figure 17-10. Format of IICA Low-Level Width Setting Register 0 (IICWL0)

Address: F0232H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWL0								

17.3.7 IICA high-level width setting register 0 (IICWH0)

This register is used to set the high-level width of the SCLA0 pin signal that is output by serial interface IICA and to control the SDAA0 pin signal.

The IICWH0 register can be set by an 8-bit memory manipulation instruction.

Set the IICWH0 register while operation of I^2C is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) is 0).

Reset signal generation sets this register to FFH.

Figure 17-11. Format of IICA High-Level Width Setting Register 0 (IICWH0)

Address: F0233H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWH0								

Remark For setting procedures of the transfer clock on master side and of the IICWL0 and IICWH0 registers on slave side, see **17.4.2 (1)** and **17.4.2 (2)**, respectively.

17.3.8 Port mode register 1 (PM1)

This register sets the input/output of port 1 in 1-bit units.

When using the P10/SCLA0/SO00/TxD0/TKCO00/INTP20/(DALITxD4) pin as clock I/O and the P11/SDAA0/SI00/RxD0/TKCO01/INTP21/(TI07)/(DALIRxD4)/(TxRx4) pin as serial data I/O, clear PM10 and PM11, and the output latches of P10 and P11 to 0.

Set the IICE0 bit (bit 7 of IICA control register 00 (IICCTL00)) to 1 before setting the output mode because the P10/SCLA0/SO00/TxD0/TKCO00/INTP20/(DALITxD4) and P11/SDAA0/SI00/RxD0/TKCO01/INTP21/(TI07)/(DALIRxD4)/(TxRx4) pins output a low level (fixed) when the IICE0 bit is 0.

The PM1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 17-12. Format of Port Mode Register 1 (PM1)

Address FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	1	1	1	1	1	PM12	PM11	PM10

PM1n	P1n pin output buffer selection (n = 0 to 2)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

Caution Be sure to set bits 3 to 7 of the PM1 register to "1".
For 30- and 20-pin products, bit 2 of the PM1 register must be set to output mode (specified by setting the port registers and port mode registers to 0) by software after reset is cancelled.

17.3.9 Port output mode register 1 (POM1)

The register sets the output mode of P10 to P12 in 1-bit units.

N-ch open drain output (V_{DD} tolerance) mode can be selected for the P11/SDAA0/SI00/RxD0/TKCO01/INTP21/(TI07)/(DALIRxD4)/(TxRx4) pin during I^2C communication.

The POM1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the POM1 register to 00H.

Figure 17-13. Format of Port Output Mode Registers 1 (POM1)

Address F0051H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
POM1	0	0	0	0	0	POM12	POM11	POM10

POM1n	P1n pin output buffer selection (n = 0 to 2)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

17.4 I²C Bus Mode Functions

17.4.1 Pin configuration

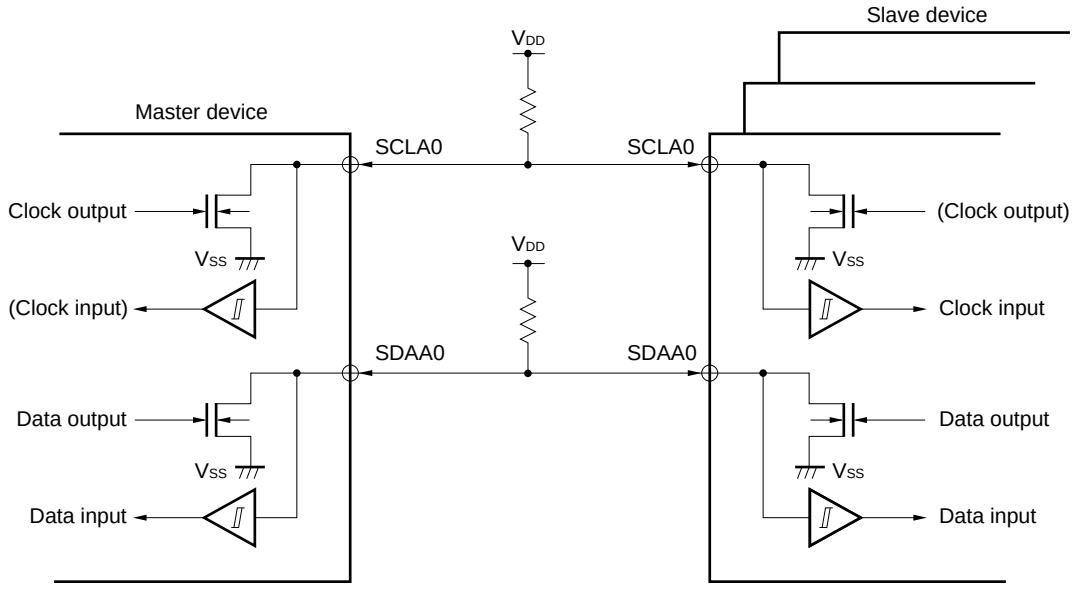
The serial clock pin (SCLA0) and the serial data bus pin (SDAA0) are configured as follows.

(1) SCLA0 .... 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) SDAA0 .... 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 17-14. Pin Configuration Diagram



17.4.2 Setting transfer clock by using IICWL0 and IICWH0 registers

(1) Setting transfer clock on master side

$$\text{Transfer clock} = \frac{f_{MCK}}{IICWL0 + IICWH0 + f_{MCK}(t_R + t_F)}$$

At this time, the optimal setting values of the IICWL0 and IICWH0 registers are as follows.
(The fractional parts of all setting values are rounded up.)

- When the fast mode

$$IICWL0 = \frac{0.52}{\text{Transfer clock}} \times f_{MCK}$$
$$IICWH0 = \left(\frac{0.48}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

- When the normal mode

$$IICWL0 = \frac{0.47}{\text{Transfer clock}} \times f_{MCK}$$
$$IICWH0 = \left(\frac{0.53}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

(2) Setting IICWL0 and IICWH0 registers on slave side

(The fractional parts of all setting values are truncated.)

- When the fast mode

$$IICWL0 = 1.3\ \mu s \times f_{MCK}$$
$$IICWH0 = (1.2\ \mu s - t_R - t_F) \times f_{MCK}$$

- When the normal mode

$$IICWL0 = 4.7\ \mu s \times f_{MCK}$$
$$IICWH0 = (5.3\ \mu s - t_R - t_F) \times f_{MCK}$$

Cautions 1. The fastest operation frequency of the IICA operation clock (f_{MCK}) is 20 MHz (Max.). Set bit 0 (PRS0) of the IICA control register 01 (IICCTL01) to "1" only when the f_{CLK} exceeds 20 MHz.

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.)

Normal mode: f_{CLK} = 1 MHz (min.)

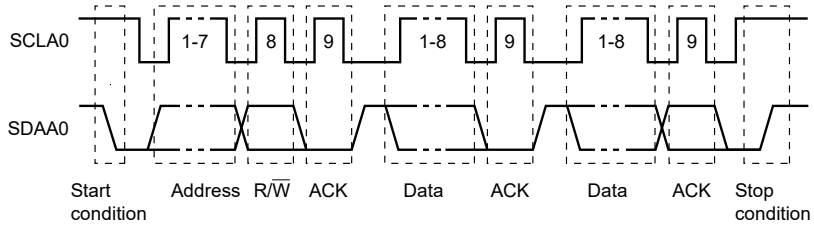
Remarks 1. Calculate the rise time (t_R) and fall time (t_F) of the SDAA0 and SCLA0 signals separately, because they differ depending on the pull-up resistance and wire load.

2. IICWL0: IICA low-level width setting register 0
IICWH0: IICA high-level width setting register 0
t_F: SDAA0 and SCLA0 signal falling times
t_R: SDAA0 and SCLA0 signal rising times
f_{MCK}: IICA operation clock frequency

17.5 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 17-15 shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I²C bus's serial data bus.

Figure 17-15. I²C Bus Serial Data Transfer Timing



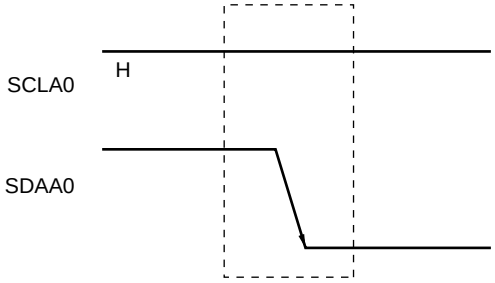
The master device generates the start condition, slave address, and stop condition.
The acknowledge (ACK) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCLA0) is continuously output by the master device. However, in the slave device, the SCLA0 pin low level period can be extended and a clock stretching can be inserted.

17.5.1 Start conditions

A start condition is met when the SCLA0 pin is at high level and the SDAA0 pin changes from high level to low level. The start conditions for the SCLA0 pin and SDAA0 pin are signals 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 17-16. Start Conditions



A start condition is output when bit 1 (STT0) of IICA control register 00 (IICCTL00) is set (1) after a stop condition has been detected (SPD0: Bit 0 of the IICA status register 0 (IICS0) = 1). When a start condition is detected, bit 1 (STD0) of the IICS0 register is set (1).

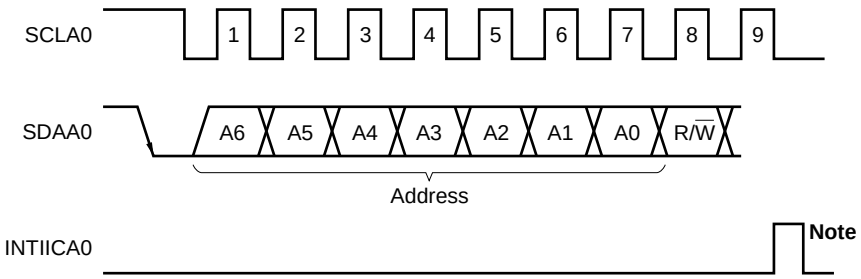
17.5.2 Addresses

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 0 (SVA0). If the address data matches the SVA0 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 17-17. Address



Note INTIICA0 is not issued if data other than a local address or extension code is received during slave device operation.

Addresses are output when a total of 8 bits consisting of the slave address and the transfer direction described in **17.5.3 Transfer direction specification** are written to the IICA shift register 0 (IICA0). The received addresses are written to the IICA0 register.

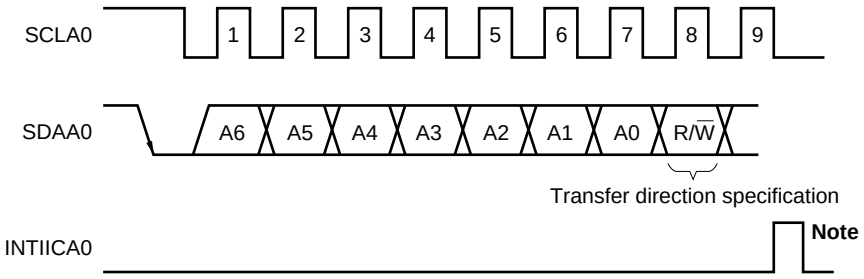
The slave address is assigned to the higher 7 bits of the IICA0 register.

17.5.3 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 17-18. Transfer Direction Specification



Note INTIICA0 is not issued if data other than a local address or extension code is received during slave device operation.

17.5.4 Acknowledge (ACK)

ACK is used to check the status 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 (ACKD0) of the IICA status register 0 (IICS0).

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.

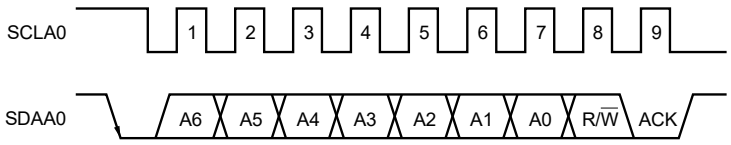
To generate ACK, the reception side makes the SDAA0 line low at the ninth clock (indicating normal reception).

Automatic generation of ACK is enabled by setting bit 2 (ACKE0) of IICA control register 00 (IICCTL00) to 1. Bit 3 (TRC0) of the IICS0 register is set by the data of the eighth bit that follows 7-bit address information. Usually, set the ACKE0 bit to 1 for reception (TRC0 = 0).

If a slave can receive no more data during reception (TRC0 = 0) or does not require the next data item, then the slave must inform the master, by clearing the ACKE0 bit to 0, that it will not receive any more data.

When the master does not require the next data item during reception (TRC0 = 0), it must clear the ACKE0 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 17-19. ACK



When the local address is received, ACK is automatically generated, regardless of the value of the ACKE0 bit. When an address other than that of the local address is received, ACK is not generated (NACK).

When an extension code is received, ACK is generated if the ACKE0 bit is set to 1 in advance.

How ACK is generated when data is received differs as follows depending on the setting of the clock stretching timing.

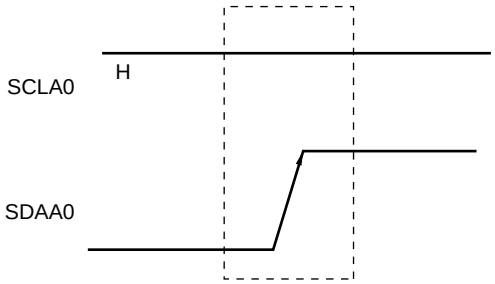
- When 8-clock clock stretching state is selected (bit 3 (WTIM0) of IICCTL00 register = 0):
  By setting the ACKE0 bit to 1 before releasing the clock stretch state, ACK is generated at the falling edge of the eighth clock of the SCLA0 pin.
- When 9-clock clock stretching state is selected (bit 3 (WTIM0) of IICCTL00 register = 1):
  ACK is generated by setting the ACKE0 bit to 1 in advance.

17.5.5 Stop condition

When the SCLA0 pin is at high level, changing the SDAA0 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 17-20. Stop Condition



A stop condition is generated when bit 0 (SPT0) of IICA control register 00 (IICCTL00) is set to 1. When the stop condition is detected, bit 0 (SPD0) of the IICA status register 0 (IICS0) is set to 1 and INTIICA0 is generated when bit 4 (SPIE0) of the IICCTL00 register is set to 1.

17.5.6 Clock stretching

The clock stretching is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a clock stretch state).

Setting the SCLA0 pin to low level notifies the communication partner of the clock stretch state. When clock stretch state has been canceled for both the master and slave devices, the next data transfer can begin.

Figure 17-21. Clock stretching (1/2)

(1) When clock stretching is set for the ninth and eighth clock cycles for the master and slave devices, respectively
(master transmits, slave receives, and ACKE0 = 1)

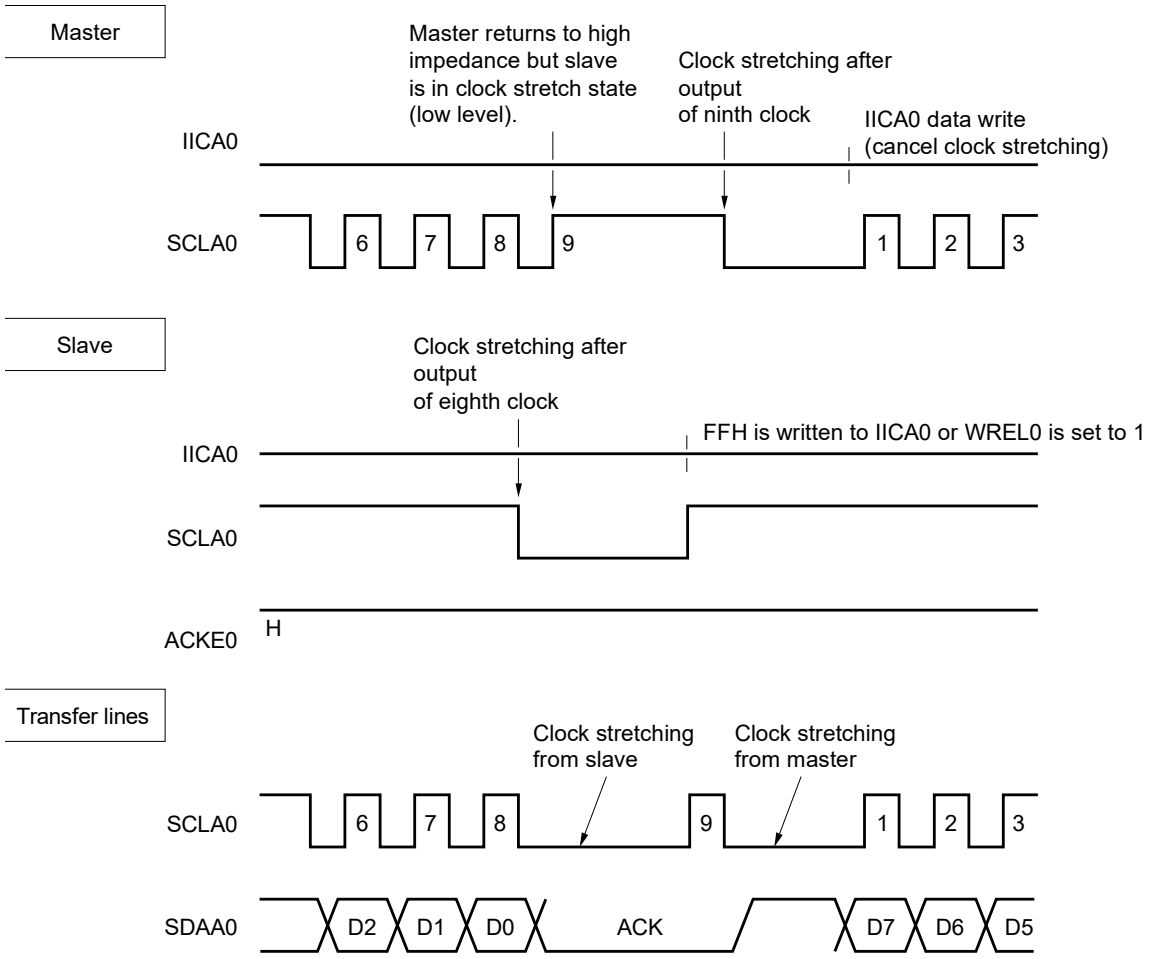
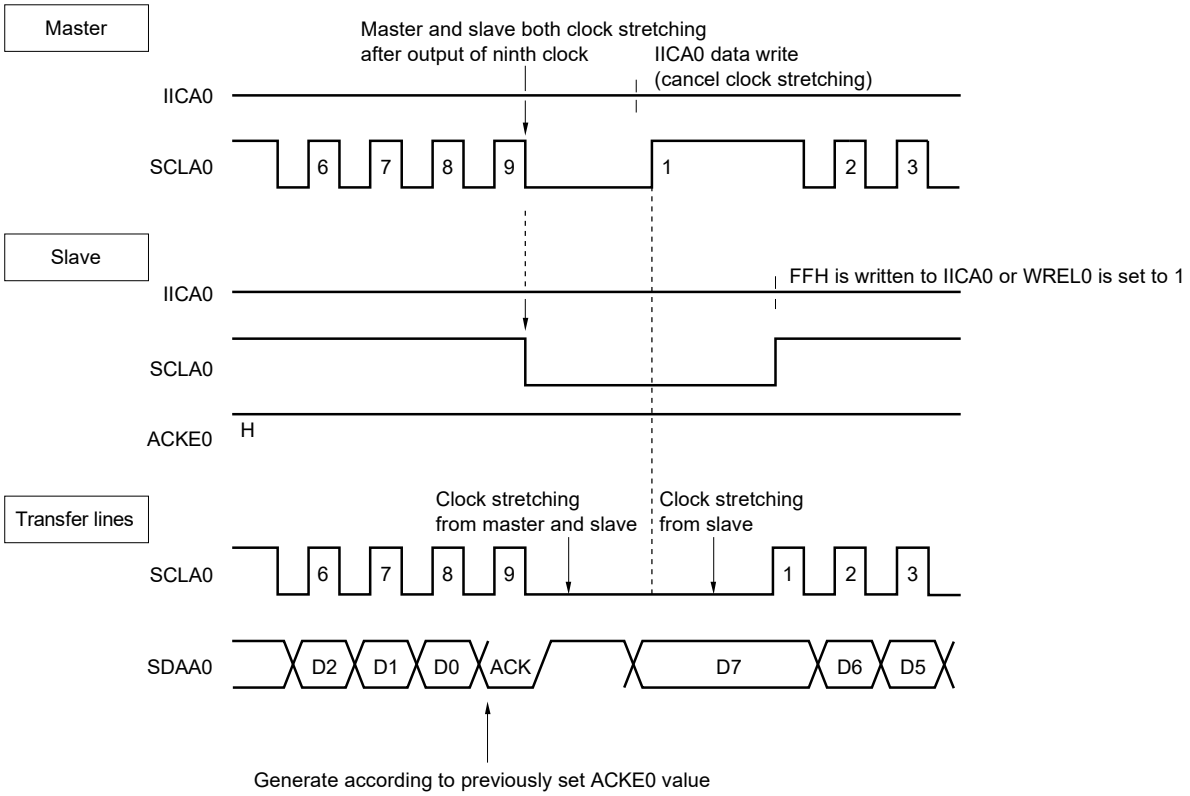


Figure 17-21. Clock stretching (2/2)

(2) When clock stretching is set for the ninth clock cycle for both the master and slave devices (master transmits, slave receives, and ACKE0 = 1)



Remark ACKE0: Bit 2 of IICA control register 00 (IICCTL00)
WREL0: Bit 5 of IICA control register 00 (IICCTL00)

A clock stretching may be automatically generated depending on the setting of bit 3 (WTIM0) of IICA control register 00 (IICCTL00).

Normally, the receiving side cancels the clock stretch state when bit 5 (WREL0) of the IICCTL00 register is set to 1 or when FFH is written to the IICA shift register 0 (IICA0), and the transmitting side cancels the clock stretch state when data is written to the IICA0 register.

The master device can also cancel the clock stretch state via either of the following methods.

- By setting bit 1 (STT0) of the IICCTL00 register to 1
- By setting bit 0 (SPT0) of the IICCTL00 register to 1

17.5.7 Canceling clock stretching

The I^2C usually cancels a clock stretch state by the following processing.

- Writing data to the IICA shift register 0 (IICA0)
- Setting bit 5 (WREL0) of IICA control register 00 (IICCTL00) (canceling clock stretching)
- Setting bit 1 (STT0) of the IICCTL00 register (generating start condition)**Note**
- Setting bit 0 (SPT0) of the IICCTL00 register (generating stop condition)**Note**

Note Master only

When the above clock stretching canceling processing is executed, the I^2C cancels the clock stretch state and communication is resumed.

To cancel a clock stretch state and transmit data (including addresses), write the data to the IICA0 register.

To receive data after canceling a clock stretch state, or to complete data transmission, set bit 5 (WREL0) of the IICCTL00 register to 1.

To generate a restart condition after canceling a clock stretch state, set bit 1 (STT0) of the IICCTL00 register to 1.

To generate a stop condition after canceling a clock stretch state, set bit 0 (SPT0) of the IICCTL00 register to 1.

Execute the canceling processing only once for one clock stretch state.

If, for example, data is written to the IICA0 register after canceling a clock stretch state by setting the WREL0 bit to 1, an incorrect value may be output to SDAA0 line because the timing for changing the SDAA0 line conflicts with the timing for writing the IICA0 register.

In addition to the above, communication is stopped if the IICE0 bit is cleared to 0 when communication has been aborted, so that the clock stretch state can be canceled.

If the I^2C bus has deadlocked due to noise, processing is saved from communication by setting bit 6 (LREL0) of the IICCTL00 register, so that the clock stretch state can be canceled.

Caution If a processing to cancel a clock stretch state is executed when WUP0 = 1, the clock stretch state will not be canceled.

17.5.8 Interrupt request (INTIICA0) generation timing and clock stretching control

The setting of bit 3 (WTIM0) of IICA control register 00 (IICCTL00) determines the timing by which INTIICA0 is generated and the corresponding clock stretching control, as shown in Table 17-2.

Table 17-2. INTIICA0 Generation Timing and Clock Stretching Control

WTIM0	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

Notes 1. The slave device's INTIICA0 signal and clock stretch period occurs at the falling edge of the ninth clock only when there is a match with the address set to the slave address register 0 (SVA0).

At this point, ACK is generated regardless of the value set to the IICCTL00 register's bit 2 (ACKE0). For a slave device that has received an extension code, INTIICA0 occurs at the falling edge of the eighth clock.

However, if the address does not match after restart, INTIICA0 is generated at the falling edge of the 9th clock, but clock stretching does not occur.

2. If the received address does not match the contents of the slave address register 0 (SVA0) and extension code is not received, neither INTIICA0 nor a clock stretching occurs.

Remark The numbers in the table indicate the number of the serial clock's clock signals. Interrupt requests and clock stretching control are both synchronized with the falling edge of these clock signals.

(1) During address transmission/reception

- Slave device operation: Interrupt and clock stretching timing are determined depending on the conditions described in Notes 1 and 2 above, regardless of the WTIM0 bit.
- Master device operation: Interrupt and clock stretching timing occur at the falling edge of the ninth clock regardless of the WTIM0 bit.

(2) During data reception

- Master/slave device operation: Interrupt and clock stretching timing are determined according to the WTIM0 bit.

(3) During data transmission

- Master/slave device operation: Interrupt and clock stretching timing are determined according to the WTIM0 bit.

(4) Clock stretching cancellation method

The four clock stretching cancellation methods are as follows.

- Writing data to the IICA shift register 0 (IICA0)
- Setting bit 5 (WREL0) of IICA control register 00 (IICCTL00) (canceling clock stretching)
- Setting bit 1 (STT0) of IICCTL00 register (generating start condition)Note
- Setting bit 0 (SPT0) of IICCTL00 register (generating stop condition)Note

Note Master only.

When an 8-clock clock stretching has been selected (WTIM0 = 0), the presence/absence of ACK generation must be determined prior to clock stretching cancellation.

(5) Stop condition detection

INTIICA0 is generated when a stop condition is detected (only when SPIE0 = 1).

17.5.9 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 (INTIICA0) occurs when the address set to the slave address register 0 (SVA0) matches the slave address sent by the master device, or when an extension code has been received.

17.5.10 Error detection

In I^2C bus mode, the status of the serial data bus (SDAA0) during data transmission is captured by the IICA shift register 0 (IICA0) 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.

17.5.11 Extension code

(1) When the higher 4 bits of the receive address are either “0000” or “1111”, the extension code reception flag (EXC0) is set to 1 for extension code reception and an interrupt request (INTIICA0) is issued at the falling edge of the eighth clock. The local address stored in the slave address register 0 (SVA0) is not affected.

(2) The settings below are specified if 11110xx0 is transferred from the master by using a 10-bit address transfer when the SVA0 register is set to 11110xx0. Note that INTIICA0 occurs at the falling edge of the eighth clock.

- Higher four bits of data match: EXC0 = 1
- Seven bits of data match: COI0 = 1

Remark EXC0: Bit 5 of IICA status register 0 (IICS0)
COI0: Bit 4 of IICA status register 0 (IICS0)

(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 while a slave device is operating, 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 bit 6 (LREL0) of IICA control register 00 (IICCTL00) to 1 to set the standby mode for the next communication operation.

Table 17-3. Bit Definitions of Major Extension Codes

Slave Address	R/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 See the I^2C bus specifications issued by NXP Semiconductors for details of extension codes other than those described above.

17.5.12 Arbitration

When several master devices simultaneously generate a start condition (when the STT0 bit is set to 1 before the STD0 bit 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, an arbitration loss flag (ALD0) in the IICA status register 0 (IICS0) is set (1) via the timing by which the arbitration loss occurred, and the SCLA0 and SDAA0 lines are both set to high impedance, which releases the bus.

The arbitration loss is detected based on the timing of the next interrupt request (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALD0 = 1 setting that has been made by software.

For details of interrupt request timing, see **17.5.8 Interrupt request (INTIICA0) generation timing and clock stretching control**.

Remark STD0: Bit 1 of IICA status register 0 (IICS0)
STT0: Bit 1 of IICA control register 00 (IICCTL00)

Figure 17-22. Arbitration Timing Example

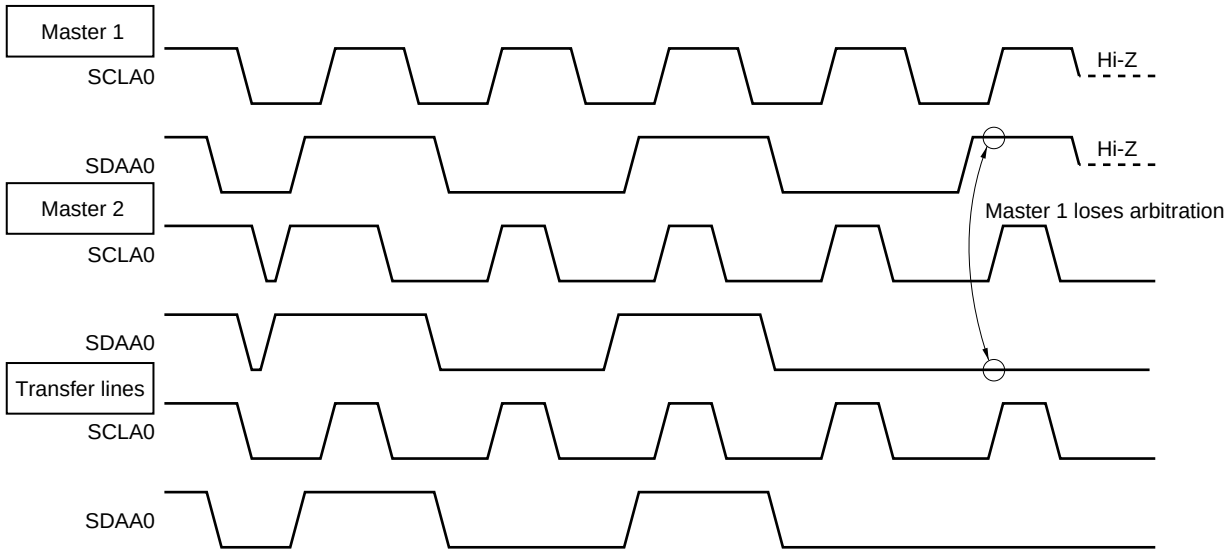


Table 17-4. Status During Arbitration and Interrupt Request Generation Timing

Status 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 SPIE0 = 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 SPIE0 = 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 SCLA0 is at low level while attempting to generate a restart condition	

Notes 1. When the WTIM0 bit (bit 3 of IICA control register 00 (IICCTL00)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIM0 = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock.

2. When there is a chance that arbitration will occur, set SPIE0 = 1 for master device operation.

Remark SPIE0: Bit 4 of IICA control register 00 (IICCTL00)

17.5.13 Wakeup function

The I^2C bus slave function is a function that generates an interrupt request signal (INTIICA0) when a local address and extension code have been received.

This function makes processing more efficient by preventing unnecessary INTIICA0 signal from occurring when addresses do not match.

When a start condition is detected, wakeup standby mode is set. This wakeup standby mode is in effect while addresses are transmitted due to the possibility that an arbitration loss may change the master device (which has generated a start condition) to a slave device.

To use the wakeup function in the STOP mode, set the WUP0 bit to 1. Addresses can be received regardless of the operation clock. An interrupt request signal (INTIICA0) is also generated when a local address and extension code have been received. Operation returns to normal operation by using an instruction to clear (0) the WUP0 bit after this interrupt has been generated.

Figure 17-23 shows the flow for setting WUP0 = 1 and Figure 17-24 shows the flow for setting WUP0 = 0 upon an address match.

Figure 17-23. Flow When Setting WUP0 = 1

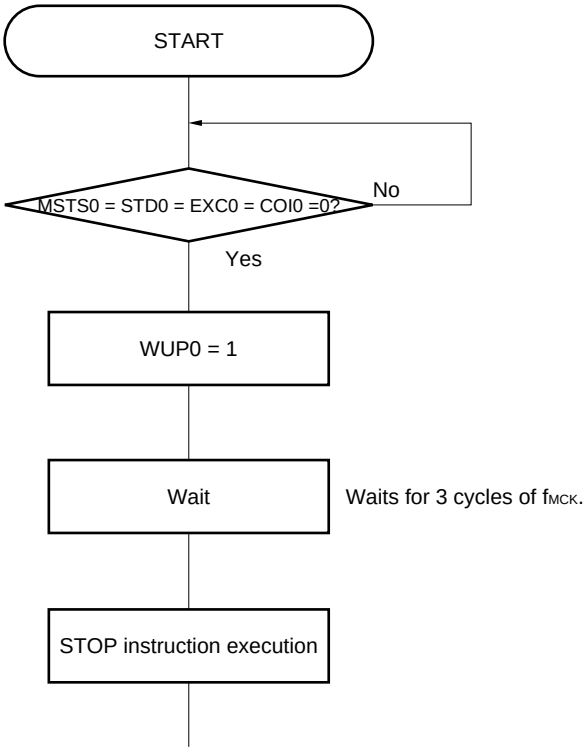
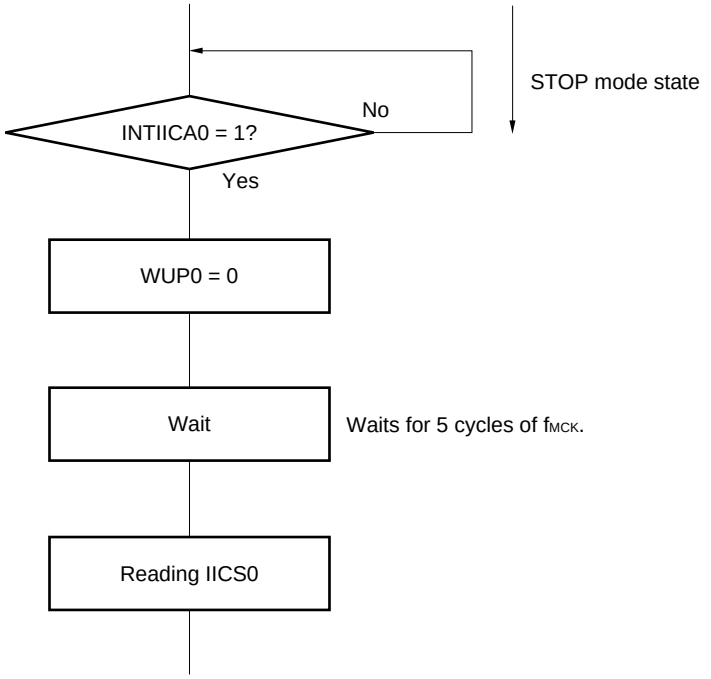


Figure 17-24. Flow When Setting WUP0 = 0 upon Address Match (Including Extension Code Reception)

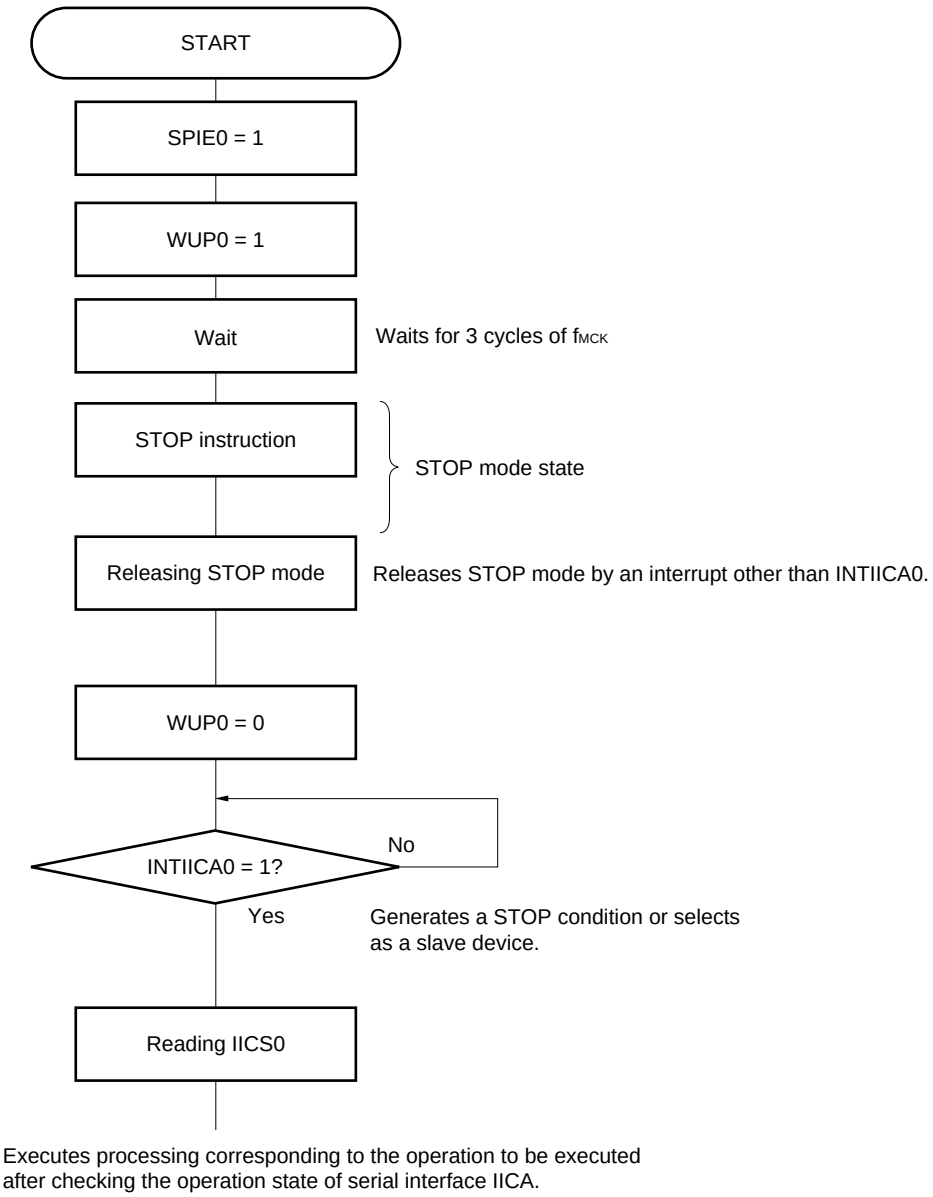


Executes processing corresponding to the operation to be executed
after checking the operation state of serial interface IICA.

Use the following flows to perform the processing to release the STOP mode other than by an interrupt request (INTIICA0) generated from serial interface IICA.

- When operating next IIC communication as master: Flow shown in Figure 17-25
- When operating next IIC communication as slave:
  - When restored by INTIICA0 interrupt: Same as the flow in Figure 17-24
  - When restored by other than INTIICA0 interrupt: Wait for INTIICA0 interrupt with WUP0 left set to 1

Figure 17-25. When Operating as Master Device after Releasing STOP Mode Other than by INTIICA0



17.5.14 Communication reservation

(1) When communication reservation function is enabled (bit 0 (IICRSV) of IICA flag register 0 (IICF0) = 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
- When an extension code is received and slave operation is disabled (ACK is not returned and the bus was released by setting bit 6 (LREL0) of IICA control register 00 (IICCTL00) to 1 and saving communication).

If bit 1 (STT0) of the IICCTL00 register is set to 1 while the bus is not used (after a stop condition is detected), a start condition is automatically generated and wait state is set.
If an address is written to the IICA shift register 0 (IICA0) after bit 4 (SPIE0) of the IICCTL00 register was set to 1, and it was detected by generation of an interrupt request signal (INTIICA0) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to the IICA0 register before the stop condition is detected is invalid.
When the STT0 bit has been set to 1, the operation mode (as start condition or as communication reservation) is determined according to the bus status.

- If the bus has been released ..... a start condition is generated
- If the bus has not been released (standby mode) ..... communication reservation

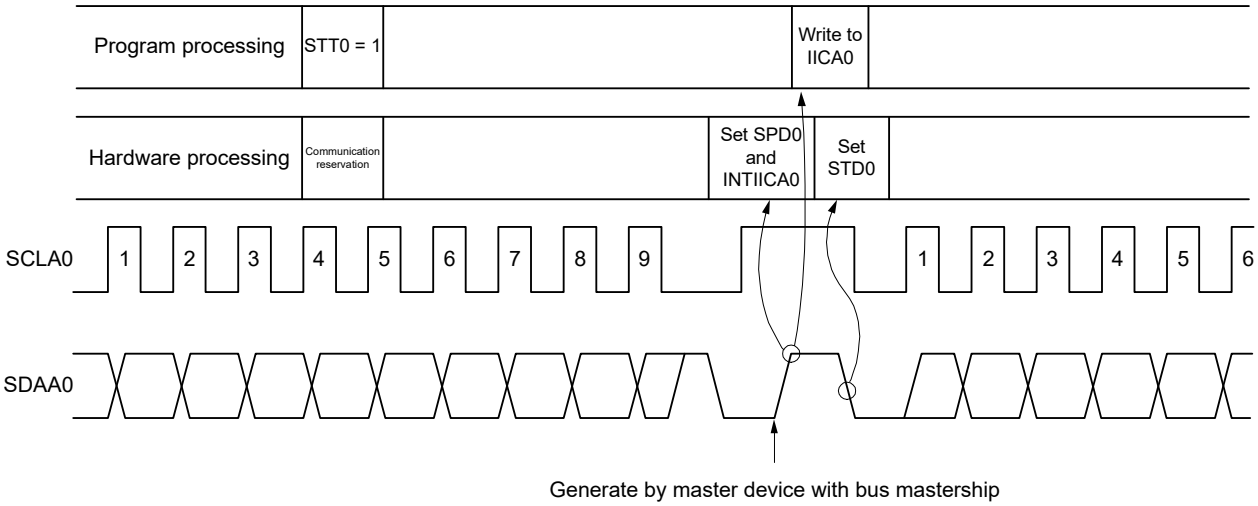
Check whether the communication reservation operates or not by using the MSTS0 bit (bit 7 of the IICA status register 0 (IICS0)) after the STT0 bit is set to 1 and the wait time elapses.
Use software to secure the wait time calculated by the following expression.

Wait time from setting STT0 = 1 to checking the MSTS0 flag: (IICWL0 setting value + IICWH0 setting value + 4) + $t_F \times 2 \times f_{MCK}$ [clocks]

Remark IICWL0: IICA low-level width setting register 0
IICWH0: IICA high-level width setting register 0
t_F: SDAA0 and SCLA0 signal falling times
f_{MCK}: IICA operation clock frequency

Figure 17-26 shows the communication reservation timing.

Figure 17-26. Communication Reservation Timing



Remark IICA0: IICA shift register 0
STT0: Bit 1 of IICA control register 00 (IICCTL00)
STD0: Bit 1 of IICA status register 0 (IICS0)
SPD0: Bit 0 of IICA status register 0 (IICS0)

Communication reservations are accepted via the timing shown in Figure 17-27. After bit 1 (STD0) of the IICA status register 0 (IICS0) is set to 1, a communication reservation can be made by setting bit 1 (STT0) of IICA control register 00 (IICCTL00) to 1 before a stop condition is detected.

Figure 17-27. Timing for Accepting Communication Reservations

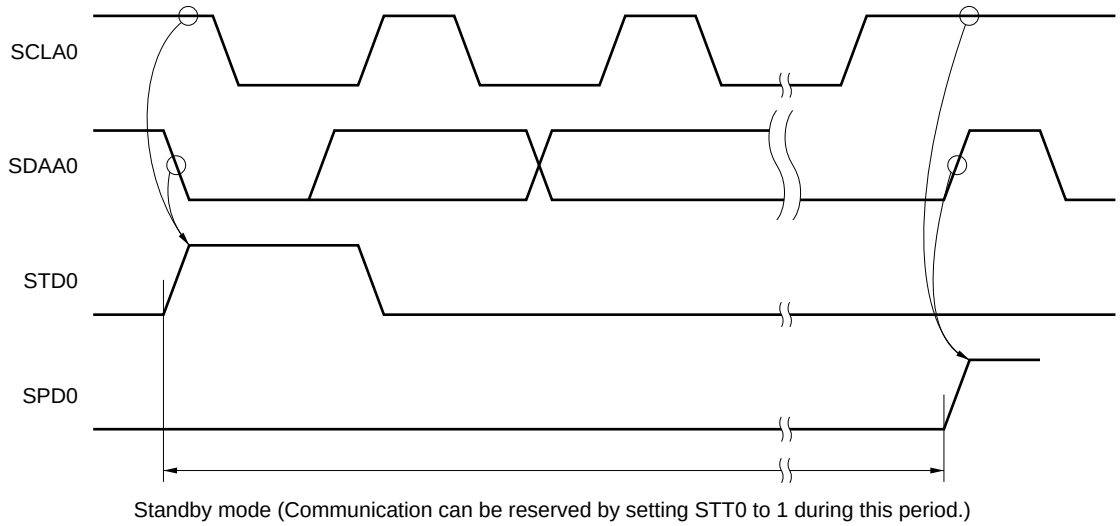
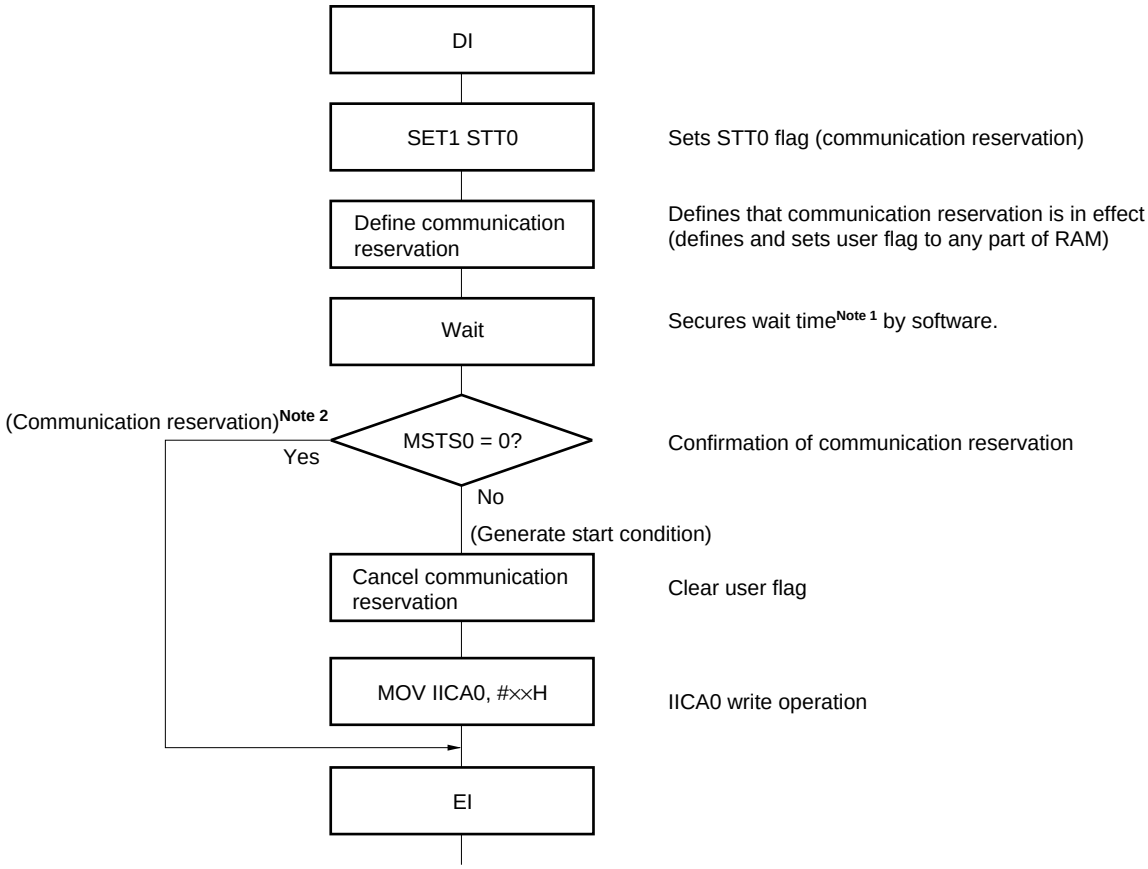


Figure 17-28 shows the communication reservation protocol.

Figure 17-28. Communication Reservation Protocol



Notes 1. The wait time is calculated as follows.
(IICWL0 setting value + IICWH0 setting value + 4) + $t_F \times 2 \times f_{MCK}$ [clocks]

2. The communication reservation operation executes a write to the IICA shift register 0 (IICA0) when a stop condition interrupt request occurs.

Remark STT0: Bit 1 of IICA control register 00 (IICCTL00)
MSTS0: Bit 7 of IICA status register 0 (IICS0)
IICA0: IICA shift register 0
IICWL0: IICA low-level width setting register 0
IICWH0: IICA high-level width setting register 0
t_F: SDAA0 and SCLA0 signal falling times
f_{MCK}: IICA operation clock frequency

(2) When communication reservation function is disabled (bit 0 (IICRSV0) of IICA flag register 0 (IICF0) = 1)

When bit 1 (STT0) of IICA control register 00 (IICCTL00) 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 following two statuses are included in the status where bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled (ACK is not returned and the bus was released by setting bit 6 (LREL0) of the IICCTL00 register to 1 and saving communication)

To confirm whether the start condition was generated or request was rejected, check STCF0 (bit 7 of the IICF0 register). It takes up to 5 cycles of f_{MCK} until the STCF0 bit is set to 1 after setting STT0 = 1. Therefore, secure the time by software.

17.5.15 Cautions

(1) When STCEN = 0

Immediately after I^2C operation is enabled (IICE0 = 1), the bus communication status (IICBSY = 1) is recognized regardless of the actual bus status. 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 01 (IICCTL01).
<2> Set bit 7 (IICE0) of IICA control register 00 (IICCTL00) to 1.
<3> Set bit 0 (SPT0) of the IICCTL00 register to 1.

(2) When STCEN = 1

Immediately after I^2C operation is enabled (IICE0 = 1), the bus released status (IICBSY = 0) is recognized regardless of the actual bus status. To generate the first start condition (STT0 = 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 SDAA0 pin is low and the SCLA0 pin is high, the macro of I^2C recognizes that the SDAA0 pin has gone low (detects a start condition). If the value on the bus at this time can be recognized as an extension code, ACK is returned, but this interferes with other I^2C communications. To avoid this, start I^2C in the following sequence.

<1> Clear bit 4 (SPIE0) of the IICCTL00 register to 0 to disable generation of an interrupt request signal (INTIICA0) when the stop condition is detected.
<2> Set bit 7 (IICE0) of the IICCTL00 register to 1 to enable the operation of I^2C.
<3> Wait for detection of the start condition.
<4> Set bit 6 (LREL0) of the IICCTL00 register to 1 before ACK is returned (4 to 72 cycles of f_{MCK} after setting the IICE0 bit to 1), to forcibly disable detection.

(4) Setting the STT0 and SPT0 bits (bits 1 and 0 of the IICCTL00 register) again after they are set and before they are cleared to 0 is prohibited.

(5) When transmission is reserved, set the SPIE0 bit (bit 4 of the IICTL0 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 0 (IICA0) 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 SPIE0 bit to 1 when the MSTS0 bit (bit 7 of the IICA status register 0 (IICS0)) is detected by software.

17.5.16 Communication operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single master system

The flowchart when using the RL78/I1A 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 multimaster system

In the I^2C bus multimaster system, whether the bus is released or used cannot be judged by the I^2C bus specifications when the bus takes part in a communication. Here, when data and clock are at a high level for a certain period (1 frame), the RL78/I1A 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/I1A 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

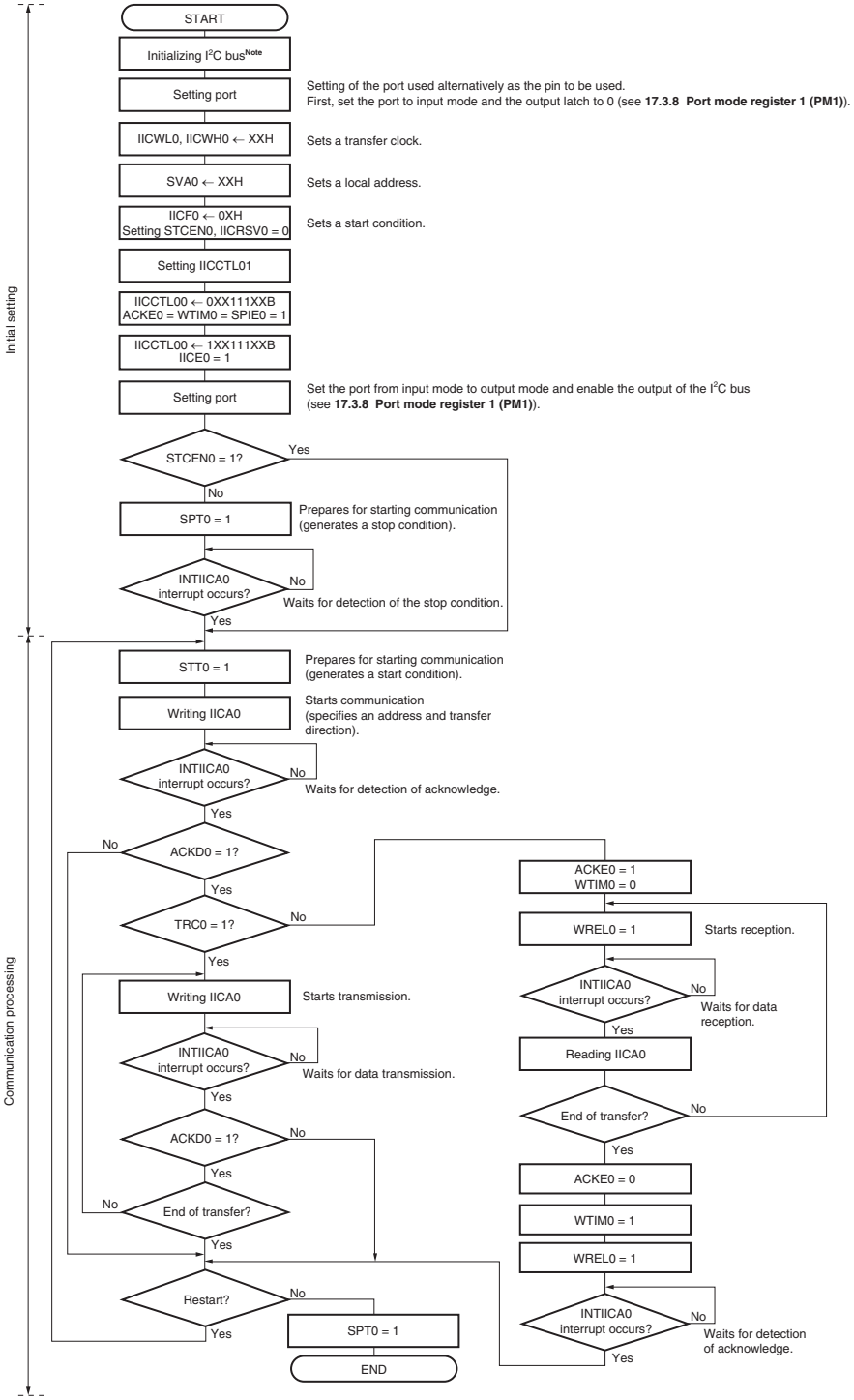
An example of when the RL78/I1A is used as the I^2C bus slave is shown below.

When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIICA0 interrupt occurrence (communication waiting). When an INTIICA0 interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.

By checking the flags, necessary communication processing is performed.

(1) Master operation in single-master system

Figure 17-29. Master Operation in Single-master System

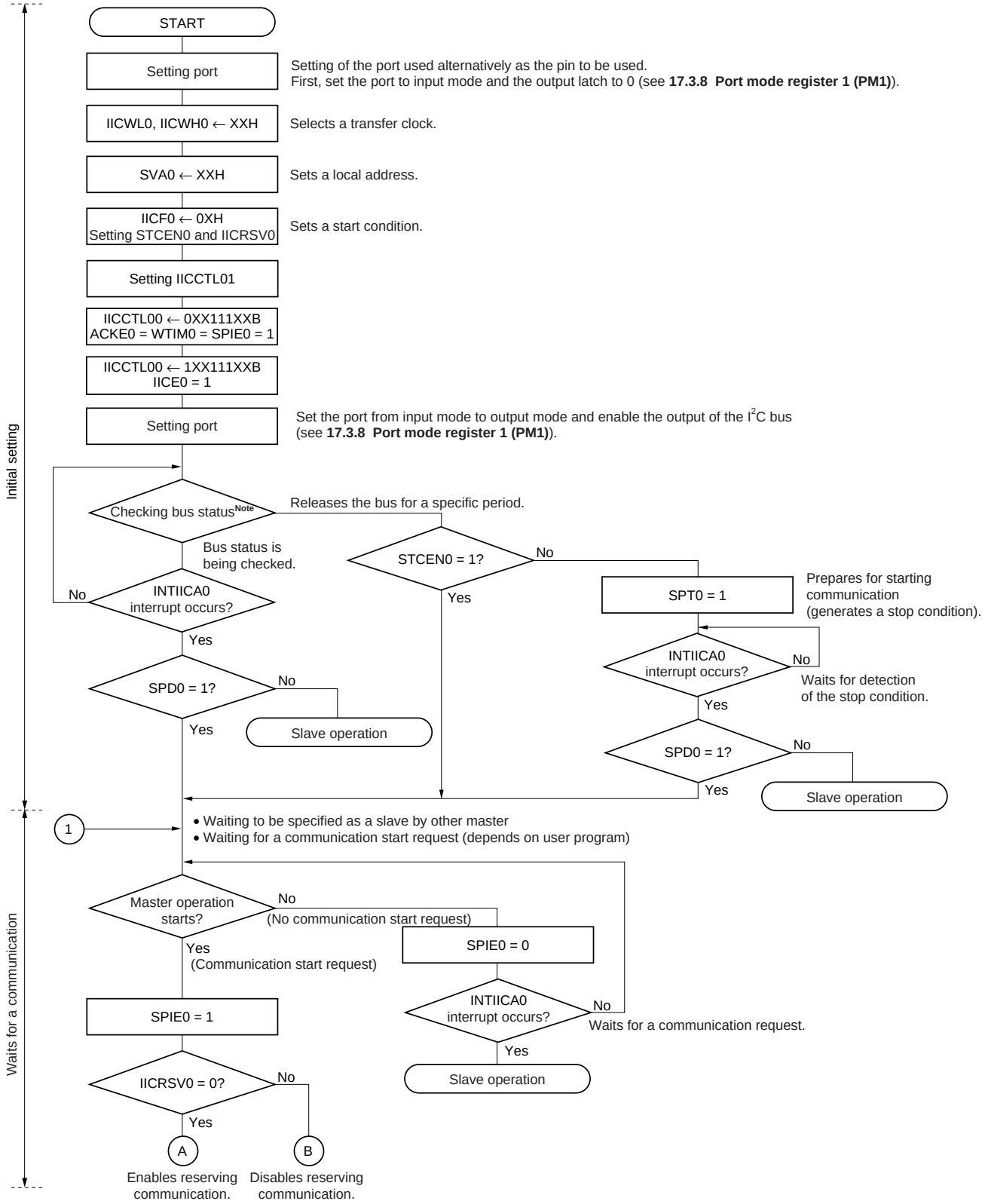


Note Release (SCLA0 and SDAA0 pins = high level) the I^2C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAA0 pin, for example, set the SCLA0 pin in the output port mode, and output a clock pulse from the output port until the SDAA0 pin is constantly at high level.

Remark Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

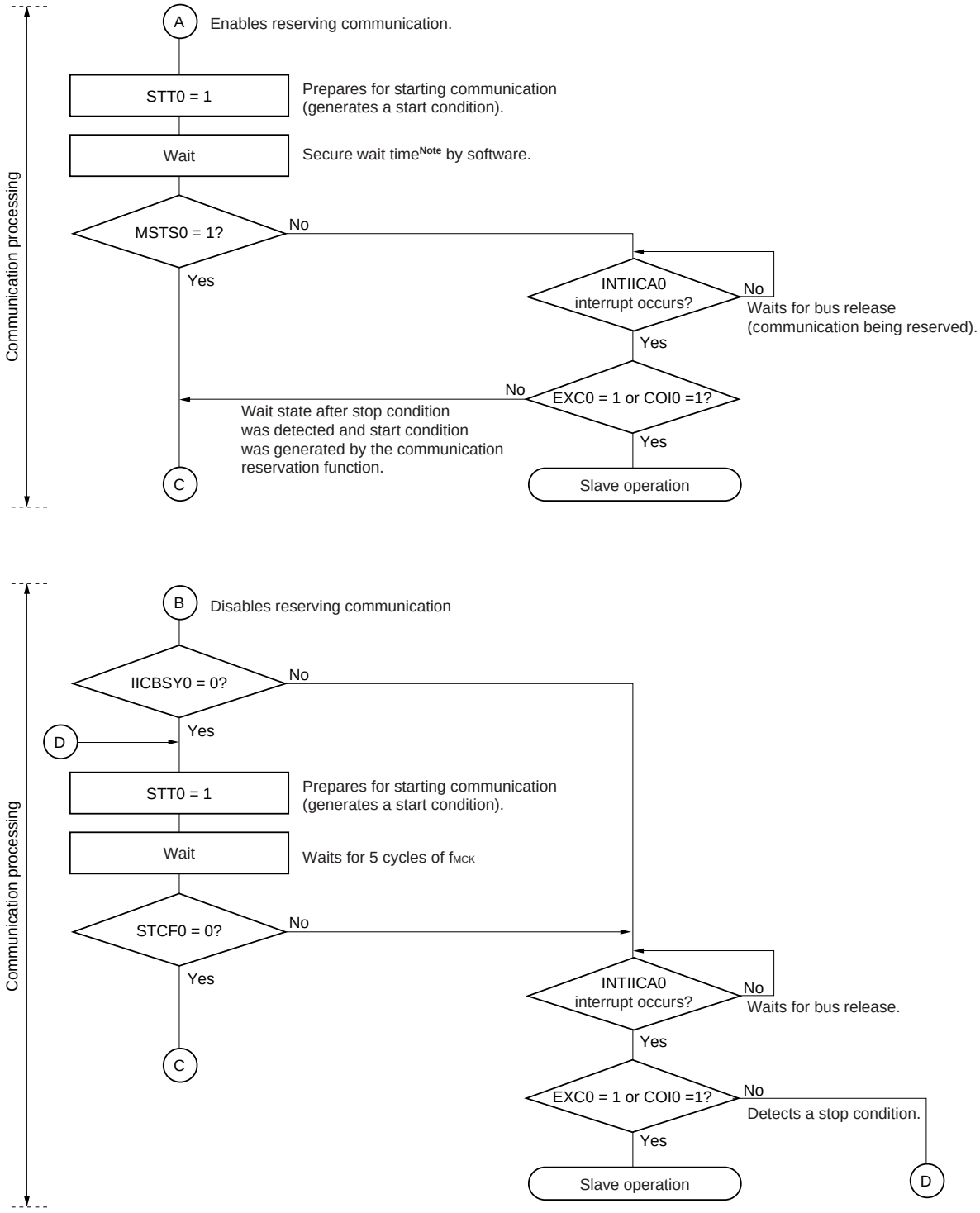
(2) Master operation in multi-master system

Figure 17-30. Master Operation in Multi-master System (1/3)



Note Confirm that the bus is released (CLD0 bit = 1, DAD0 bit = 1) for a specific period (for example, for a period of one frame). If the SDAA0 pin is constantly at low level, decide whether to release the I^2C bus (SCLA0 and SDAA0 pins = high level) in conformance with the specifications of the product that is communicating.

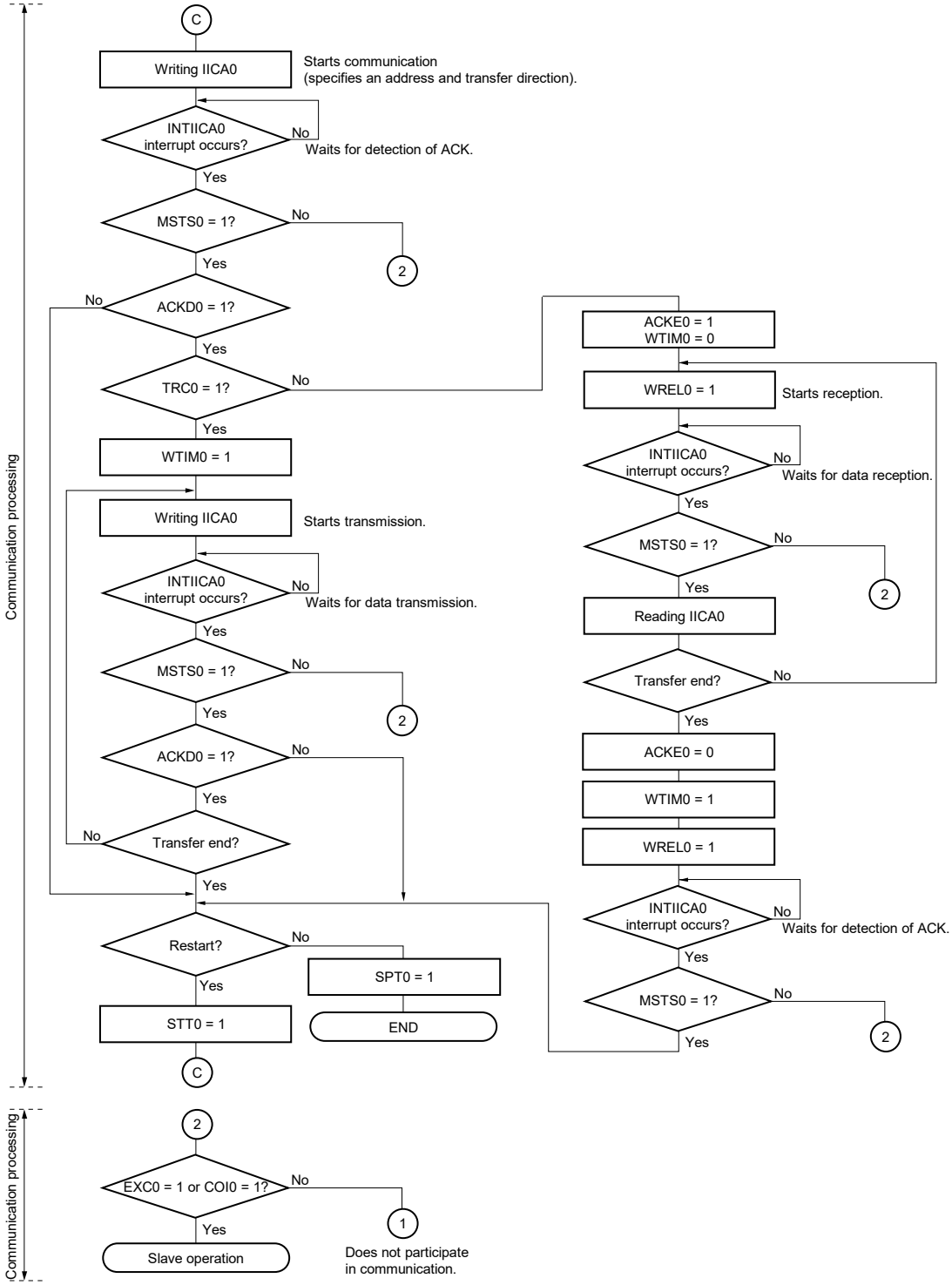
Figure 17-30. Master Operation in Multi-master System (2/3)



Note The wait time is calculated as follows.
(IICWL0 setting value + IICWH0 setting value + 4) + $t_F \times 2 \times f_{MCK}$ [clocks]

Remark IICWL0: IICA low-level width setting register 0
IICWH0: IICA high-level width setting register 0
t_F: SDAA0 and SCLA0 signal falling times
f_{MCK}: IICA operation clock frequency

Figure 17-30. Master Operation in Multi-master System (3/3)



Remarks 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

2. To use the device as a master in a multi-master system, read the MSTS0 bit each time interrupt INTIICA0 has occurred to check the arbitration result.

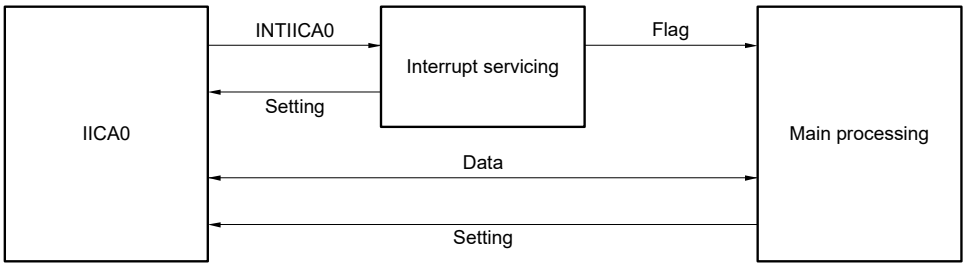
3. To use the device as a slave in a multi-master system, check the status by using the IICA status register 0 (IICS0) and IICA flag register 0 (IICF0) each time interrupt INTIICA0 has occurred, and determine the processing to be performed next.

(3) Slave operation

The processing procedure of the slave operation is as follows.

Basically, the slave operation is event-driven. Therefore, processing by the INTIICA0 interrupt (processing that must substantially change the operation status such as detection of a stop condition during communication) is necessary.

In the following explanation, it is assumed that the extension code is not supported for data communication. It is also assumed that the INTIICA0 interrupt servicing only performs status transition processing, and that actual data communication is performed by the main processing.



Therefore, data communication processing is performed by preparing the following three flags and passing them to the main processing instead of INTIICA0.

<1> Communication mode flag

This flag indicates the following two communication statuses.

- Clear mode: Status in which data communication is not performed
- Communication mode: Status in which data communication is performed (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 INTIICA0 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 TRC0 bit.

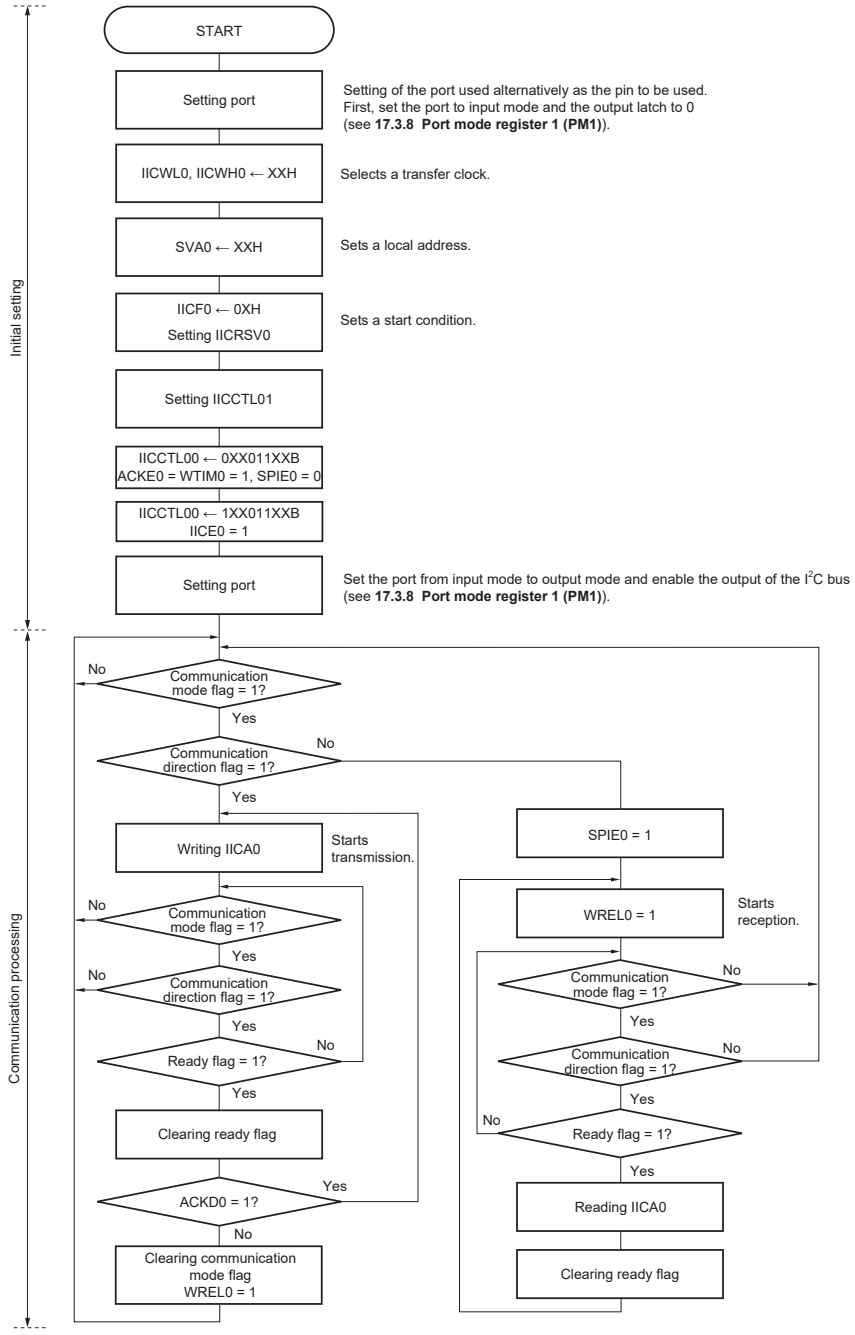
The main processing of the slave operation is explained next.

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 status 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 communication status occurs in this way.

Figure 17-31. Slave Operation Flowchart (1)



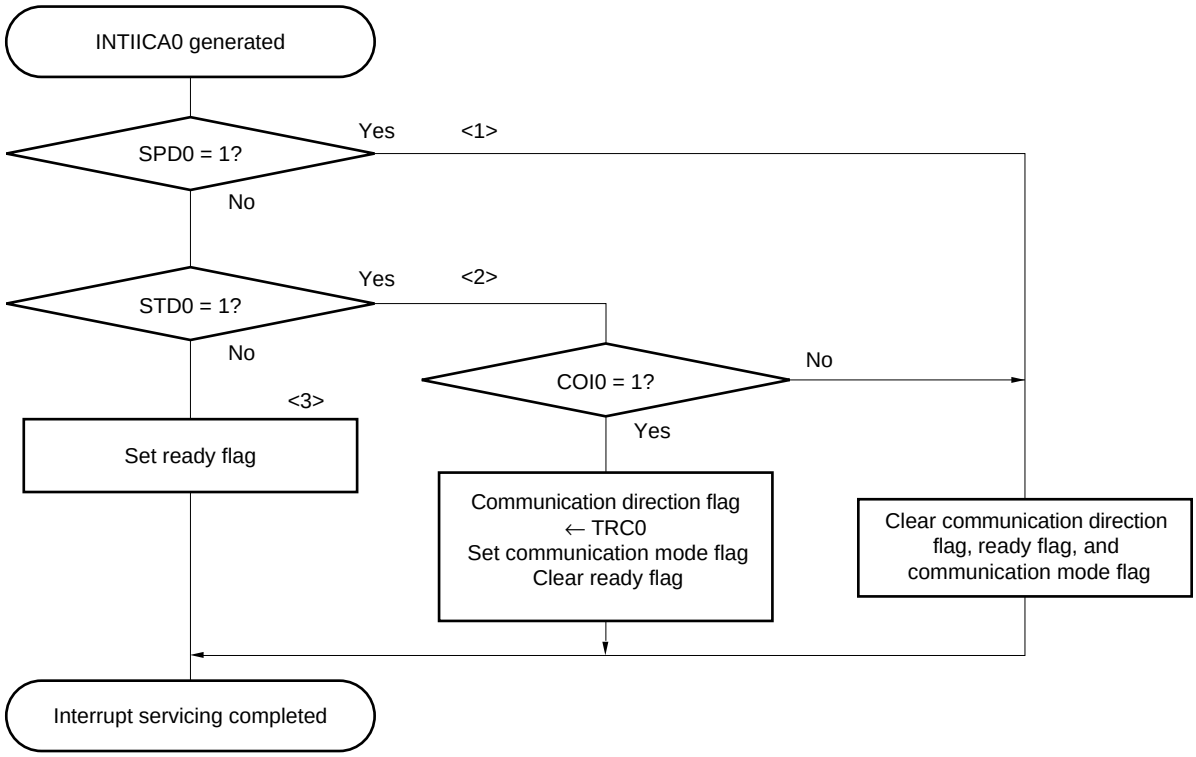
Remark Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.

An example of the processing procedure of the slave with the INTIICA0 interrupt is explained below (processing is performed assuming that no extension code is used). The INTIICA0 interrupt checks the status, 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 cancelled, 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 17-32 Slave Operation Flowchart (2)**.

Figure 17-32. Slave Operation Flowchart (2)



17.5.17 Timing of I^2C interrupt request (INTIICA0) occurrence

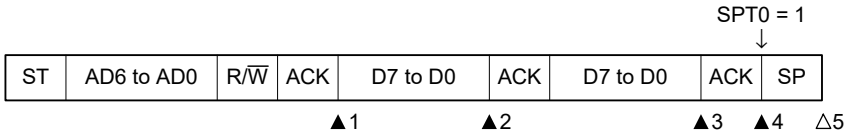
The timing of transmitting or receiving data and generation of interrupt request signal INTIICA0, and the value of the IICA status register 0 (IICS0) when the INTIICA0 signal is generated are shown below.

Remark ST: Start condition
AD6 to AD0: Address
R/$\overline{W}$: Transfer direction specification
ACK: Acknowledge
D7 to D0: Data
SP: Stop condition

(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (transmission/reception)

(i) When WTIM0 = 0

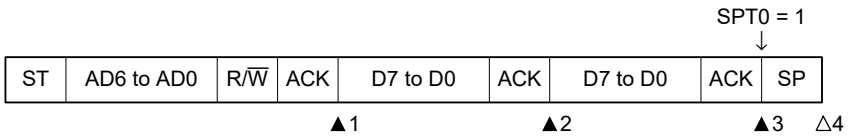


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B
▲3: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)[Note]
▲4: IICS0 = 1000××00B (Sets the SPT0 bit to 1)[Note]
△5: IICS0 = 00000001B

Note To generate a stop condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

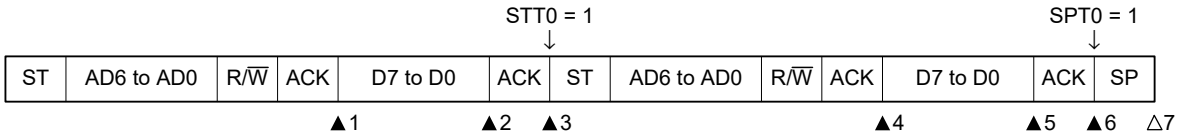


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B
▲3: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

(i) When WTIM0 = 0

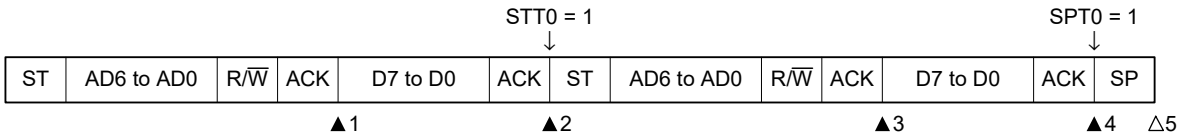


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)Note 1
▲3: IICS0 = 1000××00B (Clears the WTIM0 bit to 0Note 2, sets the STT0 bit to 1)
▲4: IICS0 = 1000×110B
▲5: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)Note 3
▲6: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△7: IICS0 = 00000001B

Notes 1. To generate a start condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.
2. Clear the WTIM0 bit to 0 to restore the original setting.
3. To generate a stop condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

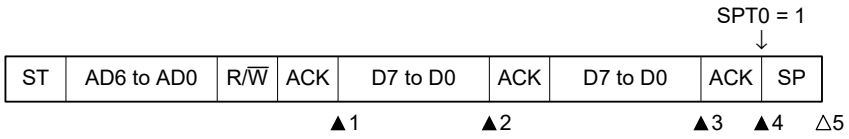


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000××00B (Sets the STT0 bit to 1)
▲3: IICS0 = 1000×110B
▲4: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

(i) When WTIM0 = 0

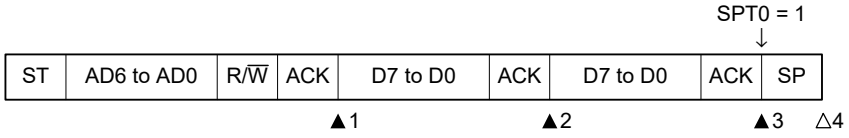


▲1: IICS0 = 1010×110B
▲2: IICS0 = 1010×000B
▲3: IICS0 = 1010×000B (Sets the WTIM0 bit to 1)[Note]
▲4: IICS0 = 1010××00B (Sets the SPT0 bit to 1)
△5: IICS0 = 00000001B

Note To generate a stop condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



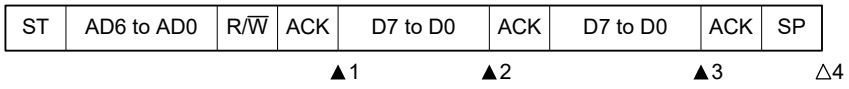
▲1: IICS0 = 1010×110B
▲2: IICS0 = 1010×100B
▲3: IICS0 = 1010××00B (Sets the SPT0 bit to 1)
△4: IICS0 = 00001001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(2) Slave device operation (slave address data reception)

(a) Start ~ Address ~ Data ~ Data ~ Stop

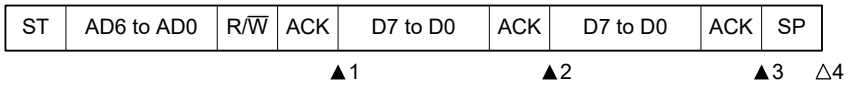
(i) When WTIM0 = 0



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

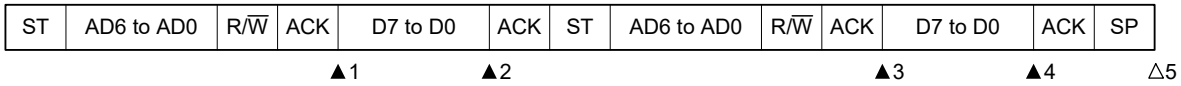


▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×100B
▲3: IICS0 = 0001××00B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

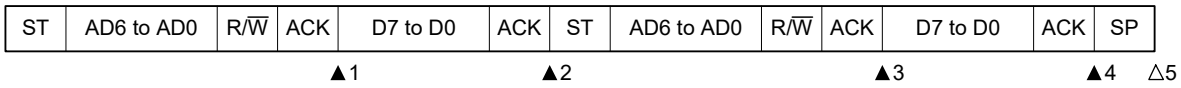
(i) When WTIM0 = 0 (after restart, matches with SVA0)



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, matches with SVA0)

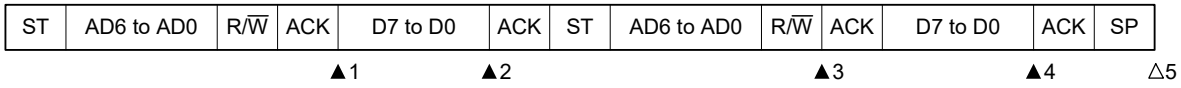


▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001××00B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

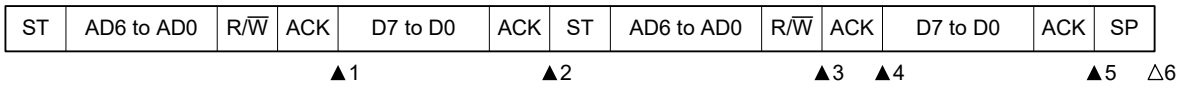
(i) When WTIM0 = 0 (after restart, does not match address (= extension code))



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, does not match address (= extension code))

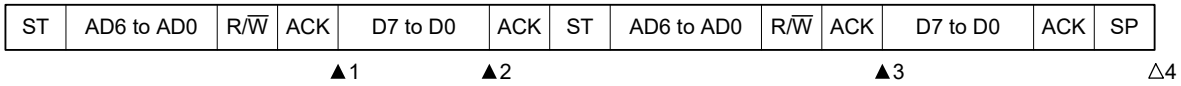


▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×110B
▲5: IICS0 = 0010××00B
△6: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, does not match address (= not extension code))



▲1 ▲2 ▲3 △4

▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 00000×10B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, does not match address (= not extension code))

▲1 ▲2 ▲3 △4

▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 00000×10B
△4: IICS0 = 00000001B

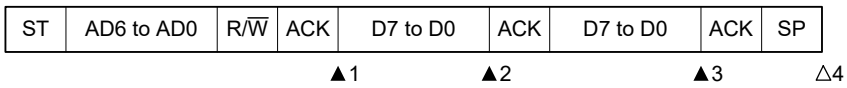
Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(3) Slave device operation (when receiving extension code)

The device is always participating in communication when it receives an extension code.

(a) Start ~ Code ~ Data ~ Data ~ Stop

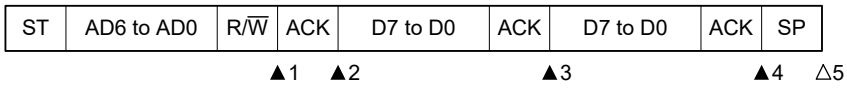
(i) When WTIM0 = 0



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(ii) When WTIM0 = 1

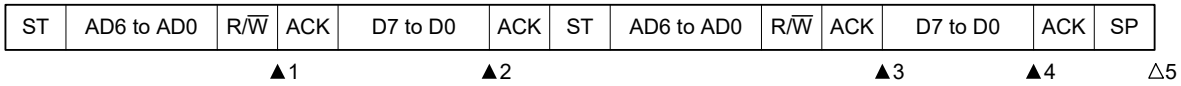


▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010×100B
▲4: IICS0 = 0010××00B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

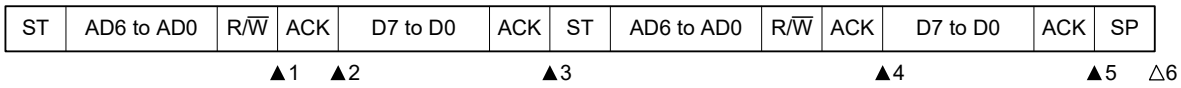
(i) When WTIM0 = 0 (after restart, matches SVA0)



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, matches SVA0)

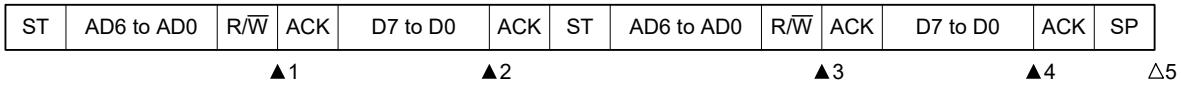


▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 0001×110B
▲5: IICS0 = 0001××00B
△6: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

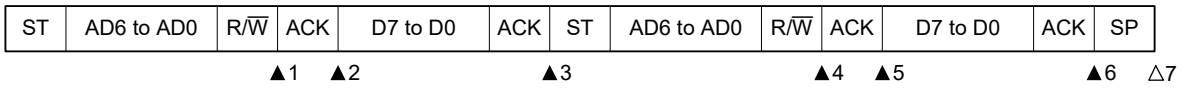
(i) When WTIM0 = 0 (after restart, extension code reception)



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, extension code reception)

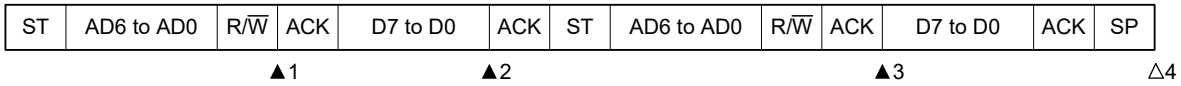


▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 0010×010B
▲5: IICS0 = 0010×110B
▲6: IICS0 = 0010××00B
△7: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

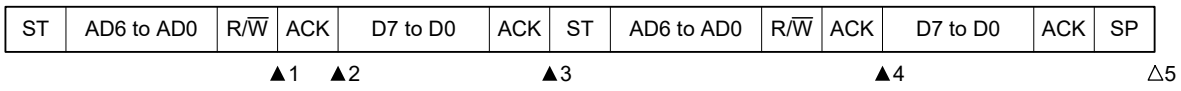
(i) When WTIM0 = 0 (after restart, does not match address (= not extension code))



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 00000×10B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, does not match address (= not extension code))

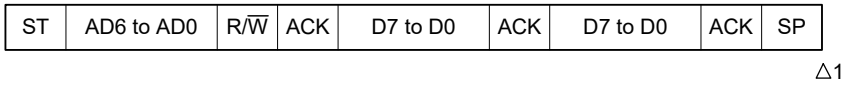


▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 00000×10B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop



△1: IICS0 = 00000001B

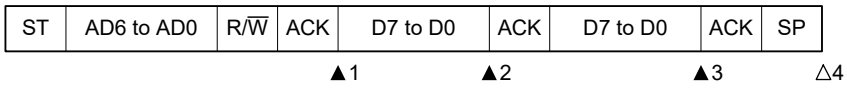
Remark △: Generated only when SPIE0 = 1

(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 MSTS0 bit each time interrupt request signal INTIICA0 has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data

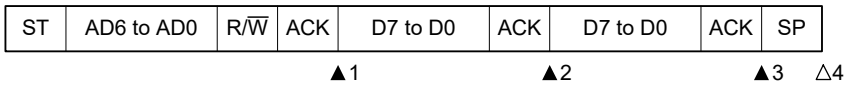
(i) When WTIM0 = 0



▲1: IICS0 = 0101×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(ii) When WTIM0 = 1

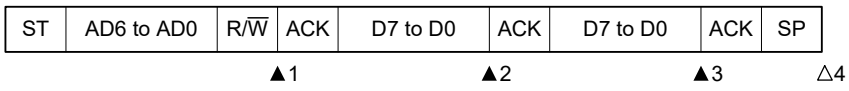


▲1: IICS0 = 0101×110B
▲2: IICS0 = 0001×100B
▲3: IICS0 = 0001××00B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(b) When arbitration loss occurs during transmission of extension code

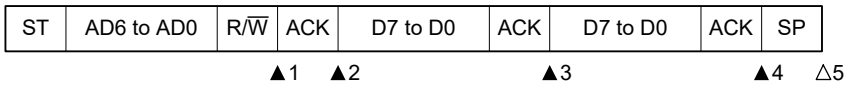
(i) When WTIM0 = 0



▲1: IICS0 = 0110×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(ii) When WTIM0 = 1



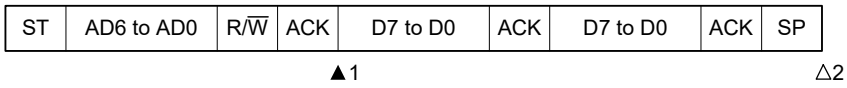
▲1: IICS0 = 0110×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010×100B
▲4: IICS0 = 0010××00B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(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 MSTS0 bit each time interrupt request signal INTIICA0 has occurred to check the arbitration result.

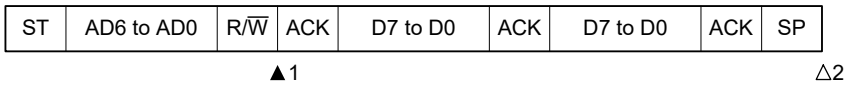
(a) When arbitration loss occurs during transmission of slave address data (when WTIM0 = 1)



▲1: IICS0 = 01000110B
△2: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(b) When arbitration loss occurs during transmission of extension code

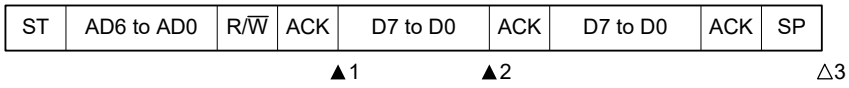


▲1: IICS0 = 0110×010B
Sets LREL0 = 1 by software
△2: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) When arbitration loss occurs during transmission of data

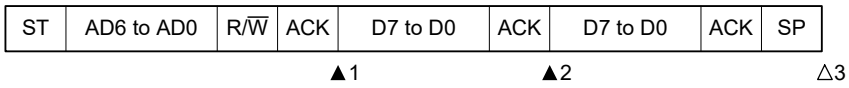
(i) When WTIM0 = 0



▲1: IICS0 = 10001110B
▲2: IICS0 = 01000000B
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(ii) When WTIM0 = 1

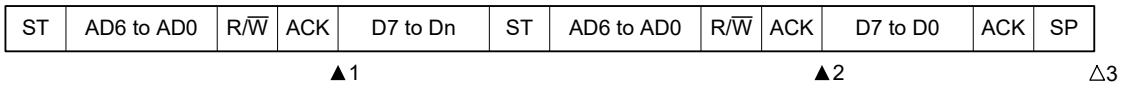


▲1: IICS0 = 10001110B
▲2: IICS0 = 01000100B
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(d) When loss occurs due to restart condition during data transfer

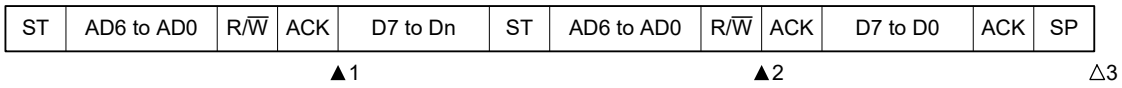
(i) Not extension code (Example: unmatches with SVA0)



▲1: IICS0 = 1000×110B
▲2: IICS0 = 01000110B
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care
n = 6 to 0

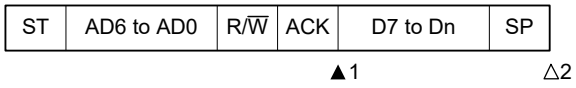
(ii) Extension code



▲1: IICS0 = 1000×110B
▲2: IICS0 = 01100010B
Sets LREL0 = 1 by software
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care
n = 6 to 0

(e) When loss occurs due to stop condition during data transfer

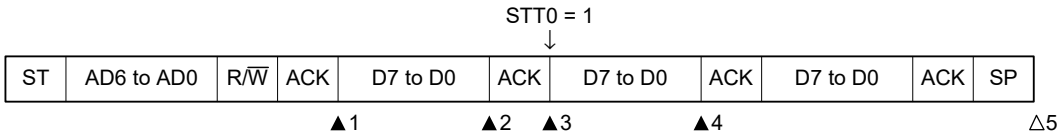


▲1: IICS0 = 10000110B
△2: IICS0 = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care
n = 6 to 0

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

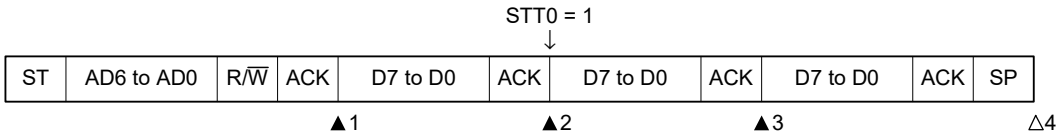
(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)
▲3: IICS0 = 1000×100B (Clears the WTIM0 bit to 0)
▲4: IICS0 = 01000000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

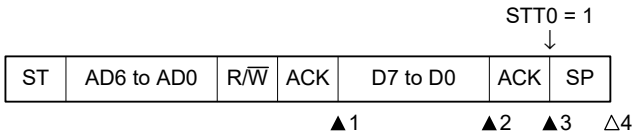


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B (Sets the STT0 bit to 1)
▲3: IICS0 = 01000100B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

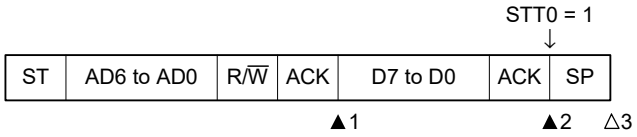
(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)
▲3: IICS0 = 1000××00B (Sets the STT0 bit to 1)
△4: IICS0 = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

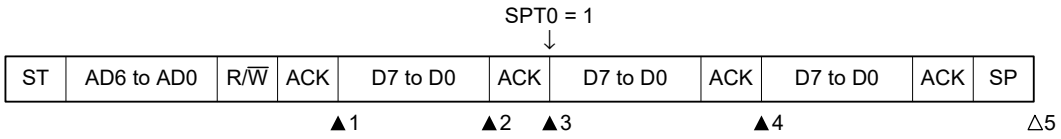


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000××00B (Sets the STT0 bit to 1)
△3: IICS0 = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

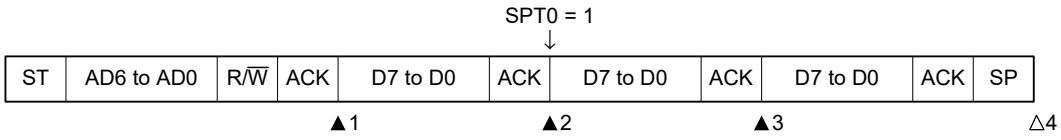
(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)
▲3: IICS0 = 1000×100B (Clears the WTIM0 bit to 0)
▲4: IICS0 = 01000100B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(ii) When WTIM0 = 1



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B (Sets the SPT0 bit to 1)
▲3: IICS0 = 01000100B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

17.6 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 TRC0 bit (bit 3 of the IICA status register 0 (IICS0)), which specifies the data transfer direction, and then starts serial communication with the slave device.

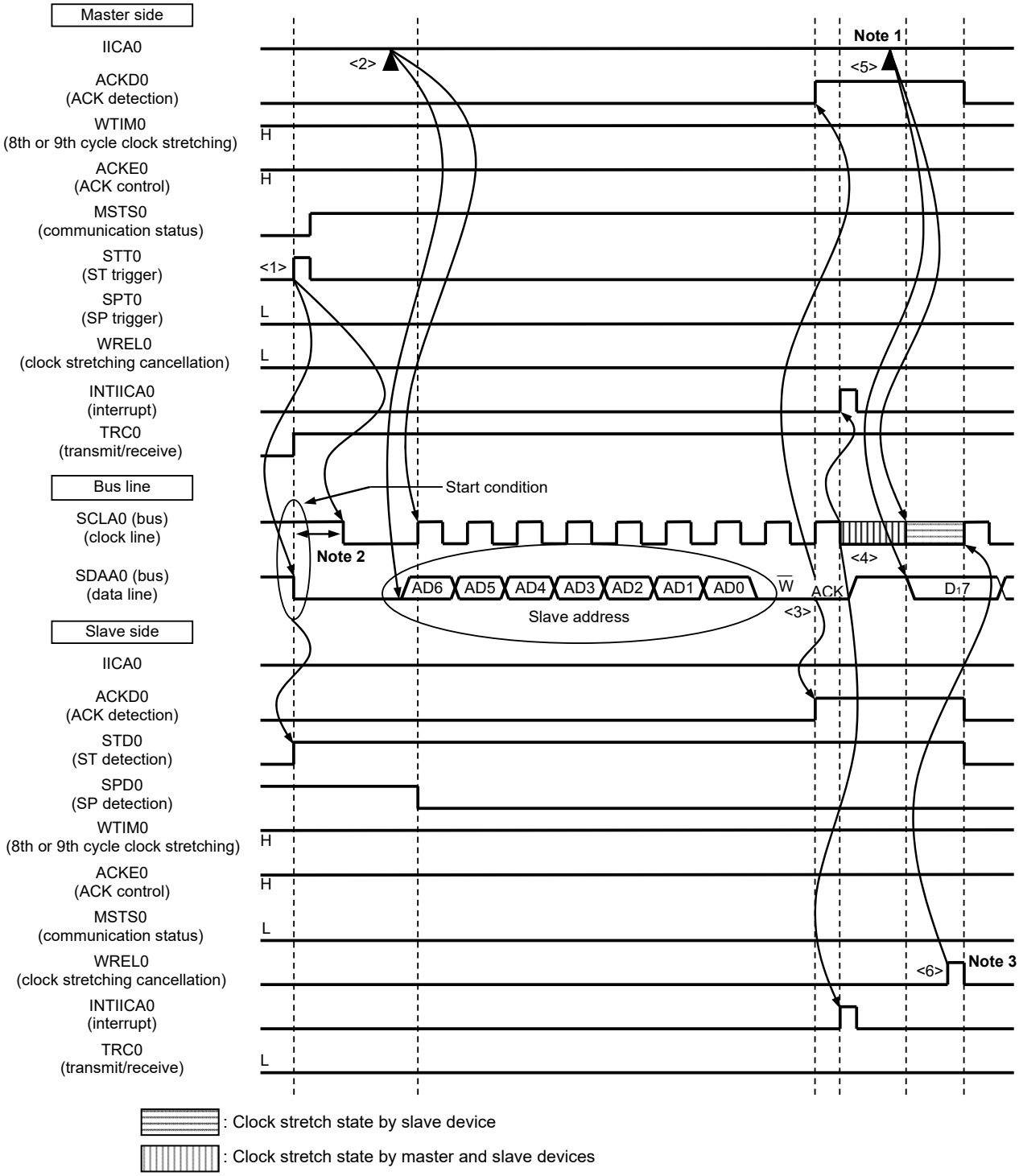
Figures 17-33 and 17-34 show timing charts of the data communication.

The IICA shift register 0 (IICA0)'s shift operation is synchronized with the falling edge of the serial clock (SCLA0). The transmit data is transferred to the SO latch and is output (MSB first) via the SDAA0 pin.

Data input via the SDAA0 pin is captured into IICA0 at the rising edge of SCLA0.

Figure 17-33. Example of Master to Slave Communication
(When 9th Cycle Clock Stretching Is Selected for Master, 9th Cycle Clock Stretching Is Selected for Slave) (1/4)

(1) Start condition ~ address ~ data



Notes 1. Write data to IICA0, not setting the WREL0 bit, in order to cancel a clock stretch state during master transmission.

2. Make sure that the time between the fall of the SDAA0 pin signal and the fall of the SCLA0 pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. To cancel slave clock stretch, write “FFH” to IICA0 or set the WREL0 bit.

The meanings of <1> to <6> in (1) Start condition ~ address ~ data in Figure 17-33 are explained below.

<1> The start condition trigger is set by the master device (STT0 = 1) and a start condition (SDAA0 = 0 and SCLA0 = 1) is generated once the bus data line goes low (SDAA0 = 0). When the start condition is subsequently detected, the master device enters the master device communication status (MSTS0 = 1). The master device is ready to communicate once the bus clock line goes low (SCLA0 = 0) after the hold time has elapsed.

<2> The master device writes the address + W (transmission) to the IICA shift register 0 (IICA0) and transmits the slave address.

<3> If the address received matches the address 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 (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA0: end of address transmission) at the falling edge of the 9th clock, and the slave device whose address matched the transmitted slave address also issues an interrupt (INTIICA0: address match). The master device and slave device also set a clock stretch status (SCLA0 = 0)**Note** when the addresses match.

<5> The master device writes the data to transmit to the IICA0 register and releases the clock stretch status that it set by the master device.

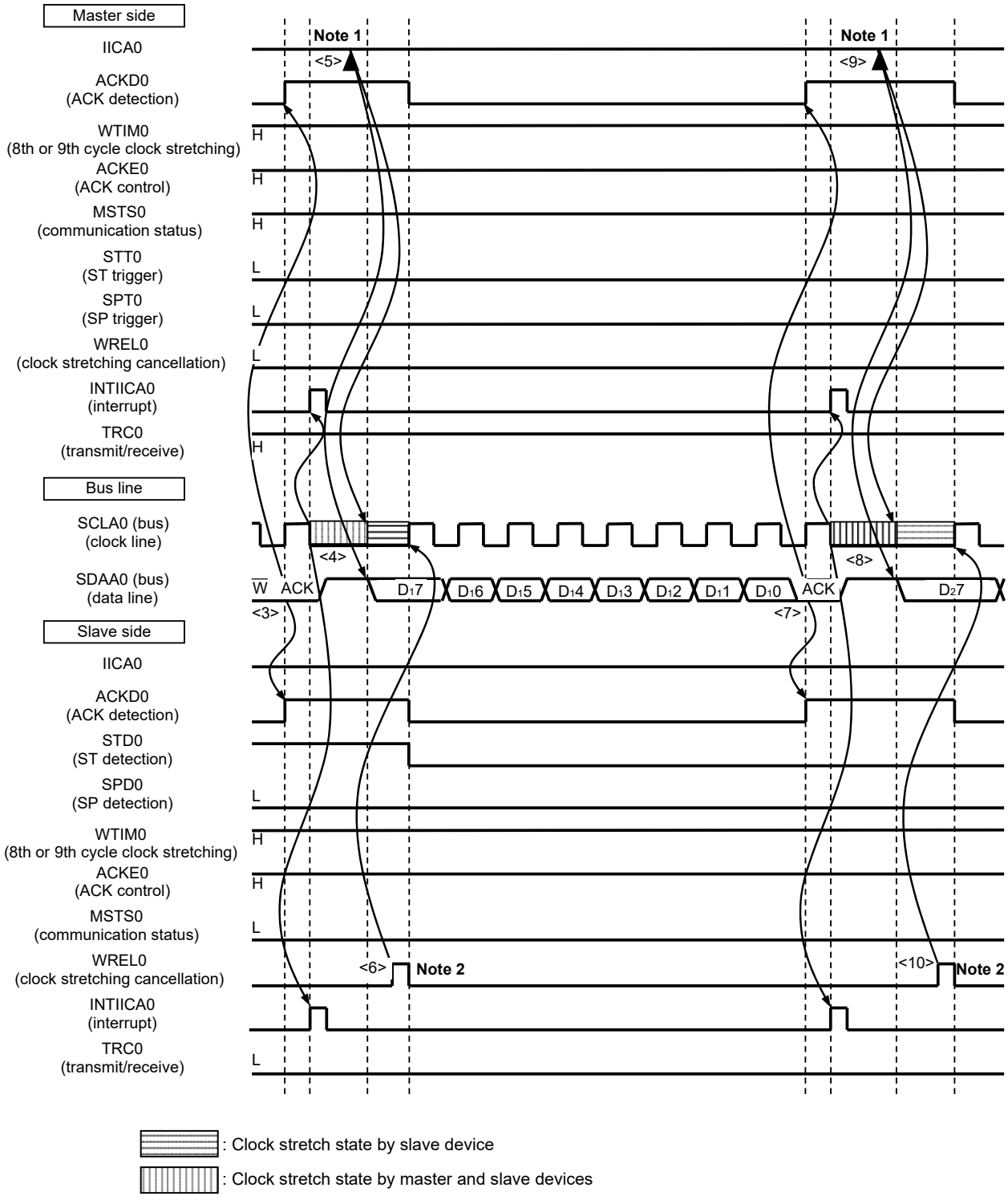
<6> If the slave device releases the clock stretch status (WREL0 = 1), the master device starts transferring data to the slave device.

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: SDAA0 = 1). The slave device also does not issue the INTIICA0 interrupt (address match) and does not set a clock stretch status. The master device, however, issues the INTIICA0 interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <15> in Figure 17-33 represent the entire procedure for communicating data using the I^2C bus. Figure 17-33 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 17-33 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 17-33 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 17-33. Example of Master to Slave Communication
(When 9th Cycle Clock Stretching Is Selected for Master, 9th Cycle Clock Stretching Is Selected for Slave) (2/4)

(2) Address ~ data ~ data



Notes 1. Write data to IICA0, not setting the WREL0 bit, in order to cancel a clock stretch state during master transmission.

2. To cancel slave clock stretch, write “FFH” to IICA0 or set the WREL0 bit.

The meanings of <3> to <10> in (2) Address ~ data ~ data in Figure 17-33 are explained below.

<3> If the address received matches the address 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 (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA0: end of address transmission) at the falling edge of the 9th clock, and the slave device whose address matched the transmitted slave address also issues an interrupt (INTIICA0: address match). The master device and slave device also set a clock stretch status (SCLA0 = 0)[Note] when the addresses match.

<5> The master device writes the data to transmit to the IICA shift register 0 (IICA0) and releases the clock stretch status that it set by the master device.

<6> If the slave device releases the clock stretch status (WREL0 = 1), the master device starts transferring data to the slave device.

<7> When data transfer is complete, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a clock stretch status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

<9> The master device writes the data to transmit to the IICA0 register and releases the clock stretch status that it set by the master device.

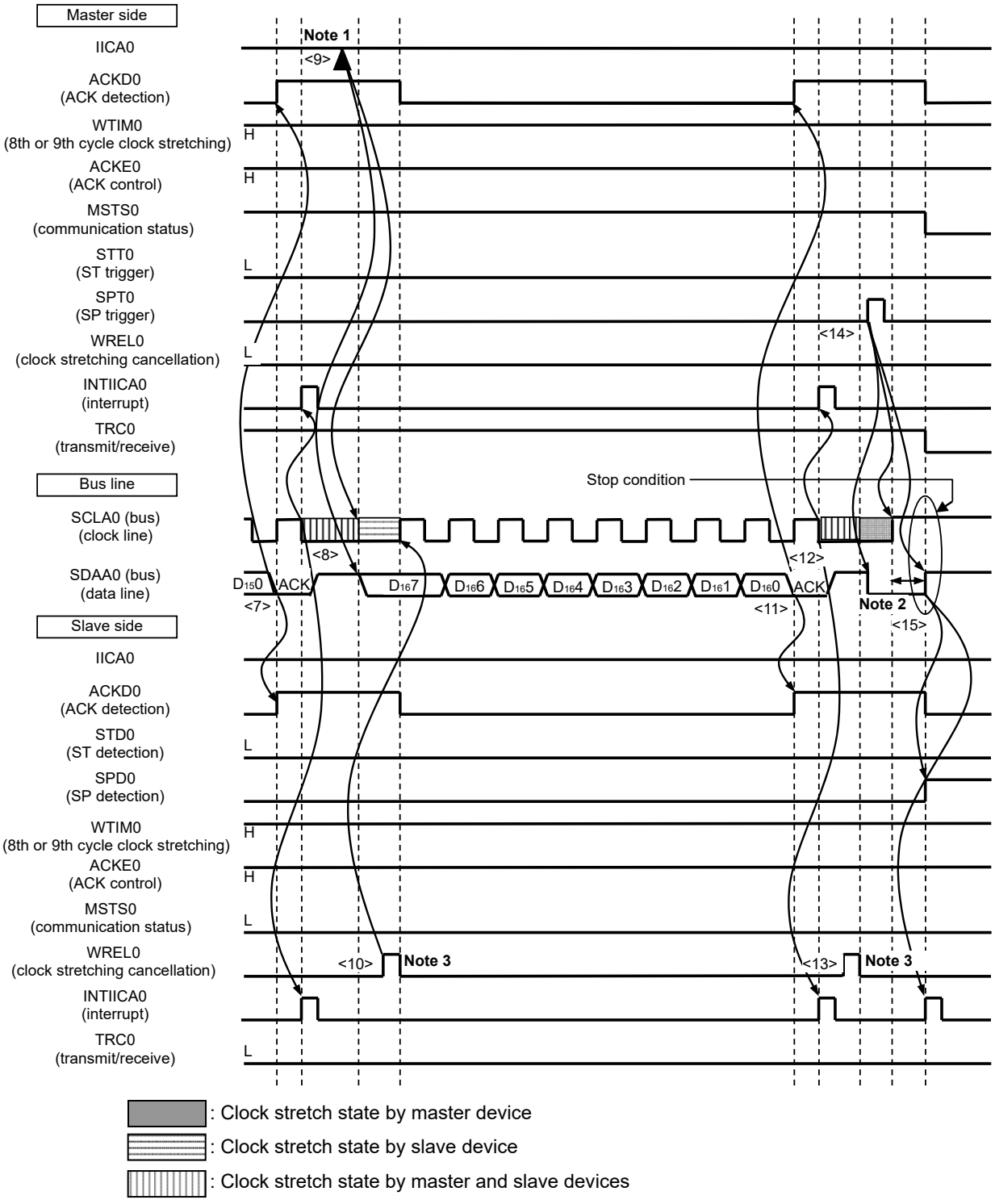
<10> The slave device reads the received data and releases the clock stretch status (WREL0 = 1). The master device then starts transferring data to the slave device.

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: SDAA0 = 1). The slave device also does not issue the INTIICA0 interrupt (address match) and does not set a clock stretch status. The master device, however, issues the INTIICA0 interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <15> in Figure 17-33 represent the entire procedure for communicating data using the I^2C bus. Figure 17-33 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 17-33 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 17-33 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 17-33. Example of Master to Slave Communication
(When 9th Cycle Clock Stretching Is Selected for Master, 9th Cycle Clock Stretching Is Selected for Slave) (3/4)

(3) Data ~ data ~ Stop condition



Notes 1. Write data to IICA0, not setting the WREL0 bit, in order to cancel a clock stretch state during master transmission.

2. Make sure that the time between the rise of the SCLA0 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.

3. To cancel slave clock stretch, write “FFH” to IICA0 or set the WREL0 bit.

The meanings of <7> to <15> in (3) Data ~ data ~ stop condition in Figure 17-33 are explained below.

<7> When data transfer is complete, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a clock stretch status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

<9> The master device writes the data to transmit to the IICA shift register 0 (IICA0) and releases the clock stretch status that it set by the master device.

<10> The slave device reads the received data and releases the clock stretch status (WREL0 = 1). The master device then starts transferring data to the slave device.

<11> When data transfer is complete, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<12> The master device and slave device set a clock stretch status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

<13> The slave device reads the received data and releases the clock stretch status (WREL0 = 1).

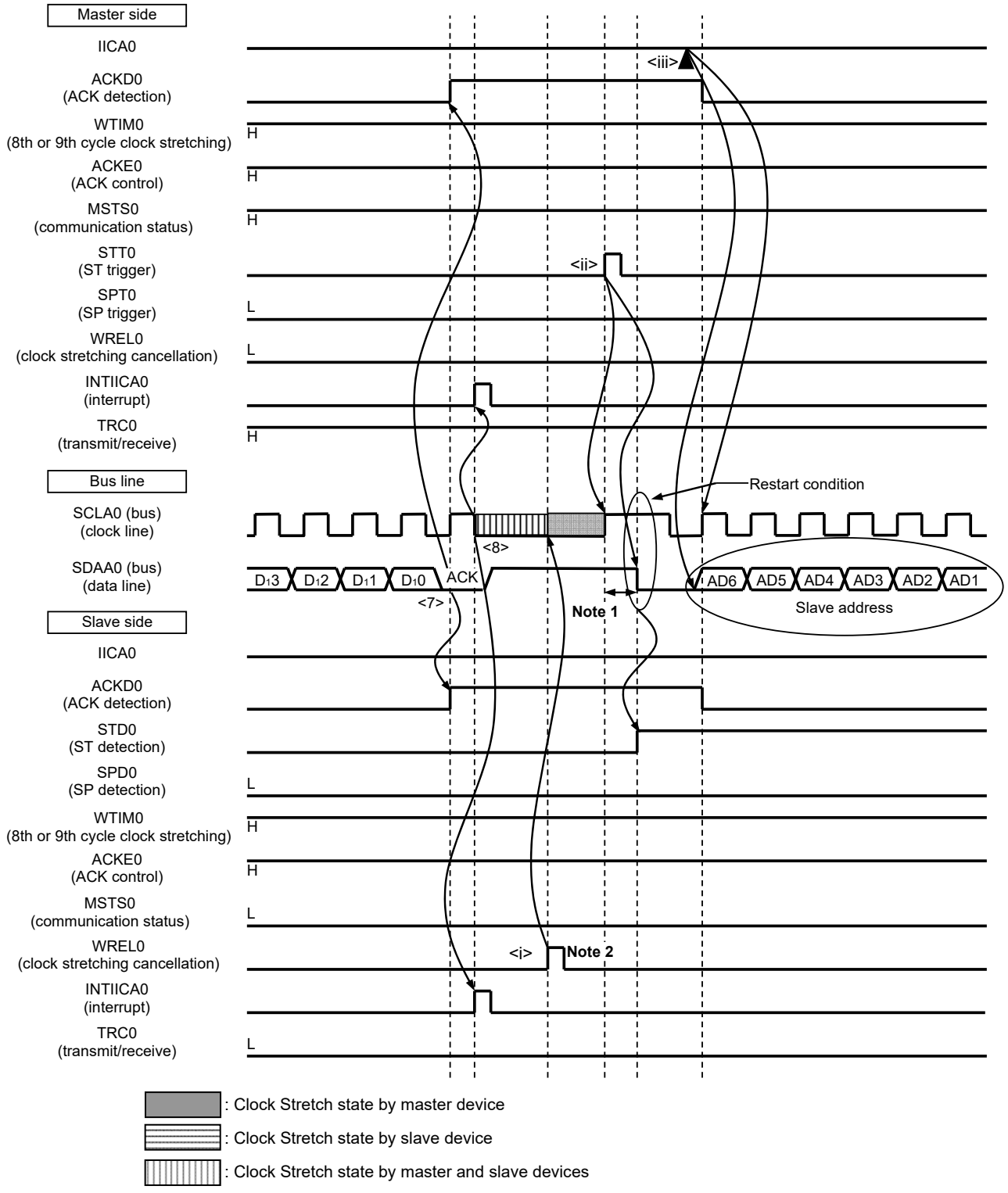
<14> After a stop condition trigger is set, the bus data line is cleared (SDAA0 = 0) and the bus clock line is set (SCLA0 = 1). The stop condition is then generated by setting the bus data line (SDAA0 = 1) after the stop condition setup time has elapsed.

<15> When a stop condition is generated, the slave device detects the stop condition and issues an interrupt (INTIICA0: stop condition).

Remark <1> to <15> in Figure 17-33 represent the entire procedure for communicating data using the I^2C bus. Figure 17-33 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 17-33 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 17-33 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 17-33. Example of Master to Slave Communication
(When 9th Cycle Clock Stretching Is Selected for Master, 9th Cycle Clock Stretching Is Selected for Slave) (4/4)

(4) Data ~ restart condition ~ address



Notes 1. Make sure that the time between the rise of the SCLA0 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.

2. To cancel slave clock stretch, write “FFH” to IICA0 or set the WREL0 bit.

The following describes the operations in Figure 17-33 (4) Data ~ restart condition ~ address. After the operations in steps <7> and <8>, the operations in steps <i> to <ii> are performed. These steps return the processing to step <3>, the data transmission step.

<7> When data transfer is complete, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a clock stretch status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

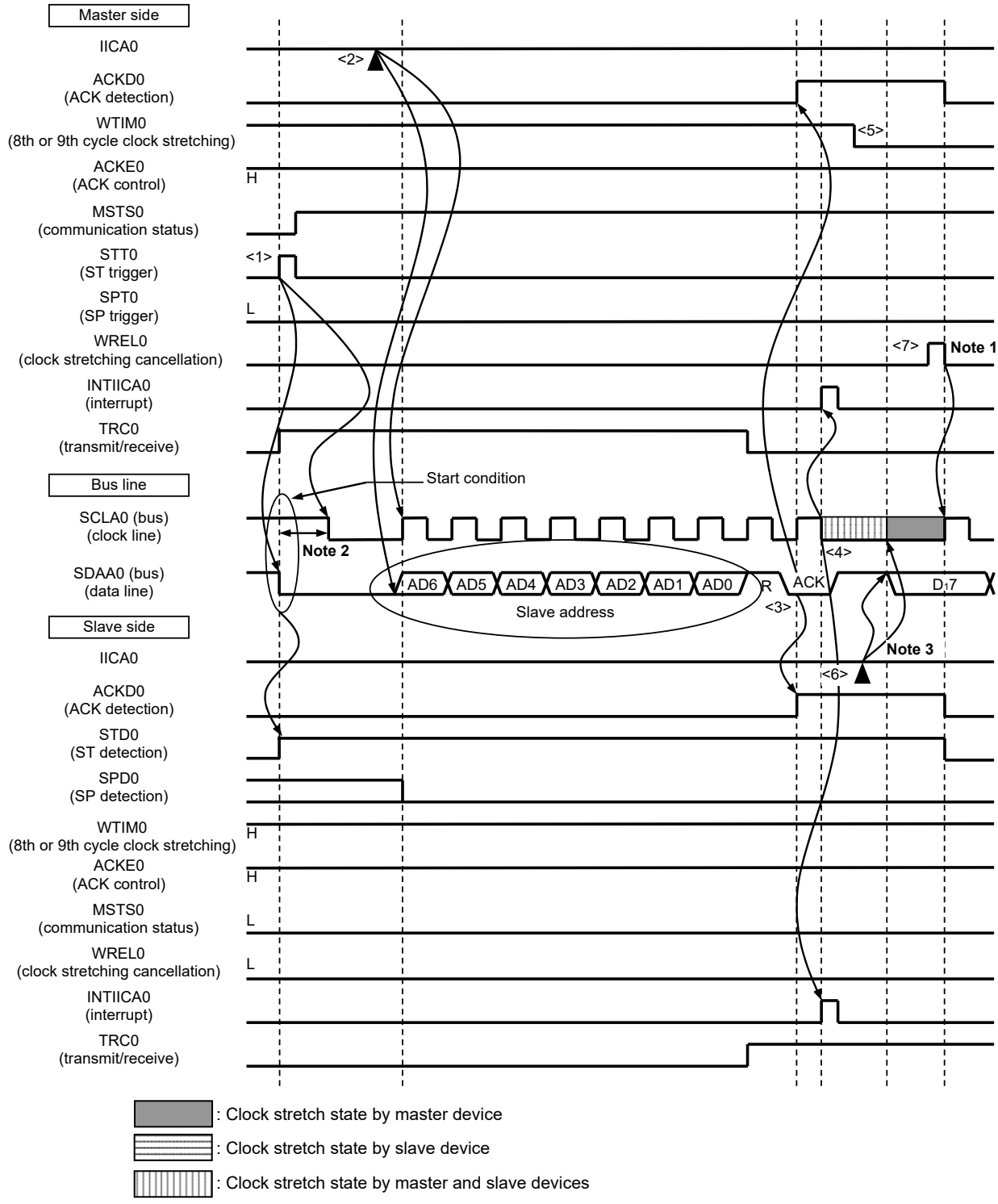
<i> The slave device reads the received data and releases the clock stretch status (WREL0 = 1).

<ii> The start condition trigger is set again by the master device (STT0 = 1) and a start condition (SDAA0 = 0 and SCLA0 = 1) is generated once the bus clock line goes high (SCLA0 = 1) and the bus data line goes low (SDAA0 = 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 (SCLA0 = 0) after the hold time has elapsed.

<iii> The master device writes the address + R/W (transmission) to the IICA shift register 0 (IICA0) and transmits the slave address.

Figure 17-34. Example of Slave to Master Communication
(When 8th Cycle Clock Stretching Is Selected for Master, 9th Cycle Clock Stretching Is Selected for Slave) (1/3)

(1) Start condition ~ address ~ data



Notes 1. To cancel master clock stretch, write “FFH” to IICA0 or set the WREL0 bit.

2. Make sure that the time between the fall of the SDAA0 pin signal and the fall of the SCLA0 pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. Write data to IICA0, not setting the WREL0 bit, in order to cancel a clock stretch state during slave transmission.

The meanings of <1> to <7> in (1) Start condition ~ address ~ data in Figure 17-34 are explained below.

<1> The start condition trigger is set by the master device (STT0 = 1) and a start condition (i.e. SCLA0 =1 changes SDAA0 from 1 to 0) is generated once the bus data line goes low (SDAA0). When the start condition is subsequently detected, the master device enters the master device communication status (MSTS0 = 1). The master device is ready to communicate once the bus clock line goes low (SCLA0 = 0) after the hold time has elapsed.

<2> The master device writes the address + R (reception) to the IICA shift register 0 (IICA0) and transmits the slave address.

<3> In the slave device if the address received matches the address (SVA0 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 (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA0: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a clock stretch status (SCLA0 = 0) and issues an interrupt (INTIICA0: address match)**Note**.

<5> The timing at which the master device sets the clock stretch status changes to the 8th clock (WTIM0 = 0).

<6> The slave device writes the data to transmit to the IICA0 register and releases the clock stretch status that it set by the slave device.

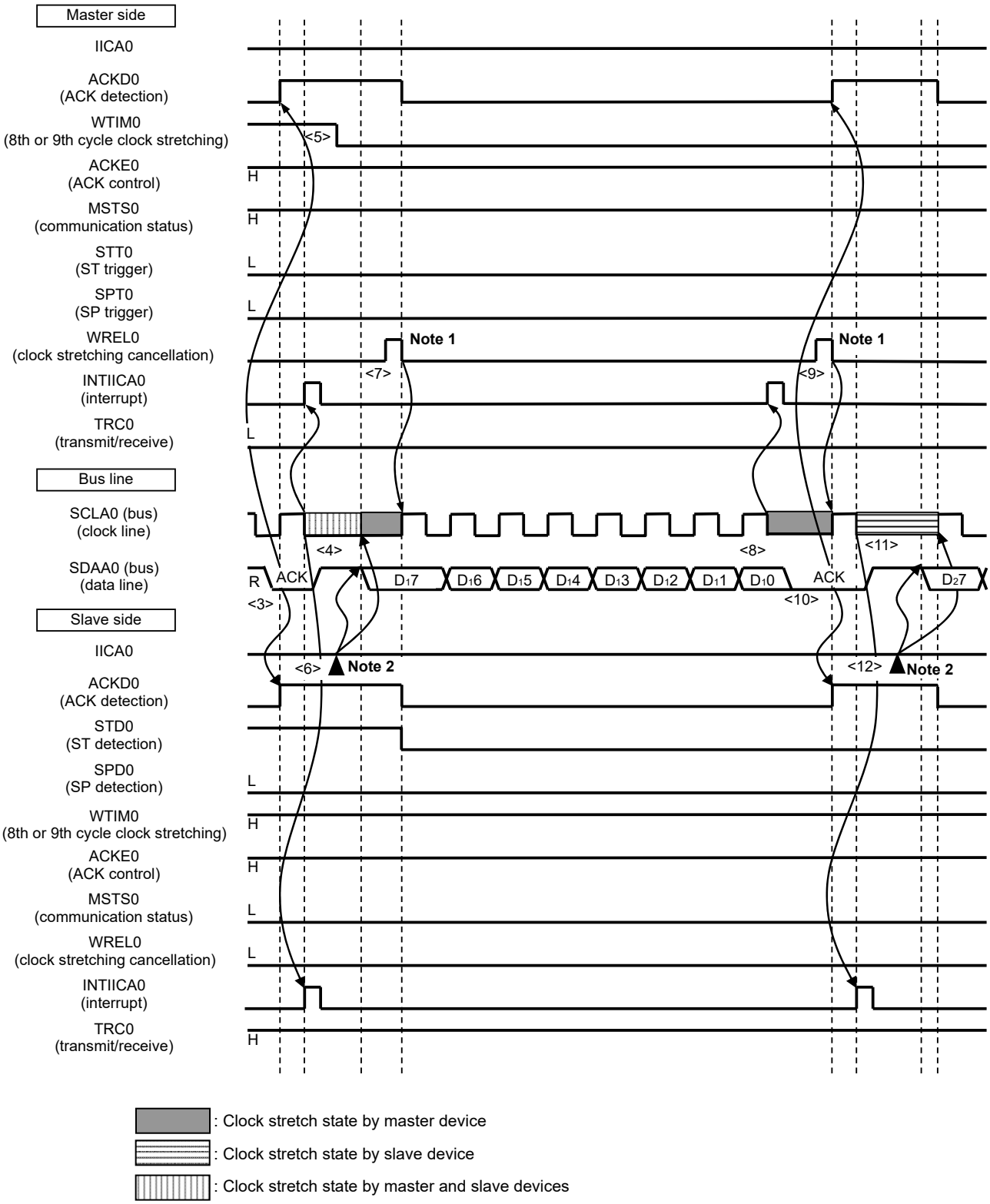
<7> If the master device releases the clock stretch status (WREL0 = 1), the slave device starts transferring data to the master device.

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: SDAA0 = 1). The slave device also does not issue the INTIICA0 interrupt (address match) and does not set a clock stretch status. The master device, however, issues the INTIICA0 interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <19> in Figure 17-34 represent the entire procedure for communicating data using the I^2C bus. Figure 17-34 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 17-34 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 17-34 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Figure 17-34. Example of Slave to Master Communication
(When 8th Cycle Clock Stretching Is Selected for Master, 9th Cycle Clock Stretching Is Selected for Slave) (2/3)

(2) Address ~ data ~ data



Notes 1. To cancel master clock stretch, write “FFH” to IICA0 or set the WREL0 bit.

2. Write data to IICA0, not setting the WREL0 bit, in order to cancel a clock stretch state during slave transmission.

The meanings of <3> to <12> in (2) Address ~ data ~ data in Figure 17-34 are explained below.

<3> If the address received matches the address 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 (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA0: end of address transmission) at the falling edge of the 9th clock, and the slave device whose address matched the transmitted slave address also issues an interrupt (INTIICA0: address match). The master device and slave device also set a clock stretch status (SCLA0 = 0)[Note] when the addresses match.

<5> The timing at which the master device sets the clock stretch status changes to the 8th clock (WTIM0 = 0).

<6> The slave device writes the data to transmit to the IICA shift register 0 (IICA0) and releases the clock stretch status that it set by the slave device.

<7> If the master device releases the clock stretch status (WREL0 = 1), the slave device starts transferring data to the master device.

<8> The master device sets a clock stretch status (SCLA0 = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICA0: end of transfer). 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 status (WREL0 = 1).

<10> The ACK is detected by the slave device (ACKD0 = 1) at the rising edge of the 9th clock.

<11> The slave device set a clock stretch status (SCLA0 = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICA0: end of transfer).

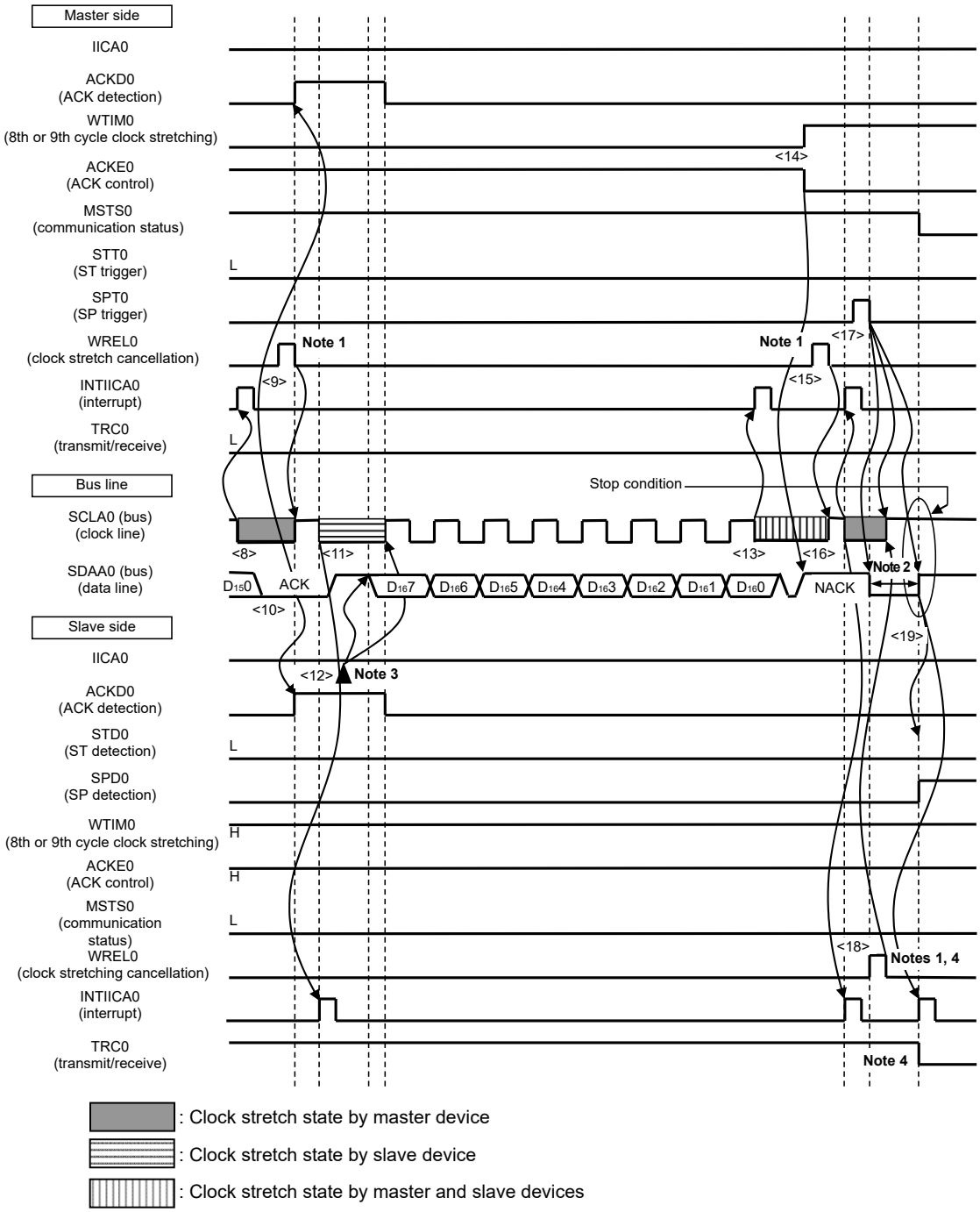
<12> The slave device writes the data to transmit to the IICA0 register and releases the clock stretch status that it set by the slave device. The slave device then starts transferring data to the master device.

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: SDAA0 = 1). The slave device also does not issue the INTIICA0 interrupt (address match) and does not set a clock stretch status. The master device, however, issues the INTIICA0 interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <19> in Figure 17-34 represent the entire procedure for communicating data using the I^2C bus. Figure 17-34 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 17-34 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 17-34 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Figure 17-34. Example of Slave to Master Communication (When 8th Cycle Clock Stretching is Changed to 9th Cycle Clock Stretching for Master, 9th Cycle Clock Stretching is Selected for Slave) (3/3)

(3) Data ~ data ~ stop condition



Notes 1. To cancel a clock stretch state, write “FFH” to IICA0 or set the WREL0 bit.

2. Make sure that the time between the rise of the SCLA0 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.

3. Write data to IICA0, not setting the WREL0 bit, in order to cancel a clock stretch state during slave transmission.

4. If a clock stretch state during slave transmission is canceled by setting the WREL0 bit, the TRC0 bit will be cleared.

The meanings of <8> to <19> in (3) Data ~ data ~ stop condition in Figure 17-34 are explained below.

<8> The master device sets a clock stretch status (SCLA0 = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICA0: end of transfer). 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 status (WREL0 = 1).

<10> The ACK is detected by the slave device (ACKD0 = 1) at the rising edge of the 9th clock.

<11> The slave device set a clock stretch status (SCLA0 = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICA0: end of transfer).

<12> The slave device writes the data to transmit to the IICA shift register 0 (IICA0) and releases the clock stretch status that it set by the slave device. The slave device then starts transferring data to the master device.

<13> The master device issues an interrupt (INTIICA0: end of transfer) at the falling edge of the 8th clock, and sets a clock stretch status (SCLA0 = 0). Because ACK control (ACKE0 = 1) is performed, the bus data line is at the low level (SDAA0 = 0) at this stage.

<14> The master device sets NACK as the response (ACKE0 = 0) and changes the timing at which it sets the clock stretch status to the 9th clock.

<15> If the master device releases the clock stretch status (WREL0 = 1), the slave device detects the NACK (ACKD0 = 0) at the rising edge of the 9th clock.

<16> The master device and slave device set a clock stretch status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

<17> When the master device issues a stop condition (SPT0 = 1), the bus data line is cleared (SDAA0 = 0) and the master device releases the clock stretch status. The master device then clock stretch until the bus clock line is set (SCLA0 = 1).

<18> The slave device acknowledges the NACK, halts transmission, and releases the clock stretch status (WREL0 = 1) to end communication. Once the slave device releases the clock stretch status, the bus clock line is set (SCLA0 = 1).

<19> Once the master device recognizes that the bus clock line is set (SCLA0 = 1) and after the stop condition setup time has elapsed, the master device sets the bus data line (SDAA0 = 1) and issues a stop condition. The slave device detects the generated stop condition and both the master device and slave device issue an interrupt (INTIICA0: stop condition).

Remark <1> to <19> in Figure 17-34 represent the entire procedure for communicating data using the I^2C bus. Figure 17-34 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 17-34 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 17-34 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

CHAPTER 18 MULTIPLIER AND DIVIDER/MULTIPLY-ACCUMULATOR

18.1 Functions of Multiplier and Divider/Multiply-Accumulator

The multiplier and divider/multiply-accumulator has the following functions.

- 16 bits × 16 bits = 32 bits (unsigned)
- 16 bits × 16 bits = 32 bits (signed)
- 16 bits × 16 bits + 32 bits = 32 bits (unsigned)
- 16 bits × 16 bits + 32 bits = 32 bits (signed)
- 32 bits ÷ 32 bits = 32 bits, 32-bits remainder (unsigned)

18.2 Configuration of Multiplier and Divider/Multiply-Accumulator

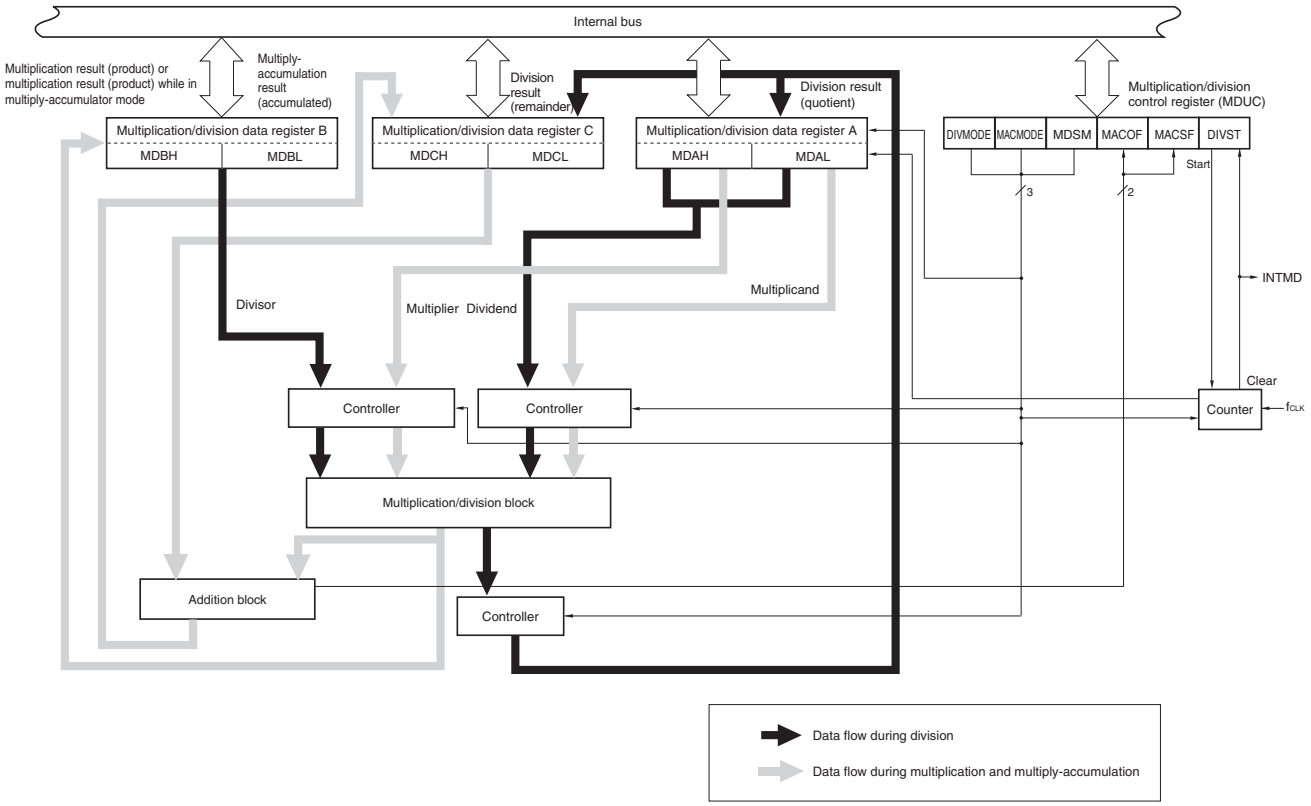
The multiplier and divider/multiply-accumulator consists of the following hardware.

Table 18-1. Configuration of Multiplier and Divider/Multiply-Accumulator

Item	Configuration
Registers	Multiplication/division data register A (L) (MDAL) Multiplication/division data register A (H) (MDAH) Multiplication/division data register B (L) (MDBL) Multiplication/division data register B (H) (MDBH) Multiplication/division data register C (L) (MDCL) Multiplication/division data register C (H) (MDCH)
Control register	Multiplication/division control register (MDUC)

Figure 18-1 shows a block diagram of the multiplier and divider/multiply-accumulator.

Figure 18-1. Block Diagram of Multiplier and Divider/Multiply-Accumulator



Remark f_{CLK}: CPU/peripheral hardware clock frequency

18.2.1 Multiplication/division data register A (MDAH, MDAL)

The MDAH and MDAL registers set the values that are used for a multiplication or division operation and store the operation result. They set the multiplier and multiplicand data in the multiplication mode or multiply-accumulator mode, and set the dividend data in the division mode. Furthermore, the operation result (quotient) is stored in the MDAH and MDAL registers in the division mode.

The MDAH and MDAL registers can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 18-2. Format of Multiplication/Division Data Register A (MDAH, MDAL)

Address: FFFF0H, FFFF1H, FFFF2H, FFFF3H After reset: 0000H, 0000H R/W

Symbol	FFFF3H								FFFF2H							
MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH	MDAH
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Symbol	FFFF1H								FFFF0H							
MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL	MDAL
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Cautions 1. Do not rewrite the MDAH and MDAL registers values during division operation processing (when the multiplication/division control register (MDUC) value is 81H or C1H). The operation will be executed in this case, but the operation result will be an undefined value.

2. The MDAH and MDAL registers values read during division operation processing (when the MDUC register value is 81H or C1H) will not be guaranteed.

3. The data is in the two's complement format in either the multiplication mode (signed) or multiply-accumulator mode (signed).

The following table shows the functions of the MDAH and MDAL registers during operation execution.

Table 18-2. Functions of MDAH and MDAL Registers During Operation Execution

Operation Mode	Setting	Operation Result
Multiplication mode (unsigned) Multiply-accumulator mode (unsigned)	MDAH: Multiplier (unsigned) MDAL: Multiplicand (unsigned)	–
Multiplication mode (signed) Multiply-accumulator mode (signed)	MDAH: Multiplier (signed) MDAL: Multiplicand (signed)	–
Division mode (unsigned)	MDAH: Dividend (unsigned) (higher 16 bits) MDAL: Dividend (unsigned) (lower 16 bits)	MDAH: Division result (unsigned) Higher 16 bits MDAL: Division result (unsigned) Lower 16 bits

18.2.2 Multiplication/division data register B (MDBL, MDBH)

The MDBH and MDBL registers set the values that are used for multiplication or division operation and store the operation result. They store the operation result (product) in the multiplication mode and multiply-accumulator mode, and set the divisor data in the division mode.

The MDBH and MDBL registers can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 18-3. Format of Multiplication/Division Data Register B (MDBH, MDBL)

Address: FFFF4H, FFFF5H, FFFF6H, FFFF7H After reset: 0000H, 0000H R/W

Symbol	FFFF5H								FFFF4H							
MDBH	MDBH 15	MDBH 14	MDBH 13	MDBH 12	MDBH 11	MDBH 10	MDBH 9	MDBH 8	MDBH 7	MDBH 6	MDBH 5	MDBH 4	MDBH 3	MDBH 2	MDBH 1	MDBH 0

Symbol	FFFF7H								FFFF6H							
MDBL	MDBL 15	MDBL 14	MDBL 13	MDBL 12	MDBL 11	MDBL 10	MDBL 9	MDBL 8	MDBL 7	MDBL 6	MDBL 5	MDBL 4	MDBL 3	MDBL 2	MDBL 1	MDBL 0

Cautions 1. Do not rewrite the MDBH and MDBL registers values during division operation processing (when the multiplication/division control register (MDUC) value is 81H or C1H) or multiply-accumulation operation processing. The operation result will be an undefined value.

2. Do not set the MDBH and MDBL registers to 0000H in the division mode. If they are set, the operation result will be an undefined value.

3. The data is in the two's complement format in either the multiplication mode (signed) or multiply-accumulator mode (signed).

The following table shows the functions of the MDBH and MDBL registers during operation execution.

Table 18-3. Functions of MDBH and MDBL Registers During Operation Execution

Operation Mode	Setting	Operation Result
Multiplication mode (unsigned) Multiply-accumulator mode (unsigned)	–	MDBH: Multiplication result (product) (unsigned) Higher 16 bits MDBL: Multiplication result (product) (unsigned) Lower 16 bits
Multiplication mode (signed) Multiply-accumulator mode (signed)	–	MDBH: Multiplication result (product) (signed) Higher 16 bits MDBL: Multiplication result (product) (signed) Lower 16 bits
Division mode (unsigned)	MDBH: Divisor (unsigned) (higher 16 bits) MDBL: Divisor (unsigned) (lower 16 bits)	–

18.2.3 Multiplication/division data register C (MDCL, MDCH)

The MDCH and MDCL registers are used to store the accumulated result while in the multiply-accumulator mode or the remainder of the operation result while in the division mode. These registers are not used while in the multiplication mode.

The MDCH and MDCL registers can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 18-4. Format of Multiplication/Division Data Register C (MDCH, MDCL)

Address: F00E0H, F00E1H, F00E2H, F00E3H After reset: 0000H, 0000H R/W

Symbol F00E3H F00E2H

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

Symbol F00E1H F00E0H

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

Cautions 1. The MDCH and MDCL registers values read during division operation processing (when the multiplication/division control register (MDUC) value is 81H or C1H) will not be guaranteed.

2. During multiply-accumulator processing, do not use software to rewrite the values of the MDCH and MDCL registers. If this is done, the operation result will be undefined.

3. The data is in the two's complement format in the multiply-accumulator mode (signed).

Table 18-4. Functions of MDCH and MDCL Registers During Operation Execution

Operation Mode	Setting	Operation Result
Multiplication mode (unsigned or signed)	–	–
Multiply-accumulator mode (unsigned)	MDCH: Initial accumulated value (unsigned) (higher 16 bits) MDCL: Initial accumulated value (unsigned) (lower 16 bits)	MDCH: accumulated value (unsigned) (higher 16 bits) MDCL: accumulated value (unsigned) (lower 16 bits)
Multiply-accumulator mode (signed)	MDCH: Initial accumulated value (signed) (higher 16 bits) MDCL: Initial accumulated value (signed) (lower 16 bits)	MDCH: accumulated value (signed) (higher 16 bits) MDCL: accumulated value (signed) (lower 16 bits)
Division mode (unsigned)	–	MDCH: Remainder (unsigned) (higher 16 bits) MDCL: Remainder (unsigned) (lower 16 bits)

The register configuration differs between when multiplication is executed and when division is executed, as follows.

- Register configuration during multiplication

  <Multiplier A> <Multiplier B> <Product>

  MDAL (bits 15 to 0) × MDAH (bits 15 to 0) = [MDBH (bits 15 to 0), MDBL (bits 15 to 0)]

- Register configuration during multiply-accumulation

  <Multiplier A> <Multiplier B> < accumulated value > < accumulated result >

  MDAL (bits 15 to 0) × MDAH (bits 15 to 0) + MDC (bits 31 to 0) = [MDCH (bits 15 to 0), MDCL (bits 15 to 0)]

  (The multiplication result is stored in the MDBH (bits 15 to 0) and MDBL (bits 15 to 0).)

- Register configuration during division

  <Dividend> <Divisor>

  [MDAH (bits 15 to 0), MDAL (bits 15 to 0)] ÷ [MDBH (bits 15 to 0), MDBL (bits 15 to 0)] =

  <Quotient> <Remainder>

  [MDAH (bits 15 to 0), MDAL (bits 15 to 0)] ··· [MDCH (bits 15 to 0), MDCL (bits 15 to 0)]

18.3 Register Controlling Multiplier and Divider/Multiply-Accumulator

The multiplier and divider/multiply-accumulator is controlled by using the multiplication/division control register (MDUC).

18.3.1 Multiplication/division control register (MDUC)

The MDUC register is an 8-bit register that controls the operation of the multiplier and divider/multiply-accumulator.

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

Note that the overflow flag (MACOF) and sign flag (MACSF) of the multiply-accumulation result (accumulated) are read-only flags.

Reset signal generation clears this register to 00H.

Figure 18-5. Format of Multiplication/Division Control Register (MDUC)

Address: F00E8H After reset: 00H R/W[Note 1]

Symbol	<7>	<6>	5	4	<3>	<2>	<1>	<0>
MDUC	DIVMODE	MACMODE	0	0	MDSM	MACOF	MACSF	DIVST

DIVMODE	MACMODE	MDSM	Operation mode selection
0	0	0	Multiplication mode (unsigned) (default)
0	0	1	Multiplication mode (signed)
0	1	0	Multiply-accumulator mode (unsigned)
0	1	1	Multiply-accumulator mode (signed)
1	0	0	Division mode (unsigned), generation of a division completion interrupt (INTMD)
1	1	0	Division mode (unsigned), not generation of a division completion interrupt (INTMD)
Other than above			Setting prohibited

MACOF	Overflow flag of multiply-accumulation result (accumulated value)
0	No overflow
1	With over flow
<Set condition> • For the multiply-accumulator mode (unsigned) The bit is set when the accumulated value goes outside the range from 00000000h to FFFFFFFFh. • For the multiply-accumulator mode (signed) The bit is set when the result of adding a positive product to a positive accumulated value exceeds 7FFFFFFFh and is negative, or when the result of adding a negative product to a negative accumulated value exceeds 80000000h and is positive.	

MACSF	Sign flag of multiply-accumulation result (accumulated value)
0	The accumulated value is positive.
1	The accumulated value is negative.
Multiply-accumulator mode (unsigned): The bit is always 0. Multiply-accumulator mode (signed): The bit indicates the sign bit of the accumulated value.	

DIVST[Note 2]	Division operation start/stop
0	Division operation processing complete
1	Starts division operation/division operation processing in progress

Notes 1. Bits 1 and 2 are read-only bits.

2. The DIVST bit can only be set (1) in the division mode. In the division mode, division operation is started by setting (1) the DIVST bit. The DIVST bit is automatically cleared (0) when the operation ends. In the multiplication mode, operation is automatically started by setting the multiplier and multiplicand to multiplication/division data register A (MDAH, MDAL), respectively.

Cautions 1. Do not rewrite the DIVMODE, MDSM bits during operation processing (while the DIVST bit is 1). If it is rewritten, the operation result will be an undefined value.

2. The DIVST bit cannot be cleared (0) by using software during division operation processing (while the DIVST bit is 1).

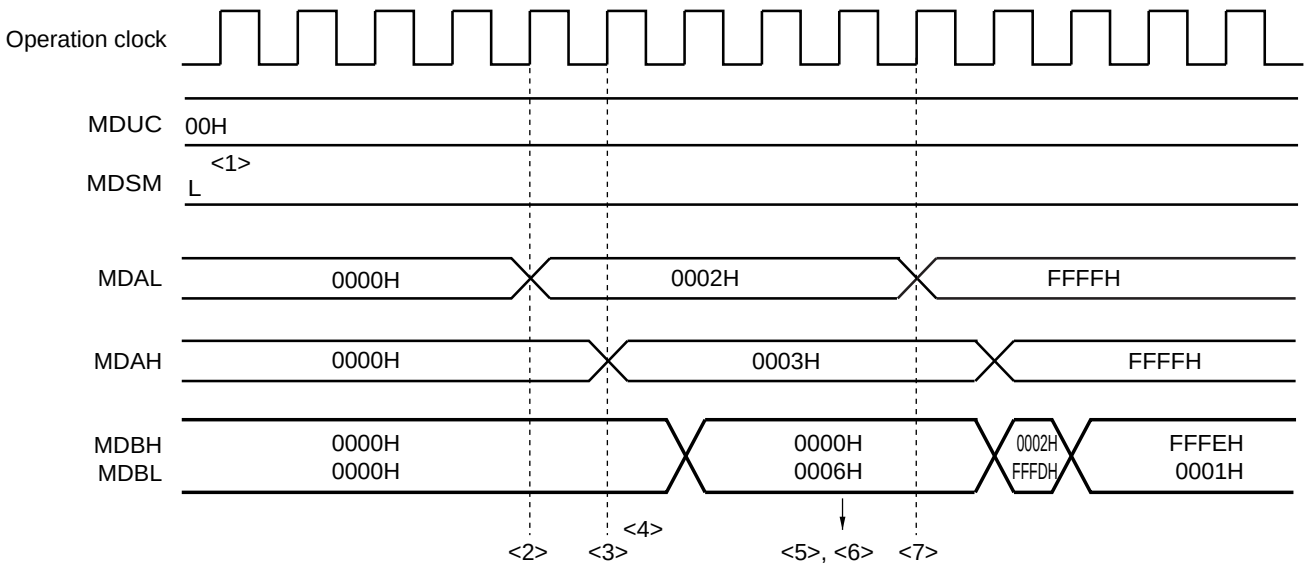
18.4 Operations of Multiplier and Divider/Multiply-Accumulator

18.4.1 Multiplication (unsigned) operation

- Initial setting
  - <1> Set the multiplication/division control register (MDUC) to 00H.
  - <2> Set the multiplicand to multiplication/division data register A (L) (MDAL).
  - <3> Set the multiplier to multiplication/division data register A (H) (MDAH).
    (There is no preference in the order of executing steps <2> and <3>. Multiplication operation is automatically started when the multiplier and multiplicand are set to the MDAH and MDAL registers, respectively.)
- During operation processing
  - <4> Wait for at least one clock. The operation will end when one clock has been issued.
- Operation end
  - <5> Read the product (lower 16 bits) from multiplication/division data register B (L) (MDBL).
  - <6> Read the product (higher 16 bits) from multiplication/division data register B (H) (MDBH).
    (There is no preference in the order of executing steps <5> and <6>.)
- Next operation
  - <7> Start with the initial settings of each step to change the operation mode.
    When the same operation mode is used sequentially, settings <1> and <2> can be omitted.

Remark Steps <1> to <7> correspond to <1> to <7> in Figure 18-6.

Figure 18-6. Timing Diagram of Multiplication (Unsigned) Operation (2 × 3 = 6)



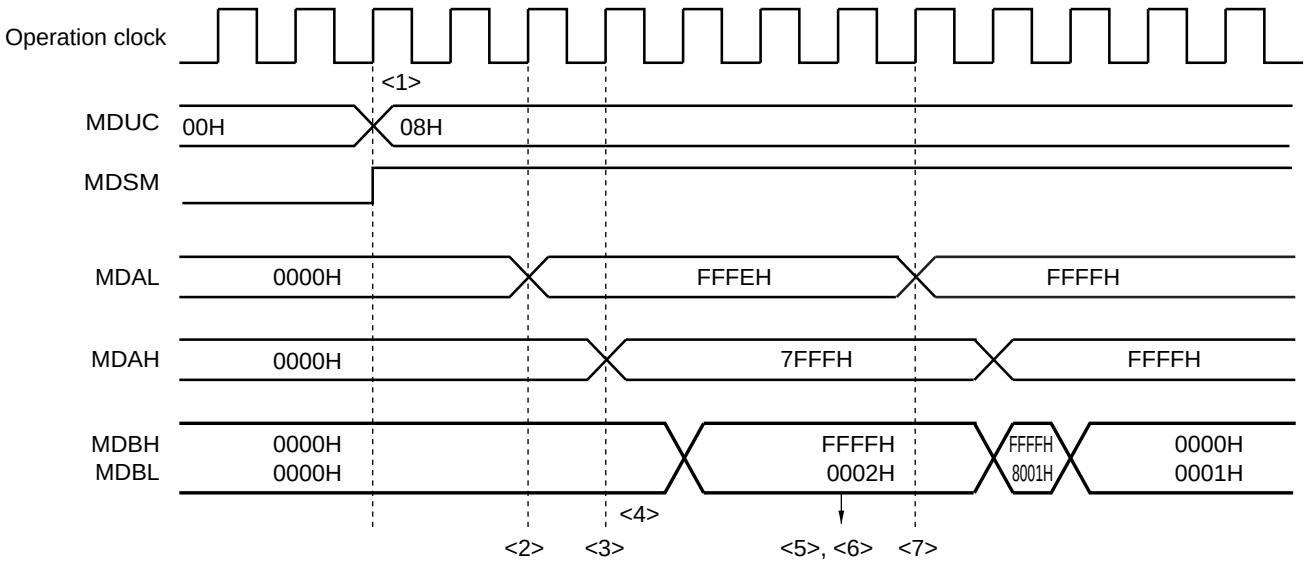
18.4.2 Multiplication (signed) operation

- Initial setting
  - <1> Set the multiplication/division control register (MDUC) to 08H.
  - <2> Set the multiplicand to multiplication/division data register A (L) (MDAL).
  - <3> Set the multiplier to multiplication/division data register A (H) (MDAH).
    (There is no preference in the order of executing steps <2> and <3>. Multiplication operation is automatically started when the multiplier and multiplicand are set to the MDAH and MDAL registers, respectively.)
- During operation processing
  - <4> Wait for at least one clock. The operation will end when one clock has been issued.
- Operation end
  - <5> Read the product (lower 16 bits) from multiplication/division data register B (L) (MDBL).
  - <6> Read the product (higher 16 bits) from multiplication/division data register B (H) (MDBH).
    (There is no preference in the order of executing steps <5> and <6>.)
- Next operation
  - <7> Start with the initial settings of each step to change the operation mode.
    When the same operation mode is used sequentially, settings <1> and <2> can be omitted.

Caution The data is in the two's complement format in multiplication mode (signed).

Remark Steps <1> to <7> correspond to <1> to <7> in Figure 18-7.

Figure 18-7. Timing Diagram of Multiplication (Signed) Operation (–2 × 32767 = –65534)

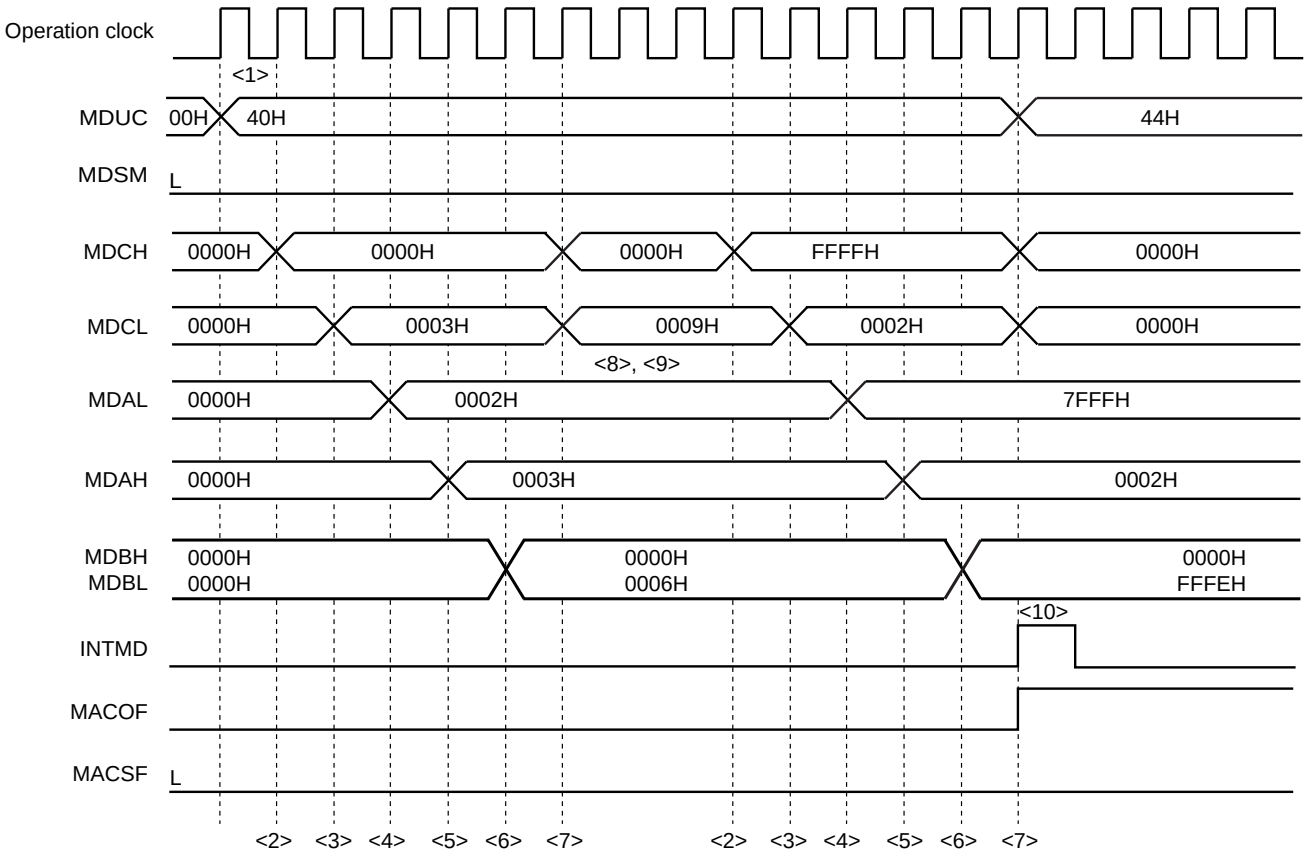


18.4.3 Multiply-accumulation (unsigned) operation

- Initial setting
  - <1> Set the multiplication/division control register (MDUC) to 40H.
  - <2> Set the initial accumulated value of higher 16 bits to multiplication/division data register C (L) (MDCL).
  - <3> Set the initial accumulated value of lower 16 bits to multiplication/division data register C (H) (MDCH).
  - <4> Set the multiplicand to multiplication/division data register A (L) (MDAL).
  - <5> Set the multiplier to multiplication/division data register A (H) (MDAH).
    (There is no preference in the order of executing steps <2>, <3>, and <4>. Multiplication operation is automatically started when the multiplier is set to the MDAH register, respectively.)
- During operation processing
  - <6> The multiplication operation finishes in one clock cycle.
    (The multiplication result is stored in multiplication/division data register B (L) (MDBL) and multiplication/division data register B (H) (MDBH).)
  - <7> After <6>, the multiply-accumulation operation finishes in one additional clock cycle. (There is a wait of at least two clock cycles after specifying the initial settings is finished (<5>).)
- Operation end
  - <8> Read the accumulated value (lower 16 bits) from the MDCL register.
  - <9> Read the accumulated value (higher 16 bits) from the MDCH register.
    (There is no preference in the order of executing steps <8> and <9>.)
  - (<10> If the result of the multiply-accumulation operation causes an overflow, the MACOF bit is set to 1, INTMD signal is occurred.)
- Next operation
  - <11> Start with the initial settings of each step to change the operation mode.
    When the same operation mode is used sequentially, settings <1> to <4> can be omitted.

Remark Steps <1> to <10> correspond to <1> to <10> in Figure 18-8.

Figure 18-8. Timing Diagram of Multiply-Accumulation (Unsigned) Operation
(2 × 3 + 3 = 9 → 32767 × 2 + 4294901762 = 0 (Over Flow Generated))



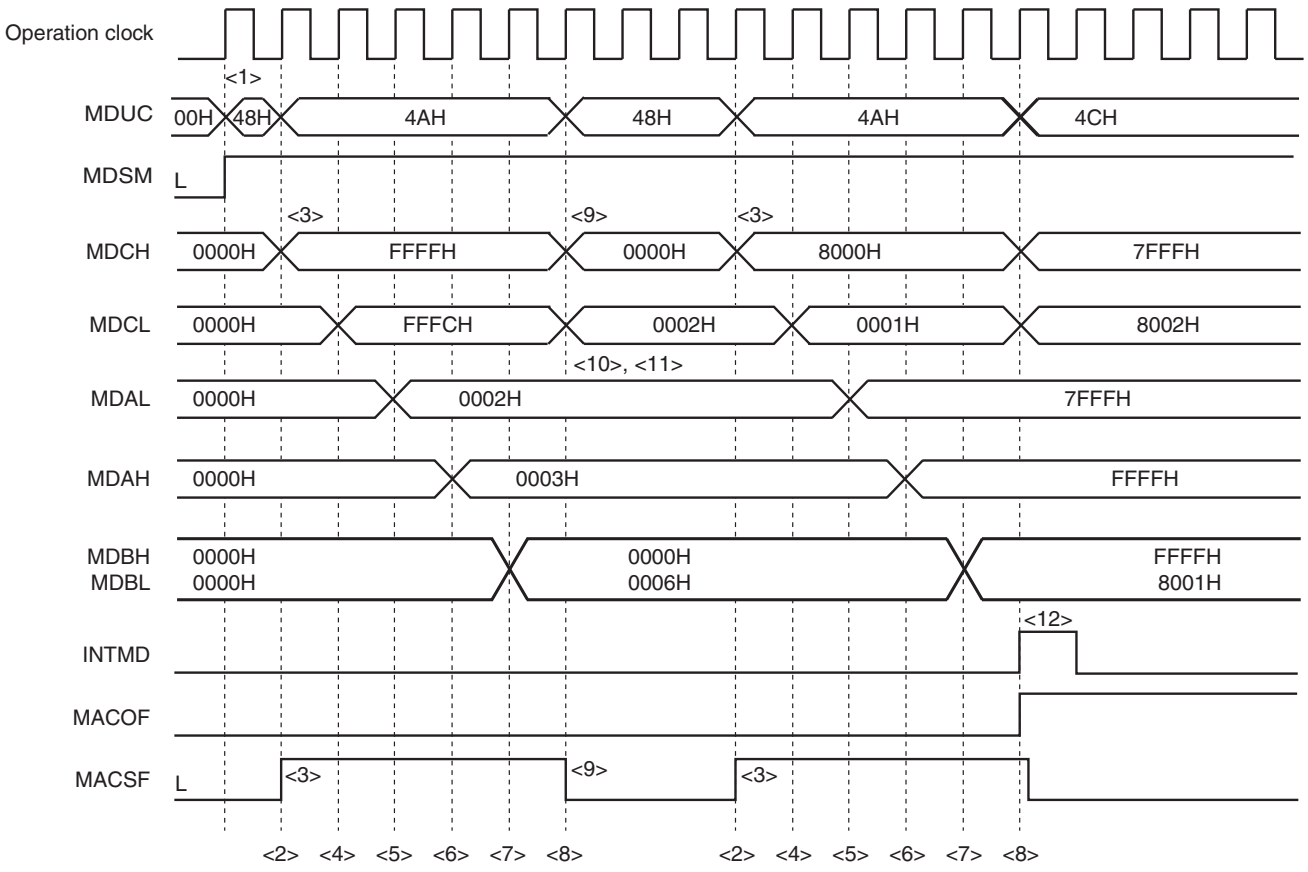
18.4.4 Multiply-accumulation (signed) operation

- Initial setting
  - <1> Set the multiplication/division control register (MDUC) to 48H.
  - <2> Set the initial accumulated value of higher 16 bits to multiplication/division data register C (H) (MDCH).
    (<3> If the accumulated value in the MDCH register is negative, the MACSF bit is set to 1.)
  - <4> Set the initial accumulated value of lower 16 bits to multiplication/division data register C (L) (MDCL).
  - <5> Set the multiplicand to multiplication/division data register A (L) (MDAL).
  - <6> Set the multiplier to multiplication/division data register A (H) (MDAH).
    (There is no preference in the order of executing steps <2>, <4>, and <5>. Multiplication operation is automatically started when the multiplier is set to the MDAH register of <6>, respectively.)
- During operation processing
  - <7> The multiplication operation finishes in one clock cycle.
    (The multiplication result is stored in multiplication/division data register B (L) (MDBL) and multiplication/division data register B (H) (MDBH).)
  - <8> After <7>, the multiply-accumulation operation finishes in one additional clock cycle. (There is a wait of at least two clock cycles after specifying the initial settings is finished (<6>).)
- Operation end
  - <9> If the accumulated value stored in the MDCL and MDCH registers is positive, the MACSF bit is cleared to 0.
  - <10> Read the accumulated value (lower 16 bits) from the MDCL register.
  - <11> Read the accumulated value (higher 16 bits) from the MDCH register.
    (There is no preference in the order of executing steps <10> and <11>.)
  - (<12> If the result of the multiply-accumulation operation causes an overflow, the MACOF bit is set to 1, INTMD signal is occurred.)
- 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.

Caution The data is in the two's complement format in multiply-accumulation (signed) operation.

Remark Steps <1> to <12> correspond to <1> to <12> in Figure 18-9.

Figure 18-9. Timing Diagram of Multiply-Accumulation (signed) Operation
(2 × 3 + (–4) = 2 → 32767 × (–1) + (–2147483647) = –2147450882 (Overflow Occurs.))



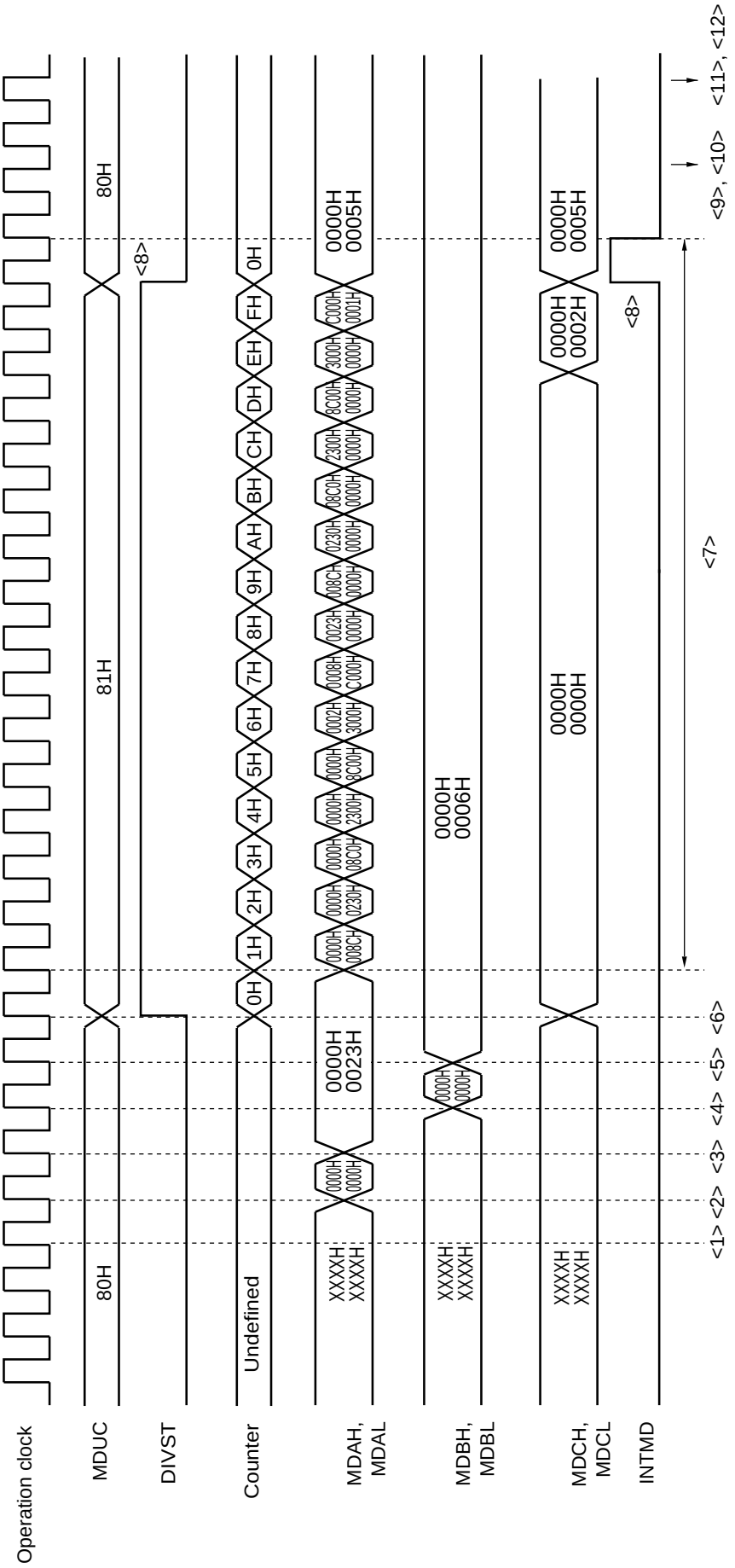
18.4.5 Division operation

- Initial setting
  - <1> Set the multiplication/division control register (MDUC) to 80H.
  - <2> Set the dividend (higher 16 bits) to multiplication/division data register A (H) (MDAH).
  - <3> Set the dividend (lower 16 bits) to multiplication/division data register A (L) (MDAL).
  - <4> Set the divisor (higher 16 bits) to multiplication/division data register B (H) (MDBH).
  - <5> Set the divisor (lower 16 bits) to multiplication/division data register B (L) (MDBL).
  - <6> Set bit 0 (DIVST) of the MDUC register to 1.
    (There is no preference in the order of executing steps <2> to <5>.)
- During operation processing
  - <7> The operation will end when one of the following processing is completed.
    - A wait of at least 16 clocks (The operation will end when 16 clocks have been issued.)
    - A check whether the DIVST bit has been cleared

    (The read values of the MDBL, MDBH, MDCL, and MDCH registers during operation processing are not guaranteed.)
- Operation end
  - <8> The DIVST bit is cleared and the operation ends. At this time, an interrupt request signal (INTMD) is generated if the operation was performed with MACMODE = 0.
  - <9> Read the quotient (lower 16 bits) from the MDAL register.
  - <10> Read the quotient (higher 16 bits) from the MDAH register.
  - <11> Read the remainder (lower 16 bits) from multiplication/division data register C (L) (MDCL).
  - <12> Read the remainder (higher 16 bits) from multiplication/division data register C (H) (MDCH).
    (There is no preference in the order of executing 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 18-10.

Figure 18-10. Timing Diagram of Division Operation (Example: 35 ÷ 6 = 5, Remainder 5)



CHAPTER 19 DMA CONTROLLER

The RL78/I1A has an internal DMA (Direct Memory Access) controller.

Data can be automatically transferred between the peripheral hardware supporting DMA, SFRs, and internal RAM without via CPU.

As a result, the normal internal operation of the CPU and data transfer can be executed in parallel with transfer between the SFR and internal RAM, and therefore, a large capacity of data can be processed. In addition, real-time control using communication, timer, and A/D can also be realized.

19.1 Functions of DMA Controller

- O Number of DMA channels: 2 channels
- O Transfer unit: 8 or 16 bits
- O Maximum transfer unit: 1024 times
- O Transfer type: 2-cycle transfer (One transfer is processed in 2 clocks and the CPU stops during that processing.)
- O Transfer mode: Single-transfer mode
- O Transfer request: Selectable from the following peripheral hardware interrupts
  - A/D converter
  - Serial interface
    (CSI00, UART0, UART1)
  - Timer (channel 0, 1, 2, 3)
- O Transfer target: Between SFR and internal RAM

Here are examples of functions using DMA.

- Successive transfer of serial interface
- Consecutive capturing of A/D conversion results
- Capturing port value at fixed interval

19.2 Configuration of DMA Controller

The DMA controller includes the following hardware.

Table 19-1. Configuration of DMA Controller

Item	Configuration
Address registers	• DMA SFR address registers 0, 1 (DSA0, DSA1) • DMA RAM address registers 0, 1 (DRA0, DRA1)
Count register	• DMA byte count registers 0, 1 (DBC0, DBC1)
Control registers	• DMA mode control registers 0, 1 (DMC0, DMC1) • DMA operation control register 0, 1 (DRC0, DRC1)

19.2.1 DMA SFR address register n (DSAn)

This is an 8-bit register that is used to set an SFR address that is the transfer source or destination of DMA channel n.

Set the lower 8 bits of the SFR addresses FFF00H to FFFFFH.

This register is not automatically incremented but fixed to a specific value.

In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

The DSAn register can be read or written in 8-bit units. However, it cannot be written during DMA transfer.

Reset signal generation clears this register to 00H.

Figure 19-1. Format of DMA SFR Address Register n (DSAn)

Address: FFFB0H (DSA0), FFFB1H (DSA1) After reset: 00H R/W

	7	6	5	4	3	2	1	0
DSAn								

Remark n: DMA channel number (n = 0, 1)

19.2.2 DMA RAM address register n (DRAn)

This is a 16-bit register that is used to set a RAM address that is the transfer source or destination of DMA channel n.

Addresses of the internal RAM area other than the general-purpose registers (see table 19-2) can be set to this register.

Set the lower 16 bits of the RAM address.

This register is automatically incremented when DMA transfer has been started. It is incremented by +1 in the 8-bit transfer mode and by +2 in the 16-bit transfer mode. DMA transfer is started from the address set to this DRAn register. When the data of the last address has been transferred, the DRAn register stops with the value of the last address +1 in the 8-bit transfer mode, and the last address +2 in the 16-bit transfer mode.

In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

The DRAn register can be read or written in 8-bit or 16-bit units. However, it cannot be written during DMA transfer.

Reset signal generation clears this register to 0000H.

Figure 19-2. Format of DMA RAM Address Register n (DRAn)

Address: FFFB2H, FFFB3H (DRA0), FFFB4H, FFFB5H (DRA1) , After reset: 0000H R/W

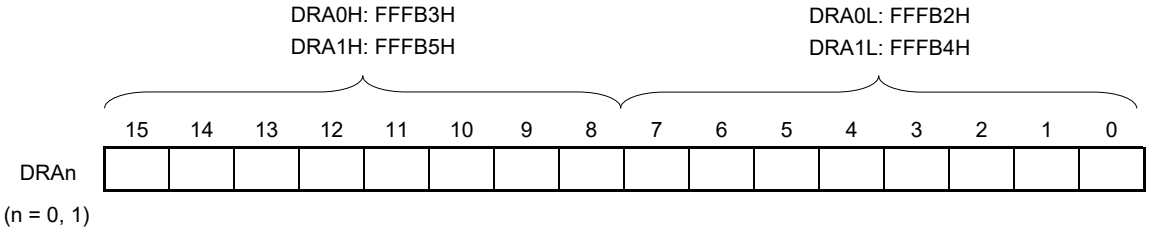


Table 19-2 Internal RAM Area Other than the General-purpose Registers

Part Number	Internal RAM Area other than the General-purpose Registers
R5F1076C, R5F107AC	FF700H to FFEDFH
R5F107AE, R5F107DE	FEF00H to FFEDFH

Remark n: DMA channel number (n = 0, 1)

19.2.3 DMA byte count register n (DBCn)

This is a 10-bit register that is used to set the number of times DMA channel n executes transfer. Be sure to set the number of times of transfer to this DBCn register before executing DMA transfer (up to 1024 times).

Each time DMA transfer has been executed, this register is automatically decremented. By reading this DBCn register during DMA transfer, the remaining number of times of transfer can be learned.

The DBCn register can be read or written in 8-bit or 16-bit units. However, it cannot be written during DMA transfer.

Reset signal generation clears this register to 0000H.

Figure 19-3. Format of DMA Byte Count Register n (DBCn)

Address: FFFB6H, FFFB7H (DBC0), FFFB8H, FFFB9H (DBC1) After reset: 0000H R/W

DBC0H: FFFB7H
DBC1H: FFFB9H

DBC0L: FFFB6H
DBC1L: FFFB8H

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

(n = 0, 1)

DBCn[9:0]	Number of Times of Transfer (When DBCn is Written)	Remaining Number of Times of Transfer (When DBCn is Read)
000H	1024	Completion of transfer or waiting for 1024 times of DMA transfer
001H	1	Waiting for remaining one time of DMA transfer
002H	2	Waiting for remaining two times of DMA transfer
003H	3	Waiting for remaining three times of DMA transfer
•	•	•
•	•	•
•	•	•
3FEH	1022	Waiting for remaining 1022 times of DMA transfer
3FFH	1023	Waiting for remaining 1023 times of DMA transfer

Cautions 1. Be sure to clear bits 15 to 10 to "0".

2. If the general-purpose register is specified or the internal RAM space is exceeded as a result of continuous transfer, the general-purpose register or SFR space are written or read, resulting in loss of data in these spaces. Be sure to set the number of times of transfer that is within the internal RAM space.

Remark n: DMA channel number (n = 0, 1)

19.3 Registers Controlling DMA Controller

DMA controller is controlled by the following registers.

- DMA mode control register n (DMCn)
- DMA operation control register n (DRCn)

Remark n: DMA channel number (n = 0, 1)

19.3.1 DMA mode control register n (DMCn)

The DMCn register is a register that is used to set a transfer mode of DMA channel n. It is used to select a transfer direction, data size, setting of pending, and start source. Bit 7 (STGn) is a software trigger that starts DMA.

Rewriting bits 6, 5, and 3 to 0 of the DMCn register is prohibited during operation (when DSTn = 1).

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

Reset signal generation clears this register to 00H.

Figure 19-4. Format of DMA Mode Control Register n (DMCn) (1/2)

Address: FFFBAH (DMC0), FFFBBH (DMC1) After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

STGn[Note 1]	DMA transfer start software trigger
0	No trigger operation
1	DMA transfer is started when DMA operation is enabled (DENn = 1).
DMA transfer is performed once by writing 1 to the STGn bit when DMA operation is enabled (DENn = 1). When this bit is read, 0 is always read.	

DRSn	Selection of DMA transfer direction
0	SFR to internal RAM
1	Internal RAM to SFR

DSn	Specification of transfer data size for DMA transfer
0	8 bits
1	16 bits

DWAITn Note 2	Pending of DMA transfer
0	Executes DMA transfer upon DMA start request (not held pending).
1	Holds DMA start request pending if any.
DMA transfer that has been held pending can be started by clearing the value of the DWAITn bit to 0. It takes 2 clocks to actually hold DMA transfer pending when the value of the DWAITn bit is set to 1.	

Notes 1. The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 bits values.

2. When DMA transfer is held pending while using both DMA channels, be sure to hold the DMA transfer pending for both channels (by setting the DWAIT0 and DWAIT1 bits to 1).

Remark n: DMA channel number (n = 0, 1)

Figure 19-4. Format of DMA Mode Control Register n (DMCn) (2/2)

Address: FFFBAH (DMC0), FFFBBH (DMC1) After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

(When n = 0 or 1)

IFCn 3	IFCn 2	IFCn 1	IFCn 0	Selection of DMA start source[Note]	
				Trigger signal	Trigger contents
0	0	0	0	–	Disables DMA transfer by interrupt. (Only software trigger is enabled.)
0	0	0	1	INTAD	A/D conversion end interrupt
0	0	1	0	INTTM00	End of timer channel 0 count or capture end interrupt
0	0	1	1	INTTM01	End of timer channel 1 count or capture end interrupt
0	1	0	0	INTTM02	End of timer channel 2 count or capture end interrupt
0	1	0	1	INTTM03	End of timer channel 3 count or capture end interrupt
0	1	1	0	INTST0/INTCSI00	UART0 transmission transfer end or buffer empty interrupt/CSI00 transfer end or buffer empty interrupt
0	1	1	1	INTSR0	UART0 reception transfer end interrupt
1	0	0	0	INTST1	UART1 transmission transfer end or buffer empty interrupt
1	0	0	1	INTSR1	UART1 reception transfer end interrupt
Other than above				Setting prohibited	

Note The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 bits values.

Remark n: DMA channel number (n = 0, 1)

19.3.2 DMA operation control register n (DRCn)

The DRCn register is a register that is used to enable or disable transfer of DMA channel n.

Rewriting bit 7 (DENn) of this register is prohibited during operation (when DSTn = 1).

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

Reset signal generation clears this register to 00H.

Figure 19-5. Format of DMA Operation Control Register n (DRCn)

Address: FFFBCH (DRC0), FFFBDH (DRC1) After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
DRCn	DENn	0	0	0	0	0	0	DSTn

DENn	DMA operation enable flag
0	Disables operation of DMA channel n (stops operating cock of DMA).
1	Enables operation of DMA channel n.
DMAC waits for a DMA trigger when DSTn = 1 after DMA operation is enabled (DENn = 1).	

DSTn	DMA transfer mode flag
0	DMA transfer of DMA channel n is completed.
1	DMA transfer of DMA channel n is not completed (still under execution).
DMAC waits for a DMA trigger when DSTn = 1 after DMA operation is enabled (DENn = 1). When a software trigger (STGn) or the start source trigger set by the IFCn3 to IFCn0 bits is input, DMA transfer is started. When DMA transfer is completed after that, this bit is automatically cleared to 0. Write 0 to this bit to forcibly terminate DMA transfer under execution.	

Caution The DSTn flag is automatically cleared to 0 when a DMA transfer is completed. Writing the DENn flag is enabled only when DSTn = 0. When a DMA transfer is terminated without waiting for generation of the interrupt (INTDMAn) of DMAn, therefore, set the DSTn bit to 0 and then the DENn bit to 0 (for details, see 19.5.5 Forced termination by software).

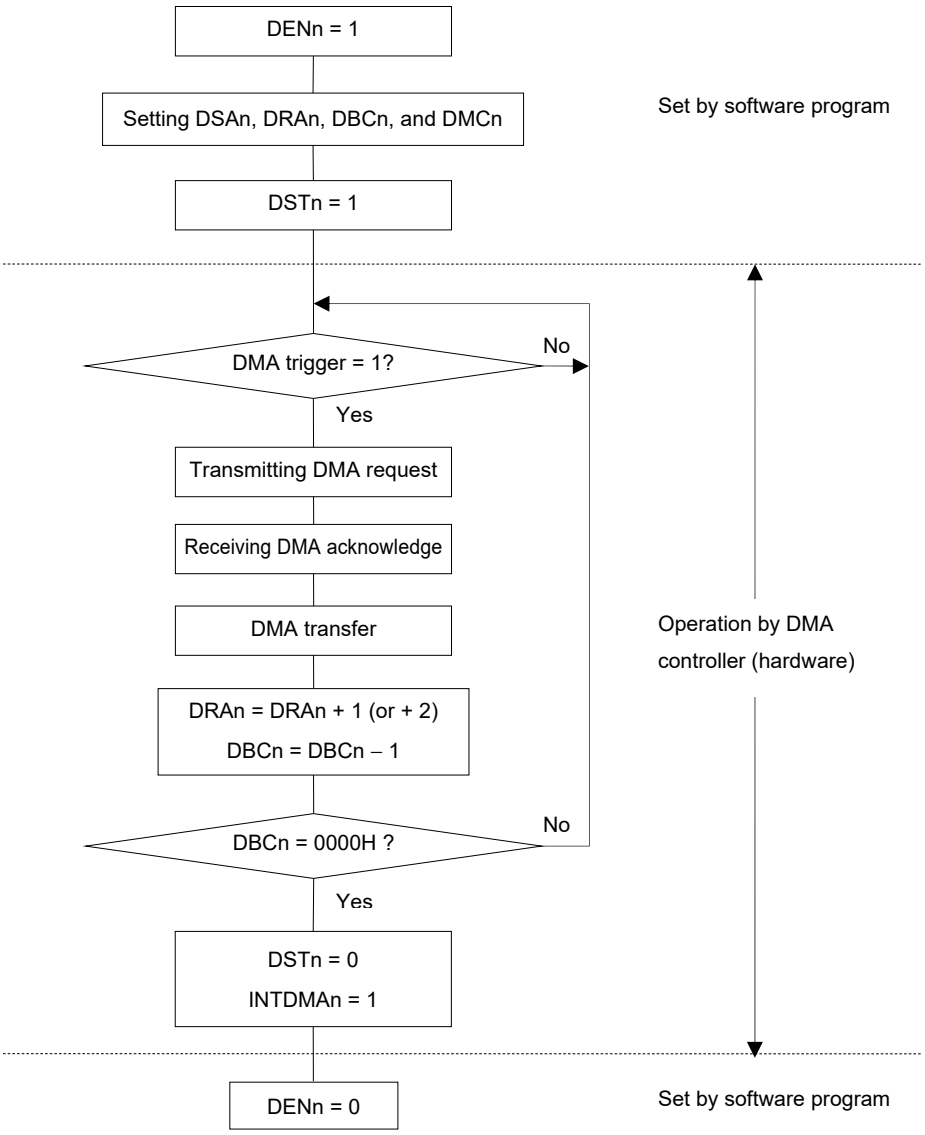
Remark n: DMA channel number (n = 0, 1)

19.4 Operation of DMA Controller

19.4.1 Operation procedure

<1> The DMA controller is enabled to operate when DENn = 1. Before writing the other registers, be sure to set the DENn bit to 1. Use 80H to write with an 8-bit manipulation instruction.
<2> Set an SFR address, a RAM address, the number of times of transfer, and a transfer mode of DMA transfer to DMA SFR address register n (DSAn), DMA RAM address register n (DRAn), DMA byte count register n (DBCn), and DMA mode control register n (DMCn).
<3> The DMA controller waits for a DMA trigger when DSTn = 1. Use 81H to write with an 8-bit manipulation instruction.
<4> When a software trigger (STGn) or a start source trigger specified by the IFCn3 to IFCn0 bits is input, a DMA transfer is started.
<5> Transfer is completed when the number of times of transfer set by the DBCn register reaches 0, and transfer is automatically terminated by occurrence of an interrupt (INTDMAn).
<6> Stop the operation of the DMA controller by clearing the DENn bit to 0 when the DMA controller is not used.

Figure 19-6. Operation Procedure



Remark n: DMA channel number (n = 0, 1)

19.4.2 Transfer mode

The following four modes can be selected for DMA transfer by using bits 6 and 5 (DRSn and DSn) of DMA mode control register n (DMCn).

DRSn	DSn	DMA Transfer Mode
0	0	Transfer from SFR of 1-byte data (fixed address) to RAM (address is incremented by +1)
0	1	Transfer from SFR of 2-byte data (fixed address) to RAM (address is incremented by +2)
1	0	Transfer from RAM of 1-byte data (address is incremented by +1) to SFR (fixed address)
1	1	Transfer from RAM of 2-byte data (address is incremented by +2) to SFR (fixed address)

By using these transfer modes, up to 1024 bytes of data can be consecutively transferred by using the serial interface, data resulting from A/D conversion can be consecutively transferred, and port data can be scanned at fixed time intervals by using a timer.

19.4.3 Termination of DMA transfer

When DBCn = 00H and DMA transfer is completed, the DSTn bit is automatically cleared to 0. An interrupt request (INTDMAn) is generated and transfer is terminated.

When the DSTn bit is cleared to 0 to forcibly terminate DMA transfer, DMA byte count register n (DBCn) and DMA RAM address register n (DRAn) hold the value when transfer is terminated.

The interrupt request (INTDMAn) is not generated if transfer is forcibly terminated.

Remark n: DMA channel number (n = 0, 1)

19.5 Example of Setting of DMA Controller

19.5.1 Simplified SPI (CSI) consecutive transmission

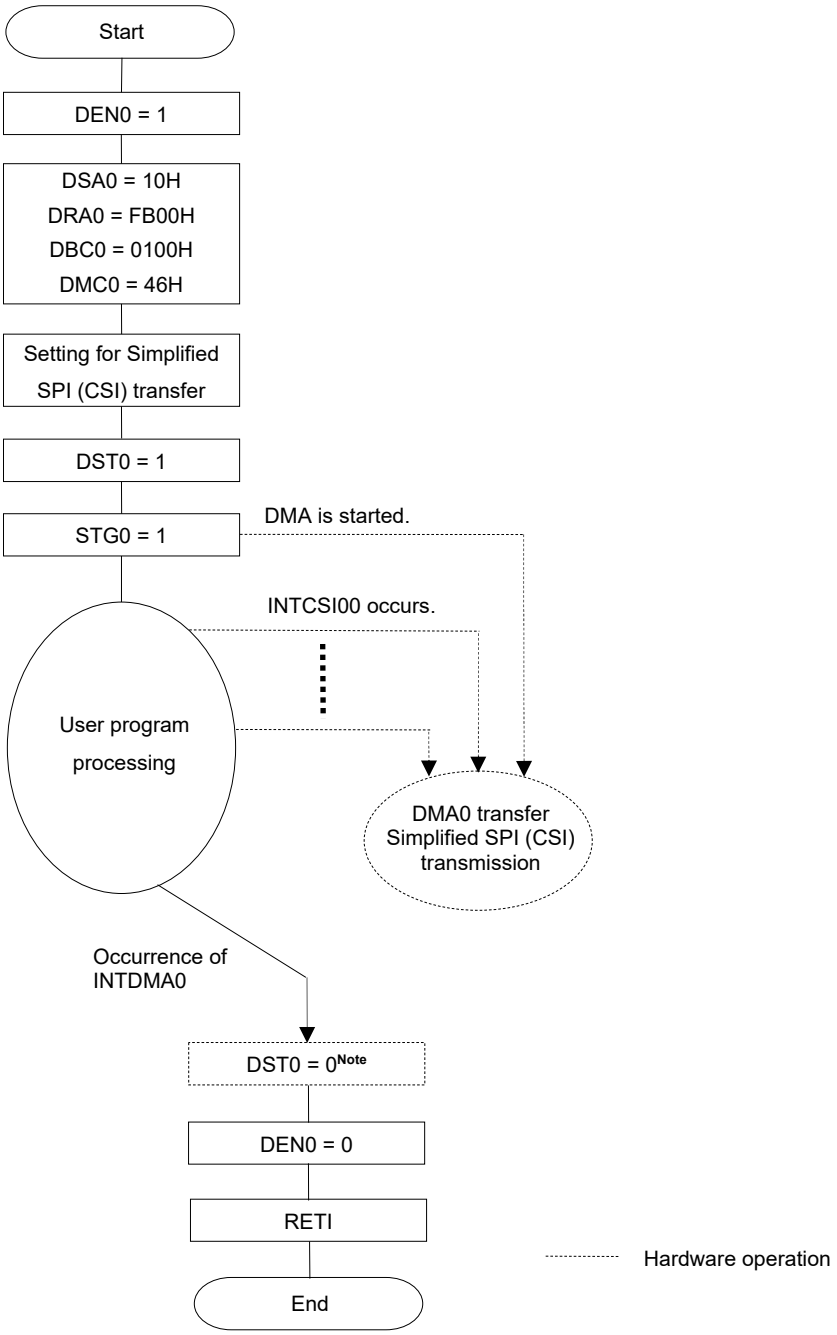
A flowchart showing an example of setting for Simplified SPI (CSI) consecutive transmission is shown below.

- Consecutive transmission of CSI00 (256 bytes)
- DMA channel 0 is used for DMA transfer.
- DMA start source: INTCSI00 (software trigger (STG0) only for the first start source)
- Interrupt of CSI00 is specified by IFC03 to IFC00 = 0110B.
- Transfers FFB00H to FFBFFH (256 bytes) of RAM to FFF10H of the data register (SIO00) of Simplified SPI (CSI).

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.

Remark IFC03 to IFC00: Bits 3 to 0 of DMA mode control registers 0 (DMC0)

Figure 19-7. Example of Setting for Simplified SPI (CSI) Consecutive Transmission



Note The DST0 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN0 flag is enabled only when DST0 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA0 (INTDMA0), set the DST0 bit to 0 and then the DEN0 bit to 0 (for details, see **19.5.5 Forced termination by software**).

The fist trigger for consecutive transmission is not started by the interrupt of Simplified SPI (CSI). In this example, it start by a software trigger.

Simplified SPI (CSI) transmission of the second time and onward is automatically executed.

A DMA interrupt (INTDMA0) occurs when the last transmit data has been written to the data register.

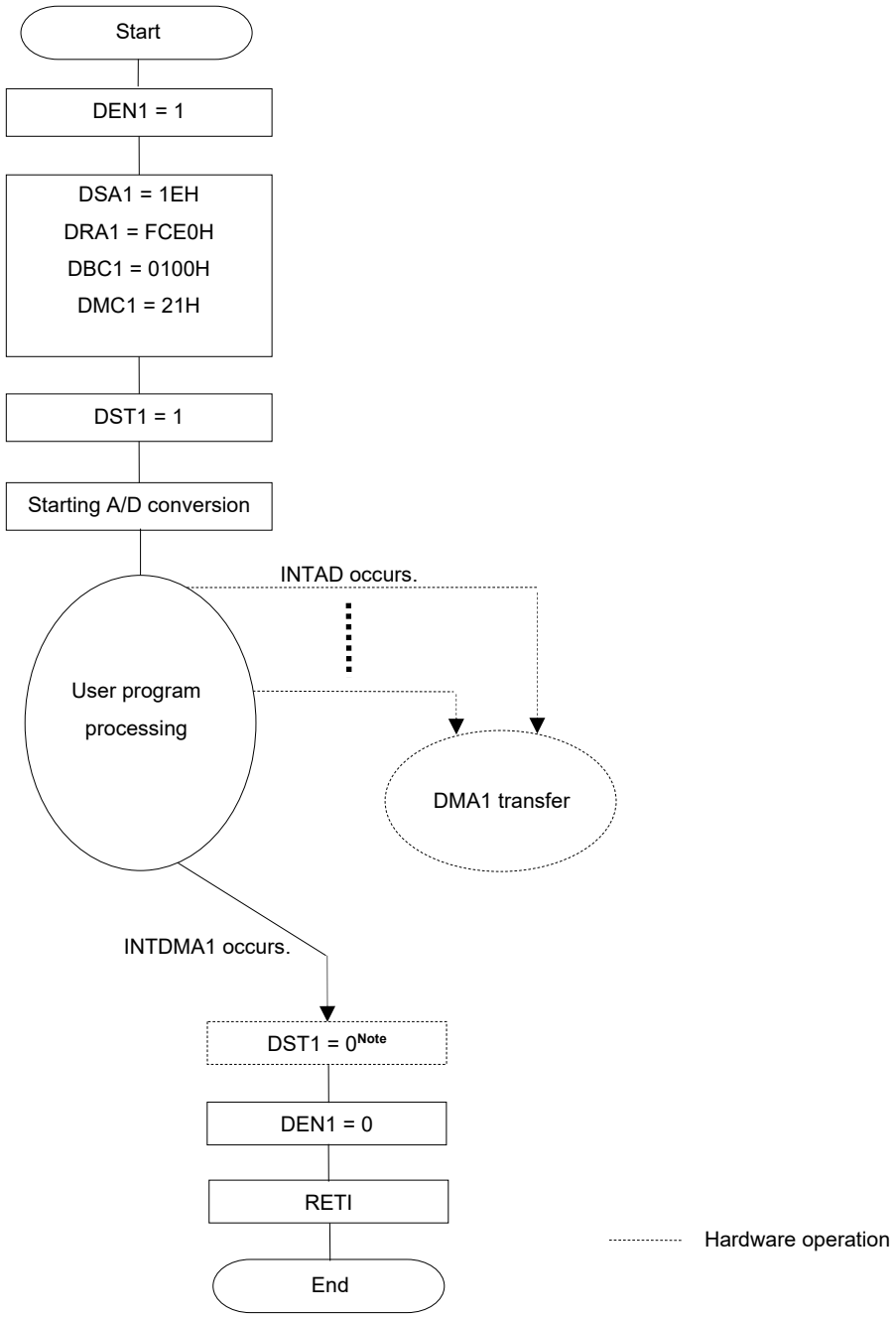
19.5.2 Consecutive capturing of A/D conversion results

A flowchart of an example of setting for consecutively capturing A/D conversion results is shown below.

- Consecutive capturing of A/D conversion results.
- DMA channel 1 is used for DMA transfer.
- DMA start source: INTAD
- Interrupt of A/D is specified by IFC13 to IFC10 = 0001B.
- Transfers FFF1EH and FFF1FH (2 bytes) of the 10-bit A/D conversion result register (ADCR) to 512 bytes of FFCE0H to FFEDFH of RAM.

Remark IFC13 to IFC10: Bits 3 to 0 of DMA mode control registers 1 (DMC1)

Figure 19-8. Example of Setting of Consecutively Capturing A/D Conversion Results



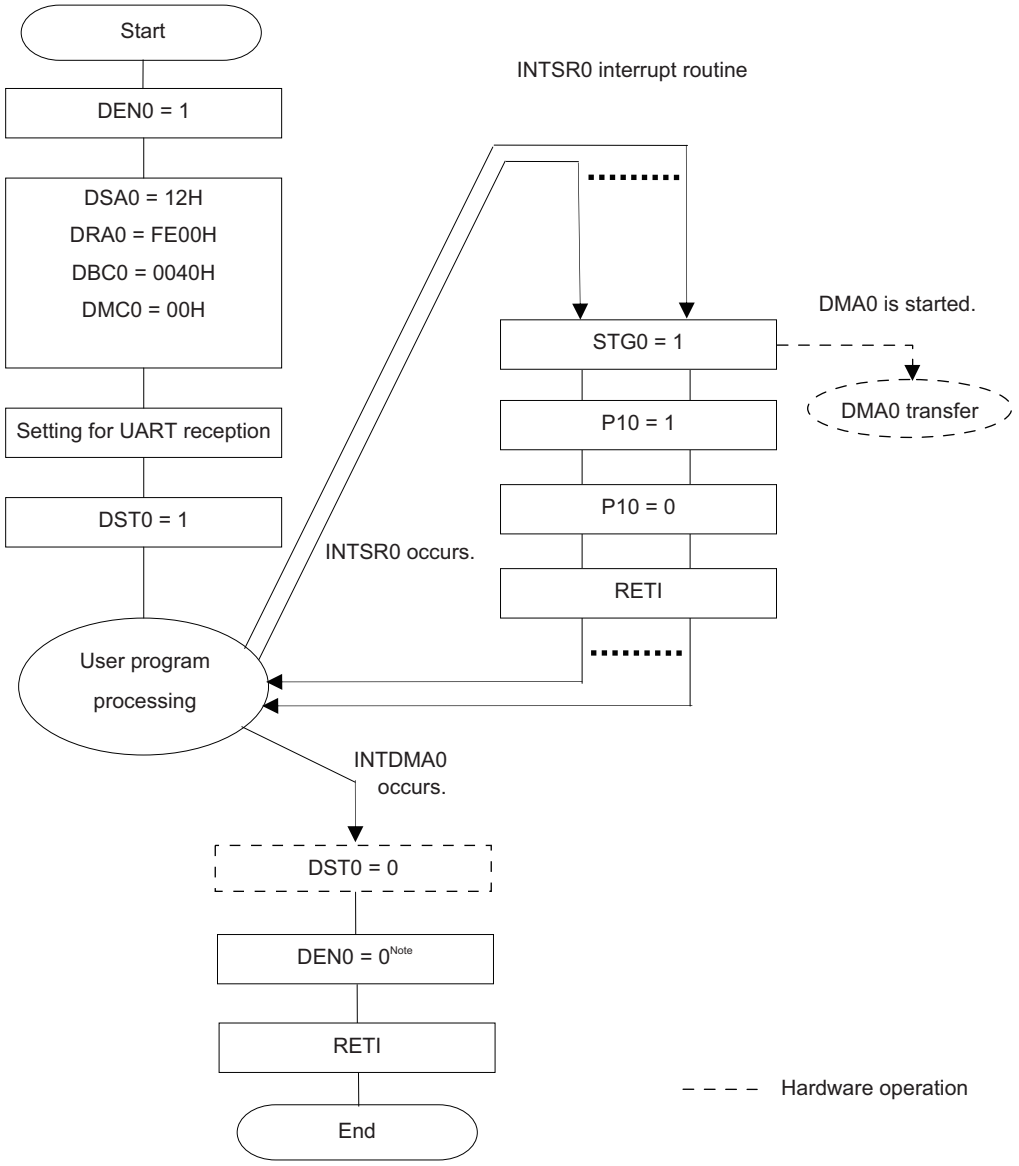
Note The DST1 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN1 flag is enabled only when DST1 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA1 (INTDMA1), set the DST1 bit to 0 and then the DEN1 bit to 0 (for details, see **19.5.5 Forced termination by software**).

19.5.3 UART consecutive reception + ACK transmission

A flowchart illustrating an example of setting for UART consecutive reception + ACK transmission is shown below.

- Consecutively receives data from UART0 and outputs ACK to P10 on completion of reception.
- DMA channel 0 is used for DMA transfer.
- DMA start source: Software trigger (DMA transfer on occurrence of an interrupt is disabled.)
- Transfers FFF12H of UART receive data register 0 (RXD0) to 64 bytes of FFE00H to FFE3FH of RAM.

Figure 19-9. Example of Setting for UART Consecutive Reception + ACK Transmission



Note The DST0 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN0 flag is enabled only when DST0 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA0 (INTDMA0), set the DST0 bit to 0 and then the DEN0 bit to 0 (for details, see **19.5.5 Forced termination by software**).

Remark This is an example where a software trigger is used as a DMA start source.
If ACK is not transmitted and if only data is consecutively received from UART, the UART reception end interrupt (INTSR0) can be used to start DMA for data reception.

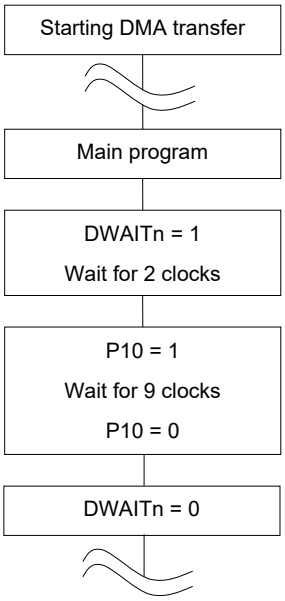
19.5.4 Holding DMA transfer pending by DWAITn bit

When DMA transfer is started, transfer is performed while an instruction is executed. At this time, the operation of the CPU is stopped and delayed for the duration of 2 clocks. If this poses a problem to the operation of the set system, a DMA transfer can be held pending by setting the DWAITn bit to 1. The DMA transfer for a transfer trigger that occurred while DMA transfer was held pending is executed after the pending status is canceled. However, because only one transfer trigger can be held pending for each channel, even if multiple transfer triggers occur for one channel during the pending status, only one DMA transfer is executed after the pending status is canceled.

To output a pulse with a width of 10 clocks of the operating frequency from the P10 pin, for example, the clock width increases to 12 if a DMA transfer is started midway. In this case, the DMA transfer can be held pending by setting the DWAITn bit to 1.

After setting the DWAITn bit to 1, it takes two clocks until a DMA transfer is held pending.

Figure 19-10. Example of Setting for Holding DMA Transfer Pending by DWAITn Bit



Caution When DMA transfer is held pending while using both DMA channels, be sure to held the DMA transfer pending for both channels (by setting DWAIT0 and DWAIT1 to 1). If the DMA transfer of one channel is executed while that of the other channel is held pending, DMA transfer might not be held pending for the latter channel.

Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

19.5.5 Forced termination by software

After the DSTn bit is set to 0 by software, it takes up to 2 clocks until a DMA transfer is actually stopped and the DSTn bit is set to 0. To forcibly terminate a DMA transfer by software without waiting for occurrence of the interrupt (INTDMAn) of DMAn, therefore, perform either of the following processes.

<When using one DMA channel>

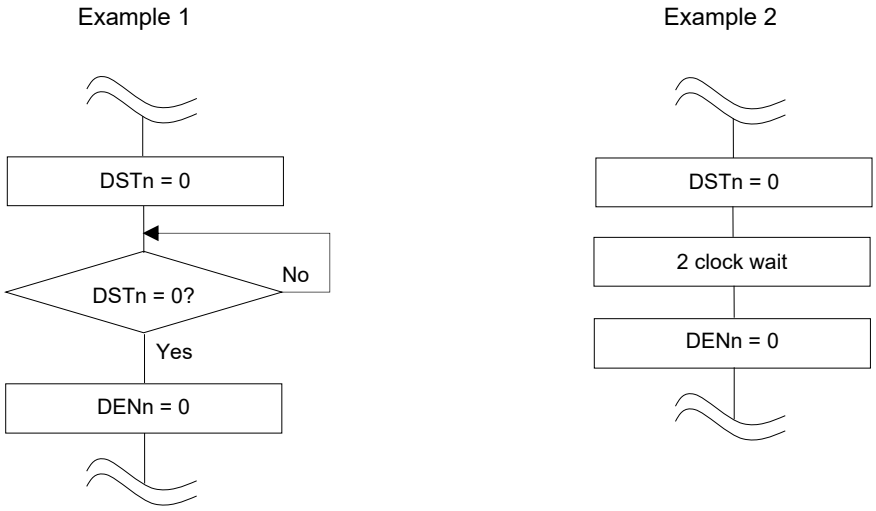
- Set the DSTn bit to 0 (use DRCn = 80H to write with an 8-bit manipulation instruction) by software, confirm by polling that the DSTn bit has actually been cleared to 0, and then set the DENn bit to 0 (use DRCn = 00H to write with an 8-bit manipulation instruction).

- Set the DSTn bit to 0 (use DRCn = 80H to write with an 8-bit manipulation instruction) by software and then set the DENn bit to 0 (use DRCn = 00H to write with an 8-bit manipulation instruction) two or more clocks after.

<When using both DMA channels>

- To forcibly terminate DMA transfer by software when using both DMA channels (by setting DSTn to 0), clear the DSTn bit to 0 after the DMA transfer is held pending by setting the DWAIT0 and DWAIT1 bits of both channels to 1. Next, clear the DWAIT0 and DWAIT1 bits of both channels to 0 to cancel the pending status, and then clear the DENn bit to 0.

Figure 19-11. Forced Termination of DMA Transfer (1/2)



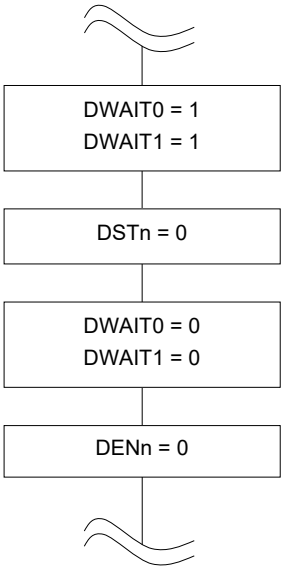
Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

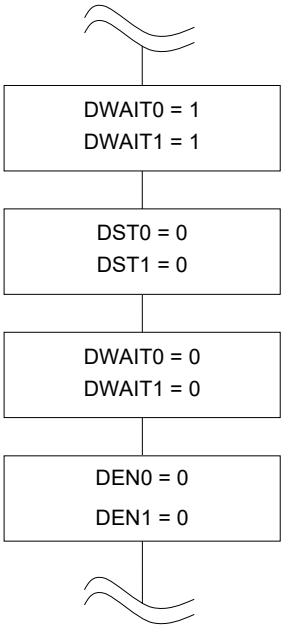
Figure 19-11. Forced Termination of DMA Transfer (2/2)

Example 3

- Procedure for forcibly terminating the DMA transfer for one channel if both channels are used



- Procedure for forcibly terminating the DMA transfer for both channels if both channels are used



Caution In example 3, the system is not required to wait two clock cycles after the DWAITn bit is set to 1. In addition, the system does not have to wait two clock cycles after clearing the DSTn bit to 0, because more than two clock cycles elapse from when the DSTn bit is cleared to 0 to when the DENn bit is cleared to 0.

Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

19.6 Cautions on Using DMA Controller

(1) Priority of DMA

During DMA transfer, a request from the other DMA channel is held pending even if generated. The pending DMA transfer is started after the ongoing DMA transfer is completed. If two or more DMA requests are generated at the same time, however, DMA channel 0 takes priority over DMA channel 1.

If a DMA request and an interrupt request are generated at the same time, the DMA transfer takes precedence, and then interrupt servicing is executed.

(2) DMA response time

The response time of DMA transfer is as follows.

Table 19-3. Response Time of DMA Transfer

	Minimum Time	Maximum Time
Response time	3 clocks	10 clocks[Note]

Note The maximum time necessary to execute an instruction from internal RAM is 16 clock cycles.

Cautions 1. The above response time does not include the two clock cycles required for a DMA transfer.

2. When executing a DMA pending instruction (see 19.6 (4)), the maximum response time is extended by the execution time of that instruction to be held pending.

3. Do not specify successive transfer triggers for a channel within a period equal to the maximum response time plus one clock cycle, because they might be ignored.

Remark 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

(3) Operation in standby mode

The DMA controller operates as follows in the standby mode.

Table 19-4. DMA Operation in Standby Mode

Status	DMA Operation
HALT mode	Normal operation
STOP mode	Stops operation. If DMA transfer and STOP instruction execution contend, DMA transfer may be damaged. Therefore, stop DMA before executing the STOP instruction.

(4) DMA pending instruction

Even if a DMA request is generated, DMA transfer is held pending immediately after the following instructions.

- CALL !addr16
- CALL $!addr20
- CALL !!addr20
- CALL rp
- CALLT [addr5]
- BRK
- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- POP PSW
- BTCLR PSW. bit, $addr20
- EI
- DI
- Write instructions for registers IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H each.
- An instruction that accesses a register placed in the 2nd SFR address range from F0500H to F06FFH
- Instruction for accessing the data flash memory

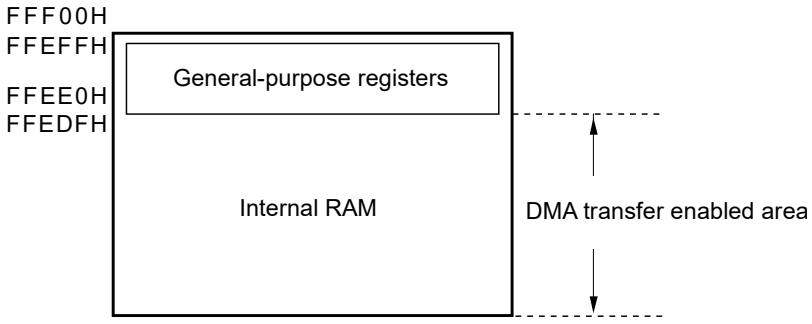
(5) Operation if address in general-purpose register area or other than those of internal RAM area is specified

The address indicated by DMA RAM address register n (DRAn) is incremented during DMA transfer. If the address is incremented to an address in the general-purpose register area or exceeds the area of the internal RAM, the following operation is performed.

- In mode of transfer from SFR to RAM
  The data of that address is lost.

- In mode of transfer from RAM to SFR
  Undefined data is transferred to SFR.

In either case, malfunctioning may occur or damage may be done to the system. Therefore, make sure that the address is within the internal RAM area other than the general-purpose register area.



(6) Access to a register placed in the 2nd SFR address range from F0500H to F06FFH

If the above register is accessed one instruction after a DMA transfer occurs, a one-clock-cycle wait is inserted.

(7) Operation if instructions for accessing the data flash area

If the data flash area is accessed after an next instruction execution from start of DMA transfer, a 3-clock wait will be inserted to the next instruction.

```
Instruction 1
DMA transfer
Instruction 2  ◄– The wait of three clock cycles occurs.
MOV A,         ! DataFlash area
```

CHAPTER 20 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	30-pin	38-pin
Maskable interrupts	External	7	10	11
	Internal	27	30	30

20.1 Interrupt Function Types

The following two types of interrupt functions are used.

(1) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into four priority groups by setting the priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H).

Multiple interrupt servicing can be applied to low-priority interrupts when high-priority interrupts are generated. If two or more interrupt requests, each having the same priority, are simultaneously generated, then they are processed according to the default priority of vectored interrupt servicing. Default priority, see **Table 20-1**.

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.

20.2 Interrupt Sources and Configuration

Interrupt sources include maskable interrupts and software interrupts. In addition, they also have up to seven reset sources (see **Table 20-1**). The vector codes that store the program start address when branching due to the generation of a reset or various interrupt requests are two bytes each, so interrupts jump to a 64 K address of 00000H to 0FFFFH.

Table 20-1. Interrupt Source List (1/3)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]	38-pin	30-pin	20-pin
		Name	Trigger						
Maskable	0	INTWDTI	Watchdog timer interval[Note 3] (75% of overflow time+1/2f_{IL})	Internal	0004H	(A)	√	√	√
	1	INTLVI	Voltage detection[Note 4]		0006H		√	√	√
	2	INTP0	Pin input edge detection	External	0008H	(B)	√	√	√
	3	INTP3			000EH		√	–	–
	4	INTP4			0010H		√	√	–
	5	INTDMA0	End of DMA0 transfer	Internal	001AH	(A)	√	√	√
	6	INTDMA1	End of DMA1 transfer		001CH		√	√	√
	7	INTST0	UART0 transmission transfer end or buffer empty interrupt		001EH		√	√	√
		INTCSI00	CSI00 transfer end or buffer empty interrupt				√	–	–
	8	INTSR0	UART0 reception transfer end		0020H		√	√	√
	9	INTSRE0	UART0 reception communication error occurrence		0022H		√	√	√
		INTTM01H	End of timer channel 1 count or capture (at higher 8-bit timer operation)				√	√	√
	10	INTST1	UART1 transmission transfer end or buffer empty interrupt		0024H		√	√	–
	11	INTSR1	UART1 reception transfer end		0026H		√	√	–
	12	INTSRE1	UART1 reception communication error occurrence		0028H		√	√	–
		INTTM03H	End of timer channel 3 count or capture (at higher 8-bit timer operation)				√	√	√
	13	INTIICA0	End of IICA0 communication		002AH		√	√	√
	14	INTTM00	End of timer channel 0 count or capture		002CH		√	√	√
	15	INTTM01	End of timer channel 1 count or capture (at 16-bit/lower 8-bit timer operation)		002EH		√	√	√
	16	INTTM02	End of timer channel 2 count or capture		0030H		√	√	√

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 40 indicates the lowest priority.

2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figure 20-1, respectively.

3. When bit 7 (WDTINT) of the option byte (000C0H) is set to 1.

4. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is cleared to 0.

Table 20-1. Interrupt Source List (2/3)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type Note 2	38-pin	30-pin	20-pin
		Name	Trigger						
Maskable	17	INTTM03	End of timer channel 3 count or capture (at 16-bit/lower 8-bit timer operation)	Internal	0032H	(A)	√	√	√
	18	INTAD	End of A/D conversion		0034H		√	√	√
	19	INTRTC	Fixed-cycle signal of real-time clock/alarm match detection		0036H		√	√	√
	20	INTIT	Interval signal of 12-bit interval timer detection		0038H		√	√	√
	21	INTSTDL4	DALI/UART4 transmission transfer end or buffer empty interrupt		003CH		√	√	√
	22	INTSRDL4	DALI/UART4 reception transfer end		003EH		√	√	√
		INTSREDL4	DALI/UART4 reception communication error occurrence				√	√	√
	23	INTP20	Pin input edge detection	External	0040H	(B)	√	√	√
		INTP22					√	√	√
	24	INTTM04	End of timer channel 4 count or capture	Internal	0042H	(A)	√	√	√
	25	INTTM05	End of timer channel 5 count or capture		0044H		√	√	√
	26	INTTM06	End of timer channel 6 count or capture		0046H		√	√	√
	27	INTTM07	End of timer channel 7 count or capture		0048H		√	√	√
	28	INTCMP0	Comparator 0 edge detection	External	004AH	(B)	√	√	√
	29	INTCMP1 Note 3	Comparator 1 edge detection		004CH		√	√	√
	30	INTCMP2	Comparator 2 edge detection		004EH		√	√	√
	31	INTP9	Pin input edge detection		0050H		√	–	–
		INTCMP3 Note 3	Comparator 3 edge detection				√	√	√
	32	INTP10	Pin input edge detection		0052H		√	–	–
		INTCMP4 Note 3	Comparator 4 edge detection				√	√	–
	33	INTP11	Pin input edge detection		0054H		√	√	–
		INTCMP5 Note 3	Comparator 5 edge detection				√	√	–

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 40 indicates the lowest priority.

2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figure 20-1.

3. INTCMP1, INTCMP3, INTCMP4, and INTCMP5 cannot be used to clear the STOP mode.
About interrupt generation timing, see **14.5 Caution for Using Timer KB Simultaneous Operation Function**.

Table 20-1. Interrupt Source List (3/3)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]	38-pin	30-pin	20-pin
		Name	Trigger						
Maskable	34	INTTMKB0	End of timer KB0 count	Internal	0056H	(A)	√	√	√
	35	INTTMKB1	End of timer KB1 count		0058H		√	√	√
	36	INTTMKB2	End of timer KB2 count		005AH		√	√	–
	37	INTTMKC0	End of timer KC0 count		005CH		√	√	√
	38	INTMD	End of division operation/overflow of multiply-accumulation result occurs		005EH		√	√	√
	39	INTP21	Pin input edge detection	External	0060H	(B)	√	√	√
		INTP23					√	√	–
	40	INTFL	Reserved	Internal	0062H	(A)	√	√	√
Software	–	BRK	Execution of BRK instruction	–	007EH	(C)	√	√	√
Reset	–	RESET	$\overline{\text{RESET}}$ pin input	–	0000H	–	√	√	√
		POR	Power-on-reset				√	√	√
		LVD	Voltage detection[Note3]				√	√	√
		WDT	Overflow of watchdog timer				√	√	√
		TRAP	Execution of illegal instruction [Note 4]				√	√	√
		IAW	Illegal-memory access				√	√	√
		RPE	RAM parity error				√	√	√

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 40 indicates the lowest priority.

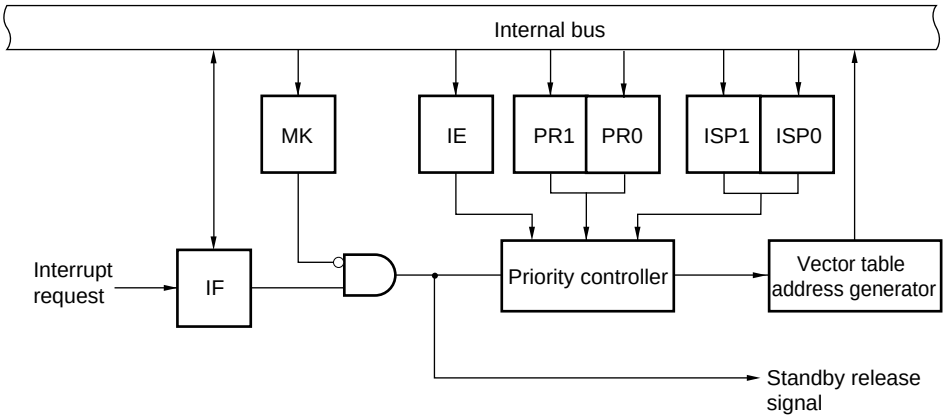
2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figure 20-1, previously.

3. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is set to 1.

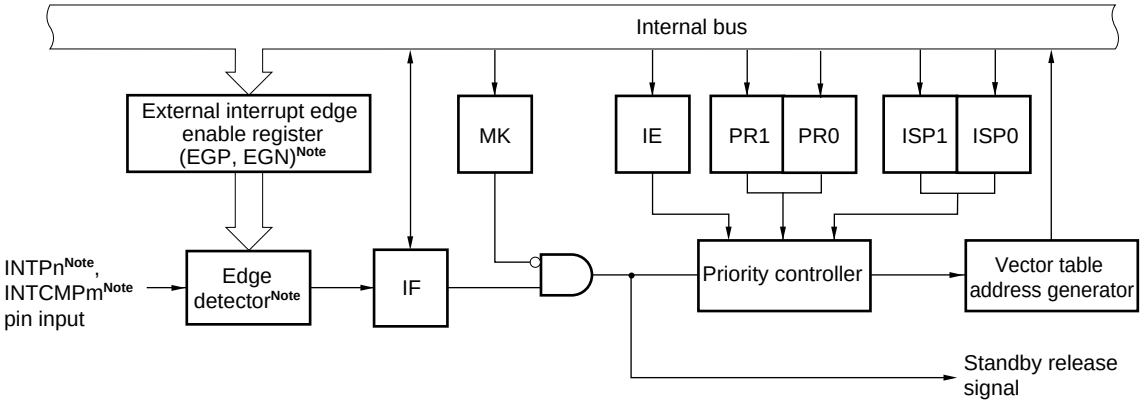
4. When the instruction code in FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Figure 20-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal maskable interrupt



(B) External maskable interrupt (INTPn, INTCMPm)



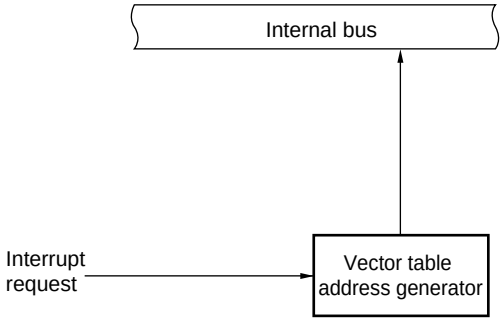
IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

Note According to setting for using of the timer KB simultaneous function (the timer KB forced output stop function and timer restart function), the interrupt signal pass, the interrupt generation timing, and the edge enable register for INTP20, INTP21, and INTCMPm vary. For details, see **14.5 Caution for Using Timer KB Simultaneous Operation Function**.

Remark 20-pin: n = 0, 20, 21, 22, m = 0 to 3
30-pin: n = 0, 4, 11, 20 to 23, m = 0 to 5
38-pin: n = 0, 3, 4, 9 to 11, 20 to 23, m = 0 to 5

Figure 20-1. Basic Configuration of Interrupt Function (2/2)

(C) Software interrupt



IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

20.3 Registers Controlling Interrupt Functions

The following types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)
- Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)
- Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)
- External interrupt rising edge enable registers (EGP0, EGP1, EGP2)
- External interrupt falling edge enable registers (EGN0, EGN1, EGN2)
- Program status word (PSW)
- Interrupt mask flag register 0 (INTMK0)
- Interrupt monitor flag register 0 (INTMF0)

Table 20-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 20-2. Flags Corresponding to Interrupt Request Sources (1/3)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		38-pin	30-pin	20-pin
		Register		Register		Register			
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L	√	√	√
INTLVI	LVIIF		LVIMK		LVIPR0, LVIPR1		√	√	√
INTP0	PIF0		PMK0		PPR00, PPR10		√	√	√
INTP3	PIF3		PMK3		PPR03, PPR13		√	–	–
INTP4	PIF4		PMK4		PPR04, PPR14		√	√	–
INTDMA0	DMAIF0	IF0H	DMAMK0	MK0H	DMAPR00, DMAPR10	PR00H, PR10H	√	√	√
INTDMA1	DMAIF1		DMAMK1		DMAPR01, DMAPR11		√	√	√
INTST0Note 1	STIF0Note 1		STMK0Note 1		STPR00, STPR10Note 1		√	√	√
INTCSI00Note 1	CSIIF00Note 1		CSIMK00Note 1		CSIPR000, CSIPR100Note 1		√	–	–
INTSR0	SRIF0		SRMK0		SRPR00, SRPR10		√	√	√
INTSRE0Note 2	SREIF0Note 2		SREMK0Note 2		SREPR00, SREPR10Note 2		√	√	√
INTTM01HNote 2	TMIF01HNote 2		TMMK01HNote 2		TMPR001H, TMPR101HNote 2		√	√	√

Notes 1. If one of the interrupt sources INTST0 and INTCSI00 is generated, bit 5 of the IF0H register is set to 1. Bit 5 of the MK0H, PR00H, and PR10H registers supports these two interrupt sources.

2. Do not use a UART0 reception error interrupt and an interrupt of channel 1 of TAU0 (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. If the UART0 reception error interrupt is not used (EOC01 = 0), UART0 and channel 1 of TAU0 (at higher 8-bit timer operation) can be used at the same time. If one 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.

Table 20-2. Flags Corresponding to Interrupt Request Sources (2/3)

Interrupt Source	Interrupt Request Flag	Register	Interrupt Mask Flag	Register	Priority Specification Flag	Register	38-pin	30-pin	20-pin
INTST1	STIF1	IF1L	STMK1	MK1L	STPR01, STPR11	PR01L, PR11L	√	√	–
INTSR1	SRIF1		SRMK1		SRPR01, SRPR11		√	√	–
INTSRE1 Note 1	SREIF1 Note 1		SREMK1 Note 1		SREPR01, SREPR11 Note 1		√	√	–
INTTM03H Note 1	TMIF03H Note 1		TMMK03H Note 1		TMPR003H, TMPR103H Note 1		√	√	√
INTIICA0	IICAIF0		IICAMK0		IICAPR00, IICAPR10		√	√	√
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100		√	√	√
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101		√	√	√
INTTM02	TMIF02		TMMK02		TMPR002, TMPR102		√	√	√
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103		√	√	√
INTAD	ADIF	IF1H	ADMK	MK1H	ADPR0, ADPR1	PR01H, PR11H	√	√	√
INTRTC	RTCIF		RTCMK		RTCPR0, RTCPR1		√	√	√
INTIT	ITIF		ITMK		ITPR0, ITPR1		√	√	√
INTSTDL4	STDLIF4		STDLMK4		STDLPR04, STDLPR14		√	√	√
INTSRDL4 Note 2	SRDLIF4 Note 2		SRDLMK4 Note 2		SRDLPR04, SRDLPR14 Note 2		√	√	√
INTSREDL4 Note 2	SREDLIF4 Note 2		SREDLMK4 Note 2		SREDLPR04, SREDLPR14 Note 2		√	√	√
INTP20 Note 3	PIF20 Note 3		PMK20 Note 3		PPR020, PPR120 Note 3		√	√	√
INTP22 Note 3	PIF22 Note 3		PMK22 Note 3		PPR022, PPR122 Note 3		√	√	√
INTTM04	TMIF04		TMMK04		TMPR004, TMPR104		√	√	√

Notes 1. Do not use a UART1 reception error interrupt and an interrupt of channel 3 of TAU0 (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. If the UART1 reception error interrupt is not used (EOC03 = 0), UART1 and channel 3 of TAU0 (at higher 8-bit timer operation) can be used at the same time. If one 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.

2. INTSRDL4 (transfer end interrupt) and INTSREDL4 (error interrupt) of DALI/UART4 share flags for the interrupt request sources. If one of the interrupt sources INTSRDL4 and INTSREDL4 is generated, bit 5 of the IF1H register is set to 1. Bit 5 of the MK1H, PR01H, and PR11H registers supports these two interrupt sources. When the error interrupt INTSREDL4 is generated, the error flag of the serial status register 41 (SSR41) is set. By the set error flag, the generation can be judged as either INTSRDL4 (transfer end interrupt) or INTSREDL4 (error interrupt)

3. The external interrupts INTP20 and INTP22 share flags for the interrupt request sources. If one of the interrupt sources INTP20 and INTP22 is generated, bit 6 of the IF1H register is set to 1. Bit 6 of the MK1H, PR01H, and PR11H registers supports these two interrupt sources.
The interrupt request of the external interrupts INTP20 and INTP22 can be masked by interrupt mask flag register 0 (INTMK0). In addition, the generation status of the external interrupt can be verified with interrupt monitor flag register 0 (INTMF0) (see **Figure 20-2**).

Table 20-2. Flags Corresponding to Interrupt Request Sources (3/3)

Interrupt Source	Interrupt Request Flag	Register	Interrupt Mask Flag	Register	Priority Specification Flag	Register	38-pin	30-pin	20-pin
INTTM05	TMIF05	IF2L	TMMK05	MK2L	TMPR005, TMPR105	PR02L, PR12L	√	√	√
INTTM06	TMIF06		TMMK06		TMPR006, TMPR106		√	√	√
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107		√	√	√
INTCMP0	CMPIF0		CMPMK0		CMPPR00, CMPPR10		√	√	√
INTCMP1	CMPIF1		CMPMK1		CMPPR01, CMPPR11		√	√	√
INTCMP2	CMPIF2		CMPMK2		CMPPR02, CMPPR12		√	√	√
INTP9[Note 1]	PIF9[Note 1]		PMK9[Note 1]		PPR09, PPR19[Note 1]		√	–	–
INTCMP3[Note 1]	CMPIF3[Note 1]		CMPMK3[Note 1]		CMPPR03, CMPPR13[Note 1]		√	√	√
INTP10[Note 2]	PIF10[Note 2]		PMK10[Note 2]		PPR010, PPR110[Note 2]		√	–	–
INTCMP4[Note 2]	CMPIF4[Note 2]		CMPMK4[Note 2]		CMPPR04, CMPPR14[Note 2]		√	√	–
INTP11[Note 3]	PIF11[Note 3]	IF2H	PMK11[Note 3]	MK2H	PPR011, PPR111[Note 3]	PR02H, PR12H	√	√	–
INTCMP5[Note 3]	CMPIF5[Note 3]		CMPMK5[Note 3]		CMPPR05, CMPPR15[Note 3]		√	√	–
INTTMKB0	TMKBIF0		TMKBMK0		TMKBPR00, TMKBPR10		√	√	√
INTTMKB1	TMKBIF1		TMKBMK1		TMKBPR01, TMKBPR11		√	√	√
INTTMKB2	TMKBIF2		TMKBMK2		TMKBPR02, TMKBPR12		√	√	–
INTTMKC0	TMKCIF0		TMKCMK0		TMKCPR00, TMKCPR10		√	√	√
INTMD	MDIF		MDMK		MDPR0, MDPR1		√	√	√
INTP21[Note 4]	PIF21[Note 4]		PMK21[Note 4]		PPR021, PPR121[Note 4]		√	√	√
INTP23[Note 4]	PIF23[Note 4]		PMK23[Note 4]		PPR023, PPR123[Note 4]		√	√	–
INTFL	FLIF		FLMK		FLPR0, FLPR1		√	√	√

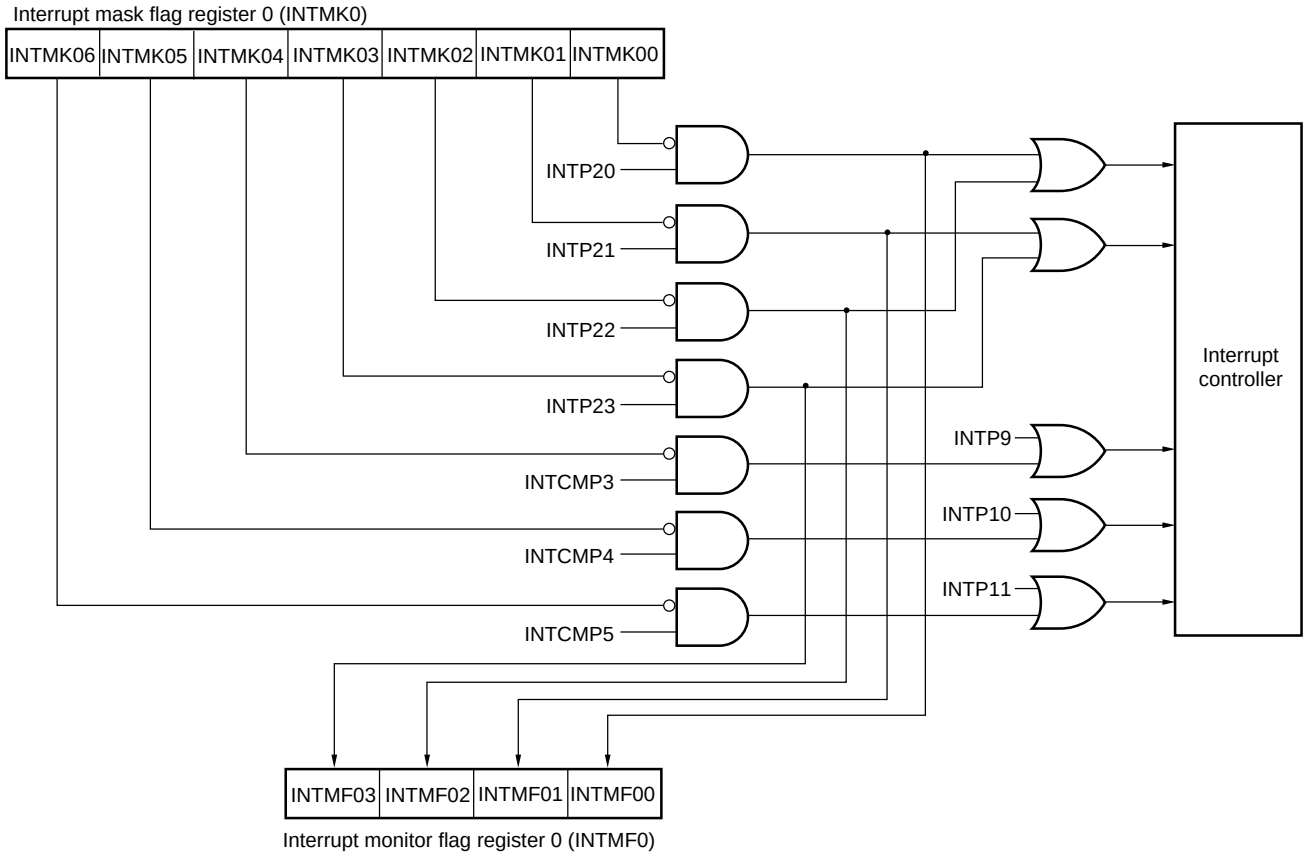
Notes 1. Do not use the external interrupt INTP9 and comparator 3 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTP9 and INTCMP3 is generated, bit 6 of the IF2L register is set to 1. Bit 6 of the MK2L, PR02L, and PR12L registers supports these two interrupt sources.
In addition, the interrupt request of INTCMP3 can be masked by the interrupt mask flag register 0 (INTMK0) (see **Figure 20-2**).

2. Do not use the external interrupt INTP10 and comparator 4 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTP10 and INTCMP4 is generated, bit 7 of the IF2L register is set to 1. Bit 7 of the MK2L, PR02L, and PR12L registers supports these two interrupt sources.
In addition, the interrupt request of INTCMP4 can be masked by the interrupt mask flag register 0 (INTMK0) (see **Figure 20-2**).

3. Do not use the external interrupt INTP11 and comparator 5 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTP11 and INTCMP5 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.
In addition, the interrupt request of INTCMP5 can be masked by the interrupt mask flag register 0 (INTMK0) (see **Figure 20-2**).

4. The external interrupts INTP21 and INTP23 share flags for the interrupt request sources. If one of the interrupt sources INTP21 and INTP23 is generated, bit 6 of the IF2H register is set to 1. Bit 6 of the MK2H, PR02H, and PR12H registers supports these two interrupt sources.
The interrupt request of the external interrupts INTP21 and INTP23 can be masked by interrupt mask flag register 0 (INTMK0). In addition, the generation status of the external interrupt can be verified by interrupt monitor flag register 0 (INTMF0) (see **Figure 20-2**).

Figure 20-2. Image of Interrupt Controller for External Interrupts INTP20 to INTP23, Comparator Interrupts INTCMP3 to INTCMP5



20.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon reset signal generation.

When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

The IF0L, IF0H, IF1L, IF1H, IF2L, and IF2H registers can be set by a 1-bit or 8-bit memory manipulation instruction. When the IF0L and IF0H registers, the IF1L and IF1H registers, and the IF2L and IF2H registers are combined to form 16-bit registers IF0, IF1, and IF2, they can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 20-3. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (38-pin) (1/2)

Address: FFFE0H After reset: 00H R/W

Symbol	7	<6>	<5>	4	3	<2>	<1>	<0>
IF0L	0	PIF4	PIF3	0	0	PIF0	LVIIF	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	0
IF0H	SREIF0 TMIF01H	SRIF0	STIF0 CSIIF00	DMAIF1	DMAIF0	0	0	0

Address: FFFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	TMIF03	TMIF02	TMIF01	TMIF00	IICAIF0	SREIF1 TMIF03H	SRIF1	STIF1

Address: FFFE3H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
IF1H	TMIF04	PIF20 PIF22	SRDLIF4 SREDLIF4	STDLIF4	0	ITIF	RTCIF	ADIF

Address: FFFD0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2L	PIF10 CMPIF4	PIF9 CMPIF3	CMPIF2	CMPIF1	CMPIF0	TMIF07	TMIF06	TMIF05

Figure 20-3. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (38-pin) (2/2)

Address: FFFD1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2H	FLIF	PIF21 PIF23	MDIF	TMKCIF0	TMKBIF2	TMKBIF1	TMKBIF1	PIF11 CMPIF5

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Cautions 1. The available registers and bits differ depending on the product. For details about the registers and bits available for each product, see Table 20-2. Be sure to set bits that are not available to the initial value.

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.

20.3.2 Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt.

The MK0L, MK0H, MK1L, MK1H, MK2L, and MK2H registers can be set by a 1-bit or 8-bit memory manipulation instruction. When the MK0L and MK0H registers, the MK1L and MK1H registers, and the MK2L and MK2H registers are combined to form 16-bit registers MK0, MK1, and MK2, they can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 20-4. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H) (38-pin)

Address: FFFE4H After reset: FFH R/W

Symbol	7	<6>	<5>	4	3	<2>	<1>	<0>
MK0L	1	PMK4	PMK3	1	1	PMK0	LVIMK	WDTIMK

Address: FFFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	0
MK0H	SREMK0 TMMK01H	SRMK0	STMK0 CSIMK00	DMAMK1	DMAMK0	1	1	1

Address: FFFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	TMMK03	TMMK02	TMMK01	TMMK00	IICAMK0	SREMK1 TMMK03H	SRMK1	STMK1

Address: FFFE7H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
MK1H	TMMK04	PMK20 PMK22	SRDLMK4 SREDLMK4	STDLMK4	1	ITMK	RTCMK	ADMK

Address: FFFD4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2L	PMK10 CMPMK4	PMK9 CMPMK3	CMPMK2	CMPMK1	CMPMK0	TMMK07	TMMK06	TMMK05

Address: FFFD5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2H	FLMK	PMK21 PMK23	MDMK	TMKCMK0	TMKBMK2	TMKBMK1	TMKBMK0	PMK11 CMPMK5

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 20-2. Be sure to set bits that are not available to the initial value.

20.3.3 Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)

The priority specification flag registers are used to set the corresponding maskable interrupt priority level.

A priority level is set by using the PR0xy and PR1xy registers in combination (xy = 0L, 0H, 1L, 1H, 2L, or 2H).

The PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and the PR12H registers can be set by a 1-bit or 8-bit memory manipulation instruction. If the PR00L and PR00H registers, the PR01L and PR01H registers, the PR02L and PR02H registers, the PR10L and PR10H registers, the PR11L and PR11H registers, and the PR12L and PR12H registers are combined to form 16-bit registers PR00, PR01, PR02, PR10, PR11, and PR12, they can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 20-5. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (38-pin) (1/2)

Address: FFFE8H After reset: FFH R/W

Symbol	7	<6>	<5>	4	3	<2>	<1>	<0>
PR00L	1	PPR04	PPR03	1	1	PPR00	LVIPR0	WDTIPR0

Address: FFFECH After reset: FFH R/W

Symbol	7	<6>	<5>	4	3	<2>	<1>	<0>
PR10L	1	PPR14	PPR13	1	1	PPR10	LVIPR1	WDTIPR1

Address: FFFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	0
PR00H	SREPR00 TMPR001H	SRPR00	STPR00 CSIPR000	DMAPR01	DMAPR00	1	1	1

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	0
PR10H	SREPR10 TMPR101H	SRPR10	STPR10 CSIPR100	DMAPR11	DMAPR10	1	1	1

Address: FFFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01L	TMPR003	TMPR002	TMPR001	TMPR000	IICAPR00	SREPR01 TMPR003H	SRPR01	STPR01

Address: FFFEEH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	TMPR103	TMPR102	TMPR101	TMPR100	IICAPR10	SREPR11 TMPR103H	SRPR11	STPR11

Figure 20-5. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (38-pin) (2/2)

Address: FFFEBH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
PR01H	TMPR004	PPR020 PPR022	SRDLPR04 SREDLPR04	STDLPR04	1	ITPR0	RTCPR0	ADPR0

Address: FFFEFH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
PR11H	TMPR104	PPR120 PPR122	SRDLPR14 SREDLPR14	STDLPR14	1	ITPR1	RTCPR1	ADPR1

Address: FFFD8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02L	PPR010 CMPPR04	PPR09 CMPPR03	CMPPR02	CMPPR01	CMPPR00	TMPR007	TMPR006	TMPR005

Address: FFFDCH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12L	PPR110 CMPPR14	PPR19 CMPPR13	CMPPR12	CMPPR11	CMPPR10	TMPR107	TMPR106	TMPR105

Address: FFFD9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02H	FLPR0	PPR021 PPR023	MDPR0	TMKCPR00	TMKBPR02	TMKBPR01	TMKBPR00	PPR011 CMPR05

Address: FFFDDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12H	FLPR1	PPR121 PPR123	MDPR1	TMKCPR10	TMKBPR12	TMKBPR11	TMKBPR10	PPR111 CMPR15

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 20-2. Be sure to set bits that are not available to the initial value.

20.3.4 External interrupt rising edge enable registers (EGP0, EGP1, EGP2), external interrupt falling edge enable registers (EGN0, EGN1, EGN2)

These registers specify the valid edge for INTP0 to INTP11.

The EGP0 to EGP2, and EGN0 to EGN2 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 20-6. Format of External Interrupt Rising Edge Enable Registers (EGP0, EGP1, EGP2) and External Interrupt Falling Edge Enable Registers (EGN0, EGN1, EGN2) (38-pin)

Address: FFF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	0	0	0	EGP4	EGP3	0	0	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	0	0	0	EGN4	EGN3	0	0	EGN0

Address: FFF3AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP1	0	0	0	0	EGP11	EGP10	EGP9	0

Address: FFF3BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN1	0	0	0	0	EGN11	EGN10	EGN9	0

Address: F0518H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP2	0	0	0	0	EGP23	EGP22	EGP21	EGP20

Address: F0519H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN2	0	0	0	0	EGN23	EGN22	EGN21	EGN20

EGPn	EGNn	INTPn pin valid edge selection (n = 0, 3, 4, 9 to 11, 20 to 23)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 20-3 shows the ports corresponding to the EGPn and EGNn bits.

Table 20-3. Interrupt Request Signal Corresponding to EGPn and EGNn Bits

Detection Enable Bit		Edge Detection Port	Interrupt Request Signal	38-pin	30-pin	20-pin
EGP0	EGN0	P137	INTP0	√	√	√
EGP3	EGN3	P30	INTP3	√	–	–
EGP4	EGN4	P31	INTP4	√	√	–
EGP9	EGN9	P75	INTP9	√	–	–
EGP10	EGN10	P76	INTP10	√	–	–
EGP11	EGN11	P77	INTP11	√	√	–
EGP20	EGN20	P10 (P203)	INTP20	√	√	√
EGP21	EGN21	P11 (P202)	INTP21	√	√	√
EGP22	EGN22	P200	INTP22	√	√	√
EGP23	EGN23	P206	INTP23	√	√	–

Caution When the input port pins used for the external interrupt functions are switched to the output mode, the INTPn interrupt might be generated upon detection of a valid edge. When switching the input port pins to the output mode, set the port mode register (PMxx) to 0 after disabling the edge detection (by setting EGPn and EGNn to 0).

Remarks 1. n = 0, 3, 4, 9 to 11, 20 to 23

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

20.3.5 Comparator rising edge enable register 0 (CMPEGP0), comparator falling edge enable register 0 (CMPEGN0)

These registers are used to set the valid edges of comparator n detection interrupt signal (INTCMPn) and external interrupts (INTP20, INTP21).

The CMPEGP0 and CMPEGN0 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 20-7. Format of Comparator Rising Edge Enable Register 0 (CMPEGP0) and Comparator Falling Edge Enable Register 0 (CMPEGN0)

Address: F0558 H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMPEGP0	CEGP7	CEGP6	CEGP5	CEGP4	CEGP3	CEGP2	CEGP1	CEGP0

Address: F0559 H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMPEGN0	CEGN7	CEGN6	CEGN5	CEGN4	CEGN3	CEGN2	CEGN1	CEGN0

CEGP7	CEGN7	INTP21 pin valid edge selection
0	0	Edge detection disabled (disables output of timer restart signal (output signal = fixed to low level))
0	1	Falling edge (enables output of timer restart signal)
1	0	Rising edge (enables output of timer restart signal)
1	1	Both rising and falling edges (enables output of timer restart signal)

CEGP6	CEGN6	INTP20 pin valid edge selection
0	0	Edge detection disabled (disables output of timer restart signal (output signal = fixed to low level))
0	1	Falling edge (enables output of timer restart signal)
1	0	Rising edge (enables output of timer restart signal)
1	1	Both rising and falling edges (enables output of timer restart signal)

CEGPn	CEGNn	INTCMPn valid edge selection
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Cautions 1. Be sure to set PGACMPEN bit in the PER2 register to 1 before setting CMPEGP0 and CMPEGN0 registers.

2. The valid edge setting is set for the signal that was non-reverse- or reverse-rotated from the comparator detect signal using the CnINV bit of the CnCTL register.

Remark n = 0 to 5

20.3.6 Interrupt mask flag register 0 (INTMK0)

The interrupt mask flag register 0 (INTMK0) is used to mask the interrupt request signals of INTP20 to INTP23, and INTCMP3 to INTCMP5.

The INTMK0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 20-8. Format of Interrupt Mask Flag Register 0 (INTMK0) (38-pin)

Address: F05C2H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
INTMK0	1	INTMK06	INTMK05	INTMK04	INTMK03	INTMK02	INTMK01	INTMK00

INTMK06	Interrupt request signal output enable/disable of INTCMP5
0	Output enable
1	Output disable

INTMK05	Interrupt request signal output enable/disable of INTCMP4
0	Output enable
1	Output disable

INTMK04	Interrupt request signal output enable/disable of INTCMP3
0	Output enable
1	Output disable

INTMK03	Interrupt request signal output enable/disable of INTP23
0	Output enable
1	Output disable

INTMK02	Interrupt request signal output enable/disable of IINTP22
0	Output enable
1	Output disable

INTMK01	Interrupt request signal output enable/disable of IINTP21
0	Output enable
1	Output disable

INTMK00	Interrupt request signal output enable/disable of IINTP20
0	Output enable
1	Output disable

20.3.7 Interrupt monitor flag register 0 (INTMF0)

The interrupt mask flag register 0 (INTMF0) is used to monitor the generation status of the interrupt request signals of INTP20 to INTP23.

Generating the interrupt request signals of INTP20 to INTP23 set the corresponding flag of the INTMF0 register to 1. However, the flag is not automatically cleared to 0. Therefore the flag must be cleared by software.

In addition, until cleared (0), interrupt request signal of INTP20 to INTP23 will not be input.

The INTMF0 register can be set and read by a 1-bit memory manipulation instruction or can be read by an 8-bit memory manipulation instruction.

When cleared (0), perform a 1-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 20-9. Format of Interrupt Monitor Flag Register 0 (INTMF0) (38-pin)

Address: F05C3H After reset: 00H R/W[Note]

Symbol	7	6	5	4	3	2	1	0
INTMF0	0	0	0	0	INTMF03	INTMF02	INTMF01	INTMF00

INTMF03	Indication of generation status of the interrupt request signal of INTP23
0	Interrupt request signal is not generated.
1	Interrupt request signal is generated.

INTMF02	Indication of generation status of the interrupt request signal of INTP22
0	Interrupt request signal is not generated.
1	Interrupt request signal is generated.

INTMF01	Indication of generation status of the interrupt request signal of INTP21
0	Interrupt request signal is not generated.
1	Interrupt request signal is generated.

INTMF00	Indication of generation status of the interrupt request signal of INTP20
0	Interrupt request signal is not generated.
1	Interrupt request signal is generated.

Note An 8-bit memory manipulation instruction can only be read.

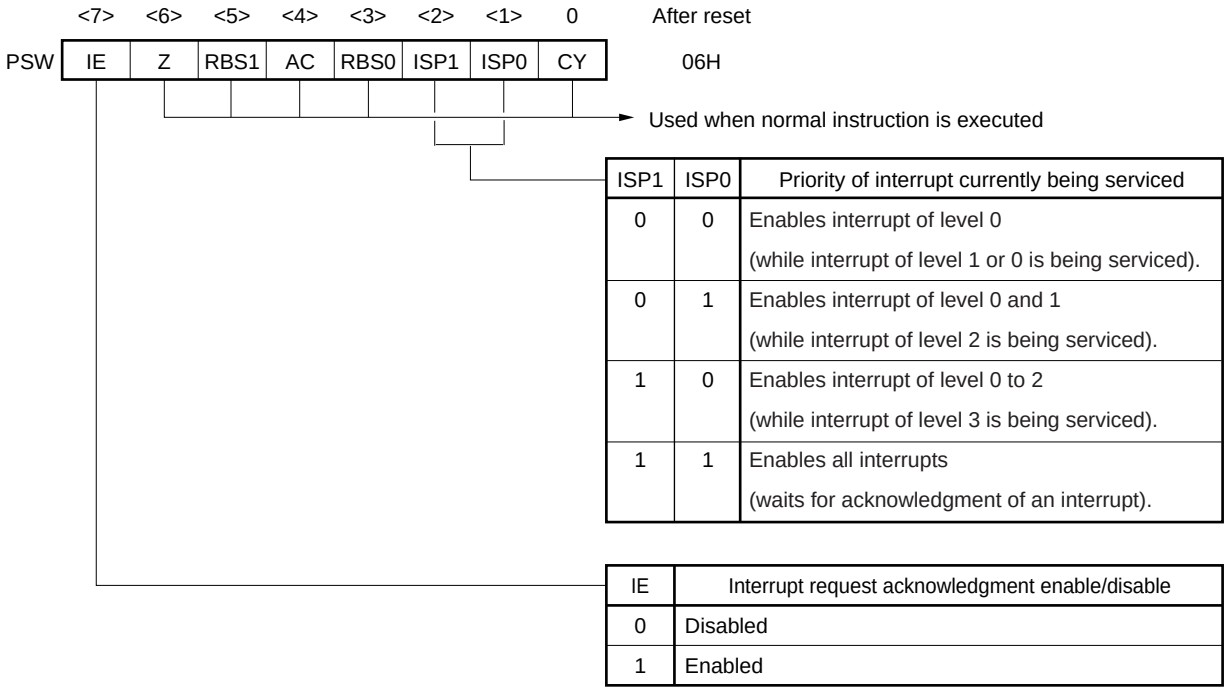
20.3.8 Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for an interrupt request. The IE flag that sets maskable interrupt enable/disable and the ISP0 and ISP1 flags that controls multiple interrupt servicing are mapped to the PSW.

Besides 8-bit read/write, this register can carry out operations using bit manipulation instructions and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged, if the BRK instruction is executed, the contents of the PSW are automatically saved into a stack and the IE flag is reset to 0. Upon acknowledgment of a maskable interrupt request, if the value of the priority specification flag register of the acknowledged interrupt is not 00, its value minus 1 is transferred to the ISP0 and ISP1 flags. The PSW contents are also saved into the stack with the PUSH PSW instruction. They are restored from the stack with the RETI, RETB, and POP PSW instructions.

Reset signal generation sets PSW to 06H.

Figure 20-10. Configuration of Program Status Word



ISP1	ISP0	Priority of interrupt currently being serviced
0	0	Enables interrupt of level 0 (while interrupt of level 1 or 0 is being serviced).
0	1	Enables interrupt of level 0 and 1 (while interrupt of level 2 is being serviced).
1	0	Enables interrupt of level 0 to 2 (while interrupt of level 3 is being serviced).
1	1	Enables all interrupts (waits for acknowledgment of an interrupt).

IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

20.4 Interrupt Servicing Operations

20.4.1 Maskable interrupt request acknowledgment

A maskable interrupt request becomes acknowledgeable when the interrupt request flag is set to 1 and the mask (MK) flag corresponding to that interrupt request is cleared to 0. A vectored interrupt request is acknowledged if interrupts are in the interrupt enabled state (when the IE flag is set to 1). However, a low-priority interrupt request is not acknowledged during servicing of a higher priority interrupt request.

The times from generation of a maskable interrupt request until vectored interrupt servicing is performed are listed in Table 20-4 below.

For the interrupt request acknowledgment timing, see **Figures 20-12** and **20-13**.

Table 20-4. Time from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time[Note]
Servicing time	9 clocks	16 clocks

Note Maximum time does not apply when an instruction from the internal RAM area is executed.

Remark 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

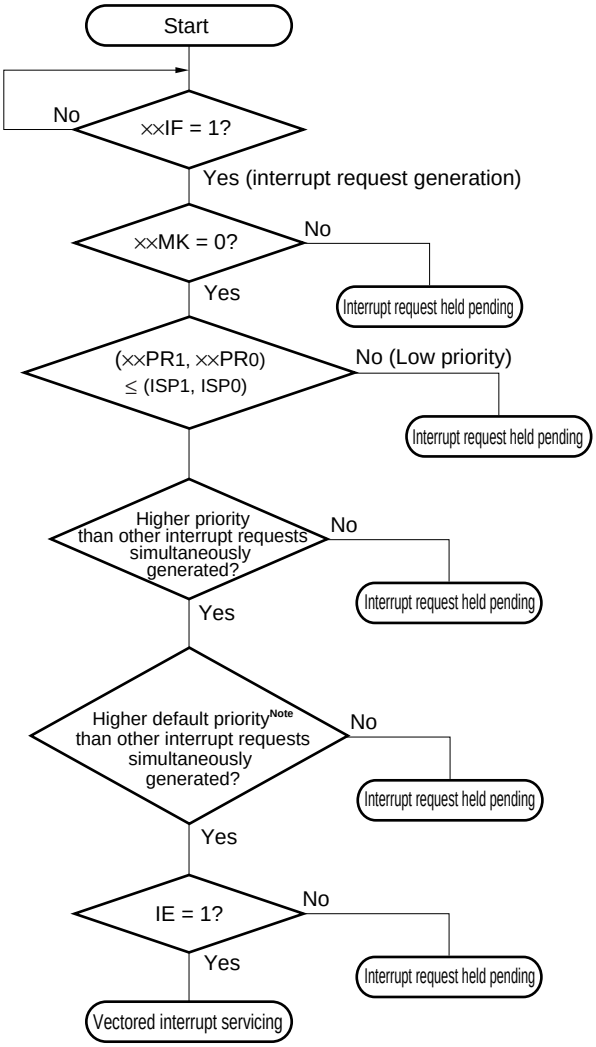
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 20-11 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 (0), and the contents of the priority specification flag corresponding to the acknowledged interrupt are transferred to the ISP1 and ISP0 flags. The vector table data determined for each interrupt request is the loaded into the PC and branched.

Restoring from an interrupt is possible by using the RETI instruction.

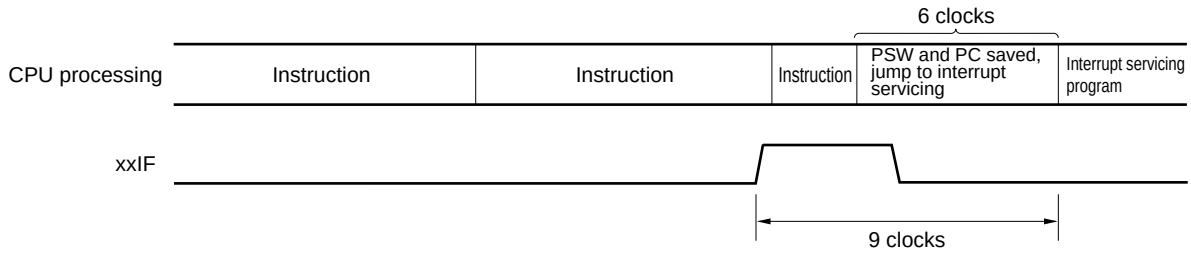
Figure 20-11. Interrupt Request Acknowledgment Processing Algorithm



××IF: Interrupt request flag
××MK: Interrupt mask flag
××PR0: Priority specification flag 0
××PR1: Priority specification flag 1
IE: Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)
ISP0, ISP1: Flag that indicates the priority level of the interrupt currently being serviced (see **Figure 20-10**)

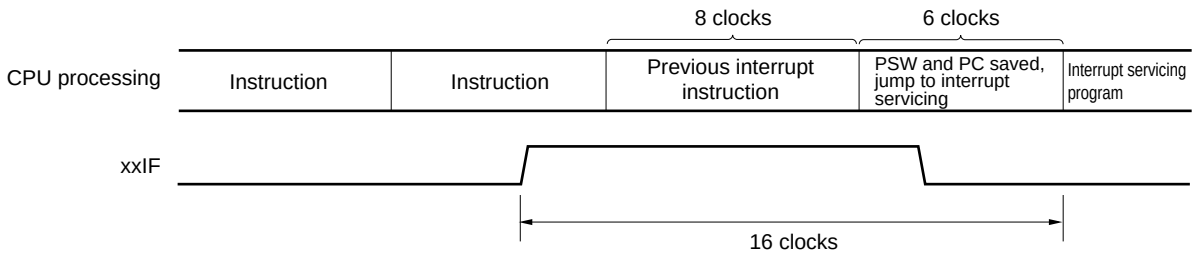
Note For the default priority, see **Table 20-1 Interrupt Source List**.

Figure 20-12. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

Figure 20-13. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

20.4.2 Software interrupt request acknowledgment

A software interrupt request is acknowledged by BRK instruction execution. Software interrupts cannot be disabled.

If a software interrupt request is acknowledged, the contents are saved into the stacks in the order of the program status word (PSW), then program counter (PC), the IE flag is reset (0), and the contents of the vector table (0007EH, 0007FH) are loaded into the PC and branched.

Restoring from a software interrupt is possible by using the RETB instruction.

Caution Do not use the RETI instruction for restoring from the software interrupt.

20.4.3 Multiple interrupt servicing

Multiple interrupt servicing occurs when another interrupt request is acknowledged during execution of an interrupt.

Multiple interrupt servicing does not occur unless the interrupt request acknowledgment enabled state is selected (IE = 1). When an interrupt request is acknowledged, interrupt request acknowledgment becomes disabled (IE = 0). Therefore, to enable multiple interrupt servicing, it is necessary to set (1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledgment.

Moreover, even if interrupts are enabled, multiple interrupt servicing may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for multiple interrupt servicing.

In the interrupt enabled state, if an interrupt request with a priority equal to or higher than that of the interrupt currently being serviced is generated, it is acknowledged for multiple interrupt servicing. If an interrupt with a priority lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for multiple interrupt servicing. Interrupt requests that are not enabled because interrupts are in the interrupt disabled state or because they have a lower priority are held pending. When servicing of the current interrupt ends, the pending interrupt request is acknowledged following execution of at least one main processing instruction execution.

Table 20-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 20-14 shows multiple interrupt servicing examples.

Table 20-5. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing

Multiple Interrupt Request / Interrupt Being Serviced		Maskable Interrupt Request								Software Interrupt Request
		Priority Level 0 (PR = 00)		Priority Level 1 (PR = 01)		Priority Level 2 (PR = 10)		Priority Level 3 (PR = 11)		
		IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	
Maskable interrupt	ISP1 = 0, ISP0 = 0	√	–	–	–	–	–	–	–	√
	ISP1 = 0, ISP0 = 1	√	–	√	–	–	–	–	–	√
	ISP1 = 1, ISP0 = 0	√	–	√	–	√	–	–	–	√
	ISP1 = 1, ISP0 = 1	√	–	√	–	√	–	√	–	√
Software interrupt		√	–	√	–	√	–	√	–	√

Remarks 1. √: Multiple interrupt servicing enabled

2. –: Multiple interrupt servicing disabled

3. ISP0, ISP1, and IE are flags contained in the PSW.

ISP1 = 0, ISP0 = 0: An interrupt of level 1 or level 0 is being serviced.

ISP1 = 0, ISP0 = 1: An interrupt of level 2 is being serviced.

ISP1 = 1, ISP0 = 0: An interrupt of level 3 is being serviced.

ISP1 = 1, ISP0 = 1: Wait for An interrupt acknowledgment (all interrupts are enabled).

IE = 0: Interrupt request acknowledgment is disabled.

IE = 1: Interrupt request acknowledgment is enabled.

4. PR is a flag contained in the PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H registers.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)

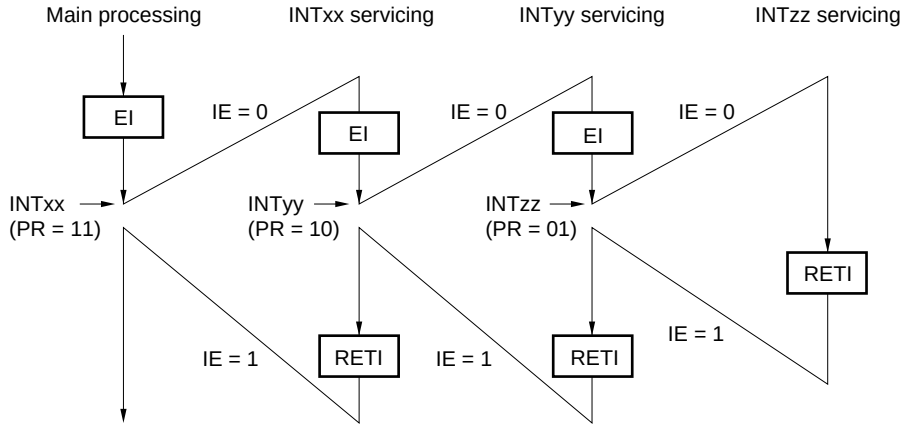
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1

PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0

PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)

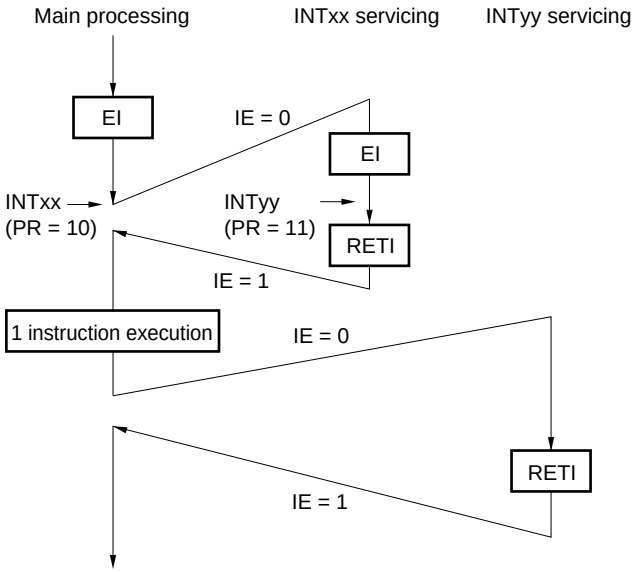
Figure 20-14. Examples of Multiple Interrupt Servicing (1/2)

Example 1. Multiple interrupt servicing occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgment.

Example 2. Multiple interrupt servicing does not occur due to priority control

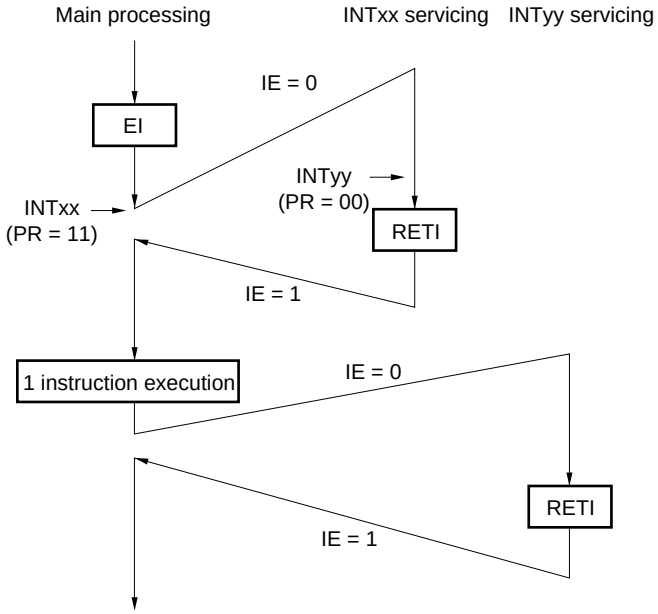


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

Figure 20-14. Examples of Multiple Interrupt Servicing (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled



Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

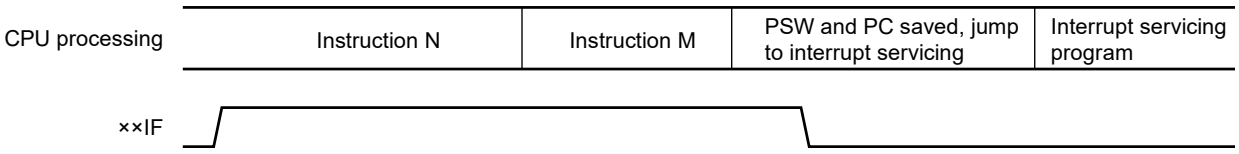
20.4.4 Interrupt request hold

There are instructions where, even if an interrupt request is issued while the instructions are being executed, interrupt request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, $addr20
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- SKH
- SKNH
- Write instructions for the IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H registers

Figure 20-15 shows the timing at which interrupt requests are held pending.

Figure 20-15. Interrupt Request Hold



Remarks 1. Instruction N: Interrupt request hold instruction

2. Instruction M: Instruction other than interrupt request hold instruction

CHAPTER 21 STANDBY FUNCTION

21.1 Standby Function

The standby function reduces the operating current of the system, and the following three modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. In the HALT mode, the CPU operation clock is stopped. If the high-speed system clock oscillator, high-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

STOP instruction execution sets the STOP mode. In the STOP mode, the high-speed system clock oscillator and high-speed on-chip oscillator stop, stopping the whole system, thereby considerably reducing the CPU operating current.

Because this mode can be cleared by an interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure the oscillation stabilization time after the STOP mode is released when the X1 clock is selected, select the HALT mode if it is necessary to start processing immediately upon interrupt request generation.

(3) SNOOZE mode

In the case of CSI00, UART0, or UART4 data reception and an A/D conversion request by the timer trigger signal (the interrupt request signal (INTRTC/INTIT)), the STOP mode is exited, the CSI00, UART0, or DALI/UART4 data is received without operating the CPU, and A/D conversion is performed. This can only be specified when the high-speed on-chip oscillator is selected for the CPU/peripheral hardware clock (f_{CLK}).

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

Cautions 1. The STOP mode can be used only when the CPU is operating on the main system clock. Do not set to the STOP mode while the CPU operates with the subsystem clock. The HALT mode can be used when the CPU is operating on either the main system clock or the subsystem clock.

2. When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction (except SNOOZE mode setting unit).

3. When using CSI00, UART0, UART4, or the A/D converter in the SNOOZE mode, set up serial standby control register 0 (SSC0), serial standby control register 4 (SSC4), and A/D converter mode register 2 (ADM2) before switching to the STOP mode. For details, see 15.3 Registers Controlling Serial Array Unit 0, 16.3 Registers Controlling Serial Array Unit 4 (DALI/UART4) and 12.3 Registers Controlling A/D Converter.

4. When UART4 is set to SNOOZE mode, CSI00, UART0, and the A/D converter cannot be set to SNOOZE mode. It is possible to set CSI00, UART0, and the A/D converter to SNOOZE mode at the same time.

5. The following sequence is recommended for power consumption reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS) and bit 0 (ADCE) of A/D converter mode register 0 (ADM0) to 0 to stop the A/D conversion operation, and then execute the STOP instruction.

6. It can be selected by the option byte whether the low-speed on-chip oscillator continues oscillating or stops in the HALT or STOP mode. For details, see CHAPTER 27 OPTION BYTE.

21.2 Registers Controlling Standby Function

The registers which control the standby function are described below.

- Subsystem clock supply mode control register (OSMC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)

Remark For details of registers described above, see **CHAPTER 5 CLOCK GENERATOR**. For registers which control the SNOOZE mode, see **CHAPTER 12 A/D CONVERTER**, **CHAPTER 15 SERIAL ARRAY UNIT**, and **CHAPTER 16 SERIAL ARRAY UNIT 4 (DALI/UART4)**.

21.3 Standby Function Operation

21.3.1 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, or subsystem clock.

The operating statuses in the HALT mode are shown below.

Caution Because the interrupt request signal is used to clear the HALT mode, if the interrupt mask flag is 0 (the interrupt processing is enabled) and the interrupt request flag is 1 (the interrupt request signal is generated), the HALT mode is not entered even if the HALT instruction is executed in such a situation.

Table 21-1. Operating Statuses in HALT Mode (1/2)

HALT Mode Setting / Item			When HALT Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on High-speed On-chip Oscillator Clock (f_{IH})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EX})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{IH}	Operation continues (cannot be stopped)	Operation disabled	
		f_X	Operation disabled	Operation continues (cannot be stopped)	Cannot operate
		f_{EX}		Cannot operate	Operation continues (cannot be stopped)
	Subsystem clock	f_{XT}	Status before HALT mode was set is retained		
		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: Oscillates • WUTMMCK0 = 0 and WDTON = 0: Stops • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops		
CPU			Operation stopped		
Code flash memory			Operation stopped		
Data flash memory					
RAM					
Port (latch)			Status before HALT mode was set is retained		
Timer array unit			Operable		
Timer KB0 to KB2					
Timer KC0					
Real-time clock (RTC)					
12-bit interval timer					
Watchdog timer			See **CHAPTER 11 WATCHDOG TIMER**		
A/D converter			Operable		
Programmable gain amplifier					
Comparator					
Serial array unit 0 (SAU0)					
Serial array unit 4 (DALI/UART4)					
Serial interface (IICA)					
Multiplier and divider/multiply-accumulator					
DMA controller					
Power-on-reset function					
Voltage detection function					
External interrupt					
CRC operation function	High-speed CRC				
	General-purpose CRC		In the calculation of the RAM area, operable when DMA is executed only		
RAM parity error detection function			Operable when DMA is executed only		
RAM guard function					
SFR guard function					
Illegal-memory access detection function					
PLL function			Operable		

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

Remark Operation stopped: Operation is automatically stopped before switching to the HALT mode.
Operation disabled: Operation is stopped before switching to the HALT mode.
f_{IH}: High-speed on-chip oscillator clock
f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock
f_{EX}: External main system clock
f_{XT}: XT1 clock
f_{EXS}: External subsystem clock

Table 21-1. Operating Statuses in HALT Mode (2/2)

HALT Mode Setting / Item			When HALT Instruction Is Executed While CPU Is Operating on Subsystem Clock	
			When CPU Is Operating on XT1 Clock (f_{XT})	When CPU Is Operating on External Subsystem Clock (f_{EXS})
System clock			Clock supply to the CPU is stopped	
	Main system clock	f_{IH}	Operation disabled	
		f_X		
		f_{EX}		
	Subsystem clock	f_{XT}	Operation continues (cannot be stopped)	Cannot operate
		f_{EXS}	Cannot operate	Operation continues (cannot be stopped)
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply mode control register (OSMC) • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 0 and WDTON = 0: Stops • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops	
CPU			Operation stopped	
Code flash memory				
Data flash memory				
RAM				
Port (latch)			Status before HALT mode was set is retained	
Timer array unit			Operable when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).	
Timer KB0 to KB2				
Timer KC0				
Real-time clock (RTC)			Operable	
12-bit interval timer				
Watchdog timer			See **CHAPTER 11 WATCHDOG TIMER**	
A/D converter			Operation disabled	
Programmable gain amplifier			Operable (However, this is not used, since the operation has been disabled for the A/D converter that is the destination for input of the PGA output signal)	
Comparator			Operable (in the low-consumption RTC mode (RTCLPC in the OSMC register is set to 1), only CMP0 and CMP2 are able to operate, on condition that the setting for release from the STOP mode in response to comparator interrupt detection has been made (CMPnSTEN in the PFSEL0 register is set to 1; n = 0, 2) and the noise filter is not in use).	
Serial array unit 0 (SAU0)			Operable when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).	
Serial array unit 4 (DALI/UART4)			Operation disabled	
Serial interface (IICA)				
Multiplier and divider/multiply-accumulator			Operable when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).	
DMA controller				
Power-on-reset function			Operable	
Voltage detection function				
External interrupt			Operable (The operations of INTP20 and INTP21 are disabled while in INTP simultaneous mode of timer KB0 to KB2.)	
CRC operation function	High-speed CRC		Operation disabled	
	General-purpose CRC		In the calculation of the RAM area, operable when DMA is executed only	
RAM parity error detection function			Operable when DMA is executed only	
RAM guard function				
SFR guard function				
Illegal-memory access detection function				
PLL function			Operation disabled	

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

Remark Operation stopped: Operation is automatically stopped before switching to the HALT mode.
Operation disabled: Operation is stopped before switching to the HALT mode.
f_{IH}: High-speed on-chip oscillator clock
f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock
f_{EX}: External main system clock
f_{XT}: XT1 clock
f_{EXS}: External subsystem clock

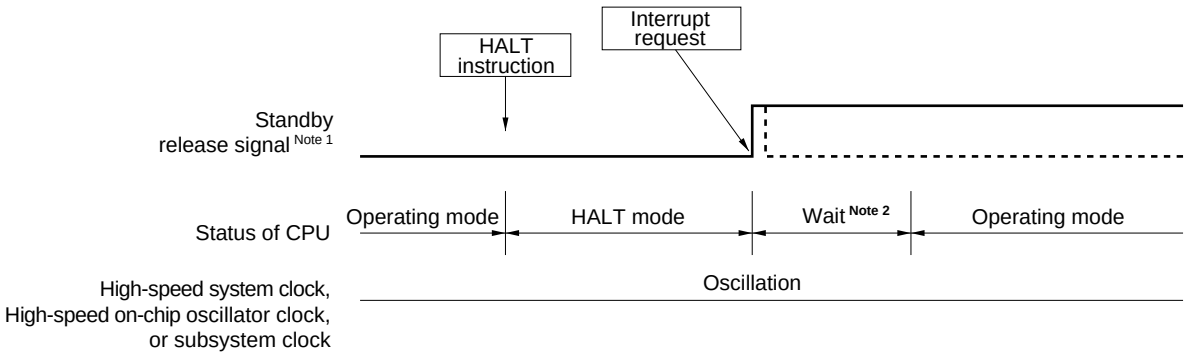
(2) HALT mode release

The HALT mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is released. If interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 21-1. HALT Mode Release by Interrupt Request Generation



Notes 1. For details of the standby release signal, see **Figure 20-1**.

2. Wait time for HALT mode release

- When vectored interrupt servicing is carried out

  Main system clock: 15 to 16 clock
  Subsystem clock (RTCLPC = 0): 10 to 11 clock
  Subsystem clock (RTCLPC = 1): 11 to 12 clock

- When vectored interrupt servicing is not carried out

  Main system clock: 9 to 10 clock
  Subsystem clock (RTCLPC = 0): 4 to 5 clock
  Subsystem clock (RTCLPC = 1): 5 to 6 clock

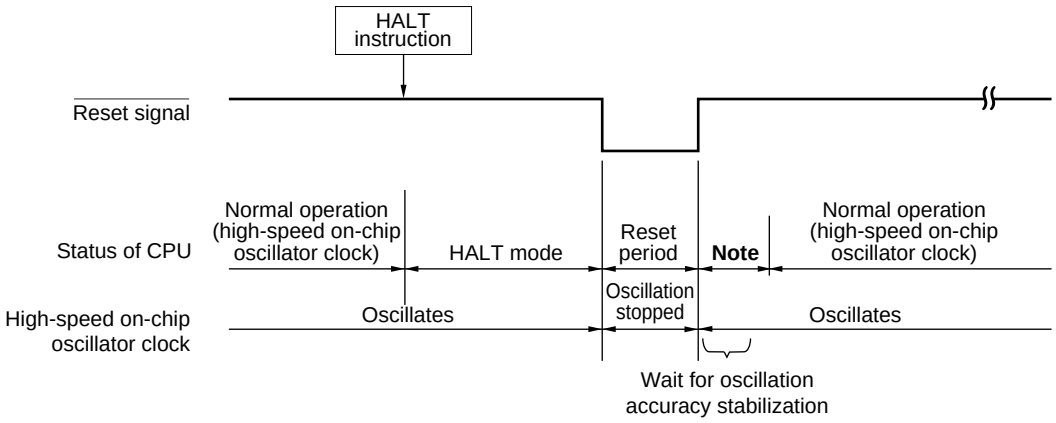
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

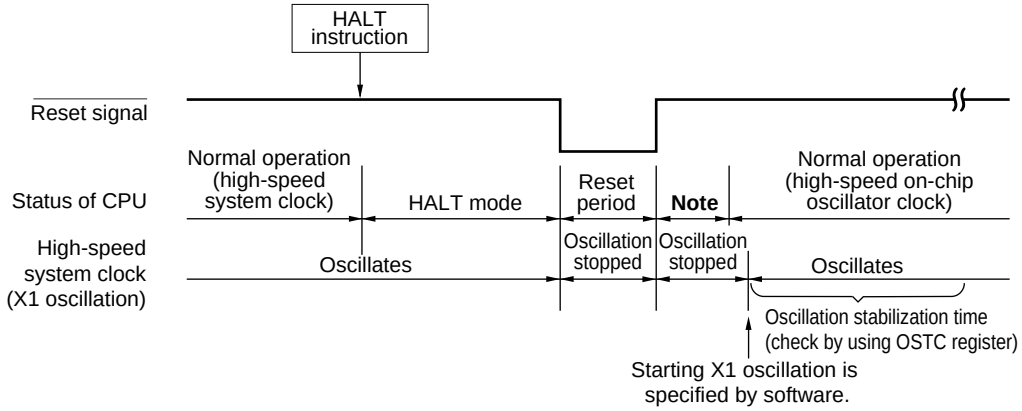
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 21-2. HALT Mode Release by Reset

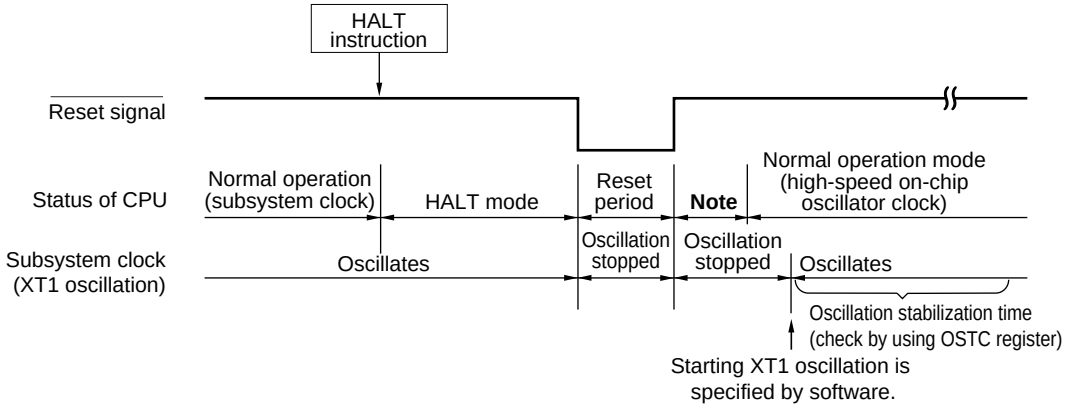
(1) When high-speed on-chip oscillator clock is used as CPU clock



(2) When high-speed system clock is used as CPU clock



(3) When subsystem clock is used as CPU clock



Note For the reset processing time, see **CHAPTER 22 RESET FUNCTION**. For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 23 POWER-ON-RESET CIRCUIT**.

21.3.2 STOP mode

(1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction, and it can be set only when the CPU clock before the setting was the high-speed on-chip oscillator clock, X1 clock, or external main system clock.

Caution Because the interrupt request signal is used to clear the STOP mode, if the interrupt mask flag is 0 (the interrupt processing is enabled) and the interrupt request flag is 1 (the interrupt request signal is generated), the STOP mode is immediately cleared if set when the STOP instruction is executed in such a situation. Accordingly, once the STOP instruction is executed, the system returns to its normal operating mode after the elapse of release time from the STOP mode.

The operating statuses in the STOP mode are shown below.

Table 21-2. Operating Statuses in STOP Mode

STOP Mode Setting / Item			When STOP Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on High-speed On-chip Oscillator Clock (f_{IH})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EX})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{IH}	Stopped		
		f_X			
		f_{EX}			
	Subsystem clock	f_{XT}	Status before STOP mode was set is retained		
		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: Oscillates • WUTMMCK0 = 0 and WDTON = 0: Stops • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops		
CPU			Operation stopped		
Code flash memory					
Data flash memory					
RAM					
Port (latch)			Status before STOP mode was set is retained		
Timer array unit			Operation disabled		
Timer KB0 to KB2					
Timer KC0					
Real-time clock (RTC)			Operable		
12-bit interval timer					
Watchdog timer			See **CHAPTER 11 WATCHDOG TIMER**		
A/D converter			Wakeup operation is enabled (switching to the SNOOZE mode)		
Programmable gain amplifier			Operable		
Comparator			Only CMP0 and CMP2 are able to operate, on condition that the setting for release from the STOP mode in response to comparator interrupt detection has been made (CMPnSTEN in the PFSEL0 register is set to 1; n = 0, 2) and the noise filter is not in use.		
Serial array unit 0 (SAU0)			Wakeup operation is enabled only for CSI00 and UART0 (switching to the SNOOZE mode) Operation disabled other than CSI00 and UART0.		
Serial array unit 4 (DALI/UART4)			Wakeup operation is enabled (switching to the SNOOZE mode).		
Serial interface (IICA)			Wakeup by address match operable		
Multiplier and divider/multiply-accumulator			Operation disabled		
DMA controller					
Power-on-reset function			Operable		
Voltage detection function					
External interrupt			Operable (The operations of INTP20 and INTP21 are disabled while in INTP simultaneous mode of timer KB0 to KB2.)		
CRC operation function	High-speed CRC		Operation stopped		
	General-purpose CRC				
RAM parity error detection function					
RAM guard function					
SFR guard function					
Illegal-memory access detection function					
PLL function			Operation disabled		

(**Remark** and **Cautions** are listed on the next page.)

Remark Operation stopped: Operation is automatically stopped before switching to the STOP mode.
Operation disabled: Operation is stopped before switching to the STOP mode.
f_{IH}: High-speed on-chip oscillator clock
f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock
f_{EX}: External main system clock
f_{XT}: XT1 clock
f_{EXS}: External subsystem clock

(2) STOP mode release

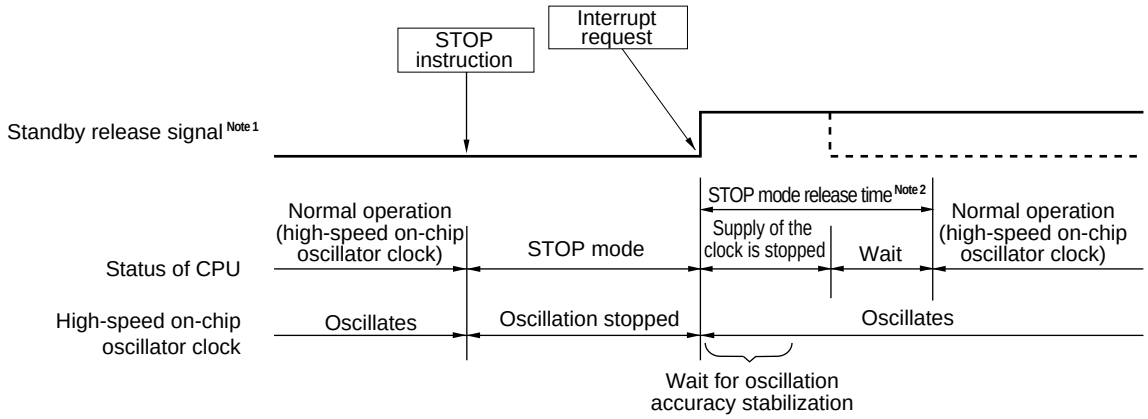
The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 21-3. STOP Mode Release by Interrupt Request Generation (1/2)

(1) When high-speed on-chip oscillator clock is used as CPU clock



Notes 1. For details of the standby release signal, see **Figure 20-1**.

2. STOP mode release time

Supply of the clock is stopped: 18 to 65 μs

Wait

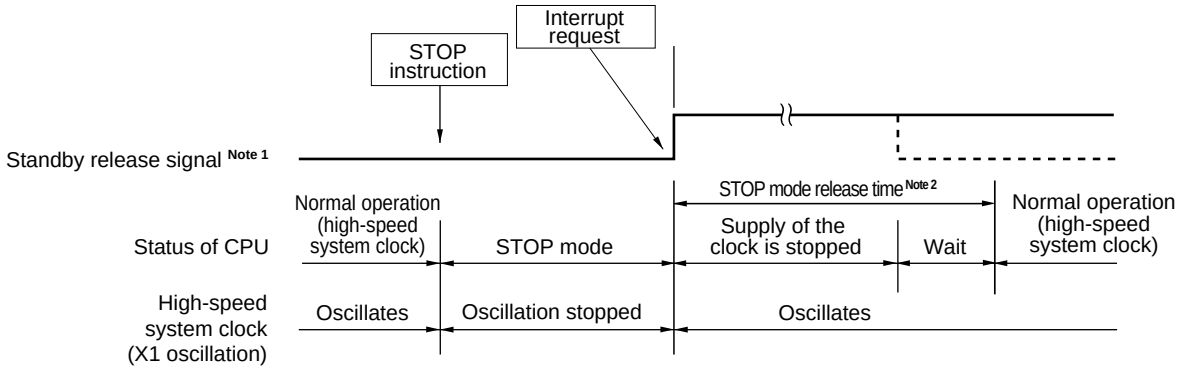
- When vectored interrupt servicing is carried out: 7 clocks
- When vectored interrupt servicing is not carried out: 1 clock

Remarks 1. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

2. The period during which clock supply stops depends on the temperature conditions and the STOP mode period.

Figure 21-3. STOP Mode Release by Interrupt Request Generation (2/2)

(2) When high-speed system clock (X1 oscillation) is used as CPU clock



Notes 1. For details of the standby release signal, see **Figure 20-1**.

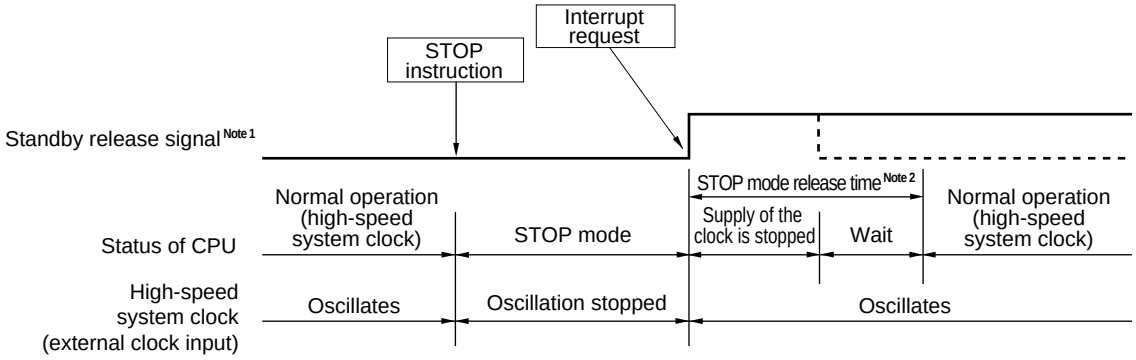
2. STOP mode release time

Supply of the clock is stopped: 18 μs to "whichever is longer 65 μs and the oscillation stabilization time (set by OSTS)"

Wait

- When vectored interrupt servicing is carried out: 10 or 11 clocks
- When vectored interrupt servicing is not carried out: 4 or 5 clocks

(3) When high-speed system clock (external clock input) is used as CPU clock



Notes 1. For details of the standby release signal, see **Figure 20-1**.

2. STOP mode release time

Supply of the clock is stopped: 18 to 65 μs

Wait

- When vectored interrupt servicing is carried out: 7 clocks
- When vectored interrupt servicing is not carried out: 1 clock

Caution To reduce the oscillation stabilization time after release from the STOP mode while CPU operates based on the high-speed system clock (X1 oscillation), switch the clock to the high-speed on-chip oscillator clock temporarily before executing the STOP instruction.

Remarks 1. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

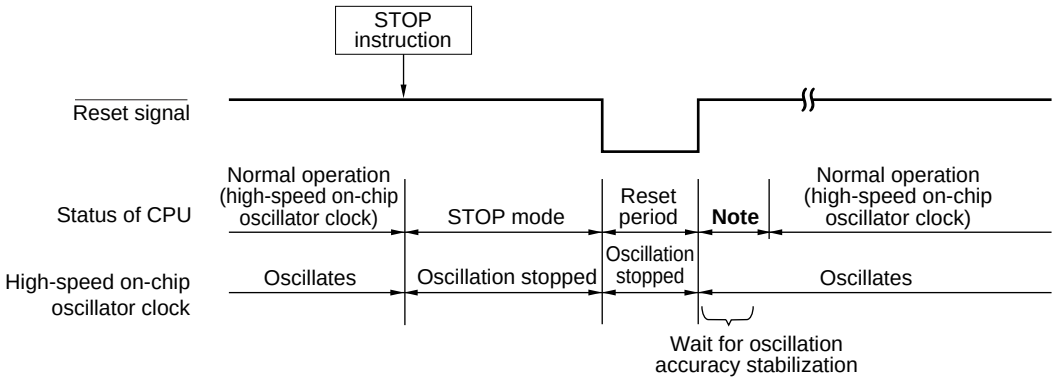
2. The period during which clock supply stops depends on the temperature conditions and the STOP mode period.

(b) Release by reset signal generation

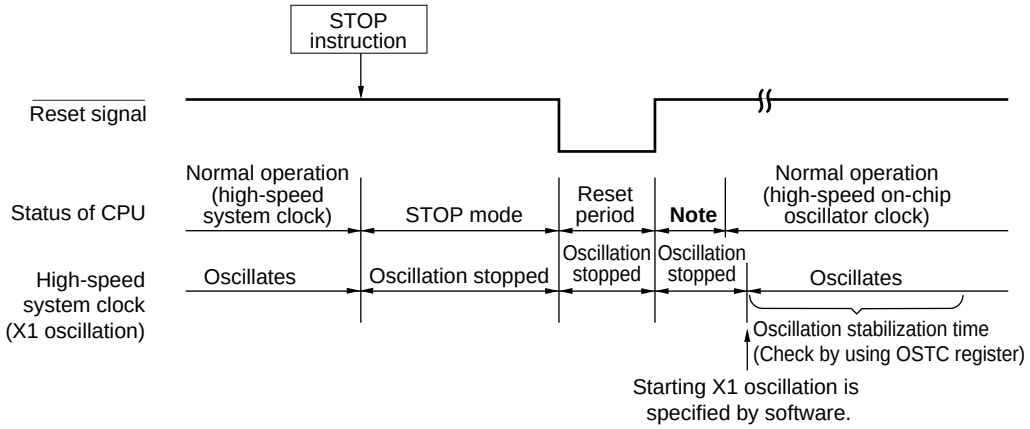
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 21-4. STOP Mode Release by Reset

(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 **CHAPTER 22 RESET FUNCTION**. For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 23 POWER-ON-RESET CIRCUIT**.

21.3.3 SNOOZE mode

(1) SNOOZE mode setting and operating statuses

The SNOOZE mode can only be specified for CSI00, UART0, UART4, or the A/D converter. In addition, this mode can only be specified if the CPU clock is the high-speed on-chip oscillator clock. Note that the PLL output cannot be used during the case described above.

When using CSI00, UART0, or UART4 in the SNOOZE mode, set the SWC0 bit of serial standby control register 0 (SSC0) or the SWC4 bit of serial standby control register 4 (SSC4) to 1 immediately before switching to the STOP mode. For details, see **15.3 Registers Controlling Serial Array Unit 0** and **16.3 Registers Controlling Serial Array Unit 4 (DALI/UART4)**.

When using the A/D converter in the SNOOZE mode, set the AWC bit of A/D converter mode register 2 (ADM2) to 1 immediately before switching to the STOP mode. For details, see **12.3 Registers Controlling A/D Converter**.

Caution When UART4 is set to SNOOZE mode, CSI00, UART0, and the A/D converter cannot be set to SNOOZE mode. It is possible to set CSI00, UART0, and the A/D converter to SNOOZE mode at the same time.

In SNOOZE mode transition, wait status to be only following time.

Transition time from STOP mode to SNOOZE mode: 18 to 65 μs

Remark The time required to transition from STOP mode to SNOOZE mode depends on the temperature conditions and the STOP mode period.

Transition time from SNOOZE mode to normal operation:

- When vectored interrupt servicing is carried out:
  HS (High-speed main) mode: 4.99 to 9.44 μs + 7 clocks
  LS (Low-speed main) mode: 1.10 to 5.08 μs + 7 clocks
- When vectored interrupt servicing is not carried out:
  HS (High-speed main) mode: 4.99 to 9.44 μs + 1 clock
  LS (Low-speed main) mode: 1.10 to 5.08 μs + 1 clock

The operating statuses in the SNOOZE mode are shown below.

Table 21-3. Operating Statuses in SNOOZE Mode

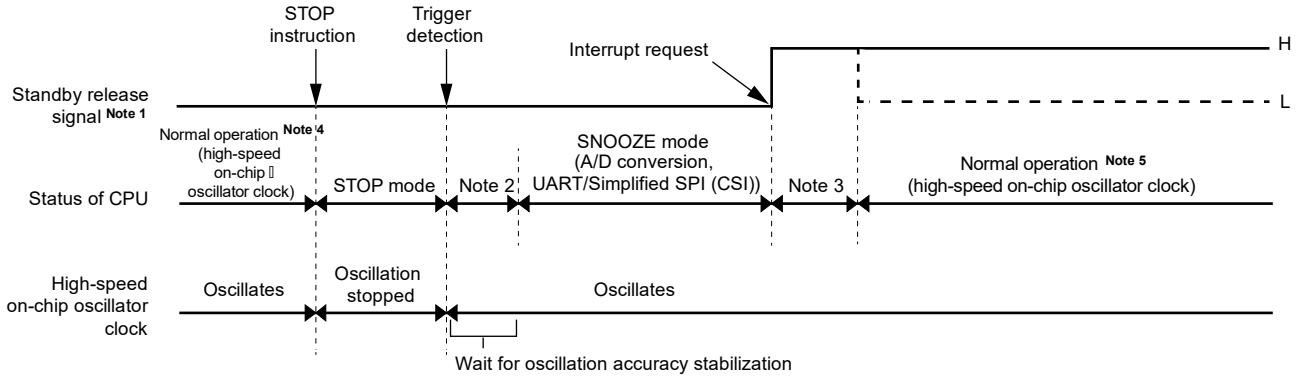
Item \ STOP Mode Setting		When Inputting CSI00/UART0 Data Reception Signal or A/D Converter Timer Trigger Signal While in STOP Mode
		When CPU Is Operating on High-speed On-chip Oscillator Clock (f_{IH})
System clock		Clock supply to the CPU is stopped
Main system clock	f_{IH}	Operation started
	f_X	Stopped
	f_{EX}	
Subsystem clock	f_{XT}	Use of the status while in the STOP mode continues
	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: Oscillates • WUTMMCK0 = 0 and WDTON = 0: Stops • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops
CPU		Operation stopped
Code flash memory		
Data flash memory		
RAM		
Port (latch)		Use of the status while in the STOP mode continues
Timer array unit		Operation disabled
Timer KB0 to KB2		
Timer KC0		
Real-time clock (RTC)		Operable
12-bit interval timer		
Watchdog timer		See **CHAPTER 11 WATCHDOG TIMER**
A/D converter		Operable
Programmable gain amplifier		Operable
Comparator		Only CMP0 and CMP2 are able to operate, on condition that the setting for release from the STOP mode in response to comparator interrupt detection has been made (CMPnSTEN in the PFSEL0 register is set to 1; n = 0, 2) and the noise filter is not in use.
Serial array unit 0 (SAU0)		Operable only CSI00 and UART0 only. Operation disabled other than CSI00 and UART0.
Serial array unit 4 (DALI/UART4)		Operable (Operation is disabled while in DALI Mode)
Serial interface (IICA)		Operation disabled
Multiplier and divider/multiply-accumulator		
DMA controller		
Power-on-reset function		Operable
Voltage detection function		
External interrupt		Operable (The operations of INTP20 and INTP21 are disabled while in INTP simultaneous mode of timer KB0 to KB2.)
CRC operation function		Operation stopped
RAM parity error detection function		
RAM guard function		
SFR guard function		
Illegal-memory access detection function		
PLL function		Operation disabled

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

Remark Operation stopped: Operation is automatically stopped before switching to the STOP mode.
Operation disabled: Operation is stopped before switching to the STOP mode.
f_{IH}: High-speed on-chip oscillator clock
f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock
f_{EX}: External main system clock
f_{XT}: XT1 clock
f_{EXS}: External subsystem clock

(2) Timing diagram when the interrupt request signal is generated in the SNOOZE mode

Figure 21-5. When the Interrupt Request Signal is Generated in the SNOOZE Mode



Notes 1. For details of the standby release signal, see **Figure 20-1**.

2. Transition time from STOP mode to SNOOZE mode

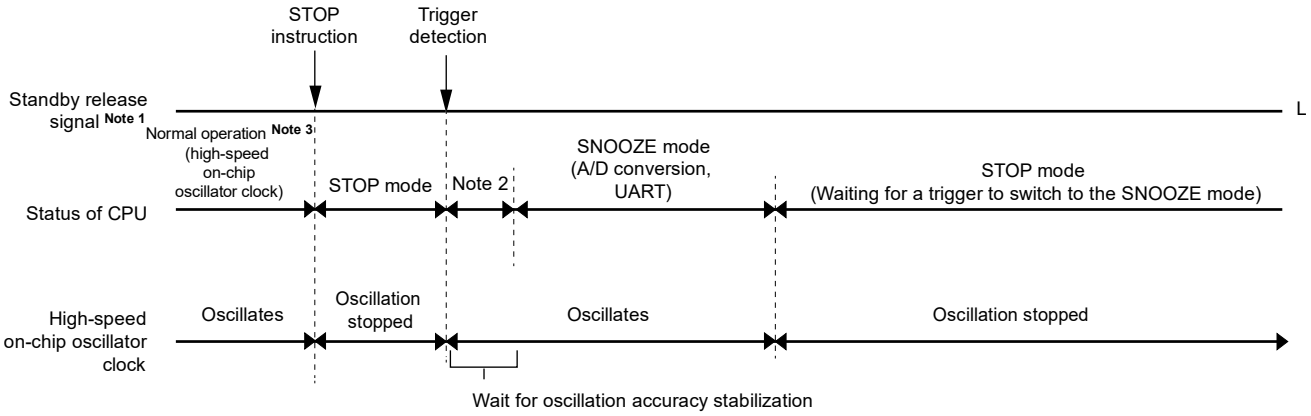
3. Transition time from SNOOZE mode to normal operation

4. Enable the SNOOZE mode (AWC = 1 or SWC = 1) immediately before switching to the STOP mode.

5. Be sure to release the SNOOZE mode (AWC = 0 or SWC = 0) immediately after return to the normal operation.

(3) Timing diagram when the interrupt request signal is not generated in the SNOOZE mode

Figure 21-6. When the Interrupt Request Signal is not Generated in the SNOOZE Mode



Notes 1. For details of the standby release signal, see **Figure 20-1**.

2. Transition time from STOP mode to SNOOZE mode

3. Enable the SNOOZE mode (AWC = 1 or SWC = 1) immediately before switching to the STOP mode.

Remark For details of the SNOOZE mode function, see **CHAPTER 12 A/D CONVERTER, CHAPTER 15 SERIAL ARRAY UNIT 0**, and **CHAPTER 16 SERIAL ARRAY UNIT 4 (DALI/UART4)**.

CHAPTER 22 RESET FUNCTION

The following seven operations are available to generate a reset signal.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer program loop detection
(3) Internal reset by comparison of supply voltage and detection voltage of power-on-reset (POR) circuit
(4) Internal reset by comparison of supply voltage of the voltage detector (LVD) and detection voltage
(5) Internal reset by execution of illegal instruction[Note]
(6) Internal reset by RAM parity error
(7) Internal reset by illegal-memory access

External and internal resets start program execution from the address at 0000H and 0001H when the reset signal is generated.

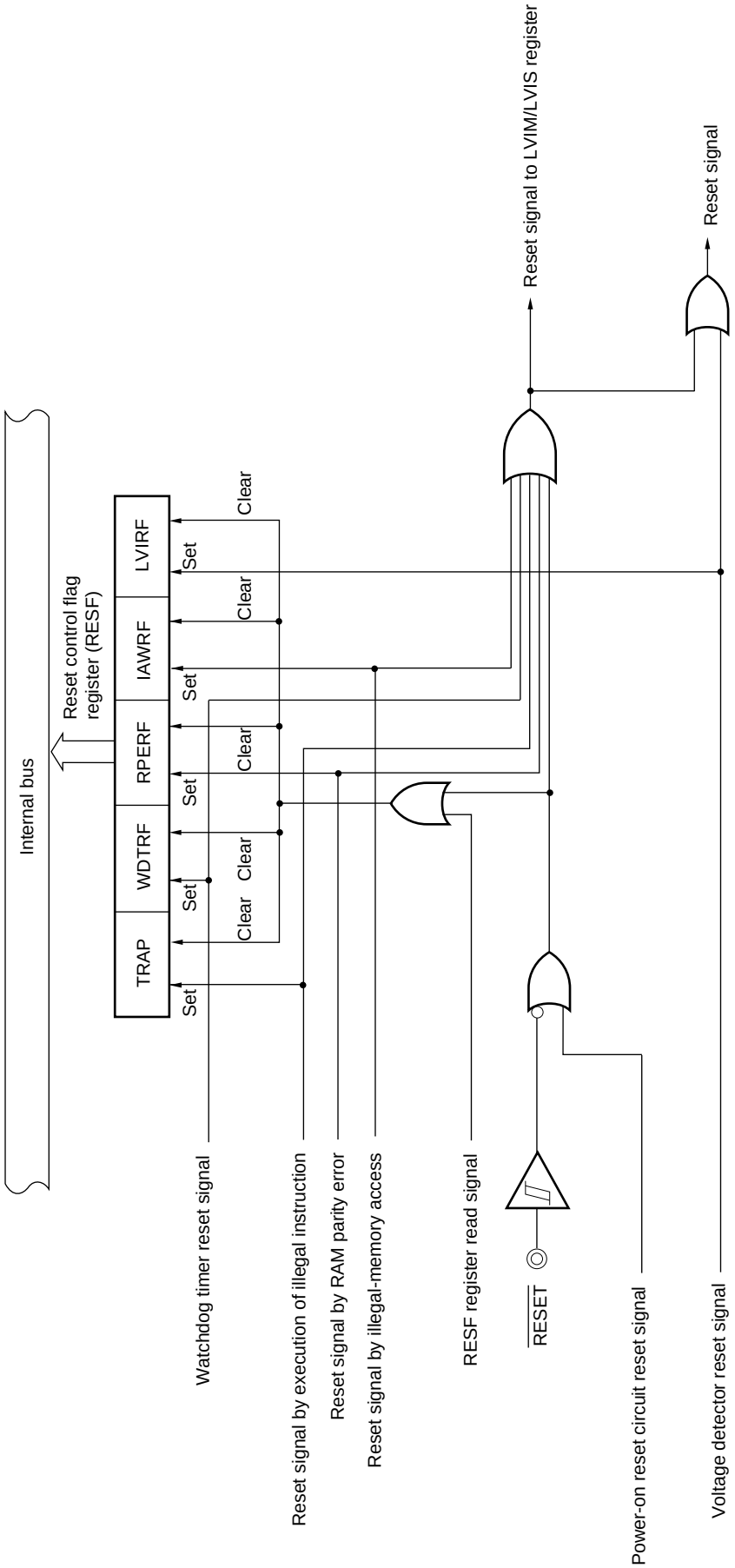
A reset is effected when a low level is input to the $\overline{\text{RESET}}$ pin, the watchdog timer overflows, or by POR and LVD circuit voltage detection, execution of illegal instruction[Note], RAM parity error or illegal-memory access, and each item of hardware is set to the status shown in Table 22-1.

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

Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
To perform an external reset upon power application, input a low level to the $\overline{\text{RESET}}$ pin, turn power on, continue to input a low level to the pin for 10 us or more within the operating voltage range shown in 32.4 or 33.4 AC Characteristics, and then input a high level to the pin.
2. During reset input, the X1 clock, XT1 clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock stop oscillating. External main system clock input and external subsystem clock input become invalid.
3. The port pins become the following state because each SFR and 2nd SFR are initialized after reset.
- **P40: High-impedance during the external reset period or reset period by the POR. High level during other types of reset or after receiving a reset signal (connected to the internal pull-up resistor).**
- **Ports other than P40: High-impedance during the reset period or after receiving a reset signal.**

Remark V_{POR}: POR power supply rise detection voltage
V_{LVD}: LVD detection voltage

Figure 22-1. Block Diagram of Reset Function



Caution An LVD circuit internal reset does not reset the LVD circuit.

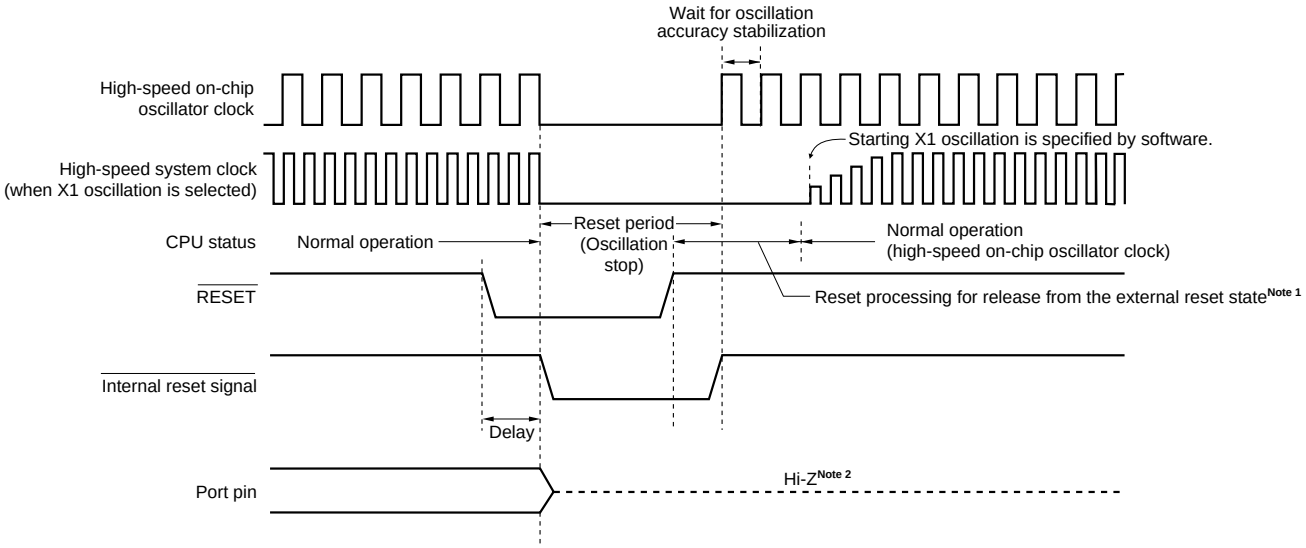
Remarks 1. LVIM: Voltage detection register

2. LVIS: Voltage detection level register

22.1 Timing of Reset Operation

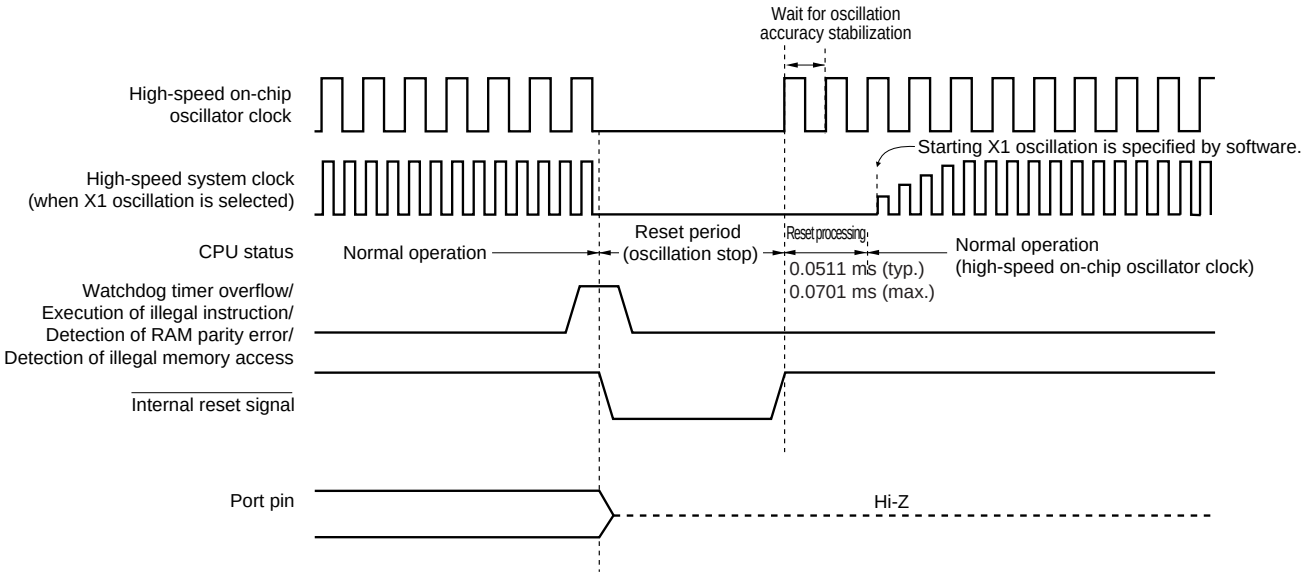
This LSI is reset by input of the low level on the $\overline{\text{RESET}}$ pin and released from the reset state by input of the high level on the $\overline{\text{RESET}}$ pin. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

Figure 22-2. Timing of Reset by $\overline{\text{RESET}}$ Input



Release from the reset state is automatic in the case of a reset due to a watchdog timer overflow, execution of an illegal instruction, detection of a RAM parity error, or detection of illegal memory access. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

Figure 22-3. Timing of Reset Due to Execution of Illegal Instruction, Watchdog Timer, RAM Parity Error, or Illegal Memory Access



(**Notes** are listed on the next page.)

Notes 1. Reset times (times for release from the external reset state)

After the first release of the POR: 0.672 ms (typ.), 0.832 ms (max.) when the LVD is in use.
0.399 ms (typ.), 0.519 ms (max.) when the LVD is off.

After the second release of the POR: 0.531 ms (typ.), 0.675 ms (max.) when the LVD is in use.
0.259 ms (typ.), 0.362 ms (max.) when the LVD is off.

After power is supplied, a voltage stabilization waiting time of about 0.99 ms (typ.) and up to 2.30 ms (max.) is required before reset processing starts after release of the external reset

2. The state of P40 is as follows.

- High-impedance during the external reset period or reset period by the POR.
- High level during other types of reset or after receiving a reset signal (connected to the internal pull-up resistor).

Reset by POR and LVD circuit supply voltage detection is automatically released when $V_{DD} \geq V_{POR}$ or $V_{DD} \geq V_{LVD}$ after the reset. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts. For details, see **CHAPTER 23 POWER-ON-RESET CIRCUIT** or **CHAPTER 24 VOLTAGE DETECTOR**.

22.2 States of Operation During Reset Periods

Table 22-1 shows the states of operation during reset periods. Table 22-2 shows the states of the hardware after receiving a reset signal.

Table 22-1. Operation Statuses During Reset Period

Item			During Reset Period
System clock			Clock supply to the CPU is stopped.
	Main system clock	f_{IH}	Operation stopped
		f_X	Operation stopped (the X1 and X2 pins are input port mode)
		f_{EX}	Clock input invalid (the pin is input port mode)
	Subsystem clock	f_{XT}	Operation stopped (the XT1 and XT2 pins are input port mode)
		f_{EXS}	Clock input invalid (the pin is input port mode)
	f_{IL}		Operation stopped
CPU			
Code flash memory			Operation stopped
Data flash memory			Operation stopped
RAM			Operation stopped
Port (latch)			High impedance[Note]
Timer array unit			Operation stopped
Timer KB0 to KB2			
Timer KC0			
Real-time clock (RTC)			
12-bit interval timer			
Watchdog timer			
A/D converter			
Programmable gain amplifier			
Comparator			
Serial array unit 0 (SAU0)			
Serial array unit 4 (DALI/UART4)			
Serial interface (IICA)			
Multiplier & divider, multiply-accumulator			
DMA controller			
Power-on-reset function			Detection operation possible
Voltage detection function			Operation is possible in the case of an LVD reset and stopped in the case of other types of reset.
External interrupt			Operation stopped
CRC operation function	High-speed CRC		
	General-purpose CRC		
RAM parity error detection function			
RAM guard function			
SFR guard function			
Illegal-memory access detection function			
PLL function			

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

Note P40 becomes the following state.

- P40: High-impedance during the external reset period or reset period by the POR. High level during other types of reset (connected to the internal pull-up resistor).

Remark f_{IH}: High-speed on-chip oscillator clock
f_X: X1 oscillation clock
f_{EX}: External main system clock
f_{XT}: XT1 oscillation clock
f_{EXS}: External subsystem clock
f_{IL}: Low-speed on-chip oscillator clock

Table 22-2. Hardware Statuses After Reset Acknowledgment

Hardware		After Reset Acknowledgment[Note 1]
Program counter (PC)		The contents of the reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		06H
RAM	Data memory	Undefined
	General-purpose registers	Undefined

Note During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

Remark The special function register (SFR) mounted depend on the product. See **3.1.4 Special function register (SFR) area** and **3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area**.

22.3 Register for Confirming Reset Source

22.3.1 Reset control flag register (RESF)

Many internal reset generation sources exist in the RL78 microcontroller. The reset control flag register (RESF) is used to store which source has generated the reset request.

The RESF register can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input, reset by power-on-reset (POR) circuit, and reading the RESF register clear TRAP, WDTRF, RPERF, IAWRF, and LVIRF flags.

Figure 22-4. Format of Reset Control Flag Register (RESF)

Address: FFFA8H After reset: Undefined[Note 1] R

Symbol	7	6	5	4	3	2	1	0
RESF	TRAP	0	0	WDTRF	0	RPERF	IAWRF	LVIRF

TRAP	Internal reset request by execution of illegal instruction[Note 2]
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

WDTRF	Internal reset request by watchdog timer (WDT)
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

RPERF	Internal reset request t by RAM parity
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

IAWRF	Internal reset request t by illegal-memory access
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

LVIRF	Internal reset request by voltage detector (LVD)
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

Notes 1. The value after reset varies depending on the reset source. See **Table 22-3**.

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

Cautions 1. Do not read data by a 1-bit memory manipulation instruction.

2. When enabling RAM parity error resets (RPERDIS = 0), be sure to initialize the used RAM area at data access or the used RAM area + 10 bytes at execution of instruction from the RAM area.
Reset generation enables RAM parity error resets (RPERDIS = 0). For details, see 25.3.3 RAM parity error detection function.

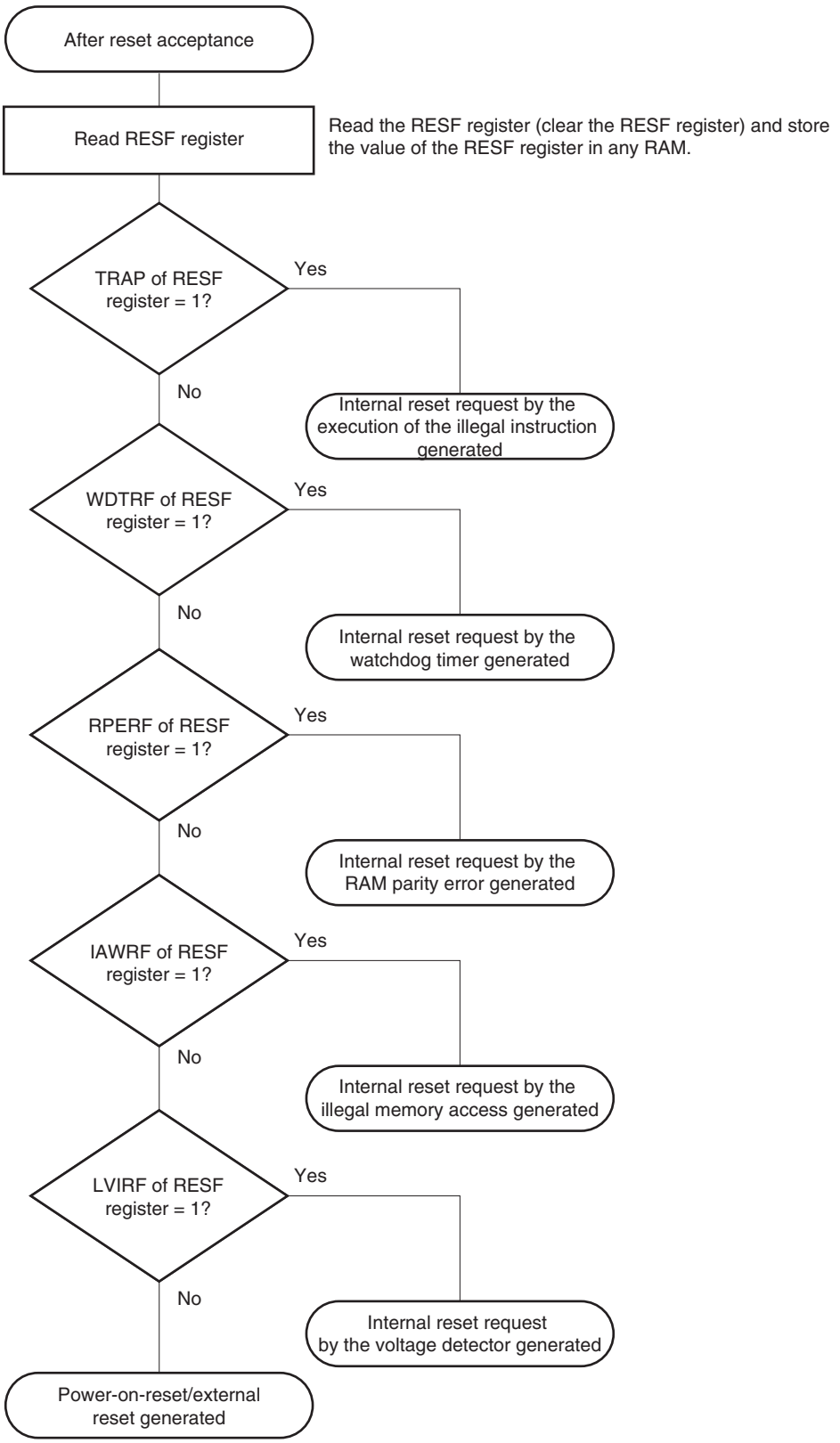
The status of the RESF register when a reset request is generated is shown in Table 22-3.

Table 22-3. RESF Register Status When Reset Request Is Generated

Flag \ Reset Source	$\overline{\text{RESET}}$ Input	Reset by POR	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by RAM Parity Error	Reset by Illegal-memory Access	Reset by LVD
TRAP bit	Cleared (0)	Cleared (0)	Set (1)	Held	Held	Held	Held
WDTRF bit			Held	Set (1)			
RPERF bit				Held	Set (1)		
IAWRF bit					Held	Set (1)	
LVIRF bit						Held	Set (1)

The RESF register is automatically cleared when it is read by an 8-bit memory manipulation instruction.
Figure 22-5 shows the procedure for checking a reset source.

Figure 22-5. Procedure for Checking Reset Source



The flow described above is an example of the procedure for checking.

CHAPTER 23 POWER-ON-RESET CIRCUIT

23.1 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 **32.4** or **33.4 AC Characteristics**. This is done by utilizing the voltage detection circuit or controlling the externally input reset signal.

- Compares supply voltage (V_{DD}) and detection voltage (V_{PDR}), generates internal reset signal when $V_{DD} < V_{PDR}$. Note that, after power is supplied, this LSI should be placed in the STOP mode, or in the reset state by utilizing the voltage detection circuit or externally input reset signal, before the operation voltage falls below the range defined in **32.4** or **33.4 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Caution If an internal reset signal is generated in the power-on-reset circuit, the reset control flag register (RESF) is cleared to 00H.

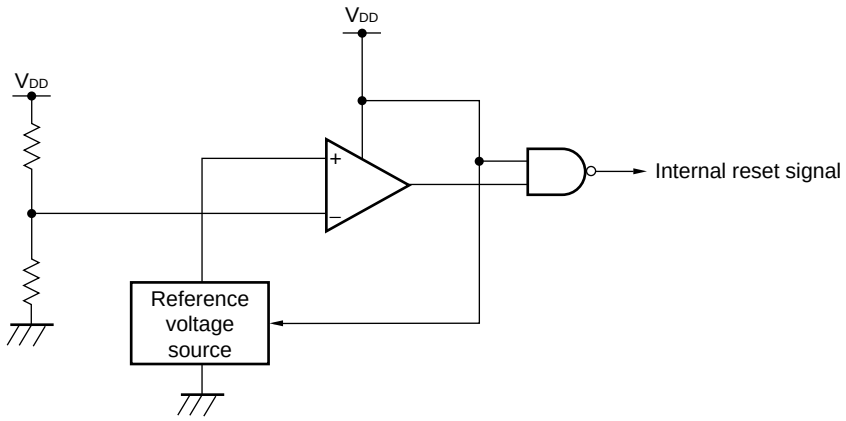
Remarks 1. The RL78 microcontroller incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset source is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT), voltage-detector (LVD), illegal instruction execution, RAM parity error, or illegal-memory access. The RESF register is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by the watchdog timer (WDT), voltage-detector (LVD), illegal instruction execution, RAM parity error, or illegal-memory access.
For details of the RESF register, see **CHAPTER 22 RESET FUNCTION**.

2. V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage
For details, see **32.6.5** or **33.6.5 POR circuit characteristics**.

23.2 Configuration of Power-on-reset Circuit

The block diagram of the power-on-reset circuit is shown in Figure 23-1.

Figure 23-1. Block Diagram of Power-on-reset Circuit

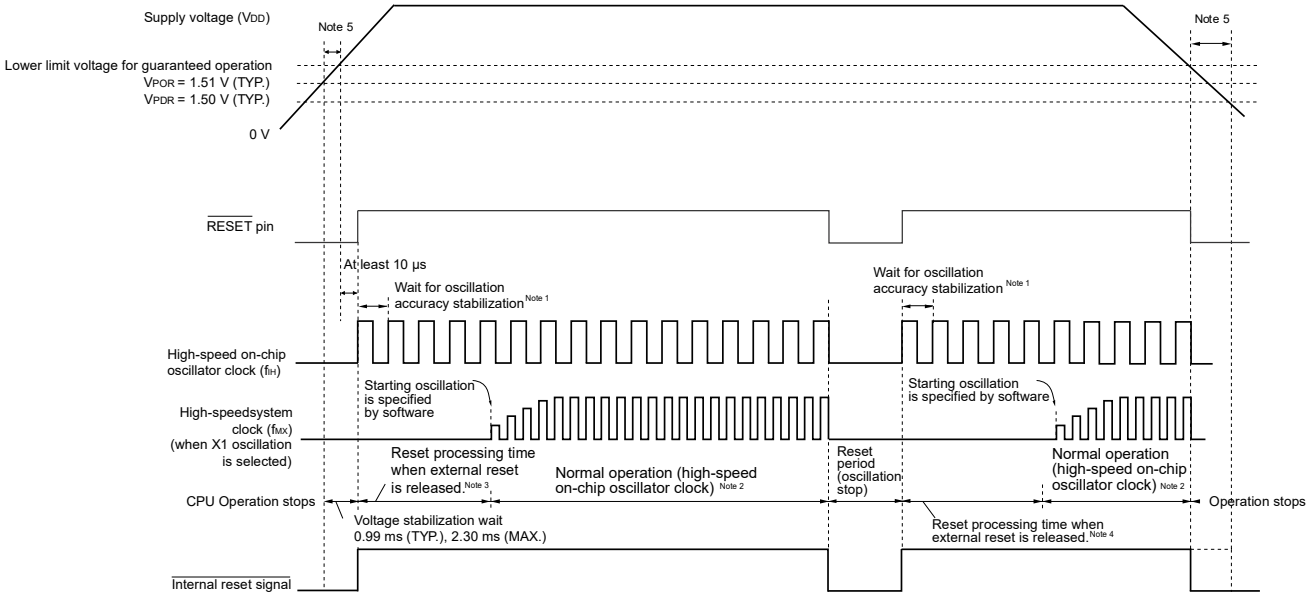


23.3 Operation of Power-on-reset Circuit

The timing of generation of the internal reset signal by the power-on-reset circuit and voltage detector is shown below.

Figure 23-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (1/3)

(1) When the externally input reset signal on the $\overline{\text{RESET}}$ pin is used



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

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.

3. The time until normal operation starts includes the following reset processing time when the external reset is released (after the first release of POR) after the $\overline{\text{RESET}}$ signal is driven high (1) as well as the voltage stabilization wait time after VPOR (1.51 V, typ.) is reached.

Reset processing time when the external reset is released is shown below.

After the first release of POR: 0.672 ms (typ.), 0.832 ms (max.) (when the LVD is in use)
0.399 ms (typ.), 0.519 ms (max.) (when the LVD is off)

4. Reset processing time when the external reset is released after the second release of POR is shown below.

After the second release of POR: 0.531 ms (typ.), 0.675 ms (max.) (when the LVD is in use)
0.259 ms (typ.), 0.362 ms (max.) (when the LVD is off)

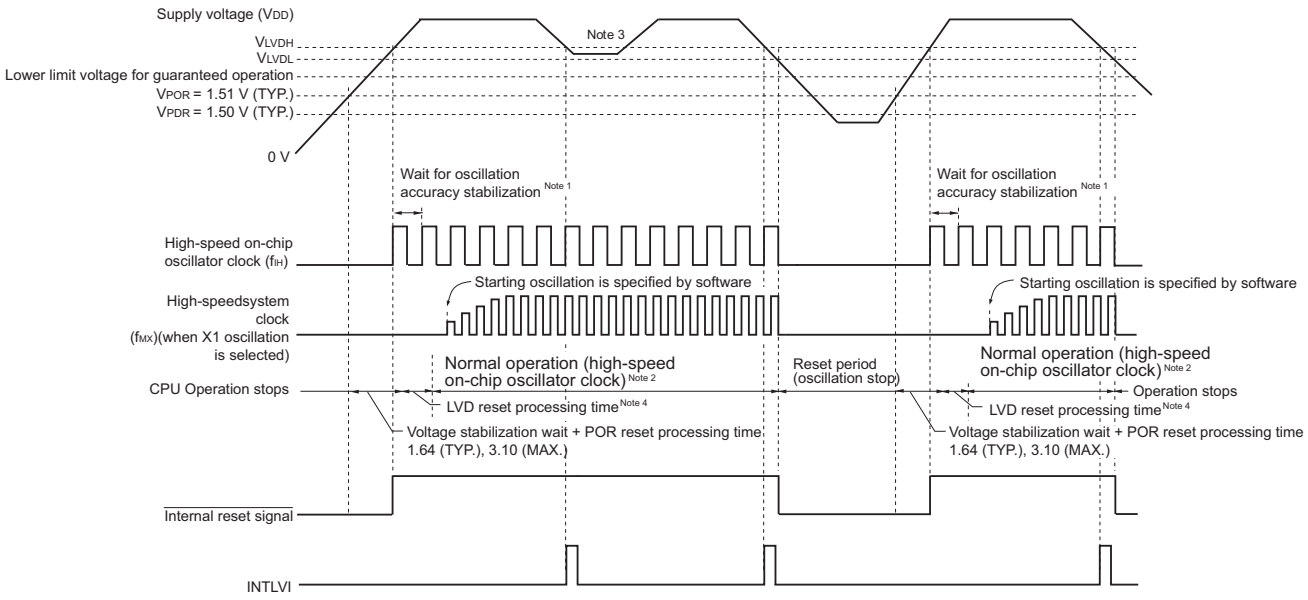
5. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **32.4** or **33.4 AC Characteristics**. This is done by controlling the externally input reset signal. After power supply is turned off, this LSI should be placed in the STOP mode, or in the reset state by utilizing the voltage detection circuit or externally input reset signal, before the voltage falls below the operating range. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Remark VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

Caution For power-on reset, be sure to use the externally input reset signal on the $\overline{\text{RESET}}$ pin when the LVD is off. For details, see CHAPTER 24 VOLTAGE DETECTOR.

Figure 23-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (2/3)

(2) LVD interrupt & reset mode (option byte 000C1H: LVIMDS1, LVIMDS0 = 1, 0)



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

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.

3. After the interrupt request signal (INTLVI) is generated, the LVILV and LVIMD bits of the voltage detection level register (LVIS) are automatically set to 1. After INTLVI is generated, appropriate settings should be made according to **Figure 24-7 Processing Procedure After an Interrupt Is Generated** and **Figure 24-8 Initial Setting of Interrupt and Reset Mode**, taking into consideration that the supply voltage might return to the high voltage detection level (V_{LVDH}) or higher without falling below the low voltage detection level (V_{LVDL}).

4. The time until normal operation starts includes the following LVD reset processing time after the LVD detection level (V_{LVDH}) is reached as well as the voltage stabilization wait + POR reset processing time after the V_{POR} (1.51 V, typ.) is reached.

LVD reset processing time: 0 ms to 0.0701 ms (max.)

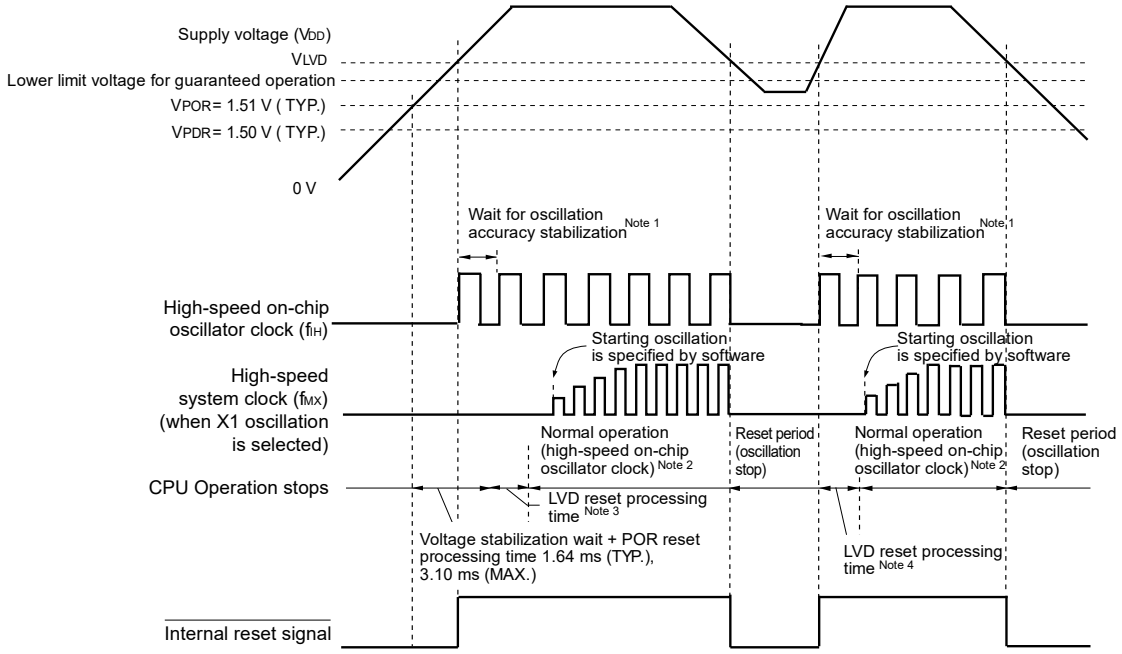
Remark V_{LVDH}, V_{LVDL}: LVD detection voltage

V_{POR}: POR power supply rise detection voltage

V_{PDR}: POR power supply fall detection voltage

Figure 23-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (3/3)

(3) LVD reset mode (option byte 000C1H: LVIMDS1, LVIMDS0 = 1, 1)



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

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.

3. The time until normal operation starts includes the following LVD reset processing time after the LVD detection level (V_{LVDH}) is reached as well as the voltage stabilization wait + POR reset processing time after the V_{POR} (1.51 V, typ.) is reached.
LVD reset processing time: 0 ms to 0.0701 ms (max.)

4. When the power supply voltage is below the lower limit for operation and the power supply voltage is then restored after an internal reset is generated only by the voltage detector (LVD), the following LVD reset processing time is required after the LVD detection level (V_{LVD}) is reached.
LVD reset processing time: 0.0511 ms (typ.), 0.0701 ms (max.)

Remarks 1. V_{LVDH}, V_{LVDL}: LVD detection voltage
V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

2. When the LVD interrupt mode is selected (option byte 000C1H: LVIMD1 = 0, LVIMD0 = 1), the time until normal operation starts after power is turned on is the same as the time specified in Note 3 of Figure 23-2 (3).

CHAPTER 24 VOLTAGE DETECTOR

24.1 Functions of Voltage Detector

The operation mode and detection voltages (V_{LVDH}, V_{LVDL}, V_{LVD}) for the voltage detector is set by using the option byte (000C1H). The voltage detector (LVD) has the following functions.

- The LVD circuit compares the supply voltage (V_{DD}) with the detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}), and generates an internal reset or internal interrupt signal.
- The detection level for the power supply detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}) can be selected by using the option byte as one of 6 levels (for details, see **CHAPTER 27 OPTION BYTE**).
- Operable in STOP mode.
- After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **32.4** or **33.4 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 010C2H).

(a) Interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0)
The two detection voltages (V_{LVDH}, V_{LVDL}) are selected by the option byte 000C1H. The high-voltage detection level (V_{LVDH}) is used for releasing resets and generating interrupts. The low-voltage detection level (V_{LVDL}) is used for generating resets.

(b) Reset mode (option byte LVIMDS1, LVIMDS0 = 1, 1)
The detection voltage (V_{LVD}) selected by the option byte 000C1H is used for triggering and ending resets.

(c) Interrupt mode (option byte LVIMDS1, LVIMDS0 = 0, 1)
The detection voltage (V_{LVD}) selected by the option byte 000C1H is used for releasing resets/generating interrupts.

The reset and internal interrupt signals are generated in each mode as follows.

Interrupt & Reset Mode (LVIMDS1, LVIMDS0 = 1, 0)	Reset Mode (LVIMDS1, LVIMDS0 = 1, 1)	Interrupt Mode (LVIMDS1, LVIMDS0 = 0, 1)
Generates an interrupt request signal by detecting $V_{DD} < V_{LVDH}$ when the operating voltage falls, and an internal reset by detecting $V_{DD} < V_{LVDL}$. Releases an internal reset by detecting $V_{DD} \geq V_{LVDH}$.	Releases an internal reset by detecting $V_{DD} \geq V_{LVD}$. Generates an interrupt request signal by detecting $V_{DD} < V_{LVD}$.	Retains the state of an internal reset by the LVD immediately after a reset until $V_{DD} \geq V_{LVD}$. Releases the LVD internal reset by detecting $V_{DD} \geq V_{LVD}$. Generates an interrupt request signal (INTLVI) by detecting $V_{DD} < V_{LVD}$ or $V_{DD} \geq V_{LVD}$ after the LVD internal reset is released.

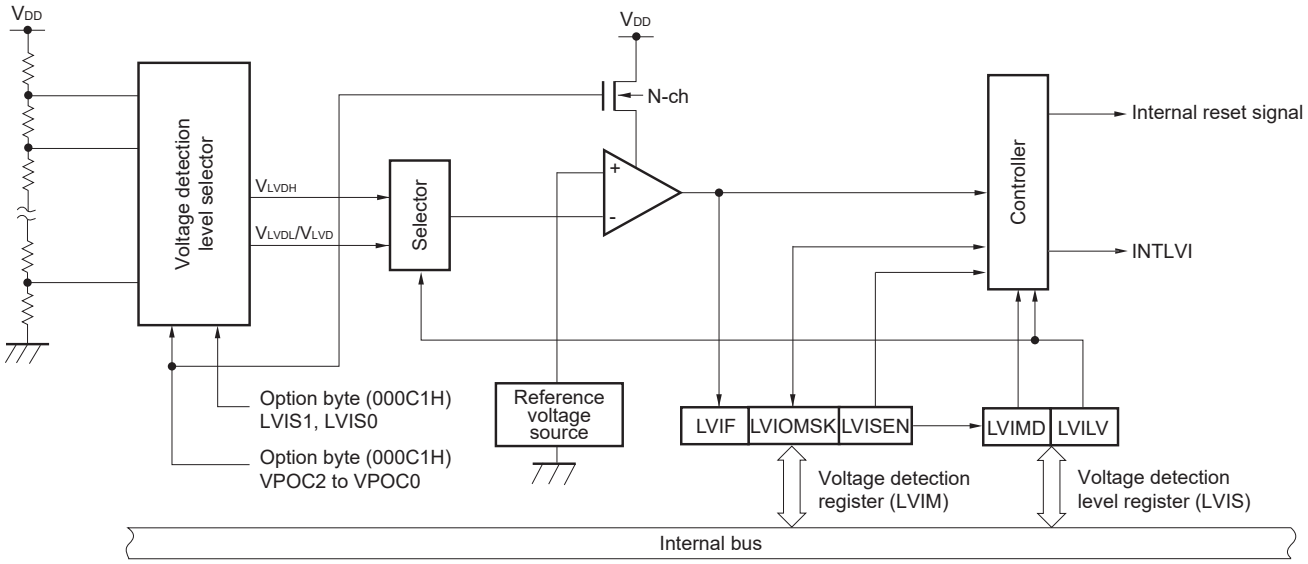
While the voltage detector is operating, whether the supply voltage is more than or less than the detection level can be checked by reading the voltage detection flag (LVIF: bit 0 of the voltage detection register (LVIM)).

Bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of the RESF register, see **CHAPTER 22 RESET FUNCTION**.

24.2 Configuration of Voltage Detector

The block diagram of the voltage detector is shown in Figure 24-1.

Figure 24-1. Block Diagram of Voltage Detector



24.3 Registers Controlling Voltage Detector

The voltage detector is controlled by the following registers.

- Voltage detection register (LVIM)
- Voltage detection level register (LVIS)

24.3.1 Voltage detection register (LVIM)

This register is used to specify whether to enable or disable rewriting the voltage detection level register (LVIS), as well as to check the LVD output mask status.

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

Reset signal generation clears this register to 00H.

Figure 24-2. Format of Voltage Detection Register (LVIM)

Address: FFFA9H After reset: 00H[Note 1] R/W[Note 2]

Symbol	<7>	6	5	4	3	2	<1>	<0>
LVIM	LVISEN Note 3	0	0	0	0	0	LVIOMSK	LVIF

LVISEN Note 3	Specification of whether to enable or disable rewriting the voltage detection level register (LVIS)
0	Disabling of rewriting the LVIS register (LVIOMSK = 0 (Mask of LVD output is invalid))
1	Enabling of rewriting the LVIS register (LVIOMSK = 1 (Mask of LVD output is valid))

LVIOMSK	Mask status flag of LVD output
0	Mask of LVD output is invalid
1	Mask of LVD output is valid[Note 4]

LVIF	Voltage detection flag
0	Supply voltage (V_{DD}) ≥ detection voltage (V_{LVD}), or when LVD is off
1	Supply voltage (V_{DD}) < detection voltage (V_{LVD})

Notes 1. The reset value changes depending on the reset source.
If the LVIS register is reset by LVD, it is not reset but holds the current value. In other reset, LVISEN is cleared to 0.

2. Bits 0 and 1 are read-only.

3. LVISEN and LVIOMSK can only be set in the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0). Do not change the initial value in other modes.

4. LVIOMSK bit is only automatically set to “1” when the interrupt & reset mode is selected (option byte LVIMDS1, LVIMDS0 = 1, 0) and reset or interrupt by LVD is masked.
- Period during LVISEN = 1
- Waiting period from the time when LVD interrupt is generated until LVD detection voltage becomes stable
- Waiting period from the time when the value of LVILV bit changes until LVD detection voltage becomes stable

24.3.2 Voltage detection level register (LVIS)

This register selects the voltage detection level.

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

Reset signal generation input sets this register to 00H/01H/81H[Note 1].

Figure 24-3. Format of Voltage Detection Level Select Register (LVIS)

Address: FFFAAH After reset: 00H/01H/81H[Note 1] R/W

Symbol	<7>	6	5	4	3	2	1	<0>
LVIS	LVIMD	0	0	0	0	0	0	LVILV

LVIMD[Note 2]	Operation mode of voltage detection
0	Interrupt mode
1	Reset mode

LVILV[Note 2]	LVD detection level
0	High-voltage detection level (V_{LVDH})
1	Low-voltage detection level (V_{LVDL} or V_{LVD})

Notes 1. The reset value changes depending on the reset source and the setting of the option byte.
This register is not cleared (00H) by LVD reset.
The generation of reset signal other than an LVD reset sets as follows.
- When option byte LVIMDS1, LVIMDS0 = 1, 0: 00H
- When option byte LVIMDS1, LVIMDS0 = 1, 1: 81H
- When option byte LVIMDS1, LVIMDS0 = 0, 1: 01H

2. Writing “0” can only be allowed in the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0). Do not set LVIMD and LVILV in other cases. The value is switched automatically when reset or interrupt is generated in the interrupt & reset mode.

Cautions 1. Rewrite the value of the LVIS register according to Figures 24-7 and 24-8.

2. Specify the LVD operation mode and detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}) of each mode by using the option byte 000C1H. Table 24-1 shows the format of the user option byte (000C1H/010C1H). For details about the option byte, see CHAPTER 27 OPTION BYTE.

Table 24-1. LVD Operation Mode and Detection Voltage Settings for User Option Byte (000C1H) (1/2)

Address: 000C1H/010C1H[Note]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD setting (interrupt & reset mode)

Detection Voltage			Option Byte Setting Value						
V_{LVDH}		V_{LVDL}	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode Setting	
Rising Edge	Falling Edge	Falling Edge						LVIMDS1	LVIMDS0
2.92 V	2.86 V	2.75 V	0	1	1	1	0	1	0
3.02 V	2.96 V					0	1		
4.06 V	3.98 V					0	0		
–			Setting of values other than above is prohibited.						

- LVD setting (reset mode)

Detection Voltage		Option Byte Setting Value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode Setting	
Rising Edge	Falling Edge						LVIMDS1	LVIMDS0
2.81 V	2.75 V	0	1	1	1	1	1	1
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
–		Setting of values other than above is prohibited.						

- LVD setting (interrupt mode)

Detection Voltage		Option Byte Setting Value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode Setting	
Rising Edge	Falling Edge						LVIMDS1	LVIMDS0
2.81 V	2.75 V	0	1	1	1	1	0	1
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
–		Setting of values other than above is prohibited.						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

(**Cautions** and **Remarks** are listed on the next page.)

Table 24-1. LVD Operation Mode and Detection Voltage Settings for User Option Byte (000C1H) (2/2)

Address: 000C1H/010C1H[Note]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD off setting (use of external reset input via RESET pin)

Detection Voltage		Option Byte Setting Value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode Setting	
Rising Edge	Falling Edge						LVIMDS1	LVIMDS0
–	–	1	×	×	×	×	×	1
–		Setting of values other than above is prohibited.						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Cautions 1. Be sure to set bit 4 to "1".

2. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 32.4 or 33.4 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 010C2H).

Remarks 1. ×: don't care

2. For details on the LVD circuit, see **CHAPTER 24 VOLTAGE DETECTOR**.

3. The detection voltage is a typical value. For details, see **32.6.6** or **33.6.6 LVD circuit characteristics**.

24.4 Operation of Voltage Detector

24.4.1 When used as reset mode

Specify the operation mode (the reset mode (LVIMDS1, LVIMDS0 = 1, 1)) and the detection voltage (V_{LVD}) by using the option byte 000C1H.

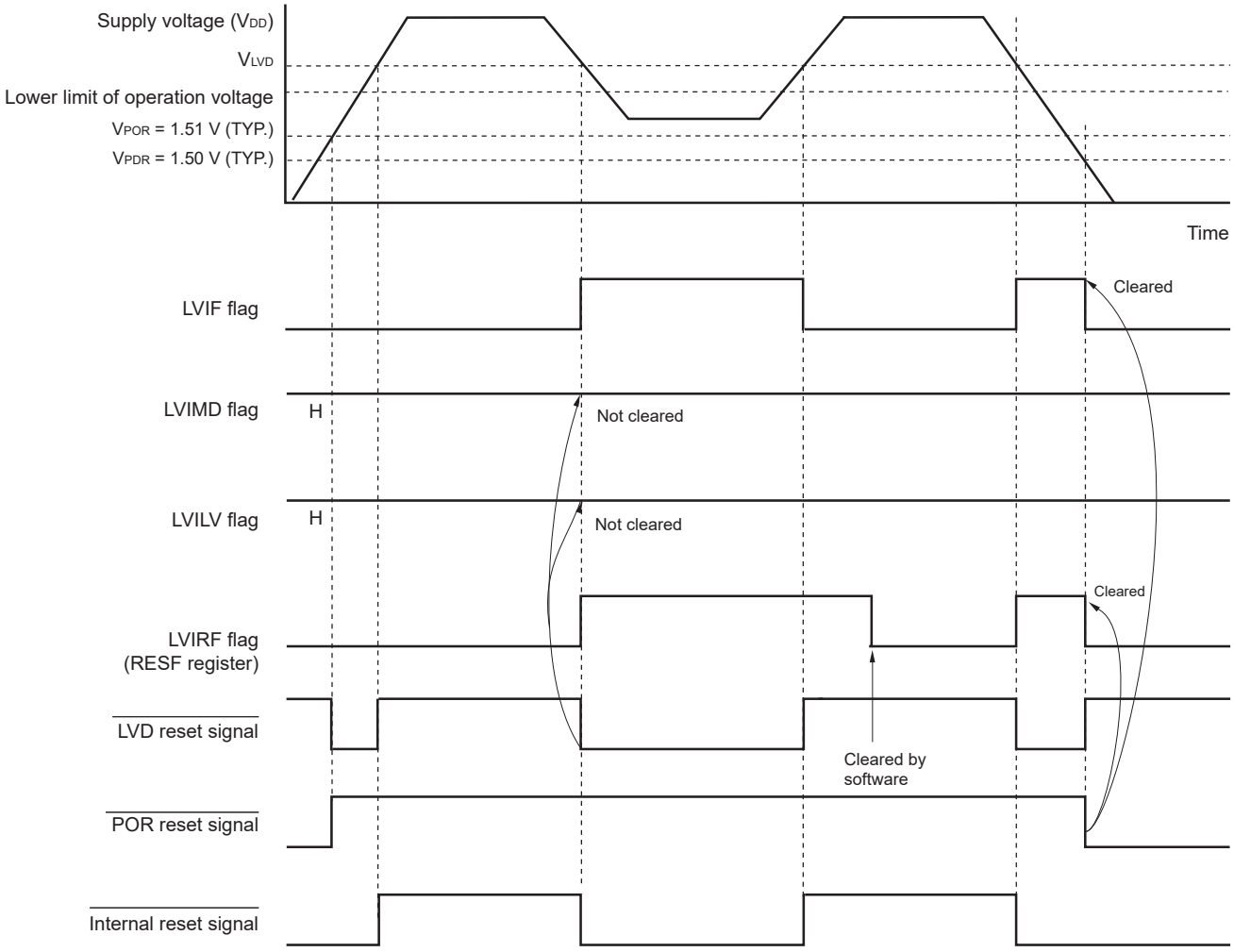
The operation is started in the following initial setting state when the reset mode is set.

- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- The initial value of the voltage detection level select register (LVIS) is set to 81H.
  Bit 7 (LVIMD) is 1 (reset mode).
  Bit 0 (LVILV) is 1 (low-voltage detection level: V_{LVD}).

- Operation in LVD reset mode
  In the reset mode (option byte LVIMDS1, LVIMDS0 = 1, 1), the state of an internal reset by LVD is retained until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}) after power is supplied. The internal reset is released when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}).
  At the fall of the operating voltage, an internal reset by LVD is generated when the supply voltage (V_{DD}) falls below the voltage detection level (V_{LVD}).

Figure 24-4 shows the timing of the internal reset signal generated in the LVD reset mode.

Figure 24-4. Timing of Voltage Detector Internal Reset Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 1)



Remark VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

24.4.2 When used as interrupt mode

Specify the operation mode (the interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)) and the detection voltage (V_{LVD}) by using the option byte 000C1H.

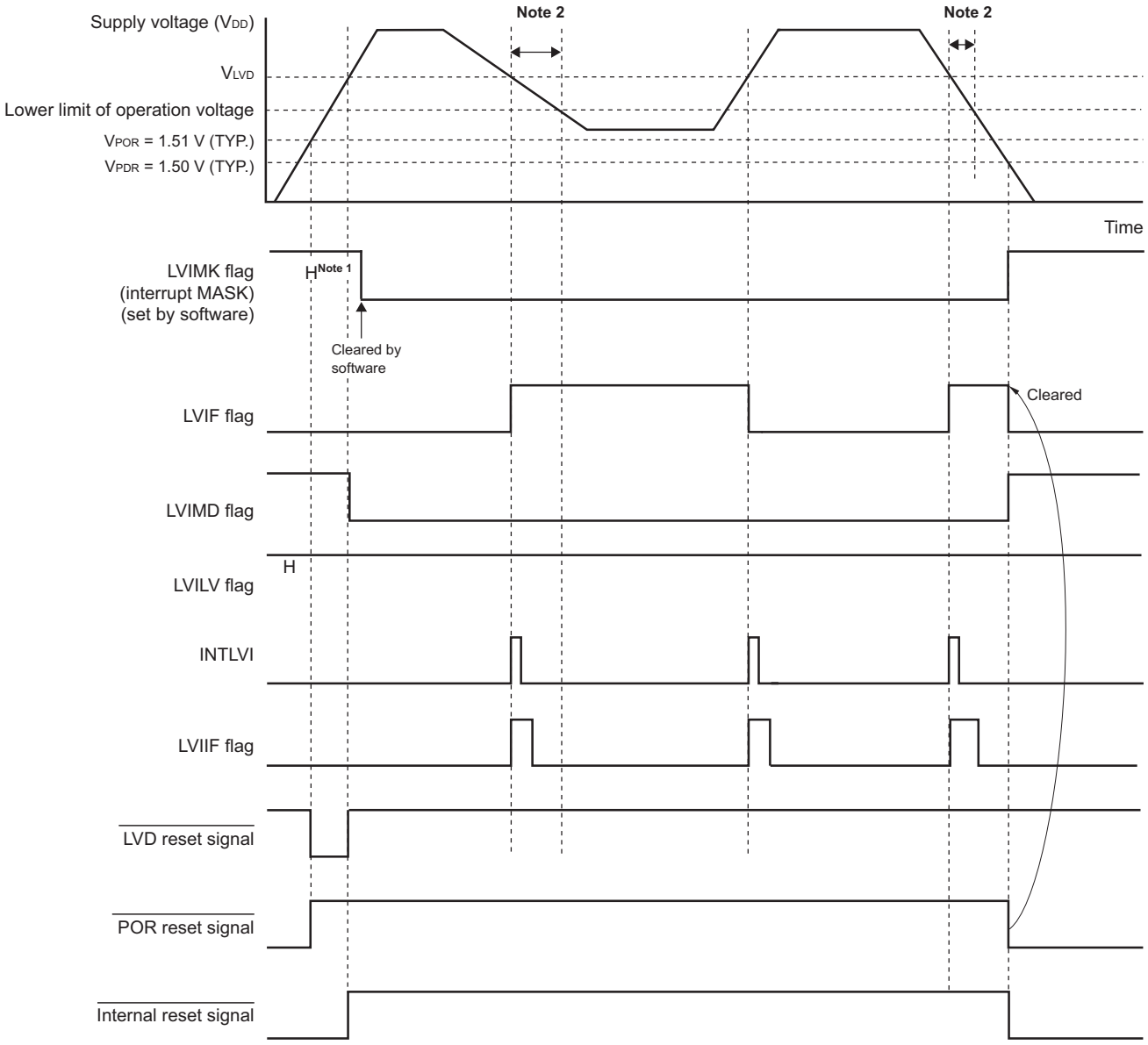
The operation is started in the following initial setting state when the interrupt mode is set.

- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- The initial value of the voltage detection level select register (LVIS) is set to 01H.
  Bit 7 (LVIMD) is 0 (interrupt mode).
  Bit 0 (LVILV) is 1 (low-voltage detection level: V_{LVD}).

- Operation in LVD interrupt mode
  In interrupt mode (LVIMDS1 and LVIMDS0 = 0 and 1 in the option byte), the state of an internal reset by the LVD is retained immediately after a reset until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}). The LVD internal reset is released when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}).
  After the LVD internal reset is released, an interrupt request signal (INTLVI) by the LVD is generated when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}). When the voltage falls, this LSI should be placed in the STOP mode, or placed in the reset state by controlling the externally input reset signal, before the voltage falls below the operating voltage range defined in **32.4** or **33.4 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Figure 24-5 shows the timing of the interrupt request signal generated in the LVD interrupt mode.

Figure 24-5. Timing of Voltage Detector Internal Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 0, 1)



Notes 1. The LVIMK flag is set to “1” by reset signal generation.

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 **32.4** or **33.4 AC characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

24.4.3 When used as interrupt and reset mode

Specify the operation mode (the interrupt & reset (LVIMDS1, LVIMDS0 = 1, 0)) and the detection voltage (V_{LVDH}, V_{LVDL}) by using the option byte 000C1H.

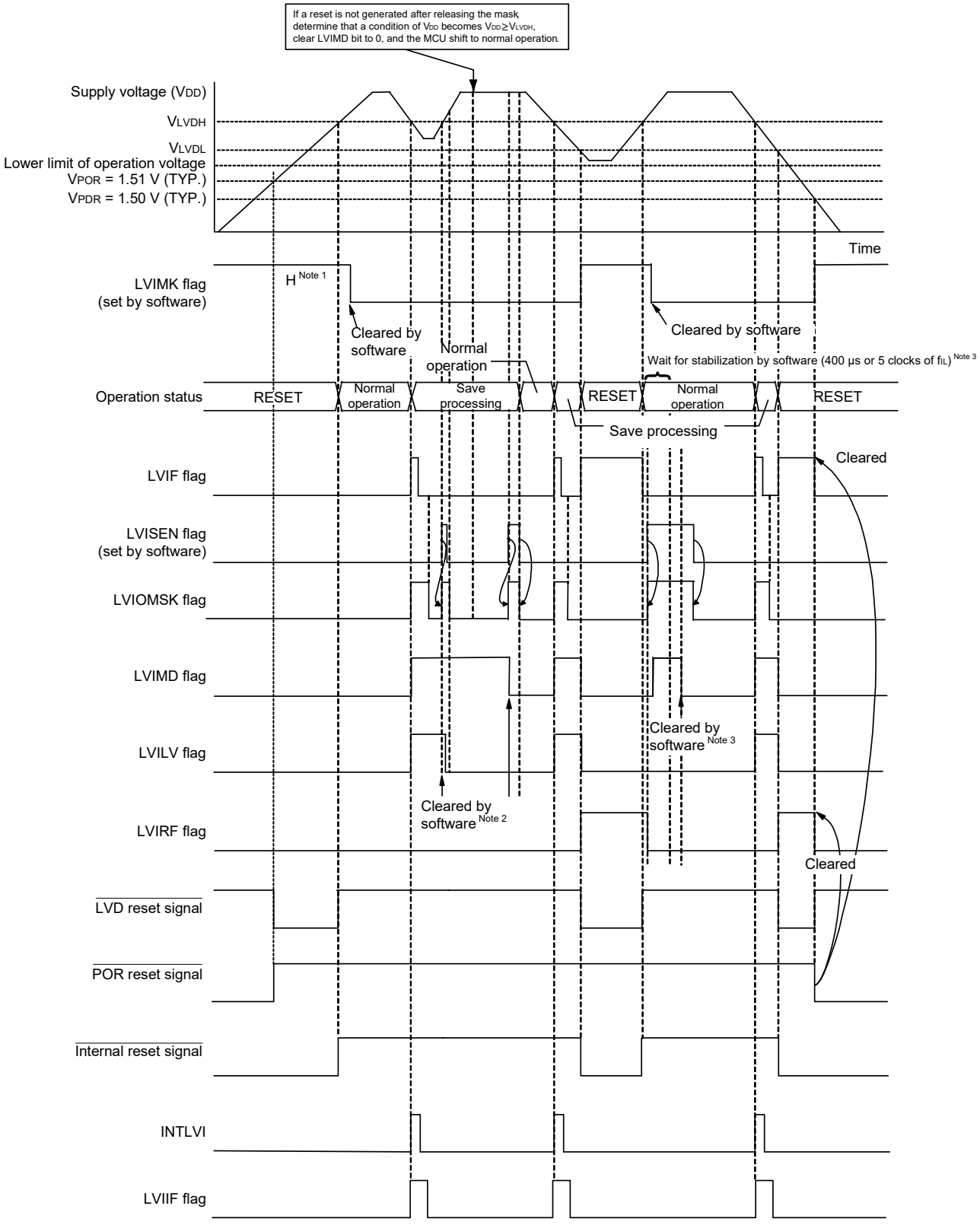
The operation is started in the following initial setting state when the interrupt & reset mode is set.

- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- The initial value of the voltage detection level select register (LVIS) is set to 00H.
  Bit 7 (LVIMD) is 0 (interrupt mode).
  Bit 0 (LVILV) is 0 (high-voltage detection level: V_{LVDH}).

- Operation in LVD interrupt & reset mode
  In the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0), the state of an internal reset by LVD is retained until the supply voltage (V_{DD}) exceeds the high-voltage detection level (V_{LVDH}) after power is supplied. The internal reset is released when the supply voltage (V_{DD}) exceeds the high-voltage detection level (V_{LVDH}).
  An interrupt request signal by LVD (INTLVD) is generated and arbitrary save processing is performed when the supply voltage (V_{DD}) falls below the high-voltage detection level (V_{LVDH}). After that, an internal reset by LVD is generated when the supply voltage (V_{DD}) falls below the low-voltage detection level (V_{LVDL}). After INTLVI is generated, an interrupt request signal is not generated even if the supply voltage becomes equal to or higher than the high-voltage detection voltage (V_{LVDH}) without falling below the low-voltage detection voltage (V_{LVDL}).
  To use the LVD reset & interrupt mode, perform the processing according to **Figure 24-7 Processing Procedure After an Interrupt Is Generated** and **Figure 24-8 Initial Setting of Interrupt and Reset Mode**.

Figure 24-6 shows the timing of the internal reset signal and interrupt signal generated in the LVD interrupt & reset mode.

Figure 24-6. Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (1/2)



(**Notes** and **Remark** are listed on the next page.)

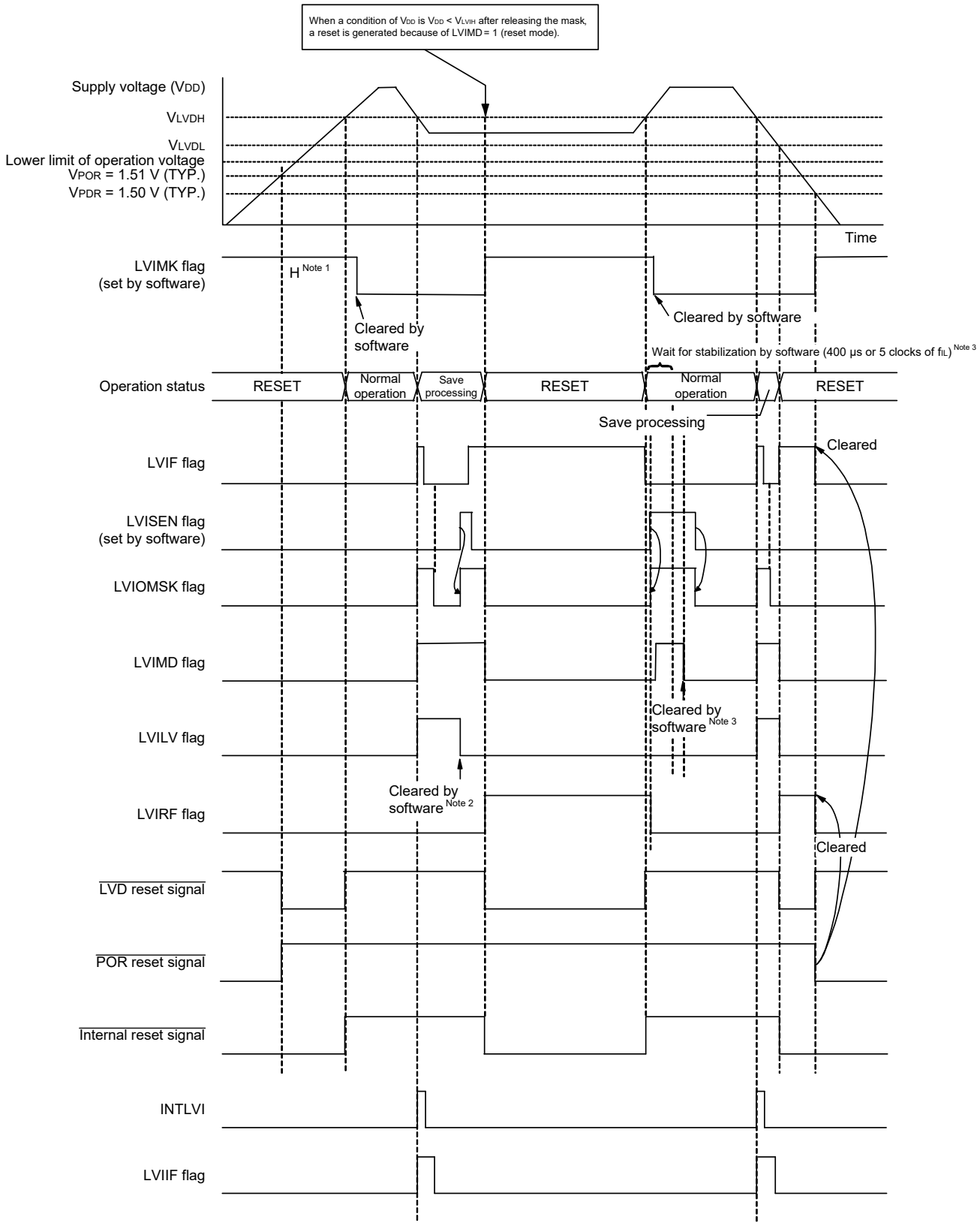
Notes 1. The LVIMK flag is set to “1” by reset signal generation.

2. After an interrupt is generated, perform the processing according to **Figure 24-7 Processing Procedure After an Interrupt Is Generated** in interrupt and reset mode.

3. After a reset is released, perform the processing according to **Figure 24-8 Initial Setting of Interrupt and Reset Mode** in interrupt and reset mode.

Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Figure 24-6. Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (2/2)



(**Notes** and **Remark** are listed on the next page.)

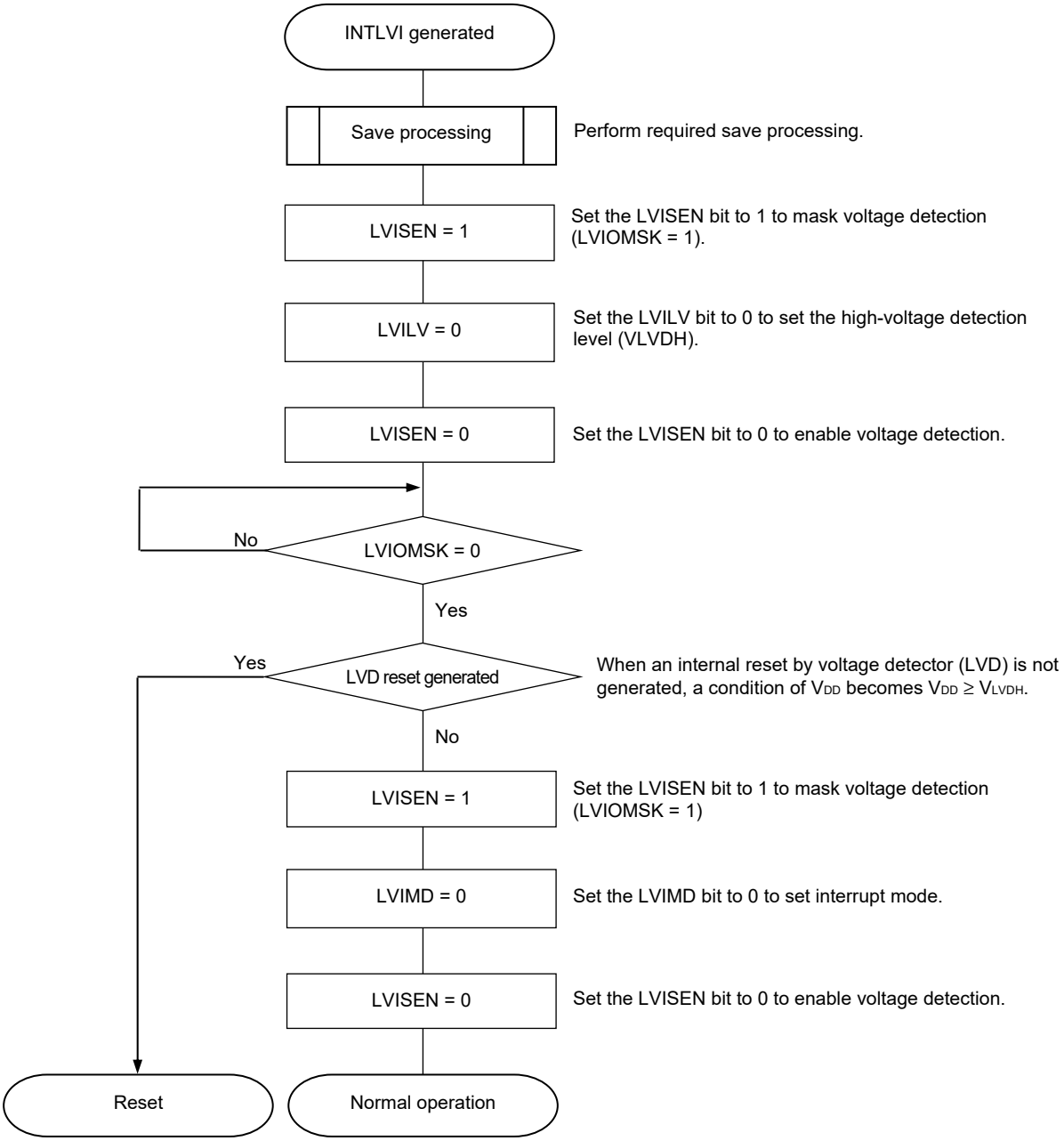
Notes 1. The LVIMK flag is set to “1” by reset signal generation.

2. After an interrupt is generated, perform the processing according to **Figure 24-7 Processing Procedure After an Interrupt Is Generated** in interrupt and reset mode.

3. After a reset is released, perform the processing according to **Figure 24-8 Initial Setting of Interrupt and Reset Mode** in interrupt and reset mode.

Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

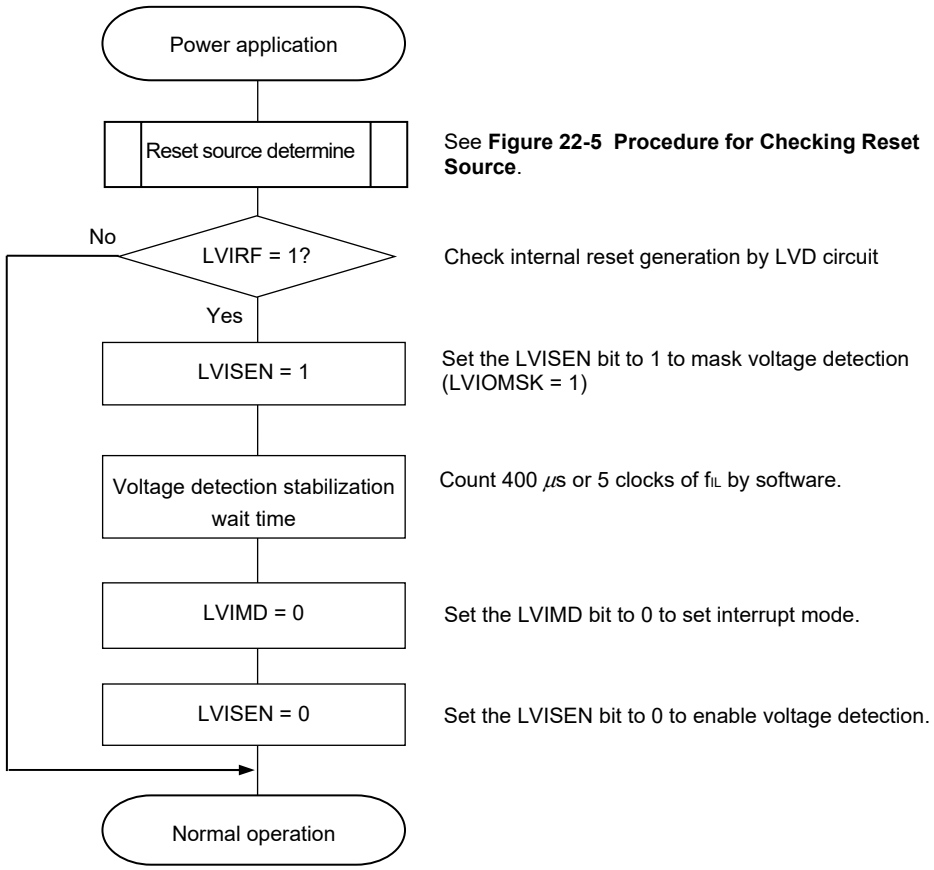
Figure 24-7. Processing Procedure After an Interrupt Is Generated



When setting an interrupt and reset mode (LVIMDS1, LVIMDS0 = 1, 0), voltage detection stabilization wait time for 400 μs or 5 clocks of fIL is necessary after LVD reset is released (LVIRF = 1). After waiting until voltage detection stabilizes, (0) clear the LVIMD bit for initialization. While voltage detection stabilization wait time is being counted and when the LVIMD bit is rewritten, set LVISEN to 1 to mask a reset or interrupt generation by LVD.

Figure 24-8 shows the procedure for initial setting of interrupt and reset mode.

Figure 24-8. Initial Setting of Interrupt and Reset Mode



Remark fIL: Low-speed on-chip oscillator clock frequency

24.5 Cautions for Voltage Detector

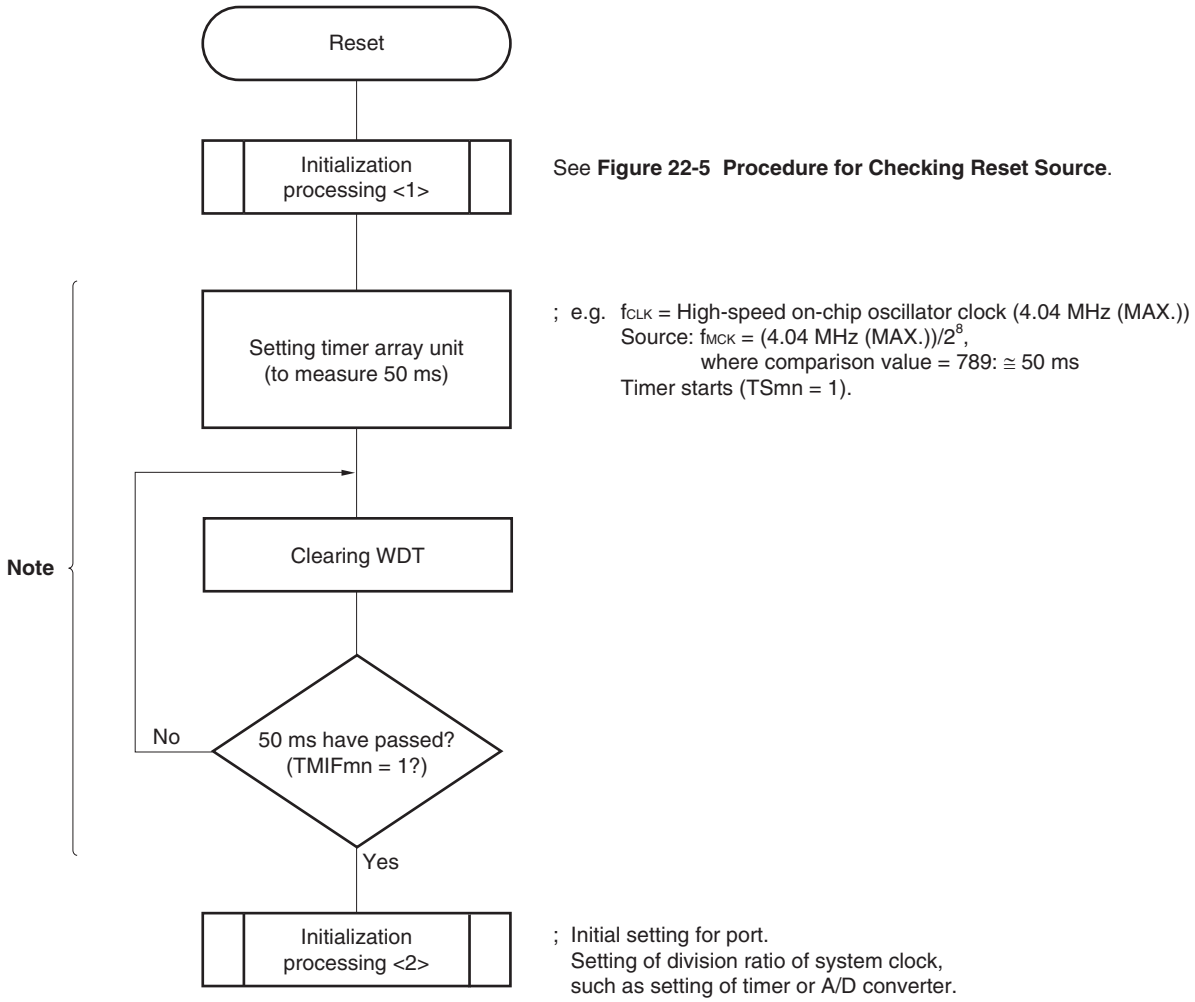
(1) Voltage fluctuation when power is supplied

In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the LVD detection voltage, the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking the following action.

<Action>

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

Figure 24-9. Example of Software Processing If Supply Voltage Fluctuation is 50 ms or Less in Vicinity of LVD Detection Voltage



Note If reset is generated again during this period, initialization processing <2> is not started.

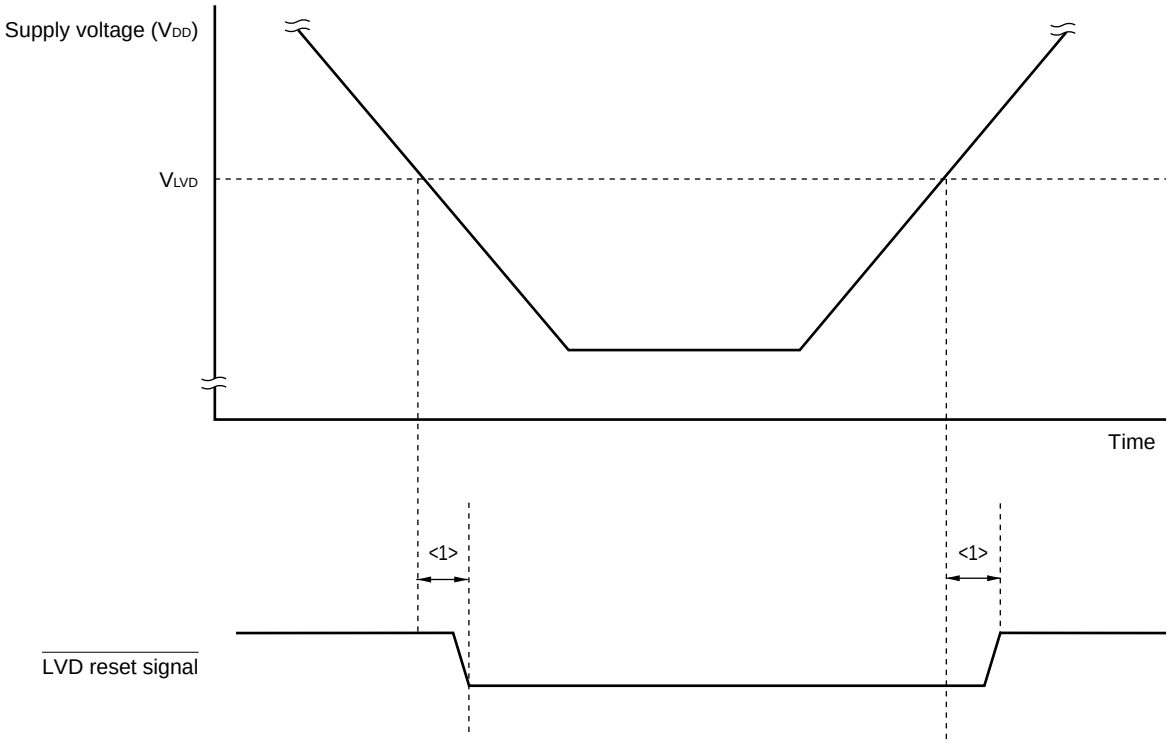
Remark m = 0, n = 0 to 7

(2) Delay from the time LVD reset source is generated until the time LVD reset has been generated or released

There is some delay from the time supply voltage (V_{DD}) < LVD detection voltage (V_{LVD}) until the time LVD reset has been generated.

In the same way, there is also some delay from the time LVD detection voltage (V_{LVD}) ≤ supply voltage (V_{DD}) until the time LVD reset has been released (see **Figure 24-10**).

Figure 24-10. Delay from the Time LVD Reset Source Is Generated Until the Time LVD Reset Has Been Generated or Released



<1>: Detection delay (300 μs (MAX.))

(3) Power on when LVD is off

Use the external rest input via the $\overline{\text{RESET}}$ pin when the LVD is off.

For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin. To perform an external reset upon power application, input a low level to the $\overline{\text{RESET}}$ pin, turn power on, continue to input a low level to the pin for 10 μs or more within the operating voltage range shown in **32.4** or **33.4 AC Characteristics**, and then input a high level to the pin.

(4) Operating voltage fall when LVD is off or LVD interrupt mode is selected

When the operating voltage falls with the LVD is off or with the LVD interrupt mode is selected, this LSI should be placed in the STOP mode, or placed in the reset state by controlling the externally input reset signal, before the voltage falls below the operating voltage range defined in **32.4** or **33.4 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

CHAPTER 25 SAFETY FUNCTIONS

25.1 Overview of Safety Functions

<R> The following safety functions are provided in the RL78/I1A to comply with the IEC60730 safety standards.

These functions enable the microcontroller to self-diagnose abnormalities and stop operating if an abnormality is detected.

(1) Flash memory CRC operation function (high-speed CRC, general-purpose CRC)

This detects data errors in the flash memory by performing CRC operations.

Two CRC functions are provided in the RL78/I1A that can be used according to the application or purpose of use.

- High-speed CRC: The CPU can be stopped and a high-speed check executed on its entire code flash memory area during the initialization routine.
- General CRC: This can be used for checking various data in addition to the code flash memory area while the CPU is running.

(2) RAM parity error detection function

This detects parity errors when reading RAM data.

(3) RAM guard function

This prevents RAM data from being rewritten when the CPU freezes.

(4) SFR guard function

This prevents SFRs from being rewritten when the CPU freezes.

(5) Invalid memory access detection function

This detects illegal accesses to invalid memory areas (such as areas where no memory is allocated and areas to which access is restricted).

(6) Frequency detection function

This function allows a self-check of the CPU/peripheral hardware clock frequencies using the timer array unit.

(7) A/D test function

This is used to perform a self-check of the A/D converter by performing A/D conversion of the A/D converter's positive and negative reference voltages, analog input channel (ANI), temperature sensor output voltage, and internal reference voltage.

Remark For more information on usage examples of the safety functions required to comply with the IEC60730 safety standards, refer to the IEC60730/60335 self-test library application notes (R01AN1062, R01AN1296) for the RL78 MCU series.

25.2 Registers Used by Safety Functions

The safety functions use the following registers for each function.

Register	Each Function of Safety Function
• Flash memory CRC control register (CRC0CTL) • Flash memory CRC operation result register (PGCRCL)	Flash memory CRC operation function (high-speed CRC)
• CRC input register (CRCIN) • CRC data register (CRCD)	CRC operation function (general-purpose CRC)
• RAM parity error control register (RPECTL)	RAM parity error detection function
• Invalid memory access detection control register (IAWCTL)	RAM guard function
	SFR guard function
	Invalid memory access detection function
• Timer input select register 0 (TIS0)	Frequency detection function
• A/D test register (ADTES)	A/D test function

The content of each register is described in **25.3 Operation of Safety Functions**.

25.3 Operation of Safety Functions

25.3.1 Flash memory CRC operation function (high-speed CRC)

The IEC60730 standard mandates the checking of data in the flash memory, and recommends using CRC to do it. The high-speed CRC provided in the RL78/I1A 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 KB 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 CRC, because the general CRC operates in LSB first order.

25.3.1.1 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.

Reset signal generation clears this register to 00H.

Figure 25-1. Format of Flash Memory CRC Control Register (CRC0CTL)

Address: F02F0H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CRC0CTL	CRC0EN	0	FEA5	FEA4	FEA3	FEA2	FEA1	FEA0

CRC0EN	Control of CRC ALU operation
0	Stop the operation.
1	Start the operation according to HALT instruction execution.

FEA5	FEA4	FEA3	FEA2	FEA1	FEA0	High-speed CRC operation range
0	0	0	0	0	0	0000H to 03FFBH (16 K to 4 bytes)
0	0	0	0	0	1	0000H to 07FFBH (32 K to 4 bytes)
0	0	0	0	1	0	0000H to 0BFFBH (48 K to 4 bytes)
0	0	0	0	1	1	0000H to 0FFFBH (64 K to 4 bytes)
Other than above						Setting prohibited

Remark Input the expected CRC operation result value to be used for comparison in the lowest 4 bytes of the flash memory. Note that the operation range will thereby be reduced by 4 bytes.

25.3.1.2 Flash memory CRC operation result register (PGCRCL)

This register is used to store the high-speed CRC operation results.

The PGCRCL register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 25-2. Format of Flash Memory CRC Operation Result Register (PGCRCL)

Address: F02F2H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
PGCRCL	PGCRC15	PGCRC14	PGCRC13	PGCRC12	PGCRC11	PGCRC10	PGCRC9	PGCRC8
	7	6	5	4	3	2	1	0
	PGCRC7	PGCRC6	PGCRC5	PGCRC4	PGCRC3	PGCRC2	PGCRC1	PGCRC0

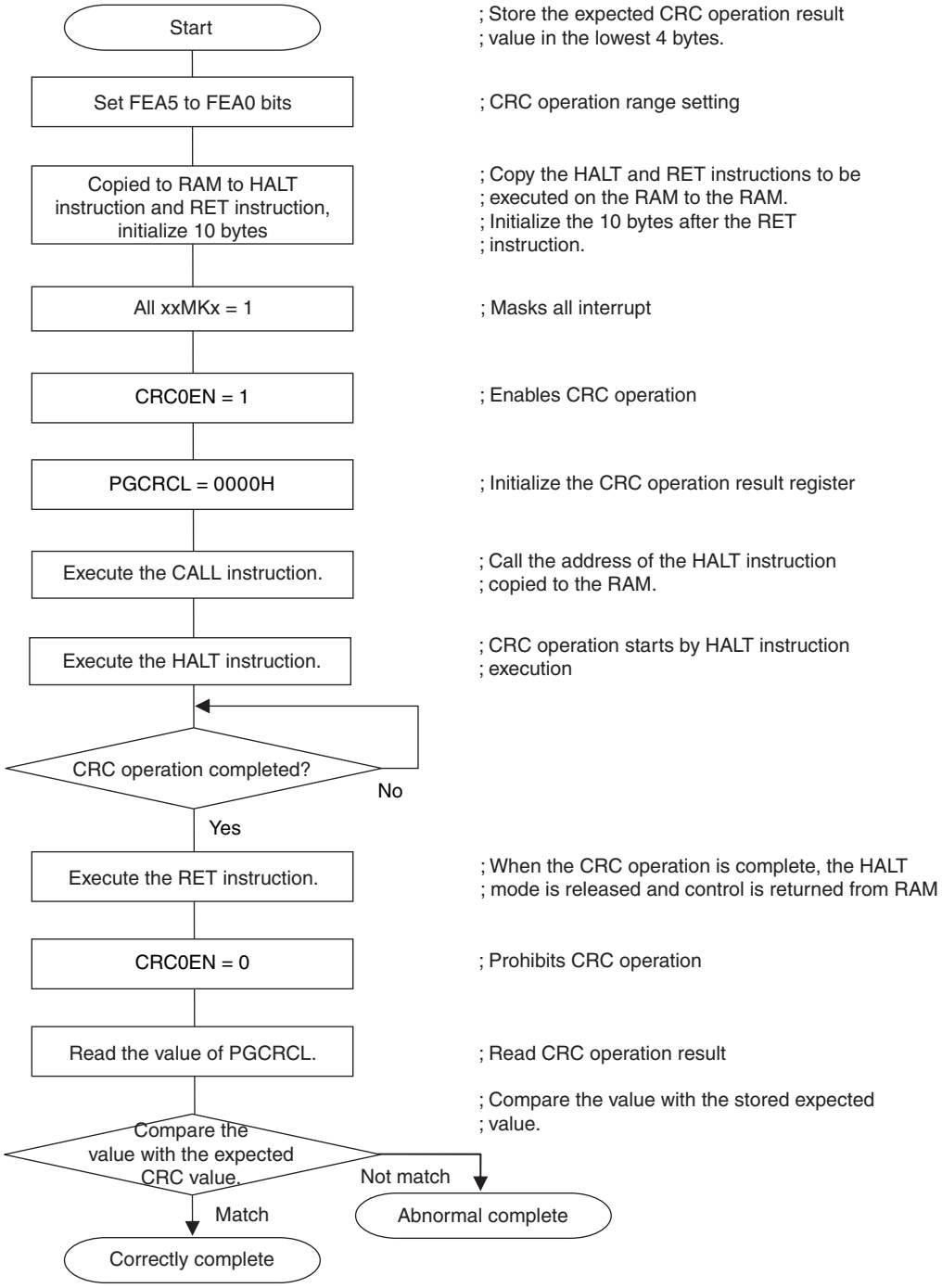
PGCRC15 to PGCRC0	High-speed CRC operation results
0000H to FFFFH	Store the high-speed CRC operation results.

Caution The PGCRCL register can only be written if CRC0EN (bit 7 of the CRC0CTL register) = 1.

Figure 25-3 shows the flowchart of flash memory CRC operation function (high-speed CRC).

<Operation flow>

Figure 25-3. Flowchart of Flash Memory CRC Operation Function (High-speed CRC)



Cautions 1. The CRC operation is executed only on the code flash.

2. Store the expected CRC operation value in the area below the operation range in the code flash.

3. The CRC operation is enabled by executing the HALT instruction in the RAM area.
Be sure to execute the HALT instruction in RAM area.

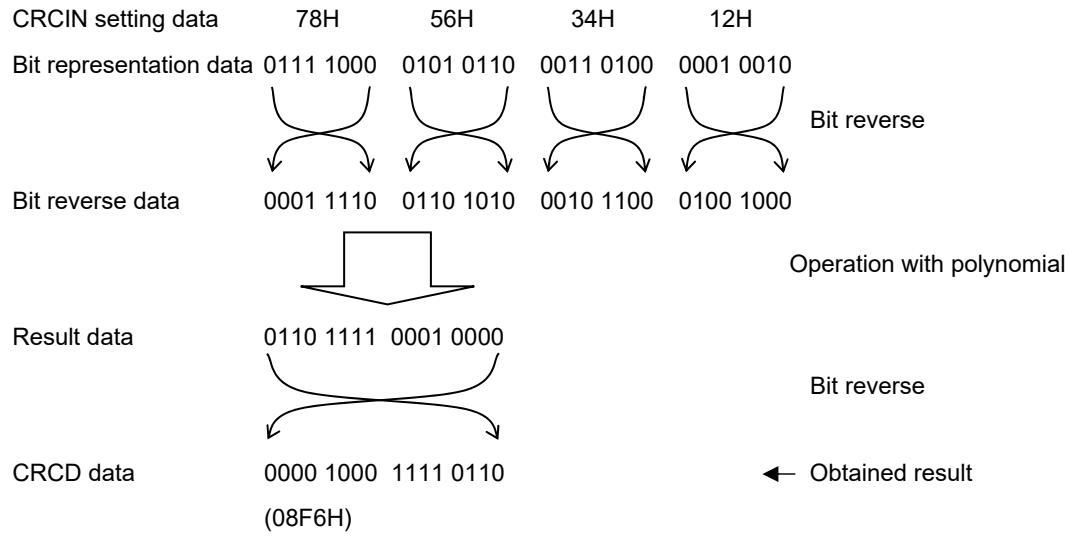
The expected CRC value can be calculated by using the Integrated Development Environment CubeSuite+. See the **Integrated Development Environment CubeSuite+ User's Manual** for details.

25.3.2 CRC operation function (general-purpose CRC)

<R> In the RL78/I1A, a general CRC operation can be executed as a peripheral function while the CPU is operating. The general CRC can be used for checking various data in addition to the code flash memory area. The data to be checked can be specified by using software (a user-created program). CRC calculation function in the HALT mode can be used only during the DMA transmission.

The general CRC operation can be executed in the main system clock operation mode as well as 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.

25.3.2.1 CRC input register (CRCIN)

CRCIN register is an 8-bit register that is used to set the CRC operation data of general-purpose CRC.

The possible setting range is 00H to FFH.

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

Reset signal generation clears this register to 00H.

Figure 25-4. Format of CRC Input Register (CRCIN)

Address: FFFACH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRCIN								

Bits 7 to 0	Function
00H to FFH	Data input.

25.3.2.2 CRC data register (CRCD)

This register is used to store the CRC operation result of the general-purpose CRC.

The setting range is 0000H to FFFFH.

After 1 clock of CPU/peripheral hardware clock (f_{CLK}) has elapsed from the time CRCIN register is written, the CRC operation result is stored to the CRCD register.

The CRCD register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 25-5. Format of CRC Data Register (CRCD)

Address: F02FAH After reset: 0000H R/W

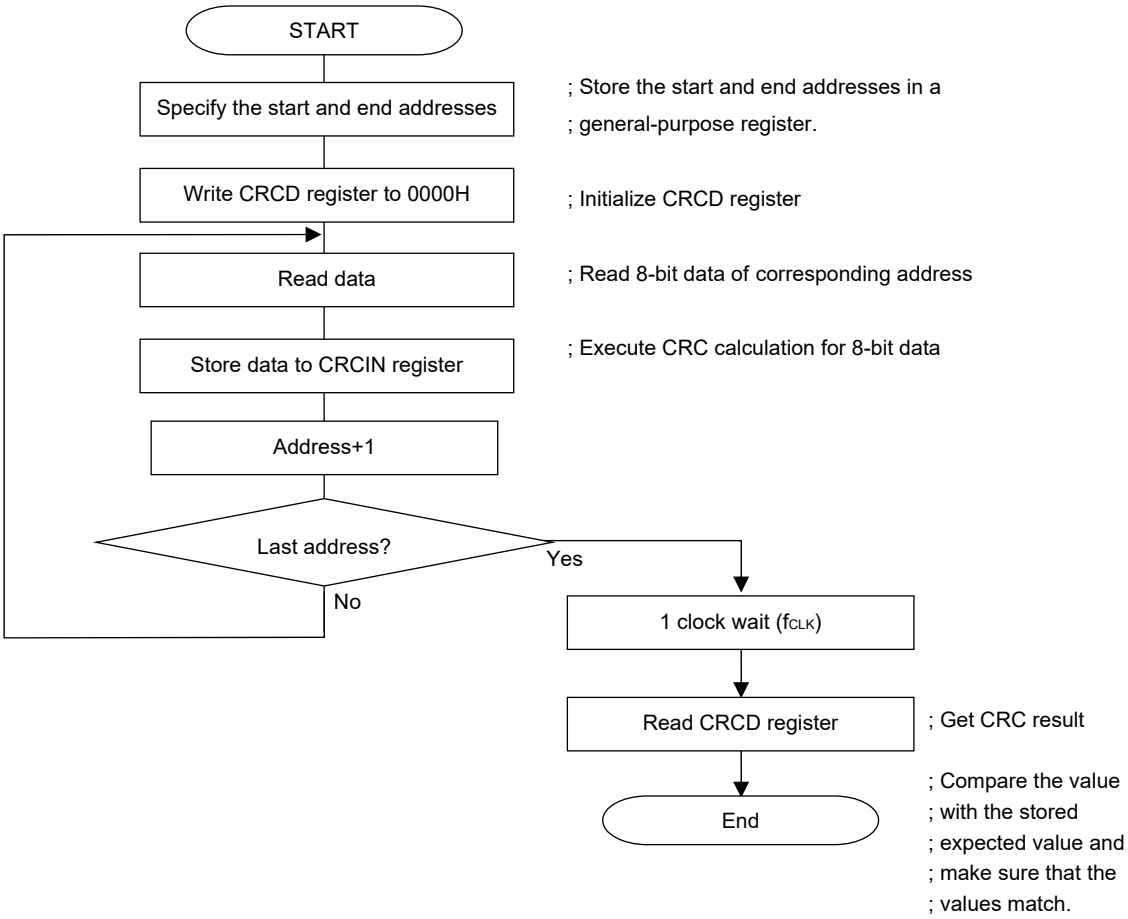
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCD																

Cautions 1. Read the value written to CRCD register before writing to CRCIN register.

2. If conflict between writing and storing operation result to CRCD register occurs, the writing is ignored.

<Operation flow>

Figure 25-6. CRC Operation Function (General-purpose CRC)



25.3.3 RAM parity error detection function

The IEC60730 standard mandates the checking of RAM data. A single-bit parity bit is therefore added to all 8-bit data in the RL78/I1A's RAM. By using this RAM parity error detection function, the parity bit is appended when data is written, and the parity is checked when the data is read. This function can also be used to trigger a reset when a parity error occurs.

25.3.3.1 RAM parity error control register (RPECTL)

This register is used to control parity error generation check bit and reset generation due to parity errors.

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

Reset signal generation clears this register to 00H.

Figure 25-7. Format of RAM Parity Error Control Register (RPECTL)

Address: F00F5H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
RPECTL	RPERDIS	0	0	0	0	0	0	RPEF

RPERDIS	Parity error reset mask flag
0	Enable parity error resets.
1	Disable parity error resets.

RPEF	Parity error status flag
0	No parity error has occurred.
1	A parity error has occurred.

Caution The parity bit is appended when data is written, and the parity is checked when the data is read. Therefore, while 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, while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the RAM area + 10 bytes when instructions are fetched from RAM areas.

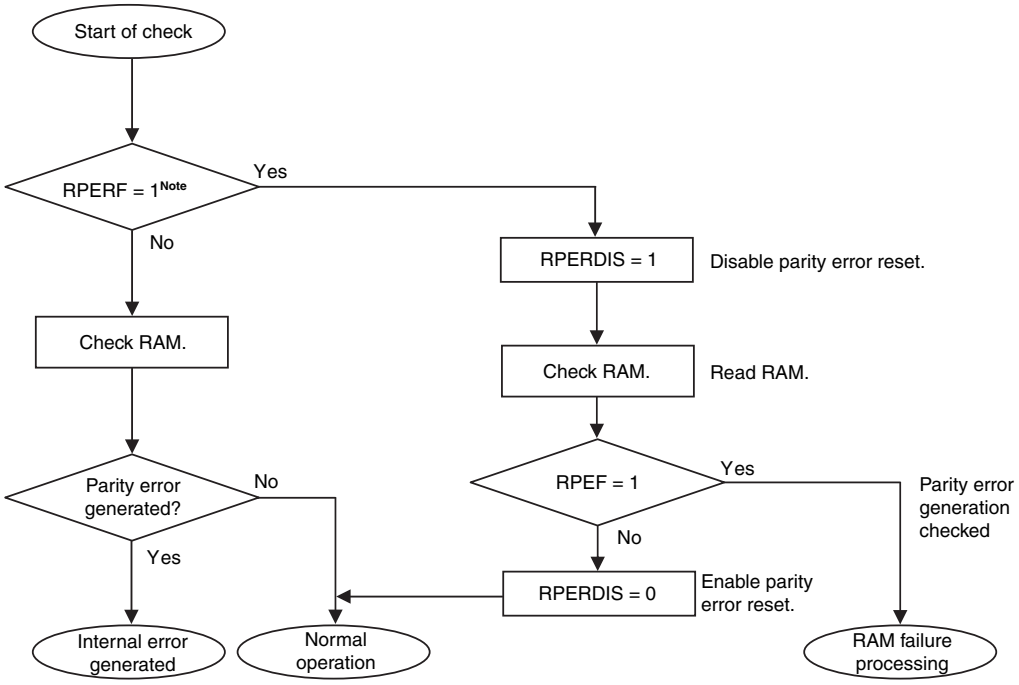
Remarks 1. The parity error reset is enabled by default (RPERDIS = 0).

2. Even if the parity error reset is disabled (RPERDIS = 1), the RPEF flag will be set (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.

3. The RPEF flag in the RPECTL register is set (1) when the RAM parity error occurs and cleared (0) by writing 0 to it or by any reset source. When RPEF = 1, the value is retained even if RAM for which no parity error has occurred is read.

4. The general-purpose registers are not included for RAM parity error detection.

Figure 25-8. Flowchart of RAM Parity Check



Note To check internal reset status using a RAM parity error, see **CHAPTER 22 RESET FUNCTION**.

25.3.4 RAM guard function

<R> This RAM guard function is used to protect data in the specified memory space.

If the RAM guard function is specified, writing to the specified RAM space is disabled, but reading from the space can be carried out as usual.

25.3.4.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

GRAM1 and GRAM0 bits are used in RAM guard function.

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

Reset signal generation clears this register to 00H.

Figure 25-9. Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GRAM1	GRAM0	RAM guard spaceNote
0	0	Disabled. RAM can be written to.
0	1	The 128 bytes starting at the start RAM address
1	0	The 256 bytes starting at the start RAM address
1	1	The 512 bytes starting at the start RAM address

Note The RAM start address differs depending on the size of the RAM provided with the product.

25.3.5 SFR guard function

<R> This SFR guard function is used to protect data in the control registers used by the port function, interrupt function, clock control function, voltage detection function, and RAM parity error detection function.

If the SFR guard function is specified, writing to the specified SFRs is disabled, but reading from the SFRs can be carried out as usual.

25.3.5.1 Invalid memory access detection control registers 0, 1 (IAWCTL0, IAWCTL1)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

GPORT, GINT and GCSC bits are used in SFR guard function.

The IAWCTL0, IAWCTL1 registers can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 25-10. Format of Invalid Memory Access Detection Control Register 0 (IAWCTL0)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL0	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GPORT	Control registers of port function guard
0	Disabled. Control registers of port function can be read or written to.
1	Enabled. Writing to control registers of port function is disabled. Reading is enabled. [Guarded SFR] PM0 to PM3, PM4, PM7, PM12, PM14, PU0, PU1, PU3, PU4, PU7, PU12, PU14, PIM0, PIM1, POM0, POM1, PMC0, PMC12, PMC14, ADPC, and PIOR1[Note]

GINT	Registers of interrupt function guard
0	Disabled. Registers of interrupt function can be read or written to.
1	Enabled. Writing to registers of interrupt function is disabled. Reading is enabled. [Guarded SFR] IF0 to IF2, MK0 to MK2, PR00, PR01, PR02, PR10, PR11, PR12, EGP0, EGP1, EGN0, and EGN1

GCSC	Control registers of clock control function, voltage detector and RAM parity error detection function guard
0	Disabled. Control registers of clock control function, voltage detector and RAM parity error detection function can be read or written to.
1	Enabled. Writing to control registers of clock control function, voltage detector and RAM parity error detection function is disabled. Reading is enabled. [Guarded SFR] CMC, CSC, OSTS, CKC, PER0, OSMC, LVIM, LVIS, and RPECTL

Note Pxx (Port register) is not guarded.

Figure 25-11. Format of Invalid Memory Access Detection Control Register 1 (IAWCTL1)

Address: F05C4H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL1	0	0	0	0	0	GDPORT1	GDINT1	GDCG1

GDPORT1	Control registers of port function guard
0	Disabled. Control registers of port function can be read or written to.
1	Enabled. Writing to control registers of port function is disabled. Reading is enabled. [Guarded SFR] PM20, PU20, POM20, PIOR1, and SUCTL[Note 1]

GDINT1	Registers of interrupt function guard
0	Disabled. Registers of interrupt function can be read or written to.
1	Enabled. Writing to registers of interrupt function is disabled. Reading is enabled. [Guarded SFR] INTMK0, EGP2, EGN2, CMPEGP0, and CMPEGN0

GDCG1	Control registers of clock control function guard
0	Disabled. Control registers of clock control function, voltage detector and RAM parity error detection function can be read or written to.
1	Enabled. Writing to control registers of clock control function, voltage detector and RAM parity error detection function is disabled. Reading is enabled. [Guarded SFR] PER1, PER2, and PLLCTL

Note Pxx (Port register) is not guarded.

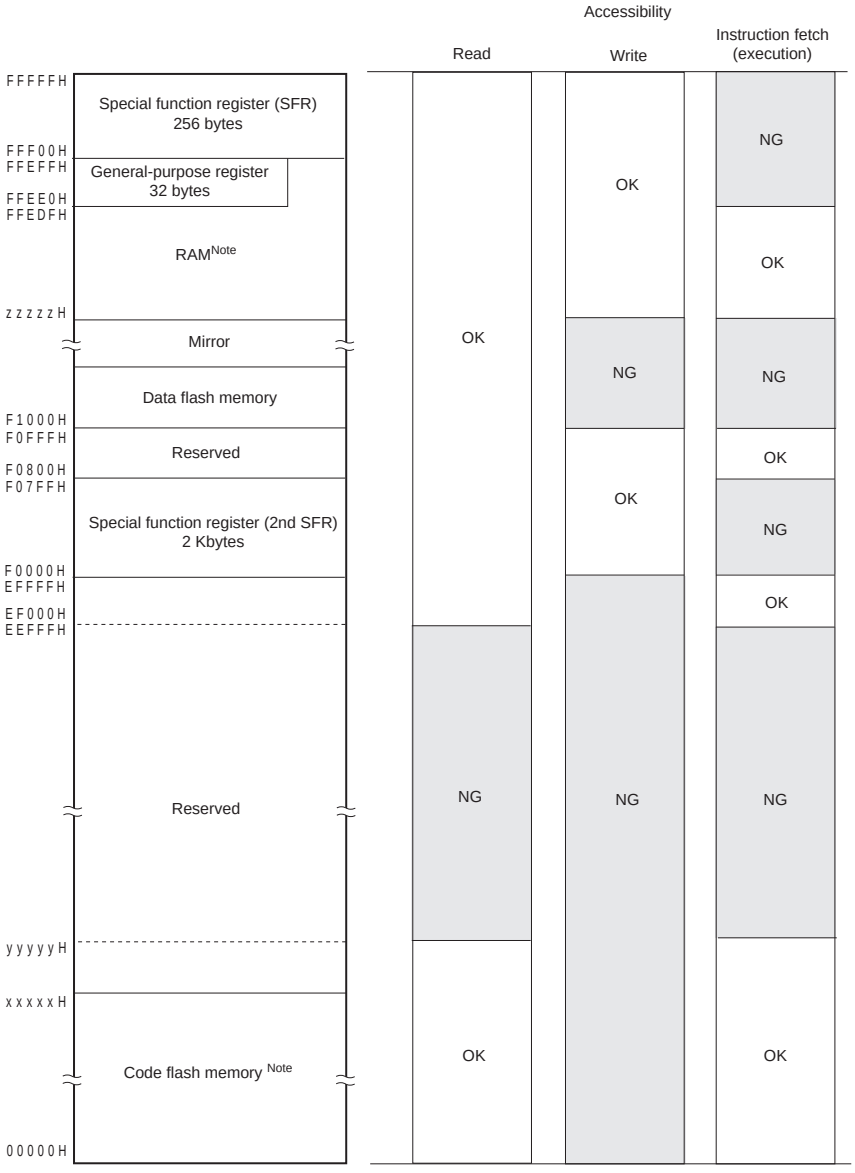
25.3.6 Invalid memory access detection function

The IEC60730 standard mandates checking that the CPU and interrupts are operating correctly.

The illegal memory access detection function triggers a reset if a memory space specified as access-prohibited is accessed.

The illegal memory access detection function applies to the areas indicated by NG in Figure 25-12.

Figure 25-12. Invalid Access Detection Area



Note The following table lists the code flash memory, RAM, and lowest detection address for each product:

Products	Code Flash Memory (00000H to xxxxxH)	RAM (zzzzzH to FFEFFH)	Detected lowest address for read/instruction fetch (execution) (yyyyyH)
R5F1076C, R5F107AC	32768 × 8 bit (00000H to 07FFFH)	2048 × 8 bit (FF700H to FFEFFH)	10000H
R5F107AE, R5F107DE	65536 × 8 bit (00000H to 0FFFFH)	4096 × 8 bit (FEF00H to FFEFFH)	10000H

25.3.6.1 Invalid memory access detection control register 0 (IAWCTL0)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

IAWEN bit is used in invalid memory access detection function.

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

Reset signal generation clears this register to 00H.

Figure 25-13. Format of Invalid Memory Access Detection Control Register 0 (IAWCTL0)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

IAWEN[Note]	Control of invalid memory access detection
0	Disable the detection of invalid memory access.
1	Enable the detection of invalid memory access.

Note Only writing 1 to the IAWEN bit is enabled, not writing 0 to it after setting it to 1.

Remark By specifying WDTON = 1 (watchdog timer operation enable) for the option byte(000C0H), the invalid memory access function is enabled even IAWEN = 0.

25.3.7 Frequency detection function

The IEC60730 standard mandates checking that the oscillation frequency is correct.

By using the CPU/peripheral hardware clock frequency (f_{CLK}) and measuring the pulse width of the input signal to channel 5 of the timer array unit 0 (TAU0), whether the proportional relationship between the two clock frequencies is correct can be determined. Note that, however, if one or both clock operations are completely stopped, the proportional relationship between the clocks cannot be determined.

<Clocks to be compared>

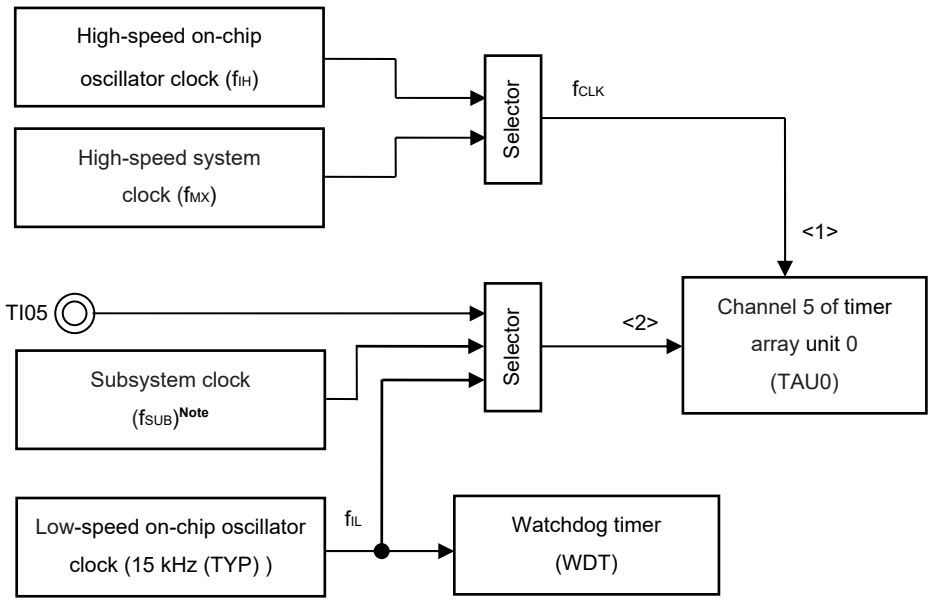
<1> CPU/peripheral hardware clock frequency (f_{CLK}):

- High-speed on-chip oscillator clock (f_{IH})
- High-speed system clock (f_{MX})

<2> Input to channel 5 of the timer array unit 0

- Timer input to channel 5 (TI05)
- Low-speed on-chip oscillator clock (f_{IL}: 15 kHz (typ.))
- Subsystem clock (f_{SUB})**Note**

Figure 25-14. Configuration of Frequency Detection Function



If input pulse interval measurement results in an abnormal value, it can be concluded that the clock frequency is abnormal.

For how to execute input pulse interval measurement, see **6.8.3 Operation as input pulse interval measurement**.

Note Can only be selected in the products incorporating the subsystem clock.

25.3.7.1 Timer input select register 0 (TIS0)

The TIS0 register is used to select the timer input of channel 5 of timer array unit 0 (TAU0).

The TIS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 25-15. Format of Timer Input Select Register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	0	0	TIS02	TIS01	TIS00

TIS02	TIS01	TIS00	Selection of timer input used with channel 5
0	0	0	Input signal of timer input pin (TI05)
0	0	1	
0	1	0	
0	1	1	
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
1	0	1	Subsystem clock (f_{SUB})
Other than above			Setting prohibited

25.3.8 A/D test function

The IEC60730 standard mandates testing the A/D converter. The A/D test function checks whether or not the A/D converter is operating normally by executing A/D conversions of the A/D converter's positive and negative reference voltages, analog input channel (ANI), temperature sensor output voltage, and the internal reference voltage. For details of the check method, see the **Safety Function (A/D Test) Application Note (R01AN0955)**.

The analog multiplexer can be checked using the following procedure.

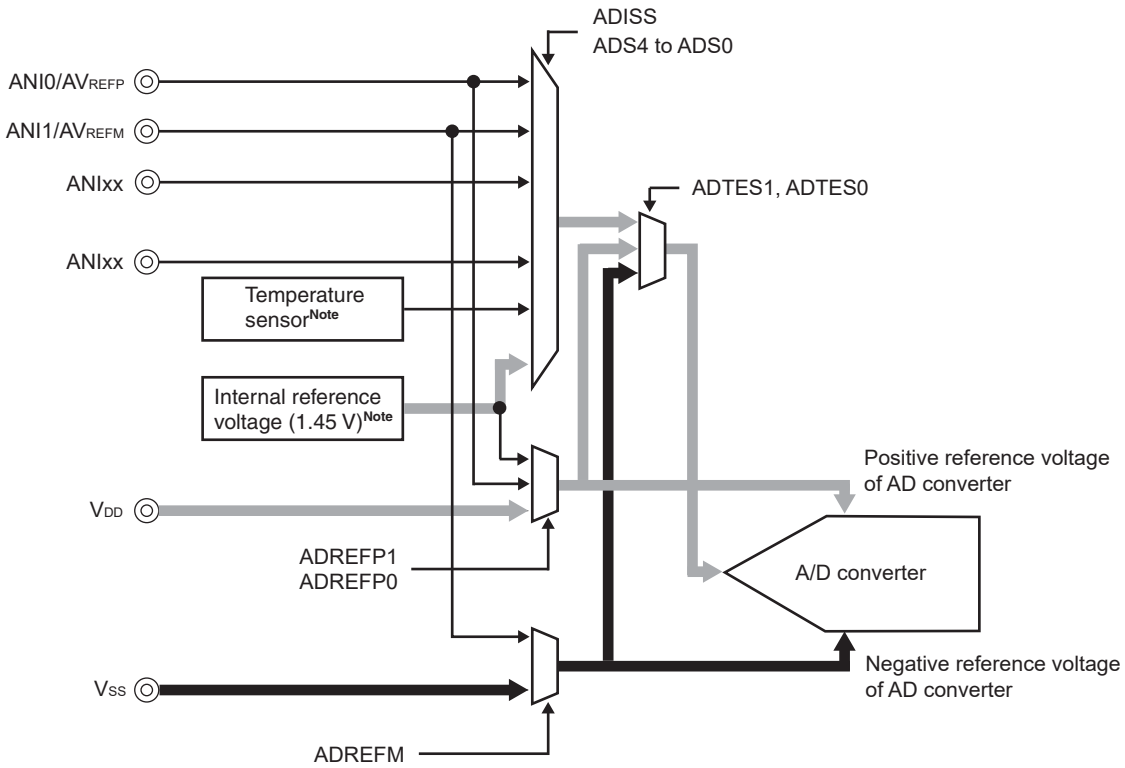
<1> Select the ANIx pin for A/D conversion using the ADTES register (ADTES1 = 0, ADTES0 = 0).
<2> Perform A/D conversion for the ANIx pin (conversion result 1-1).
<3> Select the A/D converter's negative reference voltage for A/D conversion using the ADTES register (ADTES1 = 1, ADTES0 = 0)
<4> Perform A/D conversion of the negative reference voltage of the A/D converter (conversion result 2-1).
<5> Select the ANIx pin for A/D conversion using the ADTES register (ADTES1 = 0, ADTES0 = 0).
<6> Perform A/D conversion for the ANIx pin (conversion result 1-2).
<7> Select the A/D converter's positive reference voltage for A/D conversion using the ADTES register (ADTES1 = 1, ADTES0 = 1)
<8> Perform A/D conversion of the positive reference voltage of the A/D converter (conversion result 2-2).
<9> Select the ANIx pin for A/D conversion using the ADTES register (ADTES1 = 0, ADTES0 = 0).
<10> Perform A/D conversion for the ANIx 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 conversion result 2-2 is all one.

Using the procedure above can confirm that the analog multiplexer is selected and all wiring is connected.

Remarks 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.

2. The conversion results might contain an error. Consider an appropriate level of error when comparing the conversion results.

Figure 25-16. Configuration of A/D Test Function



Note This setting can be used only in HS (high-speed main) mode.

25.3.8.1 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 channel (ANIxx), temperature sensor output voltage, or internal reference voltage (1.45 V) as the target of A/D conversion.

When using the A/D test function, specify the following settings:

- Select negative reference voltage as the target of A/D conversion for zero-scale measurement.
- Select 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.

Reset signal generation clears this register to 00H.

Figure 25-17. Format of A/D Test Register (ADTES)

Address: F0013H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	0	ADTES1	ADTES0

ADTES1	ADTES0	A/D conversion target
0	0	ANIxx/temperature sensor output voltage[Note]/internal reference voltage (1.45 V)[Note] (This is specified using the analog input channel specification register (ADS).)
1	0	Negative reference voltage (selected with the ADREFM bit in ADM2)
1	1	Positive reference voltage (selected with the ADREFP1 or ADREFP0 bit in ADM2)[Note]
Other than above		Setting prohibited

Note Temperature sensor output voltage and internal reference voltage (1.45 V) can be used only in HS (high-speed main) mode.

25.3.8.2 Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.

Set A/D test register (ADTES) to 00H when measuring the ANIxx/temperature sensor output voltage/internal reference voltage (1.45 V).

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

Reset signal generation clears this register to 00H.

Figure 25-18. Format of Analog Input Channel Specification Register (ADS) (1/2)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

OSelect mode (ADMD = 0)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P20/ANI0/AV_{REFP} pin
0	0	0	0	0	1	ANI1	P21/ANI1/AV_{REFM} pin
0	0	0	0	1	0	ANI2	P22/ANI2 pin
0	0	0	0	1	1	PGAOUT	PGA output 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
1	0	0	0	0	0	–	Temperature sensor output voltage[Note]
1	0	0	0	0	1	–	Internal reference voltage (1.45 V)[Note]
Other than above						Setting prohibited	

Note This setting can be used only in HS (high-speed main) mode.

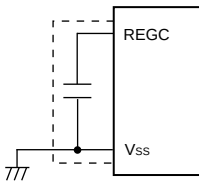
(**Cautions** are listed on the next page.)

Cautions
1. **Be sure to clear bits 5 and 6 to "0".**
2. **Select input mode for the ports which are set to analog input with the ADPC and PMC registers, using the port mode registers 0, 2, 12, and 14 (PM0, PM2, PM12, and PM14).**
3. **Do not use the ADS register to set the pins which should be set as digital I/O with the A/D port configuration register (ADPC).**
4. **Do not use the ADS register to set the pins which should be set as digital I/O with the port mode control registers 0, 12, and 14 (PMC0, PCM12, and PCM14).**
5. **Only rewrite the value of the ADISS bit while conversion operation is stopped (ADCS = 0, ADCE = 0).**
6. **If using AV_{REFP} as the positive reference voltage of the A/D converter, do not select ANI0 as an A/D conversion channel.**
7. **If using AV_{REFM} as the negative reference voltage of the A/D converter, do not select ANI1 as an A/D conversion channel.**
8. **When ADISS is 1, the internal reference voltage (1.45 V) cannot be used for the positive reference voltage. In addition, the first conversion result obtained after setting ADISS to 1 is not available. For detailed setting flow, see 12.7.4 Setup when temperature sensor output voltage/internal reference voltage is selected (example for software trigger mode and one-shot conversion mode).**
9. **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 32.3.2 or 33.3.2 Supply current characteristics is added.**

CHAPTER 26 REGULATOR

26.1 Regulator Overview

The RL78/I1A contains a circuit for operating the device with a constant voltage. At this time, in order to stabilize the regulator output voltage, connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.



Caution Keep the wiring length as short as possible for the broken-line part in the above figure.

The regulator output voltage, see **Table 26-1**.

Table 26-1. Regulator Output Voltage Conditions

Mode	Output Voltage	Condition
LS (low-speed main) mode	1.8 V	-
HS (high-speed main) mode	1.8 V	In STOP mode
		When both the high-speed system clock (f_{MX}) and the high-speed on-chip oscillator clock (f_{IH}) are stopped during CPU operation with the subsystem clock (f_{SUB})
		When both the high-speed system clock (f_{MX}) and the high-speed on-chip oscillator clock (f_{IH}) are stopped during the HALT mode when the CPU operation with the subsystem clock (f_{SUB}) has been set
	2.1 V	Other than above (include during OCD mode)**Note**

Note When it shifts to the subsystem clock operation or STOP mode during the on-chip debugging, the regulator output voltage is kept at 2.1 V (not decline to 1.8 V).

CHAPTER 27 OPTION BYTE

27.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the RL78/I1A form an option byte area.

Option bytes consist of user option byte (000C0H to 000C2H) and on-chip debug option byte (000C3H).

Upon power application or resetting and starting, an option byte is automatically referenced and a specified function is set. When using the product, be sure to set the following functions by using the option bytes.

For the bits to which no function is allocated, do not change their initial values.

To use the boot swap operation during self programming, 000C0H to 000C3H are replaced by 010C0H to 010C3H. Therefore, set the same values as 000C0H to 000C3H to 010C0H to 010C3H.

Caution The option bytes should always be set regardless of whether each function is used.

27.1.1 User option byte (000C0H to 000C2H/010C0H to 010C2H)

(1) 000C0H/010C0H

O Operation of watchdog timer
- Enabling or disabling of counter operation
- Operation is stopped or enabled in the HALT or STOP mode.

O Setting of overflow time of watchdog timer

O Setting of window open period of watchdog timer

O Setting of interval interrupt of watchdog timer
- Whether or not to use the interval interrupt is selectable.

Caution Set the same value as 000C0H to 010C0H when the boot swap operation is used because 000C0H is replaced by 010C0H.

(2) 000C1H/010C1H

O Setting of LVD operation mode
- Interrupt & reset mode.
- Reset mode.
- Interrupt mode.
- LVD off (by controlling the externally input reset signal on the $\overline{\text{RESET}}$ pin)

O Setting of LVD detection level (V_{LVDH}, V_{LVDL}, V_{LVD})

Cautions 1. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 32.4 or 33.4 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 010C2H).

2. Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

(3) 000C2H/010C2H

O Setting of flash operation mode

Set according to the main system clock frequency (f_{MAIN}) and supply voltage (V_{DD})

- LS (low speed main) mode
- HS (high speed main) mode

O Setting of the frequency of the high-speed on-chip oscillator

- Select from 32 MHz/24 MHz/16 MHz/12 MHz/8 MHz/6 MHz/4 MHz /3 MHz/2 MHz/1 MHz (TYP.).

Caution Set the same value as 000C2H to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

27.1.2 On-chip debug option byte (000C3H/010C3H)

O Control of on-chip debug operation

- On-chip debug operation is disabled or enabled.

O 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 Set the same value as 000C3H to 010C3H when the boot swap operation is used because 000C3H is replaced by 010C3H.

27.2 Format of User Option Byte

The format of user option byte is shown below.

Figure 27-1. Format of User Option Byte (000C0H/010C0H)

Address: 000C0H/010C0H[Note 1]

7	6	5	4	3	2	1	0
WDTINIT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINIT	Use of interval interrupt of watchdog timer
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% + 1/2f_{IL} of the overflow time is reached.

WINDOW1	WINDOW0	Watchdog timer window open period[Note 2]
0	0	Setting prohibited
0	1	50%
1	0	75% [Note 3]
1	1	100%

WDTON	Operation control of watchdog timer counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{IL} = 17.25 kHz (MAX.))
0	0	0	$2^6/f_{IL}$ (3.71 ms)
0	0	1	$2^7/f_{IL}$ (7.42 ms)
0	1	0	$2^8/f_{IL}$ (14.84 ms)
0	1	1	$2^9/f_{IL}$ (29.68 ms)
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)
1	0	1	$2^{13}/f_{IL}$ (474.89 ms)
1	1	0	$2^{14}/f_{IL}$ (949.79 ms)
1	1	1	$2^{16}/f_{IL}$ (3799.18 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

Notes 1. Set the same value as 000C0H to 010C0H when the boot swap operation is used because 000C0H is replaced by 010C0H.

2. The window open period is 100% when WDSTBYON = 0, regardless the value of the WINDOW1 and WINDOW0 bits.

(**Note** continues on the next page.)

Notes 3. When the window open period is set to 75%, clearing the counter of the watchdog timer (writing ACH to WDTE) must proceed outside the corresponding period from among those listed below, over which clearing of the counter is prohibited (for example, confirming that the interval timer interrupt request flag (WDTIIF) of the watchdog timer is set).

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{IL} = 17.25 kHz (MAX.))	Period over which clearing the counter is prohibited when the window open period is set to 75%
0	0	0	$2^6/f_{IL}$ (3.71 ms)	1.85 ms to 2.51 ms
0	0	1	$2^7/f_{IL}$ (7.42 ms)	3.71 ms to 5.02 ms
0	1	0	$2^8/f_{IL}$ (14.84 ms)	7.42 ms to 10.04 ms
0	1	1	$2^9/f_{IL}$ (29.68 ms)	14.84 ms to 20.08 ms
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)	56.36 ms to 80.32 ms
1	0	1	$2^{13}/f_{IL}$ (474.89 ms)	237.44 ms to 321.26 ms
1	1	0	$2^{14}/f_{IL}$ (949.79 ms)	474.89 ms to 642.51 ms
1	1	1	$2^{16}/f_{IL}$ (3799.18 ms)	1899.59 ms to 2570.04 ms

Remark f_{IL}: Low-speed on-chip oscillator clock frequency

Figure 27-2. Format of User Option Byte (000C1H/010C1H) (1/2)

Address: 000C1H/010C1H[Note]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD setting (interrupt & reset mode)

Detection voltage			Option byte setting value						
V_{LVDH}		V_{LVDL}	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge	Falling edge						LVIMDS1	LVIMDS0
2.92 V	2.86 V	2.75 V	0	1	1	1	0	1	0
3.02 V	2.96 V					0	1		
4.06 V	3.98 V					0	0		
–			Setting of values other than above is prohibited.						

- LVD setting (reset mode)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
2.81 V	2.75 V	0	1	1	1	1	1	1
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
–		Setting of values other than above is prohibited.						

- LVD setting (interrupt mode)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
2.81 V	2.75 V	0	1	1	1	1	0	1
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
–		Setting of values other than above is prohibited.						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Remarks 1. For details on the LVD circuit, see **CHAPTER 24 VOLTAGE DETECTOR.**

2. The detection voltage is a typical value. For details, see **32.6.6** or **33.6.6 LVD circuit characteristics.**

(**Cautions** are listed on the next page.)

Figure 27-2. Format of User Option Byte (000C1H/010C1H) (2/2)

Address: 000C1H/010C1H[Note]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD off setting (external reset input from the $\overline{\text{RESET}}$ pin is used)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
–	–	1	×	×	×	×	×	1
–		Setting of values other than above is prohibited.						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Cautions 1. Be sure to set bit 4 to "1".

2. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 32.4 or 33.4 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 010C2H).

Remarks 1. ×: don't care

2. For LVD setting, see **24.1 Functions of Voltage Detector**.

3. The detection voltage is a typical value. For details, see **32.6.6** or **33.6.6 LVD circuit characteristics**.

Figure 27-3. Format of Option Byte (000C2H/010C2H)

Address: 000C2H/010C2H[Note]

7	6	5	4	3	2	1	0
CMODE1	CMODE0	1	0	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

CMODE1	CMODE0	Setting of flash operation mode		
			Operating frequency range (f_{MAIN})	Operating voltage range (V_{DD})
1	0	LS (low speed main) mode	1 to 8 MHz	2.7 to 5.5 V
1	1	HS (high speed main) mode	1 to 32 MHz	2.7 to 5.5 V
Other than above		Setting prohibited		

CMODE1	CMODE0	Setting of flash operation mode
1	0	LS (low speed main) mode
1	1	HS (high speed main) mode
Other than above		Setting prohibited

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator
1	0	0	0	32 MHz
0	0	0	0	24 MHz
1	0	0	1	16 MHz
0	0	0	1	12 MHz
1	0	1	0	8 MHz
0	0	1	0	6 MHz
1	0	1	1	4 MHz
0	0	1	1	3 MHz
1	1	0	0	2 MHz
1	1	0	1	1 MHz
Other than above				Setting prohibited

Note Set the same value as 000C2H to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

Cautions 1. Be sure to set bits 5 and 4 to 10B.

2. The ranges of operation frequency and operation voltage vary depending on the flash operation mode. For details, see 32.4 or 33.4 AC Characteristics.

27.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 27-4. Format of On-chip Debug Option Byte (000C3H/010C3H)

Address: 000C3H/010C3H[Note]

7	6	5	4	3	2	1	0
OCDENSET	0	0	0	0	1	0	OCDERSD

OCDENSET	OCDERSD	Control of on-chip debug operation
0	0	Disables on-chip debug operation.
0	1	Setting prohibited
1	0	Enables on-chip debugging. Erases data of flash memory in case of failures in authenticating on-chip debug security ID.
1	1	Enables on-chip debugging. Does not erases data of flash memory in case of failures in authenticating on-chip debug security ID.

Note Set the same value as 000C3H to 010C3H when the boot swap operation is used because 000C3H is replaced by 010C3H.

Caution Bits 7 and 0 (OCDENSET and OCDERSD) can only be specified a value.
Be sure to set 000010B to bits 6 to 1.

Remark The value on bits 3 to 1 will be written over when the on-chip debug function is in use and thus it will become unstable after the setting.
However, be sure to set the default values (0, 1, and 0) to bits 3 to 1 at setting.

27.4 Setting of Option Byte

The user option byte and on-chip debug option byte can be set using the link option, in addition to describing to the source. When doing so, the contents set by using the link option take precedence, even if descriptions exist in the source, as mentioned below.

A software description example of the option byte setting is shown below.

```
OPT   CSEG        OPT_BYTE
      DB          36H     ; Does not use interval interrupt of watchdog timer,
                          ; Enables watchdog timer operation,
                          ; Window open period of watchdog timer is 50%,
                          ; Overflow time of watchdog timer is 2^9/fIL,
                          ; Stops watchdog timer operation during HALT/STOP mode
      DB          7AH     ; Select 2.75 V for VLVDL
                          ; Select rising edge 2.92 V, falling edge 2.86 V for VLVDH
                          ; Select the interrupt & reset mode as the LVD operation mode
      DB          ADH     ; Select the LS (low speed main) mode as the flash operation mode
                            and 1 MHz as the frequency of the high-speed on-chip oscillator
      DB          85H     ; Enables on-chip debug operation, does not erase flash memory
                            data when security ID authorization fails
```

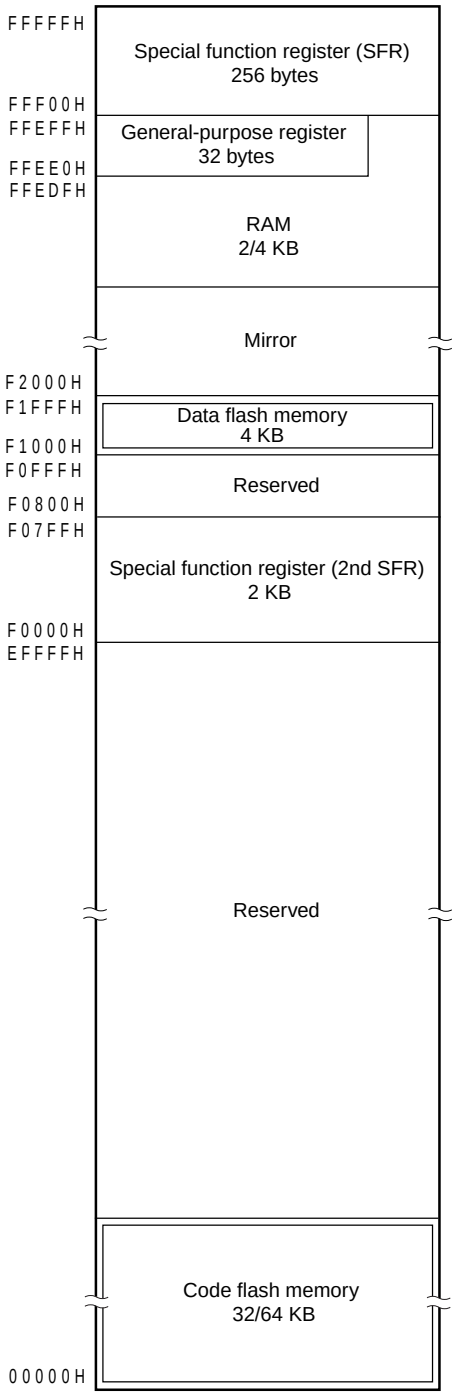
When the boot swap function is used during self programming, 000C0H to 000C3H is switched to 010C0H to 010C3H. Describe to 010C0H to 010C3H, therefore, the same values as 000C0H to 000C3H as follows.

```
OPT2  CSEG   AT   010C0H
      DB          36H     ; Does not use interval interrupt of watchdog timer,
                          ; Enables watchdog timer operation,
                          ; Window open period of watchdog timer is 50%,
                          ; Overflow time of watchdog timer is 2^9/fIL,
                          ; Stops watchdog timer operation during HALT/STOP mode
      DB          7AH     ; Select 2.75 V for VLVDL
                          ; Select rising edge 2.92 V, falling edge 2.86 V for VLVDH
                          ; Select the interrupt & reset mode as the LVD operation mode
      DB          ADH     ; Select the LS (low speed main) mode as the flash operation mode
                            and 1 MHz as the frequency of the high-speed on-chip oscillator
      DB          85H     ; Enables on-chip debug operation, does not erase flash memory
                            data when security ID authorization fails
```

Caution To specify the option byte by using assembly language, use OPT_BYTE as the relocation attribute name of the CSEG pseudo instruction. To specify the option byte to 010C0H to 010C3H in order to use the boot swap function, use the relocation attribute AT to specify an absolute address.

CHAPTER 28 FLASH MEMORY

The RL78 microcontroller incorporates the flash memory to which a program can be written, erased, and overwritten. The flash memory includes the “code flash memory”, in which programs can be executed, and the “data flash memory”, an area for storing data.



The following methods for programming the flash memory are available.

The code flash memory can be rewritten to through serial programming using a flash memory programmer, or through self-programming.

- Serial programming using flash memory programmer (see **28.1**)
  Data can be written to the flash memory on-board or off-board by using a dedicated flash memory programmer.

- Self-programming (see **28.5**)
  The user application can execute self-programming of the code flash memory by using the flash self-programming library.

The data flash memory can be rewritten to by using the data flash library during user program execution (background operation). For access and writing to the data flash memory, see **28.7 Data Flash**.

28.1 Serial Programming Using Flash Memory Programmer

The following dedicated flash memory programmer can be used to write data to the internal flash memory of the RL78 microcontroller.

- PG-FP5, FL-PR5
- E1 on-chip debugging emulator

Data can be written to the flash memory on-board or off-board, by using a dedicated flash memory programmer.

(1) On-board programming

The contents of the flash memory can be rewritten after the RL78 microcontroller has been mounted on the target system. The connectors that connect the dedicated flash memory programmer must be mounted on the target system.

(2) Off-board programming

Data can be written to the flash memory with a dedicated program adapter (FA series) before the RL78 microcontroller is mounted on the target system.

Remark FL-PR5 and FA series are products of Naito Densei Machida Mfg. Co., Ltd.

Table 28-1. Wiring Between RL78/I1A and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin No.		
					20-pin	30-pin	38-pin
Signal Name		I/O	Pin Function		SSOP	SSOP	SSOP
PG-FP5, FL-PR5	E1 on-chip Debugging Emulator						
–	TOOL0	I/O	Transmit/receive signal	TOOL0/P40	3	5	5
SI/RxD	–	I/O	Transmit/receive signal				
–	$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	4	6	6
/RESET	–	Output					
V_{DD}		I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	10	12	14
GND		–	Ground	V_{SS}	9	11	13
				REGC[Note]	8	10	12
FLMD1	EMV_{DD}	–	Driving power for TOOL0 pin	V_{DD}	10	12	14

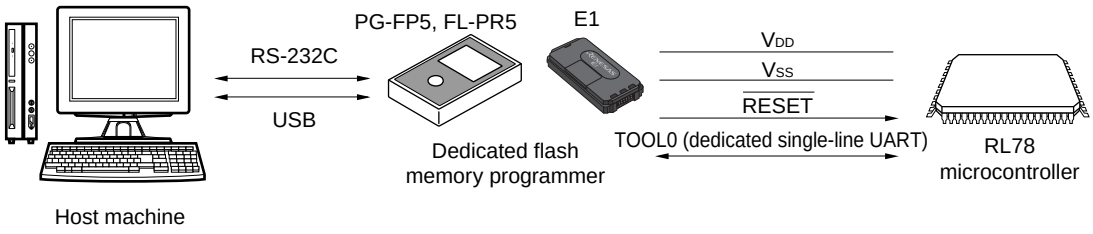
Note Connect REGC pin to ground via a capacitor (default: 0.47 to 1 μF).

Remark Pins that are not indicated in the above table can be left open when using the flash memory programmer for flash programming.

28.1.1 Programming environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 28-1. Environment for Writing Program to Flash Memory



A host machine that controls the dedicated flash memory programmer is necessary.

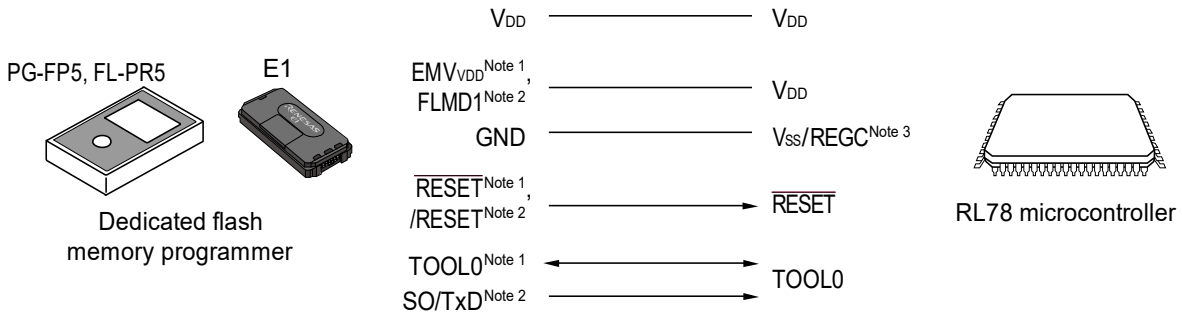
To interface between the dedicated flash memory programmer and the RL78 microcontroller, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART.

28.1.2 Communication mode

Communication between the dedicated flash memory programmer and the RL78 microcontroller is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the RL78 microcontroller.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 28-2. Communication with Dedicated Flash Memory Programmer



Notes 1. When using E1 on-chip debugging emulator.

2. When using PG-FP5 or FL-PR5.

3. Connect REGC pin to ground via a capacitor (default: 0.47 to 1 μF).

The dedicated flash memory programmer generates the following signals for the RL78/I1A. See each manual of PG-FP5, FL-PR5, or E1 on-chip debugging emulator for details.

Table 28-2. Pin Connection

Dedicated Flash Memory Programmer				RL78 Microcontroller
Signal Name		I/O	Pin Function	Pin Name[Note 2]
PG-FP5, FL-PR5	E1 on-chip Debugging Emulator			
FLMD0	–	Output	Mode signal	–
V_{DD}		I/O	V_{DD} voltage generation/power monitoring	V_{DD}
GND		–	Ground	V_{SS}, REGC[Note 1]
FLMD1	EMV_{DD}	–	Driving power for TOOL pin	V_{DD}
/RESET	–	Output	Reset signal	$\overline{\text{RESET}}$
–	$\overline{\text{RESET}}$	Output		
–	TOOL0	I/O	Transmit/receive signal	TOOL0
SI/RxD	–	I/O	Transmit/receive signal	

Notes 1. Connect REGC pin to ground via a capacitor (default: 0.47 to 1 μF).

2. Pins to be connected differ with the product. For details, see **Table 28-1**.

28.2 Connection of Pins on Board

To write the flash memory on-board by using the flash memory programmer, connectors that connect the dedicated flash memory programmer must be provided on the target system. First provide a function that selects the normal operation mode or flash memory programming mode on the board.

When the flash memory programming mode is set, all the pins not used for programming the flash memory are in the same status as immediately after reset. Therefore, if the external device does not recognize the state immediately after reset, the pins must be handled as described below.

Remark For flash programming mode, see **28.3.2 Flash memory programming mode**.

28.2.1 P40/TOOL0 pin

In the flash memory programming mode, connect this pin to the dedicated flash memory programmer via an external 1 kΩ pull-up resistor.

When this pin is used as the port pin, use that by the following method.

When used as an input pin: Input of low-level is prohibited for t_{HD} period after external reset release. Furthermore, when this pin is used via pull-down resistors, use the 500 kΩ or more resistors.

When used as an output pin: When this pin is used via pull-down resistors, use the 500 kΩ or more resistors.

Remarks 1. t_{HD}: How long to keep the TOOL0 pin at the low level from when the external and internal resets end for setting of the flash memory programming mode. See **32.10** or **33.10 Timing of Entry to Flash Memory Programming Modes**.

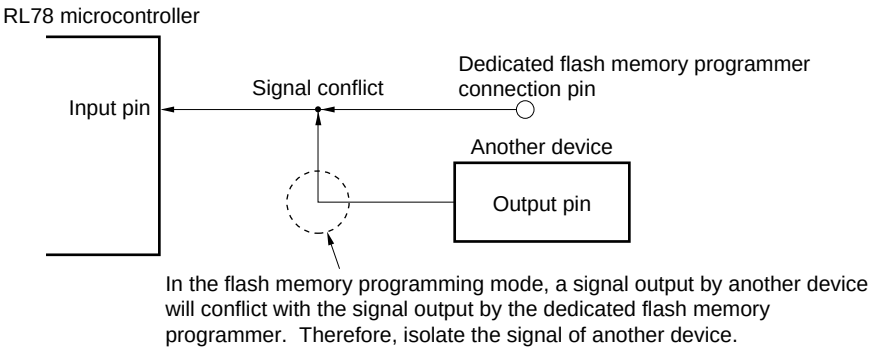
2. The SAU and IICA pins are not used for communication between the RL78 microcontroller and dedicated flash memory programmer, because single-line UART (TOOL0 pin) is used.

28.2.2 $\overline{\text{RESET}}$ pin

Signal conflict will occur if the reset signal of the dedicated flash memory programmer and external device are connected to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on the board. To prevent this conflict, isolate the connection with the reset signal generator.

The flash memory will not be correctly programmed if the reset signal is input from the user system while the flash memory programming mode is set. Do not input any signal other than the reset signal of the dedicated flash memory programmer.

Figure 28-3. Signal Conflict ($\overline{\text{RESET}}$ Pin)



28.2.3 Port pins

When the flash memory programming mode is set, all the pins not used for flash memory programming enter the same status as that immediately after reset. If external devices connected to the ports do not recognize the port status immediately after reset, the port pin must be connected to either to V_{DD} or V_{SS}, via a resistor.

28.2.4 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.

28.2.5 X1 and X2 pins

Connect X1 and X2 in the same status as in the normal operation mode.

Remark In the flash memory programming mode, the high-speed on-chip oscillator clock (f_{IH}) is used.

28.2.6 Power supply

To use the supply voltage output of the flash memory programmer, connect the V_{DD} pin to V_{DD} of the flash memory programmer, and the V_{SS} pin to GND of the flash memory programmer.

To use the on-board supply voltage, connect in compliance with the normal operation mode.

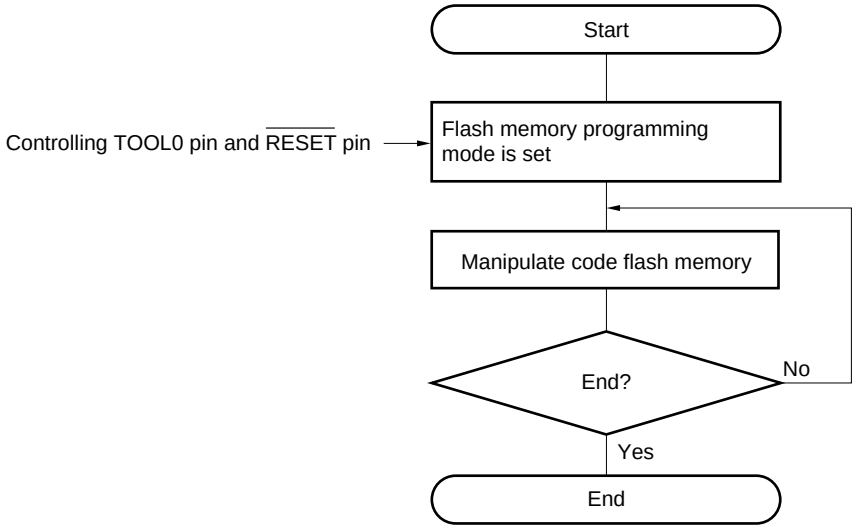
However, when writing to the flash memory by using the flash memory programmer and using the on-board supply voltage, be sure to connect the V_{DD} and V_{SS} pins to V_{DD} and GND of the flash memory programmer to use the power monitor function with the flash memory programmer.

28.3 Serial Programming Method

28.3.1 Serial programming procedure

The following figure illustrates a flow for rewriting the code flash memory through serial programming.

Figure 28-4. Code Flash Memory Manipulation Procedure



28.3.2 Flash memory programming mode

To rewrite the contents of the code flash memory through serial programming, specify flash memory programming mode. To enter the mode, set as follows.

<When serial programming by using the dedicated flash memory programmer>

Communication from the dedicated flash memory programmer is performed to automatically switch to the flash memory programming mode.

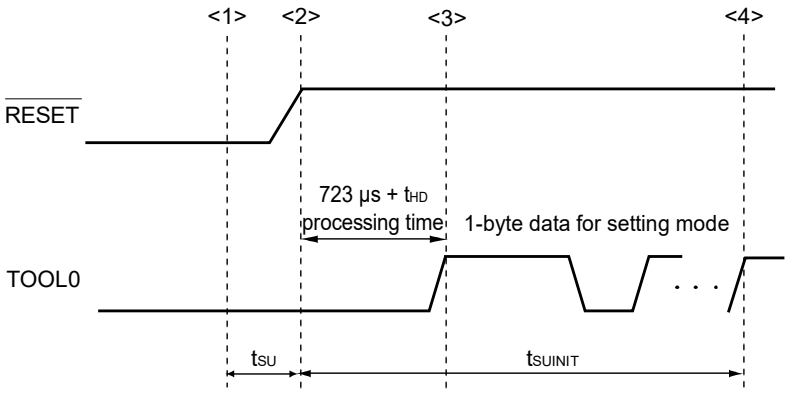
<When serial programming by using an external device (UART communication)>

Set the TOOL0 pin to the low level, and then cancel the reset (see **Table 28-3**). After that, enter flash memory programming mode according to the procedures <1> to <4> shown in Figure 28-5. For details, refer to the **RL78 Microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

Table 28-3. Relationship Between TOOL0 Pin and Operation Mode After Reset Release

TOOL0	Operation Mode
V_{DD}	Normal operation mode
0	Flash memory programming mode

Figure 28-5. Setting of Flash Memory Programming Mode



<1> The low level is input to the TOOL0 pin.
<2> The external reset ends (POR and LVD reset must end before the pin reset ends.).
<3> The TOOL0 pin is set to the high level.
<4> Baud rate setting by UART reception is completed.

Remark t_{SUINIT}: The segment shows that it is necessary to finish specifying the initial communication settings within 100 ms from when the resets end.
t_{SU}: How long from when the TOOL0 pin is placed at the low level until an external reset ends
t_{HD}: How long to keep the TOOL0 pin at the low level from when the external and internal resets end (except soft processing time)

For details, see **32.10** or **33.10 Timing of Entry to Flash Memory Programming Modes**.

There are two flash memory programming modes: wide voltage mode and full speed mode. The supply voltage value applied to the microcontroller during write operations and the setting information of the user option byte for setting of the flash memory programming mode determine which mode is selected.

When a dedicated flash memory programmer is used for serial programming, setting the voltage on GUI selects the mode automatically.

Table 28-4. Programming Modes and Voltages at Which Data Can Be Written, Erased, or Verified

Power Supply Voltage (V_{DD})	User Option Byte Setting for Switching to Flash Memory Programming Mode		Flash Programming Mode
	Flash Operation Mode	Operating Frequency	
2.7 V ≤ V_{DD} ≤ 5.5 V	HS (high-speed main) mode	1 MHz to 32 MHz	Full speed mode
	LS (low-speed main) mode	1 MHz to 8 MHz	Wide voltage mode

Remarks 1. Using both the wide voltage mode and full speed mode imposes no restrictions on writing, deletion, or verification.

2. For details about communication commands, see **28.3.4 Communication commands**.

28.3.3 Selecting communication mode

Communication mode of the RL78 microcontroller as follows.

Table 28-5. Communication Modes

Communication Mode	Standard Setting[Note 1]				Pins Used
	Port	Speed[Note 2]	Frequency	Multiply Rate	
1-line mode (when flash memory programmer is used, or when external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	–	–	TOOL0

Notes 1. Selection items for Standard settings on GUI of the flash memory programmer.

2. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

28.3.4 Communication commands

The RL78 microcontroller executes serial programming through the commands listed in **Table 28-6.**

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 **RL78 Microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

Table 28-6. Flash Memory Control Commands

Classification	Command Name	Function
Verify	Verify	Compares the contents of a specified area of the flash memory with data transmitted from the programmer.
Erase	Block Erase	Erases a specified area in the flash memory.
Blank check	Block Blank Check	Checks if a specified block in the flash memory has been correctly erased.
Write	Programming	Writes data to a specified area in the flash memory.
Getting information	Silicon Signature	Gets the RL78/I1A information (such as the part number and flash memory configuration).
	Version Get	Gets the RL78/I1A firmware version.
	Checksum	Gets the checksum data for a specified area.
Security	Security Set	Sets security information.
	Security Release	Release setting of prohibition of writing.
Others	Reset	Used to detect synchronization status of communication.
	Baud Rate Set	Sets baud rate when UART communication mode is selected.

Note Confirm that no data has been written to the write area. Because data cannot be erased after block erase is prohibited, do not write data if the data has not been erased.
Product information (such as product name and firmware version) can be obtained by executing the “Silicon Signature” command.

Table 28-7 is a list of signature data list and Table 28-8 shows an example of signature data list.

Table 28-7. Signature Data List

Field Name	Description	Number of Transmit Data
Device code	The serial number assigned to the device	3 bytes
Device name	Device name (ASCII code)	10 bytes
Code flash memory area last address	Last address of code flash memory area (Sent from lower address. Example. 00000H to 0FFFFH (64 KB) → FFH, 1FH, 00H)	3 bytes
Data flash memory area last address	Last address of data flash memory area (Sent from lower address. Example. F1000H to F1FFFH (4 KB) → FFH, 1FH, 0FH)	3 bytes
Firmware version	Version information of firmware for programming (Sent from upper address. Example. From Ver. 1.23 → 01H, 02H, 03H)	3 bytes

Table 28-8. Example of Signature Data

Field Name	Description	Number of Transmit Data	Data (Hexadecimal)
Device code	RL78 protocol A	3 bytes	10 00 06
Device name	R5F107AE	10 bytes	52 = “R” 35 = “5” 46 = “F” 31 = “1” 30 = “0” 37 = “7” 41 = “A” 45 = “E” 20 = “ ” 20 = “ ”
Code flash memory area last address	Code flash memory area 00000H to 0FFFFH (64 KB)	3 bytes	FF FF 00
Data flash memory area last address	Data flash memory area F1000H to F1FFFH (4 KB)	3 bytes	FF 1F 0F
Firmware version	Ver.1.23	3 bytes	01 02 03

28.4 Processing Time for Each Command When PG-FP5 Is in Use (Reference Value)

The following shows the processing time for each command (reference value) when PG-FP5 is used as a dedicated flash memory programmer.

Table 28-9. Processing Time for Each Command When PG-FP5 Is in Use (Reference Value)

PG-FP5 Command	Code Flash	
	32 KB	64 KB
Erasing	1 s	1.5 s
Writing	1.5 s	2.5 s
Verification	1.5 s	2 s
Writing after erasing	2 s	3 s

Remark The command processing times (reference values) shown in the table are typical values under the following conditions.

Port: TOOL0 (single-line UART)

Speed: 1,000,000 bps

Mode: Full speed mode (flash operation mode: HS (high-speed main) mode)

28.5 Self-Programming

The RL78 microcontroller supports a self-programming function that can be used to rewrite the code flash memory via a user program. Because this function allows a user application to rewrite the code flash memory by using the flash self-programming library, it can be used to upgrade the program in the field.

Cautions 1. The self-programming function cannot be used when the CPU operates with the subsystem clock.

2. To prohibit an interrupt during self-programming, in the same way as in the normal operation mode, execute the flash self-programming library in the state where the IE flag is cleared (0) by the DI instruction. To enable an interrupt, clear (0) the interrupt mask flag to accept in the state where the IE flag is set (1) by the EI instruction, and then execute the flash self-programming library.

3. The high-speed on-chip oscillator should be kept operating during self-programming. If it is kept stopping, the clock of the high-speed on-chip oscillator should be operated (HIOSTOP = 0). The flash self-programming library should be executed after 30 μs have elapsed.

Remarks 1. For details of the self-programming function, refer to the **RL78 Microcontroller Flash Self Programming Library Type01 User's Manual (R01US0050)**.

2. For details of the time required to execute self-programming, see the notes on use that accompany the flash self-programming library tool.

The self-programming function has two flash memory programming modes; wide voltage mode and full speed mode.

Specify the mode that corresponds to the flash operation mode specified in bits CMODE1 and CMODE0 in option byte 000C2H. Specify the full speed mode when the HS (high-speed main) mode is specified. Specify the wide voltage mode when the LS (low-speed main) mode is specified.

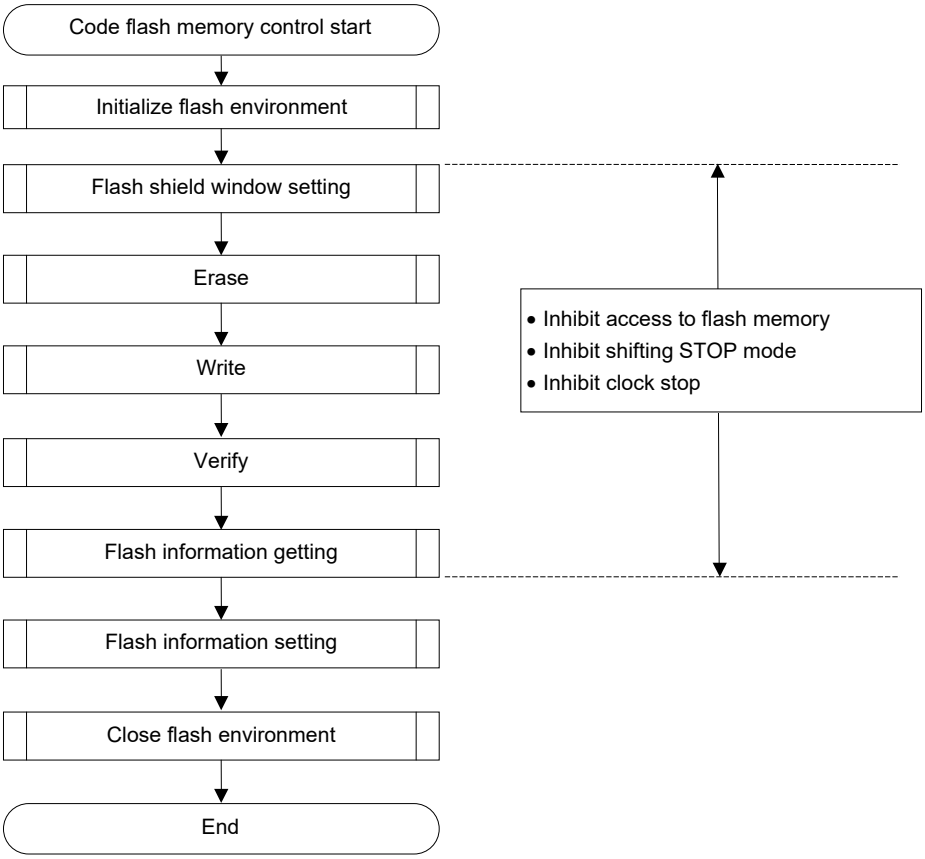
If the argument fsl_flash_voltage_u08 is 00H when the FSL_Init function of the flash self-programming library provided by Renesas Electronics is executed, full speed mode is specified. If the argument is other than 00H, the wide voltage mode is specified.

Remark Using both the wide voltage mode and full speed mode imposes no restrictions on writing, erasing, or verification.

28.5.1 Self-programming procedure

The following figure illustrates a flow of rewriting the code flash memory by using a flash self-programming library.

Figure 28-6. Flow of Self Programming (Rewriting Flash Memory)



28.5.2 Boot swap function

If rewriting the boot area 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.

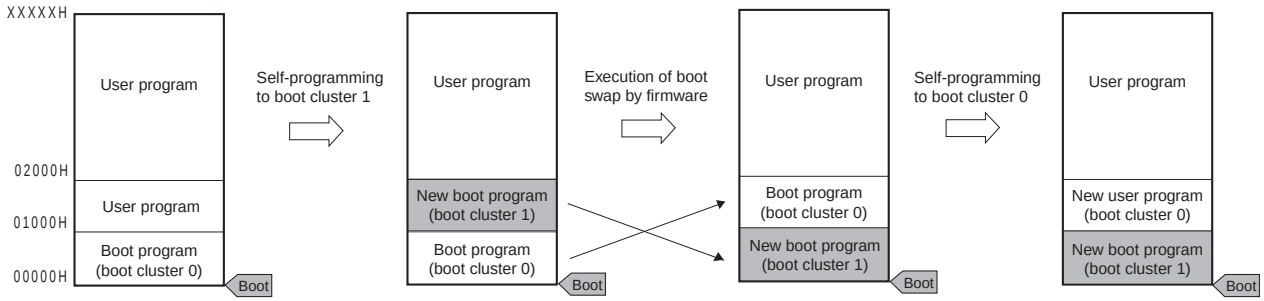
The boot swap function is used to avoid this problem.

Before erasing boot cluster 0[Note], which is a boot area, by self-programming, write a new boot program to boot cluster 1 in advance. When the program has been correctly written to boot cluster 1, swap this boot cluster 1 and boot cluster 0 by using the set information function of the firmware of the RL78 microcontroller, so that boot cluster 1 is used as a boot area. After that, erase or write the original area, 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 to be swapped when the program is reset and started next.

Note A boot cluster is a 4 KB area and boot clusters 0 and 1 are swapped by the boot swap function.

Figure 28-7. Boot Swap Function

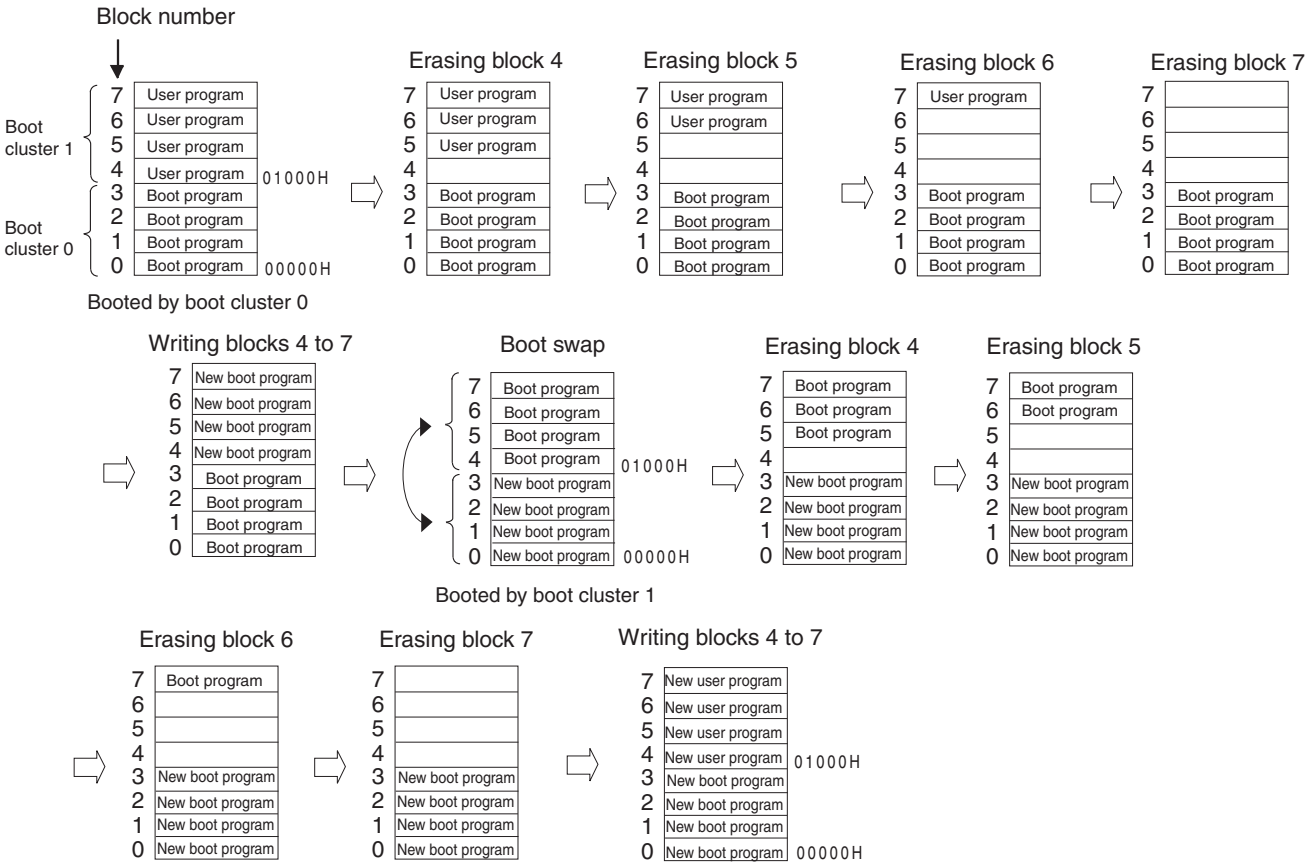


In an example of above figure, it is as follows.

Boot cluster 0: Boot area before boot swap

Boot cluster 1: Boot area after boot swap

Figure 28-8. Example of Executing Boot Swapping



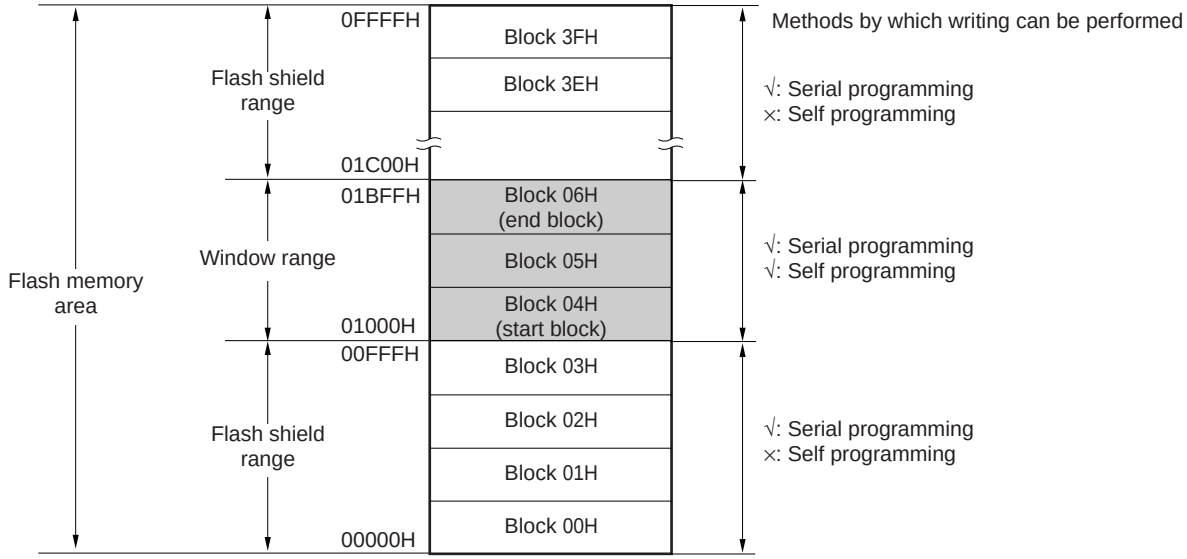
28.5.3 Flash shield window function

The flash shield window function is provided as one of the security functions for self-programming. It disables writing to and erasing areas outside the range specified as a window only during self-programming.

The window range can be set by specifying the start and end blocks. The window range can be set or changed during both serial programming and self-programming.

Writing to and erasing areas outside the window range are disabled during self-programming. During serial programming, however, areas outside the range specified as a window can be written and erased.

Figure 28-9. Flash Shield Window Setting Example
(Target Devices: R5F107AE, Start Block: 04H, End Block: 06H)



Cautions 1. If the rewrite-prohibited area of the boot cluster 0 overlaps with the flash shield window range, prohibition to rewrite the boot cluster 0 takes priority.

2. The flash shield window can only be used for the code flash memory (and is not supported for the data flash memory).

Table 28-10. Relationship Between Flash Shield Window Function Setting/Change Methods and Commands

Programming Conditions	Window Range Setting/Change Methods	Execution Commands	
		Block Erase	Write
Self-programming	Specify the starting and ending blocks by the set flash self-programming library.	Block erasing is enabled only within the window range.	Writing is enabled only within the range of window range.
Serial programming	Specify the starting and ending blocks on GUI of dedicated flash memory programmer, etc.	Block erasing is enabled also outside the window range.	Writing is enabled also outside the window range.

Remark See **28.6 Security Settings** to prohibit writing/erasing during serial programming.

28.6 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 using the Security Set command.

- Disabling block erase
  Execution of the block erase command for a specific block in the flash memory is prohibited during serial programming. However, blocks can be erased by means of self-programming.

- Disabling write
  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 of prohibition of writing is specified, releasing the setting by the Security Release command is enabled by a reset.

- Disabling rewriting boot cluster 0
  Execution of the block erase command and write command on boot cluster 0 (00000H to 00FFFH) in the code flash memory is prohibited by this setting.

After the security settings are specified, releasing the security settings by the Security Release command is enabled by a reset.

The block erase, write commands and rewriting boot cluster 0 are enabled by the default setting when the flash memory is shipped. Security can be set by serial programming and self-programming. Each security setting can be used in combination.

Table 28-11 shows the relationship between the erase and write commands when the RL78 microcontroller security function is enabled.

Caution The security function of the flash programmer does not support self-programming.

Remark To prohibit writing and erasing during self-programming, use the flash shield window function (see **28.5.3** for detail).

Table 28-11. Relationship Between Enabling Security Function and Command

(1) During serial programming

Valid Security	Executed Command	
	Block Erase	Write
Prohibition of block erase	Blocks cannot be erased.	Can be performed.[Note]
Prohibition of writing	Blocks can be erased.	Cannot be performed.
Prohibition of rewriting boot cluster 0	Boot cluster 0 cannot be erased.	Boot cluster 0 cannot be written.

Note Confirm that no data has been written to the write area. Because data cannot be erased after block erase is prohibited, do not write data if the data has not been erased.

(2) During self-programming

Valid Security	Executed Command	
	Block Erase	Write
Prohibition of block erase	Blocks can be erased.	Can be performed.
Prohibition of writing		
Prohibition of rewriting boot cluster 0	Boot cluster 0 cannot be erased.	Boot cluster 0 cannot be written.

Remark To prohibit writing and erasing during self-programming, use the flash shield window function (see **28.5.3** for detail).

Table 28-12. Setting Security in Each Programming Mode

(1) During serial programming

Security	Security Setting	How to Disable Security Setting
Prohibition of block erase	Set via GUI of dedicated flash memory programmer, etc.	Cannot be disabled after set.
Prohibition of writing		Set via GUI of dedicated flash memory programmer, etc.
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

Caution Releasing the setting of prohibition of writing is enabled only when the security is not set as the block erase prohibition and the boot cluster 0 rewrite prohibition with code flash memory area and data flash memory area being blanks.

(2) During self-programming

Security	Security Setting	How to Disable Security Setting
Prohibition of block erase	Set by using flash self-programming library.	Cannot be disabled after set.
Prohibition of writing		Cannot be disabled during self-programming (set via GUI of dedicated flash memory programmer, etc. during serial programming).
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

28.7 Data Flash

28.7.1 Data flash overview

An overview of the data flash memory is provided below.

- The user program can rewrite the data flash memory by using the flash data library. For details, refer to the **RL78 Family Data Flash Library User's Manual**.
- The data flash memory can also be written to through serial programming using the dedicated flash memory programmer or an external device
- The data flash can be erased in 1-block (1 KB) units.
- The data flash can be accessed only in 8-bit units.
- The data flash can be directly read by CPU instructions.
- Instructions can be executed from the code flash memory while rewriting the data flash memory (that is, back ground operation (BGO) is supported).
- Because the data flash memory is an area exclusively used for data, it cannot be used to execute instructions.
- Accessing the data flash memory is not possible while rewriting the code flash memory (during self-programming).
- Manipulating the DFLCTL register is not possible while rewriting the data flash memory.
- Transition to the STOP mode is not possible while rewriting the data flash memory.

Cautions 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.

2. The high-speed on-chip oscillator should be kept operating during data flash rewrite. If it is kept stopping, the high-speed on-chip oscillator clock should be operated (HIOSTOP = 0). The flash data library should be executed after 30 μs have elapsed.

Remark For rewriting the code flash memory via a user program, see **28.5 Self-Programming**.

28.7.2 Register controlling data flash memory

28.7.2.1 Data flash control register (DFLCTL)

This register is used to enable or disable accessing to the data flash.

The DFLCTL register is set by a 1-bit or 8-bit memory manipulation instruction.

Reset input sets this register to 00H.

Figure 28-10. Format of Data Flash Control Register (DFLCTL)

Address: F0090H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
DFLCTL	0	0	0	0	0	0	0	DFLEN

DFLEN	Data flash access control
0	Disables data flash access
1	Enables data flash access

Caution Manipulating the DFLCTL register is not possible while rewriting the data flash memory.

28.7.3 Procedure for accessing data flash memory

The data flash memory is stopped after a reset ends. To access the data flash, make initial settings according to the following procedure.

<1> Set bit 0 (DFLEN) of the data flash control register (DFLCTL) to 1.

<2> Wait for the setup to finish for software timer, etc.

The time setup takes differs for each main clock mode.

<Setup time for each main clock mode>

- HS (High-speed main): 5 μs
- LS (Low-speed main): 720 ns

<3> After the wait, the data flash memory can be accessed.

Cautions 1. Accessing the data flash memory is not possible during the setup time.

2. Transition to the STOP mode is not possible during the setup time. To enter the STOP mode during the setup time, clear DFLEN to 0 and then execute the STOP instruction.

3. The high-speed on-chip oscillator should be kept operating during data flash rewrite. If it is kept stopping, the high-speed on-chip oscillator clock should be operated (HIOSTOP = 0). The flash data library should be executed after 30 μs have elapsed.

<R> **4. Once the data flash memory is read while the subsystem clock is selected as the CPU/peripheral hardware clock (CLS = 1), follow the procedure listed as steps (1) to (3) below, in that order, to read the data flash area after switching the CPU/peripheral hardware clock from the subsystem clock to the main system clock.**

(1) Make sure the main system clock is selected as the CPU/peripheral hardware clock (CLS = 0).

(2) Read data from any location in the data flash area. The value read at this point is undefined.

(3) Wait for the time listed below according to the operating mode, then read data from the desired parts of the data flash area.

HS (high-speed main) mode: 5 µs

LS (low-speed main) mode: 1 µs

After initialization, the data flash memory can be read or rewritten by using a CPU instruction or flash data library.
If the DMA controller operates when the data flash memory is accessed, follow either of these procedures:

(A) Hold pending/forced termination of DMA transfer

Before reading the data flash memory, hold DMA transfer pending of all the channels in use. Note that, wait for at least 3 clocks (f_{CLK}) after setting the DWAITn bit to 1 and before reading the data flash memory. After reading the data flash, cancel holding of DMA transfer pending by clearing the DWAITn bit to 0, or before reading the data flash memory, forcibly terminate DMA transfer in accordance with **19.5.5 Forced termination by software**. Resume DMA transfer after reading the data flash memory.

(B) Accessing data flash memory by using library

Access the data flash memory by using the latest flash data library.

(C) Inserting NOP

Insert a NOP instruction immediately before the instruction that reads the data flash memory.

<Example>

```
MOVW    HL, !addr16     ; Reads RAM.
NOP                     ; Insert NOP instruction before reading the data flash memory.
MOV     A,[DE]          ; Reads the data flash memory.
```

If a high-level language such as C is used, however, the compiler may generate two instructions per code. In this case, read the data flash memory in procedure (A) or (B) because the NOP instruction is not inserted immediately before the data flash reading instruction.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

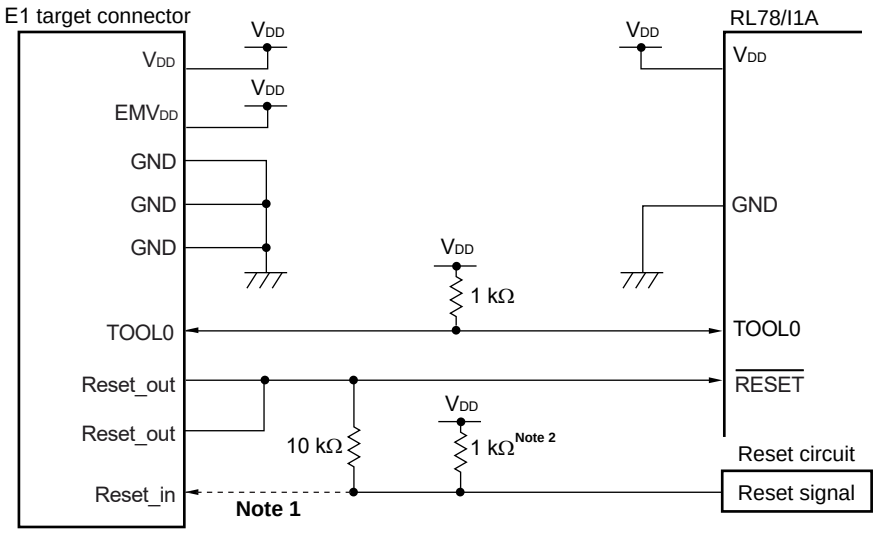
CHAPTER 29 ON-CHIP DEBUG FUNCTION

29.1 Connecting E1 On-chip Debugging Emulator

The RL78 microcontroller uses the V_{DD}, $\overline{RESET}$, TOOL0, and V_{SS} pins to communicate with the host machine via an E1 on-chip debugging emulator. Serial communication is performed by using a single-line UART that uses the TOOL0 pin.

Caution The RL78 microcontroller has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

Figure 29-1. Connection Example of E1 On-chip Debugging Emulator



Notes 1. Connecting the dotted line is not necessary during serial programming.

2. If the reset circuit on the target system does not have a buffer and generates a reset signal only with resistors and capacitors, this pull-up resistor is not necessary.

Caution This circuit diagram is assumed that the reset signal outputs from an N-ch O.D. buffer (output resistor: 100 Ω or less)

29.2 On-Chip Debug Security ID

The RL78 microcontroller has an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 27 OPTION BYTE**) and an on-chip debug security ID setting area at 000C4H to 000CDH, to prevent third parties from reading memory content.

When the boot swap function is used, also set a value that is the same as that of 010C3H and 010C4H to 010CDH in advance, because 000C3H, 000C4H to 000CDH and 010C3H, and 010C4H to 010CDH are switched.

Table 29-1. On-chip Debug Security ID

Address	On-chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytes (exclude all FFH)
010C4H to 010CDH	

29.3 Securing of User Resources

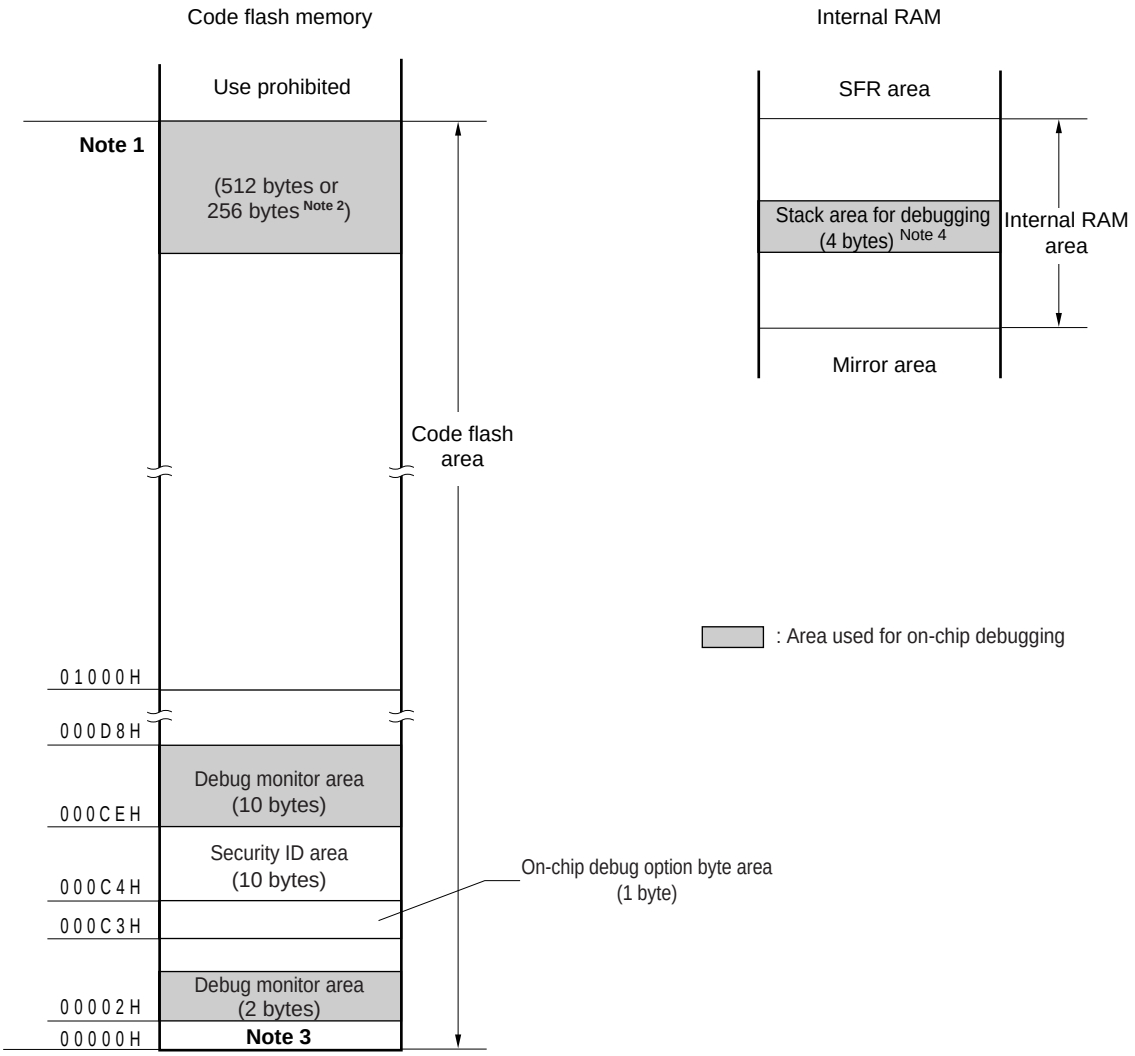
To perform communication between the RL78 microcontroller and E1 on-chip debugging emulator, as well as each debug function, the securing of memory space must be done beforehand.

If Renesas Electronics assembler or compiler is used, the items can be set by using link options.

(1) Securement of memory space

The shaded portions in Figure 29-2 are the areas reserved for placing the debug monitor program, so user programs or data cannot be allocated in these spaces. When using the on-chip debug function, these spaces must be secured so as not to be used by the user program. Moreover, this area must not be rewritten by the user program.

Figure 29-2. Memory Spaces Where Debug Monitor Programs Are Allocated



Notes 1. Address differs depending on products as follows.

Products	Address of **Note 1**
R5F1076C, R5F107AC	07FFFH
R5F107AE, R5F107DE	0FFFFH

2. When real-time RAM monitor (RRM) function and dynamic memory modification (DMM) function are not used, it is 256 bytes.

3. In debugging, reset vector is rewritten to address allocated to a monitor program.

4. Since this area is allocated immediately before the stack area, the address of this area varies depending on the stack increase and decrease. That is, 4 extra bytes are consumed for the stack area used.
When using self-programming, 12 extra bytes are consumed for the stack area used.

CHAPTER 30 BCD CORRECTION CIRCUIT

30.1 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).

30.2 Registers Used by BCD Correction Circuit

The BCD correction circuit uses the following registers.

- BCD correction result register (BCDADJ)

30.2.1 BCD correction result register (BCDADJ)

The BCDADJ register stores correction values for obtaining the add/subtract result as BCD code through add/subtract instructions using the A register as the operand.

The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags.

The BCDADJ register is read by an 8-bit memory manipulation instruction.

Reset input sets this register to undefined.

Figure 30-1. Format of BCD Correction Result Register (BCDADJ)

Address: F00FEH After reset: Undefined R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

30.3 BCD Correction Circuit Operation

The basic operation of the BCD correction circuit is as follows.

(1) Addition: Calculating the result of adding a BCD code value and another BCD code value by using a BCD code value

<1> The BCD code value to which addition is performed is stored in the A register.

<2> By adding the value of the A register and the second operand (value of one more BCD code to be added) as are in binary, the binary operation result is stored in the A register and the correction value is stored in the BCD correction result register (BCDADJ).

<3> Decimal correction is performed by adding in binary the value of the A register (addition result in binary) and the BCDADJ register (correction value), and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

An example is shown below.

Examples 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	–

Examples 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	–

Examples 3: 80 + 80 = 160

Instruction	A Register	CY Flag	AC Flag	BCDADJ Register
`MOV A, #80H ; <1>`	80H	–	–	–
`ADD A, #80H ; <2>`	00H	1	0	60H
`ADD A, !BCDADJ ; <3>`	60H	1	0	–

(2) Subtraction: Calculating the result of subtracting a BCD code value from another BCD code value by using a BCD code value

<1> The BCD code value from which subtraction is performed is stored in the A register.

<2> By subtracting the value of the second operand (value of BCD code to be subtracted) from the A register as is in binary, the calculation result in binary is stored in the A register, and the correction value is stored in the BCD correction result register (BCDADJ).

<3> Decimal correction is performed by subtracting the value of the BCDADJ register (correction value) from the A register (subtraction result in binary) in binary, and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

An example is shown below.

Example: 91 – 52 = 39

Instruction	A Register	CY Flag	AC Flag	BCDADJ Register
`MOV A, #91H ; <1>`	91H	–	–	–
`SUB A, #52H ; <2>`	3FH	0	1	06H
`SUB A, !BCDADJ ; <3>`	39H	0	0	–

CHAPTER 31 INSTRUCTION SET

This chapter lists the instructions in the RL78 microcontroller instruction set. For details of each operation and operation code, refer to the separate document **RL78 Family Microcontrollers User's Manual: Software (R01US0015)**.

31.1 Conventions Used in Operation List

31.1.1 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 31-1. 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 SFR List** 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 Extended SFR (2nd SFR) List** for the symbols of the extended special function registers.

31.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 31-2. 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)
addr15	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)

Caution When accessing the extended SFR area, the wait clock is necessary. For details, see CHAPTER 35 CAUTIONS FOR WAIT.

31.1.3 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 31-3. 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

31.1.4 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 DMA transfer are not acknowledged between a PREFIX instruction code and the instruction immediately after.

Table 31-4. 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.

31.1.5 Number of operating clock cycles

In the RL78/I1A, wait cycles are inserted in the CPU processing in the following cases.

In all other cases, the number of operating clock cycles is the value shown in **31.3 Operation List.**

(1) When accessing a part of the extended SFR area

When accessing the registers in the extended SFR (2nd SFR) area of F0500H to F06FFH, the CPU does not move to the next instruction processing and instead enters the wait state.

For details, see **CHAPTER 35 CAUTIONS FOR WAIT.**

(2) When a hazard occurs due to the combination of instructions

If the data in a register is indirectly accessed immediately after writing to the register that is to be used for the indirect access, a one-clock wait is inserted.

For details, see **31.2 Hazards Related to Combined Instructions**.

(3) Cautions concerning 1-bit memory manipulation instructions

Care is required concerning waits when using 1-bit manipulation instructions.

In the RL78/I1A, 1-bit memory manipulation instructions are executed by using the following procedure.

Example: Executing the SET1 instruction on the Pn register with an output latch

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

When executing a 1-bit memory manipulation instruction on an SFR that causes a wait to occur, the wait will occur for both the reading and writing operations.

31.2 Hazards Related to Combined Instructions

If the data in the register is indirectly accessed immediately after the writing to the register that is to be used for the indirect access, a one-clock wait is inserted.

Table 31-5. Hazards Related to Combined Instructions

Register Name	Previous Instruction	Operand of Next Instruction (or Instruction)
DE	Write instruction to D register[Note] Write instruction to E register[Note] Write instruction to DE register[Note] SEL RBn	[DE], [DE+byte]
HL	Write instruction to H register[Note] Write instruction to L register[Note] Write instruction to HL register[Note] SEL RBn	[HL], [HL+byte], [HL+B], [HL+C], [HL].bit
B	Write instruction to B register[Note] SEL RBn	Word[B], [HL+B]
C	Write instruction to C register[Note] SEL RBn	Word[C], [HL+C]
BC	Write instruction to B register[Note] Write instruction to C register[Note] Write instruction to BC register[Note] SEL RBn	Word[BC], [HL+B], [HL+C]

SP	MOVW SP, #word
	MOVW SP, AX
	ADDW SP, #byte
	SUBW SP, #byte
		[SP+byte]
		CALL instruction, CALLT instruction, BRK instruction, SOFT instruction, RET instruction, RETI instruction, RETB instruction, interrupt, PUSH instruction, POP instruction

CS	MOV CS, #byte MOV CS, A	CALL rp BR AX
AX	Write instruction to A register[Note] Write instruction to X register[Note] Write instruction to AX register[Note] SEL RBn	BR AX
AX BC DE HL	Write instruction to A register[Note] Write instruction to X register[Note] Write instruction to B register[Note] Write instruction to C register[Note] Write instruction to D register[Note] Write instruction to E register[Note] Write instruction to H register[Note] Write instruction to L register[Note] Write instruction to AX register[Note] Write instruction to BC register[Note] Write instruction to DE register[Note] Write instruction to HL register[Note] SEL RBn	CALL rp

Note Register write instructions also require wait cycles when overwriting the target register values during by using addressing, short direct addressing, register indirect addressing, based addressing, or based indexed addressing.

31.3 Operation List

Table 31-6. Operation List (1/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	r, #byte	2	1	–	r ← byte			
		PSW, #byte	3	3	–	PSW ← byte	×	×	×
		CS, #byte	3	1	–	CS ← byte			
		ES, #byte	2	1	–	ES ← byte			
		!addr16, #byte	4	1	–	(addr16) ← byte			
		ES:!addr16, #byte	5	2	–	(ES, addr16) ← byte			
		saddr, #byte	3	1	–	(saddr) ← byte			
		sfr, #byte	3	1	–	sfr ← byte			
		[DE+byte], #byte	3	1	–	(DE+byte) ← byte			
		ES:[DE+byte],#byte	4	2	–	((ES, DE)+byte) ← byte			
		[HL+byte], #byte	3	1	–	(HL+byte) ← byte			
		ES:[HL+byte],#byte	4	2	–	((ES, HL)+byte) ← byte			
		[SP+byte], #byte	3	1	–	(SP+byte) ← byte			
		word[B], #byte	4	1	–	(B+word) ← byte			
		ES:word[B], #byte	5	2	–	((ES, B)+word) ← byte			
		word[C], #byte	4	1	–	(C+word) ← byte			
		ES:word[C], #byte	5	2	–	((ES, C)+word) ← byte			
		word[BC], #byte	4	1	–	(BC+word) ← byte			
		ES:word[BC], #byte	5	2	–	((ES, BC)+word) ← byte			
		A, r Note 3	1	1	–	A ← r			
		r, A Note 3	1	1	–	r ← A			
		A, PSW	2	1	–	A ← PSW			
		PSW, A	2	3	–	PSW ← A	×	×	×
		A, CS	2	1	–	A ← CS			
		CS, A	2	1	–	CS ← A			
		A, ES	2	1	–	A ← ES			
		ES, A	2	1	–	ES ← A			
		A, !addr16	3	1	4	A ← (addr16)			
		A, ES:!addr16	4	2	5	A ← (ES, addr16)			
		!addr16, A	3	1	–	(addr16) ← A			
		ES:!addr16, A	4	2	–	(ES, addr16) ← A			
		A, saddr	2	1	–	A ← (saddr)			
		saddr, A	2	1	–	(saddr) ← A			

Notes 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.

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.

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 31-6. Operation List (2/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	MOV	A, sfr	2	1	–	A ← sfr	
		sfr, A	2	1	–	sfr ← A	
		A, [DE]	1	1	4	A ← (DE)	
		[DE], A	1	1	–	(DE) ← A	
		A, ES:[DE]	2	2	5	A ← (ES, DE)	
		ES:[DE], A	2	2	–	(ES, DE) ← A	
		A, [HL]	1	1	4	A ← (HL)	
		[HL], A	1	1	–	(HL) ← A	
		A, ES:[HL]	2	2	5	A ← (ES, HL)	
		ES:[HL], A	2	2	–	(ES, HL) ← A	
		A, [DE+byte]	2	1	4	A ← (DE + byte)	
		[DE+byte], A	2	1	–	(DE + byte) ← A	
		A, ES:[DE+byte]	3	2	5	A ← ((ES, DE) + byte)	
		ES:[DE+byte], A	3	2	–	((ES, DE) + byte) ← A	
		A, [HL+byte]	2	1	4	A ← (HL + byte)	
		[HL+byte], A	2	1	–	(HL + byte) ← A	
		A, ES:[HL+byte]	3	2	5	A ← ((ES, HL) + byte)	
		ES:[HL+byte], A	3	2	–	((ES, HL) + byte) ← A	
		A, [SP+byte]	2	1	–	A ← (SP + byte)	
		[SP+byte], A	2	1	–	(SP + byte) ← A	
		A, word[B]	3	1	4	A ← (B + word)	
		word[B], A	3	1	–	(B + word) ← A	
		A, ES:word[B]	4	2	5	A ← ((ES, B) + word)	
		ES:word[B], A	4	2	–	((ES, B) + word) ← A	
		A, word[C]	3	1	4	A ← (C + word)	
		word[C], A	3	1	–	(C + word) ← A	
		A, ES:word[C]	4	2	5	A ← ((ES, C) + word)	
		ES:word[C], A	4	2	–	((ES, C) + word) ← A	
		A, word[BC]	3	1	4	A ← (BC + word)	
		word[BC], A	3	1	–	(BC + word) ← A	
		A, ES:word[BC]	4	2	5	A ← ((ES, BC) + word)	
		ES:word[BC], A	4	2	–	((ES, BC) + word) ← A	

Notes 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.

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 31-6. Operation List (3/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	MOV	A, [HL+B]	2	1	4	A ← (HL + B)	
		[HL+B], A	2	1	–	(HL + B) ← A	
		A, ES:[HL+B]	3	2	5	A ← ((ES, HL) + B)	
		ES:[HL+B], A	3	2	–	((ES, HL) + B) ← A	
		A, [HL+C]	2	1	4	A ← (HL + C)	
		[HL+C], A	2	1	–	(HL + C) ← A	
		A, ES:[HL+C]	3	2	5	A ← ((ES, HL) + C)	
		ES:[HL+C], A	3	2	–	((ES, HL) + C) ← A	
		X, !addr16	3	1	4	X ← (addr16)	
		X, ES:!addr16	4	2	5	X ← (ES, addr16)	
		X, saddr	2	1	–	X ← (saddr)	
		B, !addr16	3	1	4	B ← (addr16)	
		B, ES:!addr16	4	2	5	B ← (ES, addr16)	
		B, saddr	2	1	–	B ← (saddr)	
		C, !addr16	3	1	4	C ← (addr16)	
		C, ES:!addr16	4	2	5	C ← (ES, addr16)	
		C, saddr	2	1	–	C ← (saddr)	
		ES, saddr	3	1	–	ES ← (saddr)	
	XCH	A, r Note 3	1 (r = X) 2 (other than r = X)	1	–	A ←→ r	
		A, !addr16	4	2	–	A ←→ (addr16)	
		A, ES:!addr16	5	3	–	A ←→ (ES, addr16)	
		A, saddr	3	2	–	A ←→ (saddr)	
		A, sfr	3	2	–	A ←→ sfr	
		A, [DE]	2	2	–	A ←→ (DE)	
		A, ES:[DE]	3	3	–	A ←→ (ES, DE)	
		A, [HL]	2	2	–	A ←→ (HL)	
		A, ES:[HL]	3	3	–	A ←→ (ES, HL)	
		A, [DE+byte]	3	2	–	A ←→ (DE + byte)	
		A, ES:[DE+byte]	4	3	–	A ←→ ((ES, DE) + byte)	
		A, [HL+byte]	3	2	–	A ←→ (HL + byte)	
		A, ES:[HL+byte]	4	3	–	A ←→ ((ES, HL) + byte)	

Notes 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.

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.

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 31-6. Operation List (4/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	XCH	A, [HL+B]	2	2	–	A ←→ (HL+B)	
		A, ES:[HL+B]	3	3	–	A ←→ ((ES, HL)+B)	
		A, [HL+C]	2	2	–	A ←→ (HL+C)	
		A, ES:[HL+C]	3	3	–	A ←→ ((ES, HL)+C)	
	ONEB	A	1	1	–	A ← 01H	
		X	1	1	–	X ← 01H	
		B	1	1	–	B ← 01H	
		C	1	1	–	C ← 01H	
		!addr16	3	1	–	(addr16) ← 01H	
		ES:!addr16	4	2	–	(ES, addr16) ← 01H	
		saddr	2	1	–	(saddr) ← 01H	
	CLRB	A	1	1	–	A ← 00H	
		X	1	1	–	X ← 00H	
		B	1	1	–	B ← 00H	
		C	1	1	–	C ← 00H	
		!addr16	3	1	–	(addr16) ← 00H	
		ES:!addr16	4	2	–	(ES,addr16) ← 00H	
		saddr	2	1	–	(saddr) ← 00H	
	MOVS	[HL+byte], X	3	1	–	(HL+byte) ← X	× ×
		ES:[HL+byte], X	4	2	–	(ES, HL+byte) ← X	× ×
16-bit data transfer	MOVW	rp, #word	3	1	–	rp ← word	
		saddrp, #word	4	1	–	(saddrp) ← word	
		sfrp, #word	4	1	–	sfrp ← word	
		AX, rp Note 3	1	1	–	AX ← rp	
		rp, AX Note 3	1	1	–	rp ← AX	
		AX, !addr16	3	1	4	AX ← (addr16)	
		!addr16, AX	3	1	–	(addr16) ← AX	
		AX, ES:!addr16	4	2	5	AX ← (ES, addr16)	
		ES:!addr16, AX	4	2	–	(ES, addr16) ← AX	
		AX, saddrp	2	1	–	AX ← (saddrp)	
		saddrp, AX	2	1	–	(saddrp) ← AX	
		AX, sfrp	2	1	–	AX ← sfrp	
		sfrp, AX	2	1	–	sfrp ← AX	

Notes 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.

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.

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 31-6. Operation List (5/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag
				Note 1	Note 2		Z AC CY
16-bit data transfer	MOVW	AX, [DE]	1	1	4	AX ← (DE)	
		[DE], AX	1	1	–	(DE) ← AX	
		AX, ES:[DE]	2	2	5	AX ← (ES, DE)	
		ES:[DE], AX	2	2	–	(ES, DE) ← AX	
		AX, [HL]	1	1	4	AX ← (HL)	
		[HL], AX	1	1	–	(HL) ← AX	
		AX, ES:[HL]	2	2	5	AX ← (ES, HL)	
		ES:[HL], AX	2	2	–	(ES, HL) ← AX	
		AX, [DE+byte]	2	1	4	AX ← (DE+byte)	
		[DE+byte], AX	2	1	–	(DE+byte) ← AX	
		AX, ES:[DE+byte]	3	2	5	AX ← ((ES, DE) + byte)	
		ES:[DE+byte], AX	3	2	–	((ES, DE) + byte) ← AX	
		AX, [HL+byte]	2	1	4	AX ← (HL + byte)	
		[HL+byte], AX	2	1	–	(HL + byte) ← AX	
		AX, ES:[HL+byte]	3	2	5	AX ← ((ES, HL) + byte)	
		ES:[HL+byte], AX	3	2	–	((ES, HL) + byte) ← AX	
		AX, [SP+byte]	2	1	–	AX ← (SP + byte)	
		[SP+byte], AX	2	1	–	(SP + byte) ← AX	
		AX, word[B]	3	1	4	AX ← (B + word)	
		word[B], AX	3	1	–	(B+ word) ← AX	
		AX, ES:word[B]	4	2	5	AX ← ((ES, B) + word)	
		ES:word[B], AX	4	2	–	((ES, B) + word) ← AX	
		AX, word[C]	3	1	4	AX ← (C + word)	
		word[C], AX	3	1	–	(C + word) ← AX	
		AX, ES:word[C]	4	2	5	AX ← ((ES, C) + word)	
		ES:word[C], AX	4	2	–	((ES, C) + word) ← AX	
		AX, word[BC]	3	1	4	AX ← (BC + word)	
		word[BC], AX	3	1	–	(BC + word) ← AX	
		AX, ES:word[BC]	4	2	5	AX ← ((ES, BC) + word)	
		ES:word[BC], AX	4	2	–	((ES, BC) + word) ← AX	

Notes 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.

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 31-6. Operation List (6/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	MOVW	BC, !addr16	3	1	4	BC ← (addr16)			
		BC, ES:!addr16	4	2	5	BC ← (ES, addr16)			
		DE, !addr16	3	1	4	DE ← (addr16)			
		DE, ES:!addr16	4	2	5	DE ← (ES, addr16)			
		HL, !addr16	3	1	4	HL ← (addr16)			
		HL, ES:!addr16	4	2	5	HL ← (ES, addr16)			
		BC, saddrp	2	1	–	BC ← (saddrp)			
		DE, saddrp	2	1	–	DE ← (saddrp)			
		HL, saddrp	2	1	–	HL ← (saddrp)			
	XCHW	AX, rp Note 3	1	1	–	AX ←→ rp			
	ONEW	AX	1	1	–	AX ← 0001H			
		BC	1	1	–	BC ← 0001H			
	CLRW	AX	1	1	–	AX ← 0000H			
		BC	1	1	–	BC ← 0000H			
8-bit operation	ADD	A, #byte	2	1	–	A, CY ← A + byte	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr)+byte	×	×	×
		A, r Note 4	2	1	–	A, CY ← A + r	×	×	×
		r, A	2	1	–	r, CY ← r + A	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16)	×	×	×
		A, saddr	2	1	–	A, 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)	×	×	×

Notes 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.

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.

3. Except rp = AX

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 31-6. Operation List (7/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	ADDC	A, #byte	2	1	–	A, CY ← A+byte+CY	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) +byte+CY	×	×	×
		A, rv Note 3	2	1	–	A, CY ← A + r + CY	×	×	×
		r, A	2	1	–	r, CY ← r + A + CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16)+CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16)+CY	×	×	×
		A, saddr	2	1	–	A, CY ← A + (saddr)+CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A+ (HL) + CY	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A+ (ES, HL) + CY	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A+ (HL+byte) + CY	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A+ ((ES, HL)+byte) + CY	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A+ (HL+B) +CY	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A+((ES, HL)+B)+CY	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A+ (HL+C)+CY	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A+ ((ES, HL)+C)+CY	×	×	×
	SUB	A, #byte	2	1	–	A, CY ← A – byte	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) – byte	×	×	×
		A, r Note 3	2	1	–	A, CY ← A – r	×	×	×
		r, A	2	1	–	r, CY ← r – A	×	×	×
		A, !addr16	3	1	4	A, CY ← A – (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A – (ES, addr16)	×	×	×
		A, saddr	2	1	–	A, CY ← A – (saddr)	×	×	×
		A, [HL]	1	1	4	A, CY ← A – (HL)	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A – (ES, HL)	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A – (HL+byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A – ((ES, HL)+byte)	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A – (HL+B)	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A – ((ES, HL)+B)	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A – (HL+C)	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A – ((ES, HL)+C)	×	×	×

Notes 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.

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.

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 31-6. Operation List (8/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	SUBC	A, #byte	2	1	–	A, CY ← A – byte – CY	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) – byte – CY	×	×	×
		A, r Note 3	2	1	–	A, CY ← A – r – CY	×	×	×
		r, A	2	1	–	r, CY ← r – A – CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A – (addr16) – CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A – (ES, addr16) – CY	×	×	×
		A, saddr	2	1	–	A, CY ← A – (saddr) – CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A – (HL) – CY	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A – (ES, HL) – CY	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A – (HL+byte) – CY	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A – ((ES, HL)+byte) – CY	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A – (HL+B) – CY	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A – ((ES, HL)+B) – CY	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A – (HL+C) – CY	×	×	×
		A, ES:[HL+C]	3	2	5	A, CY ← A – ((ES:HL)+C) – CY	×	×	×
	AND	A, #byte	2	1	–	A ← A ∧ byte	×		
		saddr, #byte	3	2	–	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 3	2	1	–	A ← A ∧ r	×		
		r, A	2	1	–	R ← r ∧ A	×		
		A, !addr16	3	1	4	A ← A ∧ (addr16)	×		
		A, ES:!addr16	4	2	5	A ← A ∧ (ES:addr16)	×		
		A, saddr	2	1	–	A ← A ∧ (saddr)	×		
		A, [HL]	1	1	4	A ← A ∧ (HL)	×		
		A, ES:[HL]	2	2	5	A ← A ∧ (ES:HL)	×		
		A, [HL+byte]	2	1	4	A ← A ∧ (HL+byte)	×		
		A, ES:[HL+byte]	3	2	5	A ← A ∧ ((ES:HL)+byte)	×		
		A, [HL+B]	2	1	4	A ← A ∧ (HL+B)	×		
		A, ES:[HL+B]	3	2	5	A ← A ∧ ((ES:HL)+B)	×		
		A, [HL+C]	2	1	4	A ← A ∧ (HL+C)	×		
		A, ES:[HL+C]	3	2	5	A ← A ∧ ((ES:HL)+C)	×		

Notes 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.

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.

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 31-6. Operation List (9/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag
				Note 1	Note 2		Z AC CY
8-bit operation	OR	A, #byte	2	1	–	A ← A∨byte	×
		saddr, #byte	3	2	–	(saddr) ← (saddr)∨byte	×
		A, r Note 3	2	1	–	A ← A∨r	×
		r, A	2	1	–	r ← r∨A	×
		A, !addr16	3	1	4	A ← A∨(addr16)	×
		A, ES:!addr16	4	2	5	A ← A∨(ES:addr16)	×
		A, saddr	2	1	–	A ← A∨(saddr)	×
		A, [HL]	1	1	4	A ← A∨(H)	×
		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$\veebar$byte	×
		saddr, #byte	3	2	–	(saddr) ← (saddr)$\veebar$byte	×
		A, r Note 3	2	1	–	A ← A$\veebar$r	×
		r, A	2	1	–	r ← r$\veebar$A	×
		A, !addr16	3	1	4	A ← A$\veebar$(addr16)	×
		A, ES:!addr16	4	2	5	A ← A$\veebar$(ES:addr16)	×
		A, saddr	2	1	–	A ← A$\veebar$(saddr)	×
		A, [HL]	1	1	4	A ← A$\veebar$(HL)	×
		A, ES:[HL]	2	2	5	A ← A$\veebar$(ES:HL)	×
		A, [HL+byte]	2	1	4	A ← A$\veebar$(HL+byte)	×
		A, ES:[HL+byte]	3	2	5	A ← A$\veebar$((ES:HL)+byte)	×
		A, [HL+B]	2	1	4	A ← A$\veebar$(HL+B)	×
		A, ES:[HL+B]	3	2	5	A ← A$\veebar$((ES:HL)+B)	×
		A, [HL+C]	2	1	4	A ← A$\veebar$(HL+C)	×
		A, ES:[HL+C]	3	2	5	A ← A$\veebar$((ES:HL)+C)	×

Notes 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.

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.

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 31-6. Operation List (10/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	CMP	A, #byte	2	1	–	A – byte	×	×	×
		!addr16, #byte	4	1	4	(addr16) – byte	×	×	×
		ES:!addr16, #byte	5	2	5	(ES:addr16) – byte	×	×	×
		saddr, #byte	3	1	–	(saddr) – byte	×	×	×
		A, r Note3	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)	×	×	×

Notes 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.

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.

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 31-6. Operation List (11/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	ADDW	AX, #word	3	1	–	AX, CY ← AX+word	×	×	×
		AX, AX	1	1	–	AX, CY ← AX+AX	×	×	×
		AX, BC	1	1	–	AX, CY ← AX+BC	×	×	×
		AX, DE	1	1	–	AX, CY ← AX+DE	×	×	×
		AX, HL	1	1	–	AX, CY ← AX+HL	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX+(addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX+(ES:addr16)	×	×	×
		AX, saddrp	2	1	–	AX, CY ← AX+(saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX+(HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX+((ES:HL)+byte)	×	×	×
	SUBW	AX, #word	3	1	–	AX, CY ← AX – word	×	×	×
		AX, BC	1	1	–	AX, CY ← AX – BC	×	×	×
		AX, DE	1	1	–	AX, CY ← AX – DE	×	×	×
		AX, HL	1	1	–	AX, CY ← AX – HL	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX – (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX – (ES:addr16)	×	×	×
		AX, saddrp	2	1	–	AX, CY ← AX – (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX – (HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX – ((ES:HL)+byte)	×	×	×
	CMPW	AX, #word	3	1	–	AX – word	×	×	×
		AX, BC	1	1	–	AX – BC	×	×	×
		AX, DE	1	1	–	AX – DE	×	×	×
		AX, HL	1	1	–	AX – HL	×	×	×
		AX, !addr16	3	1	4	AX – (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX – (ES:addr16)	×	×	×
		AX, saddrp	2	1	–	AX – (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX – (HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX – ((ES:HL)+byte)	×	×	×
Multiply	MULU	X	1	1	–	AX ← A×X			

Notes 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.

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 31-6. Operation List (12/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag
				Note 1	Note 2		Z AC CY
Increment/ decrement	INC	r	1	1	–	r ← r+1	× ×
		!addr16	3	2	–	(addr16) ← (addr16)+1	× ×
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16)+1	× ×
		saddr	2	2	–	(saddr) ← (saddr)+1	× ×
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte)+1	× ×
		ES: [HL+byte]	4	3	–	((ES:HL)+byte) ← ((ES:HL)+byte)+1	× ×
	DEC	r	1	1	–	r ← r – 1	× ×
		!addr16	3	2	–	(addr16) ← (addr16) – 1	× ×
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) – 1	× ×
		saddr	2	2	–	(saddr) ← (saddr) – 1	× ×
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) – 1	× ×
		ES: [HL+byte]	4	3	–	((ES:HL)+byte) ← ((ES:HL)+byte) – 1	× ×
	INCW	rp	1	1	–	rp ← rp+1	
		!addr16	3	2	–	(addr16) ← (addr16)+1	
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16)+1	
		saddrp	2	2	–	(saddrp) ← (saddrp)+1	
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte)+1	
		ES: [HL+byte]	4	3	–	((ES:HL)+byte) ← ((ES:HL)+byte)+1	
	DECW	rp	1	1	–	rp ← rp – 1	
		!addr16	3	2	–	(addr16) ← (addr16) – 1	
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) – 1	
		saddrp	2	2	–	(saddrp) ← (saddrp) – 1	
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) – 1	
		ES: [HL+byte]	4	3	–	((ES:HL)+byte) ← ((ES:HL)+byte) – 1	
Shift	SHR	A, cnt	2	1	–	(CY ← A_0, A_{m-1} ← A_m, A_7 ← 0) ×cnt	×
	SHRW	AX, cnt	2	1	–	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← 0) ×cnt	×
	SHL	A, cnt	2	1	–	(CY ← A_7, A_m ← A_{m-1}, A_0 ← 0) ×cnt	×
		B, cnt	2	1	–	(CY ← B_7, B_m ← B_{m-1}, B_0 ← 0) ×cnt	×
		C, cnt	2	1	–	(CY ← C_7, C_m ← C_{m-1}, C_0 ← 0) ×cnt	×
	SHLW	AX, cnt	2	1	–	(CY ← AX_{15}, AX_m ← AX_{m-1}, AX_0 ← 0) ×cnt	×
		BC, cnt	2	1	–	(CY ← BC_{15}, BC_m ← BC_{m-1}, BC_0 ← 0) ×cnt	×
	SAR	A, cnt	2	1	–	(CY ← A_0, A_{m-1} ← A_m, A_7 ← A_7) ×cnt	×
	SARW	AX, cnt	2	1	–	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← AX_{15}) ×cnt	×

Notes 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.

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.

Remarks 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.

2. cnt indicates the bit shift count.

Table 31-6. Operation List (13/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Rotate	ROR	A, 1	2	1	–	(CY, $A_7 \leftarrow A_0$, $A_{m-1} \leftarrow A_m$)×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			×

Notes 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.

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 31-6. Operation List (14/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipulate	XOR1	CY, A.bit	2	1	–	CY ← CY ⩛ 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 ← $\overline{CY}$			×

Notes 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

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 31-6. Operation List (15/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag
				Note 1	Note 2		Z AC CY
Call/ return	CALL	rp	2	3	–	(SP – 2) ← (PC+2)S, (SP – 3) ← (PC+2)H, (SP – 4) ← (PC+2)L, PC ← CS, rp, SP ← SP – 4	
		$!addr20	3	3	–	(SP – 2) ← (PC+3)S, (SP – 3) ← (PC+3)H, (SP – 4) ← (PC+3)L, PC ← PC+3+jdisp16, SP ← SP – 4	
		!addr16	3	3	–	(SP – 2) ← (PC+3)S, (SP – 3) ← (PC+3)H, (SP – 4) ← (PC+3)L, PC ← 0000, addr16, SP ← SP – 4	
		!!addr20	4	3	–	(SP – 2) ← (PC+4)S, (SP – 3) ← (PC+4)H, (SP – 4) ← (PC+4)L, PC ← addr20, SP ← SP – 4	
	CALLT	[addr5]	2	5	–	(SP – 2) ← (PC+2)S , (SP – 3) ← (PC+2)H, (SP – 4) ← (PC+2)L , PCS ← 0000, PCH ← (0000, addr5+1), PCL ← (0000, addr5), SP ← SP – 4	
	BRK	-	2	5	–	(SP – 1) ← PSW, (SP – 2) ← (PC+2)S, (SP – 3) ← (PC+2)H, (SP – 4) ← (PC+2)L, PCS ← 0000, PCH ← (0007FH), PCL ← (0007EH), SP ← SP – 4, IE ← 0	
	RET	-	1	6	–	PCL ← (SP), PCH ← (SP+1), PCS ← (SP+2), SP ← SP+4	
	RETI	-	2	6	–	PCL ← (SP), PCH ← (SP+1), PCS ← (SP+2), PSW← (SP+3), SP ← SP+4	R R R
	RETB	-	2	6	–	PCL ← (SP), PCH ← (SP+1), PCS ← (SP+2), PSW ← (SP+3), SP ← SP+4	R R R

Notes 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.

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 31-6. Operation List (16/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag
				Note 1	Note 2		Z AC CY
Stack manipulate	PUSH	PSW	2	1	–	(SP – 1) ← PSW, (SP – 2) ← 00H, SP ← SP–2	
		rp	1	1	–	(SP – 1) ← rp_H, (SP – 2) ← rp_L, SP ← SP – 2	
	POP	PSW	2	3	–	PSW ← (SP+1), SP ← SP + 2	R R R
		rp	1	1	–	rp_L ←(SP), rp_H ← (SP+1), SP ← SP + 2	
	MOVW	SP, #word	4	1	–	SP ← word	
		SP, AX	2	1	–	SP ← AX	
		AX, SP	2	1	–	AX ← SP	
		HL, SP	3	1	–	HL ← SP	
		BC, SP	3	1	–	BC ← SP	
		DE, SP	3	1	–	DE ← SP	
	ADDW	SP, #byte	2	1	–	SP ← SP + byte	
	SUBW	SP, #byte	2	1	–	SP ← SP – byte	
Unconditional branch	BR	AX	2	3	–	PC ← CS, AX	
		$addr20	2	3	–	PC ← PC + 2 + jdisp8	
		$!addr20	3	3	–	PC ← PC + 3 + jdisp16	
		!addr16	3	3	–	PC ← 0000, addr16	
		!!addr20	4	3	–	PC ← addr20	
Conditional branch	BC	$addr20	2	2/4 Note3	–	PC ← PC + 2 + jdisp8 if CY = 1	
	BNC	$addr20	2	2/4 Note3	–	PC ← PC + 2 + jdisp8 if CY = 0	
	BZ	$addr20	2	2/4 Note3	–	PC ← PC + 2 + jdisp8 if Z = 1	
	BNZ	$addr20	2	2/4 Note3	–	PC ← PC + 2 + jdisp8 if Z = 0	
	BH	$addr20	3	2/4 Note3	–	PC ← PC + 3 + jdisp8 if (Z∨CY)=0	
	BNH	$addr20	3	2/4 Note3	–	PC ← PC + 3 + jdisp8 if (Z∨CY)=1	
	BT	saddr.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1	
		sfr.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1	
		A.bit, $addr20	3	3/5 Note3	–	PC ← PC + 3 + jdisp8 if A.bit = 1	
		PSW.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1	
		[HL].bit, $addr20	3	3/5 Note3	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 1	
		ES:[HL].bit, $addr20	4	4/6 Note3	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1	

Notes 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.

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.

3. This indicates the number of clocks “when condition is not met/when condition is met”.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 31-6. Operation List (17/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag
				Note 1	Note 2		Z AC CY
Conditional branch	BF	saddr.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 0	
		sfr.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if sfr.bit = 0	
		A.bit, $addr20	3	3/5 Note3	–	PC ← PC + 3 + jdisp8 if A.bit = 0	
		PSW.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if PSW.bit = 0	
		[HL].bit, $addr20	3	3/5 Note3	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 0	
		ES:[HL].bit, $addr20	4	4/6 Note3	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0	
	BTCLR	saddr.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit	
		sfr.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit	
		A.bit, $addr20	3	3/5 Note3	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit	
		PSW.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	× × ×
		[HL].bit, $addr20	3	3/5 Note3	–	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit	
		ES:[HL].bit, $addr20	4	4/6 Note3	–	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1 then reset (ES, HL).bit	
Conditional skip	SKC	–	2	1	–	Next instruction skip if CY = 1	
	SKNC	–	2	1	–	Next instruction skip if CY = 0	
	SKZ	–	2	1	–	Next instruction skip if Z = 1	
	SKNZ	–	2	1	–	Next instruction skip if Z = 0	
	SKH	–	2	1	–	Next instruction skip if (Z∨CY)=0	
	SKNH	–	2	1	–	Next instruction skip if (Z∨CY)=1	
CPU control	SEL Note4	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	

Notes 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.

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.

3. This indicates the number of clocks “when condition is not met/when condition is met”.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

CHAPTER 32 ELECTRICAL SPECIFICATIONS (G: Industrial applications, T_A = –40 to +105°C)

In this chapter, shows the electrical specifications of the target products.

Target products (G: Industrial applications): T_A = – 40 to + 105°C
R5F107xxGxx

Cautions 1. The RL78/I1A has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

2. The pins mounted depend on the product. See 2.1 Port Function to 2.2.1 Functions for each product.

32.1 Absolute Maximum Ratings

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}		–0.5 to +6.5	V
REGC pin input voltage	V_{IREGC}	REGC	–0.3 to +2.8 and –0.3 to V_{DD} +0.3 Note 1	V
Input voltage	V_{I1}	P02, P03, P05, P06, P10 to P12, P20 to P22, P24 to P27, P30, P31, P40, P75 to P77, P120 to P124, P137, P147, P200 to P206, EXCLK, EXCLKS, $\overline{\text{RESET}}$	–0.3 to V_{DD} +0.3 Note 2	V
Output voltage	V_{O1}	P02, P03, P05, P06, P10 to P12, P20 to P22, P24 to P27, P30, P31, P40, P75 to P77, P120, P147, P200 to P206	–0.3 to V_{DD} +0.3 Note 2	V
Analog input voltage	V_{AI1}	ANI0 to ANI2, ANI4 to ANI7, ANI16 to ANI19	–0.3 to V_{DD} +0.3 and –0.3 to $AV_{REF(+)}$ +0.3 Notes 2, 3	V

Notes 1. Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

2. Must be 6.5 V or lower.

3. Do not exceed $AV_{REF(+)}$ + 0.3 V in case of A/D conversion target pin.

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

2. $AV_{REF(+)}$: + side reference voltage of the A/D converter.

3. V_{SS}: Reference voltage

Absolute Maximum Ratings (T_A = 25°C) (2/2)

Parameter	Symbols	Conditions		Ratings	Unit
Output current, high	I_{OH1}	Per pin	P02, P03, P05, P06, P10 to P12, P30, P31, P40, P75 to P77, P120, P147, P200 to P206	–40	mA
		Total of all pins –170 mA	P02, P03, P40, P120	–70	mA
			P05, P06, P10 to P12, P30, P31, P75 to P77, P147, P200 to P206	–100	mA
	I_{OH2}	Per pin	P20 to P22, P24 to P27	–0.5	mA
		Total of all pins		–2	mA
Output current, low	I_{OL1}	Per pin	P02, P03, P05, P06, P10 to P12, P30, P31, P40, P75 to P77, P120, P147, P200 to P206	40	mA
		Total of all pins 170 mA	P02, P03, P40, P120	70	mA
			P05, P06, P10 to P12, P30, P31, P75 to P77, P147, P200 to P206	100	mA
	I_{OL2}	Per pin	P20 to P22, P24 to P27	1	mA
		Total of all pins		5	mA
Operating ambient temperature	T_A	In normal operation mode		–40 to +105	°C
		In flash memory programming mode			
Storage temperature	T_{stg}			–65 to +150	°C

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

32.2 Oscillator Characteristics

32.2.1 X1, XT1 oscillator characteristics

(T_A = –40 to +105°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Resonator	Conditions	MIN.	TYP.	MAX.	Unit
X1 clock oscillation frequency (f_X)Note	Ceramic resonator/crystal resonator		1.0		20.0	MHz
XT1 clock oscillation frequency (f_{XT})Note	Crystal resonator		32	32.768	35	kHz

Note Indicates only permissible oscillator frequency ranges. See **AC Characteristics** for instruction execution time. Request evaluation by the manufacturer of the oscillator circuit mounted on a board to check the oscillator characteristics.

Caution Since the CPU is started by the high-speed on-chip oscillator clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and the oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.

Remark When using the X1 oscillator and XT1 oscillator, see **5.4 System Clock Oscillator**.

32.2.2 On-chip oscillator characteristics

(T_A = –40 to +105°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Oscillators	Parameters	Conditions	MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency[Note 1]	f_{IH}		1		32	MHz
High-speed on-chip oscillator clock frequency accuracy[Note 2]		T_A = –20 to 85°C	–1		+1	%
		T_A = –40 to 105°C	–1.5		+1.5	%
Low-speed on-chip oscillator clock frequency	f_{IL}			15		kHz
Low-speed on-chip oscillator clock frequency accuracy			–15		+15	%

Notes 1. Frequency can be selected in a high-speed on-chip oscillator. Selected by bits 0 to 3 of option byte (000C2H/010C2H).

2. This indicates the oscillator characteristics only. See **AC Characteristics** for instruction execution time.

32.2.3 PLL characteristics

(T_A = –40 to +105°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
PLL input clock frequency[Note]	f_{PLLIN}	High-speed system clock is selected (f_{MX} = 4 MHz)	3.94	4.00	4.06	MHz
		High-speed on-chip oscillator clock is selected (f_{IH} = 4 MHz)	3.94	4.00	4.06	MHz
PLL output clock frequency[Note]	f_{PLL}		f_{PLLIN} × 16			MHz

Note This only indicates the oscillator characteristics. See **AC Characteristics** for instruction execution time.

32.3 DC Characteristics

32.3.1 Pin characteristics

(T_A = –40 to +105°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high[Note 1]	I_{OH1}	Per pin for P02, P03, P05, P06, P10 to P12, P30, P31, P40, P75 to P77, P120, P147, P200 to P206	4.0 V ≤ V_{DD} ≤ 5.5 V			–3.0[Note 2]	mA
			2.7 V ≤ V_{DD} < 4.0 V			–1.0	mA
		Total of P02, P03, P40, P120 (When duty ≤ 70%[Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			–12.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–4.0	mA
		Total of P05, P06, P10 to P12, P30, P31, P75 to P77, P147, P200 to P206 (When duty ≤ 70%[Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			–30.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–10.0	mA
		Total of all pins (When duty ≤ 70%[Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			–30.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–14.0	mA
	I_{OH2}	Per pin for P20 to P22, P24 to P27	2.7 V ≤ V_{DD} ≤ 5.5 V			–0.1[Note 2]	mA
		Total of all pins (When duty ≤ 70%[Note 3])	2.7 V ≤ V_{DD} ≤ 5.5 V			–0.7	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from the V_{DD} pin to an output pin.

2. However, do not exceed the total current value.

3. Specification under conditions where the duty factor ≤ 70%.
The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).

- Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OH} = –10.0 mA

Total output current of pins = (–10.0 × 0.7)/(80 × 0.01) ≅ –8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Caution P02, P10 to P12 do not output high level in N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = –40 to +105°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low[Note 1]	I_{OL1}	Per pin for P02, P03, P05, P06, P10 to P12, P30, P31, P40, P75 to P77, P120, P147, P200 to P206	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 P02, P03, P40, P120 (When duty ≤ 70%[Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			40.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			7.5	mA
		Total of P05, P06, P10 to P12, P30, P31, P75 to P77, P147, P200 to P206 (When duty ≤ 70%[Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			40.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			17.5	mA
		Total of all pins (When duty ≤ 70%[Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			80.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			25.0	mA
	I_{OL2}	Per pin for P20 to P22, P24 to P27	2.7 V ≤ V_{DD} ≤ 5.5 V			0.4[Note 2]	mA
		Total of all pins (When duty ≤ 70%[Note 3])	2.7 V ≤ V_{DD} ≤ 5.5 V			2.8	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to the V_{SS} pin.

2. However, do not exceed the total current value.

3. Specification under conditions where the duty factor ≤ 70%.
The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).

- Total output current of pins = (I_{OL} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OL} = –10.0 mA

Total output current of pins = (–10.0 × 0.7)/(80 × 0.01) ≅ – 8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = –40 to +105°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P02, P03, P05, P06, P10 to P12, P20 to P22, P24 to P27, P30, P31, P40, P75 to P77, P120 to P124, P137, P147, P200 to P206, EXCLK, EXCLKS, $\overline{\text{RESET}}$	Normal input buffer	0.8V_{DD}		V_{DD}	V
	V_{IH2}	P03, P10, P11	TTL input buffer 4.0 V ≤ V_{DD} ≤ 5.5 V	2.1		V_{DD}	V
			TTL input buffer 3.3 V ≤ V_{DD} < 4.0 V	2.0		V_{DD}	V
			TTL input buffer 2.7 V ≤ V_{DD} < 3.3 V	1.5		V_{DD}	V
Input voltage, low	V_{IL1}	P02, P03, P05, P06, P10 to P12, P20 to P22, P24 to P27, P30, P31, P40, P75 to P77, P120 to P124, P137, P147, P200 to P206, EXCLK, EXCLKS, $\overline{\text{RESET}}$	Normal input buffer	0		0.2V_{DD}	V
	V_{IL2}	P03, P10, P11	TTL input buffer 4.0 V ≤ V_{DD} ≤ 5.5 V	0		0.8	V
			TTL input buffer 3.3 V ≤ V_{DD} < 4.0 V	0		0.5	V
			TTL input buffer 2.7 V ≤ V_{DD} < 3.3 V	0		0.32	V

Caution The maximum value of V_{IH} of pins P02, P10 to P12 is V_{DD}, even in the N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = –40 to +105°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH1}	P02, P03, P05, P06, P10 to P12, P30, P31, P40, P75 to P77, P120, P147, P200 to P206	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –3.0 mA	V_{DD} – 0.7			V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –1.0 mA	V_{DD} – 0.5			V
	V_{OH2}	P20 to P22, P24 to P27	2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OH2} = –100 μA	V_{DD} – 0.5			V
Output voltage, low	V_{OL1}	P02, P03, P05, P06, P10 to P12, P30, P31, P40, P75 to P77, P120, P147, P200 to P206	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 8.5 mA			0.7	V
			4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 4.0 mA			0.4	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 1.5 mA			0.4	V
	V_{OL2}	P20 to P22, P24 to P27	2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL2} = 400 μA			0.4	V

Caution P02, P10 to P12 do not output high level in N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = –40 to +105°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P02, P03, P05, P06, P10 to P12, P20 to P22, P24 to P27, P30, P31, P40, P75 to P77, P120, P137, P147, P200 to P206, $\overline{\text{RESET}}$	$V_I = V_{DD}$				1	μA
	I_{LIH2}	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	$V_I = V_{DD}$	In input port or external clock input			1	μA
				In resonator connection			10	μA
Input leakage current, low	I_{LIL1}	P02, P03, P05, P06, P10 to P12, P20 to P22, P24 to P27, P30, P31, P40, P75 to P77, P120, P137, P147, P200 to P206, $\overline{\text{RESET}}$	$V_I = V_{SS}$				–1	μA
	I_{LIL2}	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	$V_I = V_{SS}$	In input port or external clock input			–1	μA
				In resonator connection			–10	μA
On-chip pull-up resistance	R_U	P02, P03, P05, P06, P10 to P12, P30, P31, P40, P75 to P77, P120, P147, P200 to P206	$V_I = V_{SS}$, In input port		10	20	100	kΩ

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

32.3.2 Supply current characteristics

(T_A = –40 to +105°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (1/2)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD1}	Operating mode	HS (high-speed main) mode Note 5	f_{IH} = 32 MHz Note 3	V_{DD} = 5.0 V		5.0	7.5	mA
					V_{DD} = 3.0 V		5.0	7.5	mA
				f_{IH} = 24 MHz Note 3	V_{DD} = 5.0 V		3.9	5.8	mA
					V_{DD} = 3.0 V		3.9	5.8	mA
				f_{IH} = 16 MHz Note 3	V_{DD} = 5.0 V		2.9	4.2	mA
					V_{DD} = 3.0 V		2.9	4.2	mA
			LS (low-speed main) mode Note 5	f_{IH} = 8 MHz Note 3, T_A = –40 to + 85°C	V_{DD} = 3.0 V		1.3	2.0	mA
			HS (high-speed main) mode Note 5	f_{MX} = 20 MHz Note 2, V_{DD} = 5.0 V	Square wave input		3.2	4.9	mA
					Resonator connection		3.3	5.0	mA
				f_{MX} = 20 MHz Note 2, V_{DD} = 3.0 V	Square wave input		3.2	4.9	mA
					Resonator connection		3.3	5.0	mA
				f_{MX} = 10 MHz Note 2, V_{DD} = 5.0 V	Square wave input		2.0	2.9	mA
					Resonator connection		2.0	2.9	mA
				f_{MX} = 10 MHz Note 2, V_{DD} = 3.0 V	Square wave input		2.0	2.9	mA
					Resonator connection		2.0	2.9	mA
			LS (low-speed main) mode Note 5	f_{MX} = 8 MHz Note 2, V_{DD} = 3.0 V, T_A = –40 to + 85°C	Square wave input		1.2	1.8	mA
					Resonator connection		1.2	1.8	mA
			HS (high-speed main) mode Note 5	f_{IH} = 4 MHz Note 3 f_{PLL} = 64 MHz, f_{CLK} = 32 MHz	V_{DD} = 5.0 V		5.4	8.5	mA
					V_{DD} = 3.0 V		5.4	8.5	mA
				f_{IH} = 4 MHz Note 3 f_{PLL} = 64 MHz, f_{CLK} = 16 MHz	V_{DD} = 5.0 V		3.3	5.7	mA
					V_{DD} = 3.0 V		3.3	5.7	mA
			Subsystem clock operation	f_{SUB} = 32.768 kHz Note 4 T_A = –40°C	Square wave input		4.2	6.0	μA
					Resonator connection		4.4	6.2	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +25°C	Square wave input		4.2	6.0	μA
					Resonator connection		4.4	6.2	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +50°C	Square wave input		4.3	7.2	μA
					Resonator connection		4.5	7.4	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +70°C	Square wave input		4.4	8.1	μA
					Resonator connection		4.6	8.3	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +85°C	Square wave input		5.2	11.4	μA
					Resonator connection		5.4	11.6	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +105°C	Square wave input		6.9	20.8	μA
					Resonator connection		7.1	21.0	μA

(**Notes** and **Remarks** are listed on the next page.)

Notes 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The following points apply in the HS (high-speed main), and LS (low-speed main) modes
- The currents in the “TYP.” column do not include the operating currents of the peripheral modules.
- The currents in the “MAX.” column include the operating currents of the peripheral modules, except for those flowing into the A/D converter, comparator, programmable gain amplifier, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors, and those flowing while the data flash memory is being rewritten.

In the subsystem clock operation, the currents in both the “TYP.” and “MAX.” columns do not include the operating currents of the peripheral modules. However, in HALT mode, including the current flowing into the RTC.

2. When high-speed on-chip oscillator and subsystem clock are stopped.

3. When high-speed system clock and subsystem clock are stopped.

4. When high-speed on-chip oscillator and high-speed system clock are stopped. When AMPHS1 = 1 (Ultra-low power consumption oscillation).

5. Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.

HS (high-speed main) mode: 2.7 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 32 MHz

LS (low-speed main) mode: 2.7 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 8 MHz

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: High-speed on-chip oscillator clock frequency

3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

4. Except subsystem clock operation, temperature condition of the TYP. value is T_A = 25°C

(T_A = –40 to +105°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (2/2)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD2} Note 2	HALT mode	HS (high-speed main) mode Note 6	f_{IH} = 32 MHz Note 4	V_{DD} = 5.0 V		0.72	2.9	mA
					V_{DD} = 3.0 V		0.72	2.9	mA
				f_{IH} = 24 MHz Note 4	V_{DD} = 5.0 V		0.57	2.3	mA
					V_{DD} = 3.0 V		0.57	2.3	mA
				f_{IH} = 16 MHz Note 4	V_{DD} = 5.0 V		0.50	1.7	mA
					V_{DD} = 3.0 V		0.50	1.7	mA
			LS (low-speed main) mode Note 6	f_{IH} = 8 MHz Note 4, T_A = –40 to +85°C	V_{DD} = 3.0 V		320	910	μA
			HS (high-speed main) mode Note 6	f_{MX} = 20 MHz Note 3, V_{DD} = 5.0 V	Square wave input		0.40	1.9	mA
					Resonator connection		0.50	2.0	mA
				f_{MX} = 20 MHz Note 3, V_{DD} = 3.0 V	Square wave input		0.40	1.9	mA
					Resonator connection		0.50	2.0	mA
				f_{MX} = 10 MHz Note 3, V_{DD} = 5.0 V	Square wave input		0.24	1.02	mA
					Resonator connection		0.30	1.08	mA
				f_{MX} = 10 MHz Note 3, V_{DD} = 3.0 V	Square wave input		0.24	1.02	mA
					Resonator connection		0.30	1.08	mA
			LS (low-speed main) mode Note 6	f_{MX} = 8 MHz Note 3, V_{DD} = 3.0 V, T_A = –40 to +85°C	Square wave input		130	720	μA
					Resonator connection		170	760	μA
			HS (high-speed main) mode Note 6	f_{IH} = 4 MHz Note 4 f_{PLL} = 64 MHz, f_{CLK} = 32 MHz	V_{DD} = 5.0 V		1.15	4.0	mA
					V_{DD} = 3.0 V		1.15	4.0	mA
				f_{IH} = 4 MHz Note 4 f_{PLL} = 64 MHz, f_{CLK} = 16 MHz	V_{DD} = 5.0 V		0.95	3.2	mA
					V_{DD} = 3.0 V		0.95	3.2	mA
			Subsystem clock operation	f_{SUB} = 32.768 kHz Note 5 T_A = –40°C	Square wave input		0.28	0.70	μA
					Resonator connection		0.47	0.89	μA
				f_{SUB} = 32.768 kHz Note 5 T_A = +25°C	Square wave input		0.33	0.70	μA
					Resonator connection		0.52	0.89	μA
				f_{SUB} = 32.768 kHz Note 5 T_A = +50°C	Square wave input		0.41	1.90	μA
					Resonator connection		0.60	2.09	μA
				f_{SUB} = 32.768 kHz Note 5 T_A = +70°C	Square wave input		0.54	2.80	μA
					Resonator connection		0.73	2.99	μA
				f_{SUB} = 32.768 kHz Note 5 T_A = +85°C	Square wave input		1.27	6.10	μA
					Resonator connection		1.46	6.29	μA
				f_{SUB} = 32.768 kHz Note 5 T_A = +105°C	Square wave input		3.04	15.5	μA
					Resonator connection		3.23	15.7	μA
	I_{DD3}	STOP mode Note 7	T_A = –40°C				0.18	0.50	μA
			T_A = +25°C				0.23	0.50	μA
			T_A = +50°C				0.27	1.70	μA
			T_A = +70°C				0.44	2.60	μA
			T_A = +85°C				1.17	5.90	μA
			T_A = +105°C				2.94	15.3	μA

(**Notes** and **Remarks** are listed on the next page.)

Notes 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The following points apply in the HS (high-speed main) and LS (low-speed main) modes.
- The currents in the "TYP." column do not include the operating currents of the peripheral modules.
- The currents in the "MAX." column include the operating currents of the peripheral modules, except for those flowing into the A/D converter, comparator, programmable gain amplifier, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors, and those flowing while the data flash memory is being rewritten.

In the subsystem clock operation, the currents in both the "TYP." and "MAX." columns do not include the operating currents of the peripheral modules. However, in HALT mode, including the current flowing into the RTC.

In the STOP mode, the currents in both the "TYP." and "MAX." columns do not include the operating currents of the peripheral modules.

2. During HALT instruction execution by flash memory.

3. When high-speed on-chip oscillator and subsystem clock are stopped.

4. When high-speed system clock and subsystem clock are stopped.

5. When high-speed on-chip oscillator and high-speed system clock are stopped. When RTCLPC = 1 and setting ultra-low current consumption (AMPHS1 = 1).

6. Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.

HS (high-speed main) mode: 2.7 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 32 MHz

LS (low-speed main) mode: 2.7 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 8 MHz

7. Regarding the value for current operate the subsystem clock in STOP mode, refer to that in HALT mode.

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: High-speed on-chip oscillator clock frequency

3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

4. Except subsystem clock operation and STOP mode, temperature condition of the TYP. value is T_A = 25°C

(T_A = –40 to +105°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Low-speed on-chip oscillator operating current	I_{FIL} Note 1				0.20		μA
RTC operating current	I_{RTC} Notes 1, 2, 3				0.02		μA
12-bit interval timer operating current	I_{IT} Notes 1, 2, 4				0.02		μA
Watchdog timer operating current	I_{WDT} Notes 1, 2, 5	f_{IL} = 15 kHz			0.22		μA
A/D converter operating current	I_{ADC} Notes 1, 6	When conversion at maximum speed	Normal mode, AV_{REFP} = V_{DD} = 5.0 V		1.3	1.7	mA
			Low voltage mode, AV_{REFP} = V_{DD} = 3.0 V		0.5	0.7	mA
A/D converter reference voltage current	I_{ADREF} Note 1				75.0		μA
Temperature sensor operating current	I_{TMPS} Note 1				75.0		μA
LVD operating current	I_{LVD} Notes 1, 7				0.08		μA
Self-programming operating current	I_{FSP} Notes 1, 8				2.50	12.2	mA
Programmable gain amplifier operating current	I_{PGA} Note 9		AV_{REFP} = V_{DD} = 5.0 V		0.21	0.31	mA
			AV_{REFP} = V_{DD} = 3.0 V		0.18	0.29	mA
Comparator operating current	I_{CMP} Note 10	When one comparator channel is operating	AV_{REFP} = V_{DD} = 5.0 V		41.4	62	μA
			AV_{REFP} = V_{DD} = 3.0 V		37.2	59	μA
	I_{VREF}	When one internal reference voltage circuit is operating	AV_{REFP} = V_{DD} = 5.0 V		14.8	26	μA
			AV_{REFP} = V_{DD} = 3.0 V		8.9	20	μA
Programmable gain amplifier/ comparator reference current source	I_{IREF} Note 11		AV_{REFP} = V_{DD} = 5.0 V		3.2	5.1	μA
			AV_{REFP} = V_{DD} = 3.0 V		2.9	4.9	μA
BGO operating current	I_{BGO} Note 12				2.50	12.2	mA
SNOOZE operating current	I_{SNOZ} Note 1	ADC operation	The mode is performed Note 13		0.50	1.1	mA
			The A/D conversion operations are performed, Standard mode, AV_{REFP} = V_{DD} = 5.0 V		2.0	3.04	mA
		Simplified SPI (CSI)/UART operation			0.70	1.54	mA

(**Notes** and **Remarks** are listed on the next page.)

Notes 1. Current flowing to the V_{DD}.

2. When the high-speed on-chip oscillator and high-speed system clock are stopped.

3. Current flowing only to the real-time clock (RTC) (excluding the operating current of the low-speed on-chip oscillator and the XT1 oscillator). The supply current of the RL78 microcontrollers is the sum of the values of either I_{DD1} or I_{DD2}, and I_{RTC}, when the real-time clock operates in operation mode or HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be added. I_{DD2} subsystem clock operation includes the operational current of the real-time clock.

4. Current flowing only to the 12-bit interval timer (excluding the operating current of the XT1 oscillator and f_{IL} operating current). The current of the RL78 microcontrollers is the sum of the values of either I_{DD1} or I_{DD2}, and I_{IT}, when the 12-bit interval timer operates in operation mode or HALT mode.

5. Current flowing only to the watchdog timer (including 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 in operation.

6. Current flowing only 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 operates in an operation mode or the HALT mode.

7. Current flowing only 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 in operation.

8. Current flowing during self-programming operation.

9. Current flowing only to the programmable gain amplifier. The supply current value of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3}, and I_{PGA}, when the programmable gain amplifier is operating in operating mode or in HALT mode.

10. Current flowing only to the comparator. The supply current value of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3}, and I_{CMP}, when the comparator is operating.

11. This is the current required to flow to V_{DD} pin of the current circuit that is used as the programmable gain amplifier and the comparator.

12. Current flowing only during data flash rewrite.

13. See **21.3.3 SNOOZE mode** for shift time to the SNOOZE mode.

Remarks 1. f_{IL}: Low-speed on-chip oscillator clock frequency

2. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

3. f_{CLK}: CPU/peripheral hardware clock frequency

4. Temperature condition of the TYP. value is T_A = 25°C

5. Example of calculating current value when using programmable gain amplifier and comparator.

Examples 1) TYP. operating current value when three comparator channels, one internal reference voltage generator, and PGA are operating (when AV_{REFP} = V_{DD} = 5.0 V)

$I_{CMP} \times 3 + I_{VREF} + I_{PGA} + I_{IREF}$
= 41.4 [μA] × 3 + 14.8 [μA] × 1 + 210 [μA] + 3.2 [μA]
= 352.2 [μA]

Examples 2) TYP. operating current value when using two comparator channels, without using internal reference voltage generator (when AV_{REFP} = V_{DD} = 5.0 V)

$I_{CMP} \times 2 + I_{IREF}$
= 41.4 [μA] × 2 + 3.2 [μA]
= 86.0 [μA]

32.4 AC Characteristics

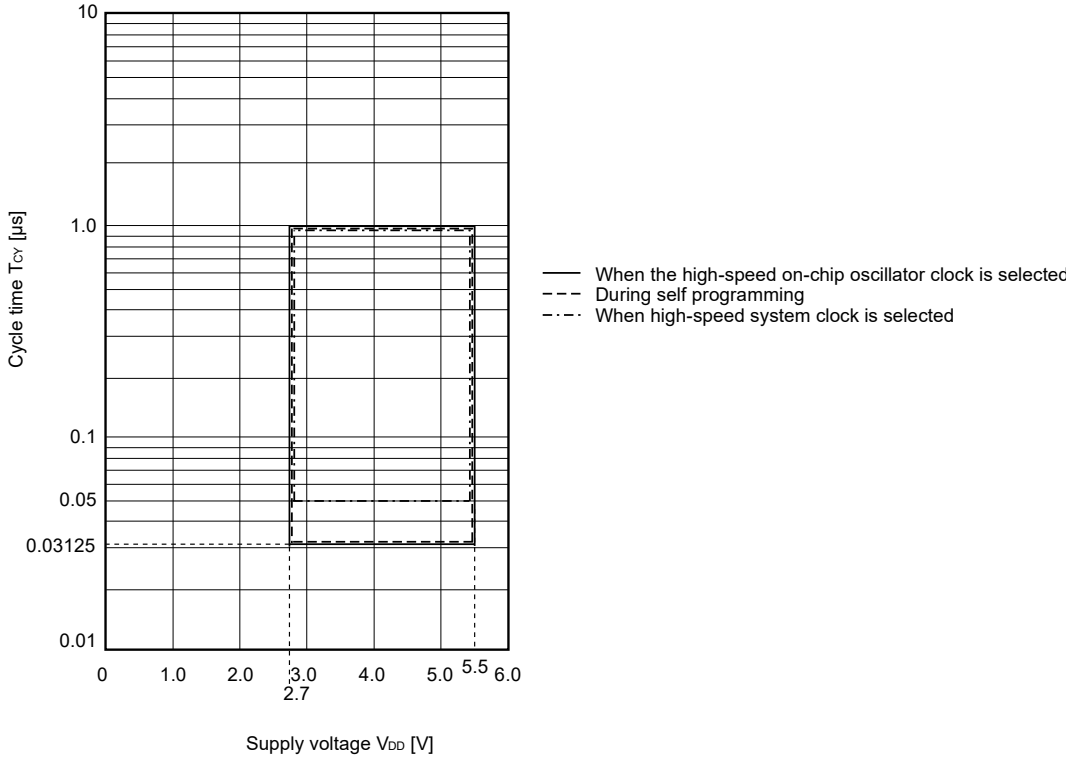
(T_A = –40 to +105°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	HS (high-speed main) mode		0.03125		1	μs
			LS (low-speed main) mode	T_A = –40 to +85°C	0.125		1	μs
		Subsystem clock (f_{SUB}) operation			28.5	30.5	31.3	μs
		In the self programming mode	HS (high-speed main) mode		0.03125		1	μs
			LS (low-speed main) mode	T_A = –40 to +85°C	0.125		1	μs
External system clock frequency	f_{EX}				1.0		20.0	MHz
	f_{EXS}				32		35	kHz
External system clock input high-level width, low-level width	t_{EXH}, t_{EXL}				24			ns
	t_{EXHS}, t_{EXLS}				13.7			μs
TI03, TI05, TI06, TI07 input high-level width, low-level width	t_{TIH}, t_{TIL}				2/f_{MCK}+10			ns
TO03, TO05, TO06, TKBO00, TKBO01, TKBO10, TKBO11, TKBO20, TKBO21, TKCO00 to TKCO05 output frequency (When duty = 50%)	f_{TO}	HS (high-speed main) mode		4.0 V ≤ V_{DD} ≤ 5.5 V			8	MHz
				2.7 V ≤ V_{DD} < 4.0 V			4	MHz
		LS (low-speed main) mode, T_A = –40 to +85°C		4.0 V ≤ V_{DD} ≤ 5.5 V			4	MHz
				2.7 V ≤ V_{DD} < 4.0 V			2	MHz
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}	INTP0, INTP3, INTP4, INTP9 to INTP11, INTP20 to INTP23			1			μs
$\overline{RESET}$ low-level width	t_{RSL}				10			μs

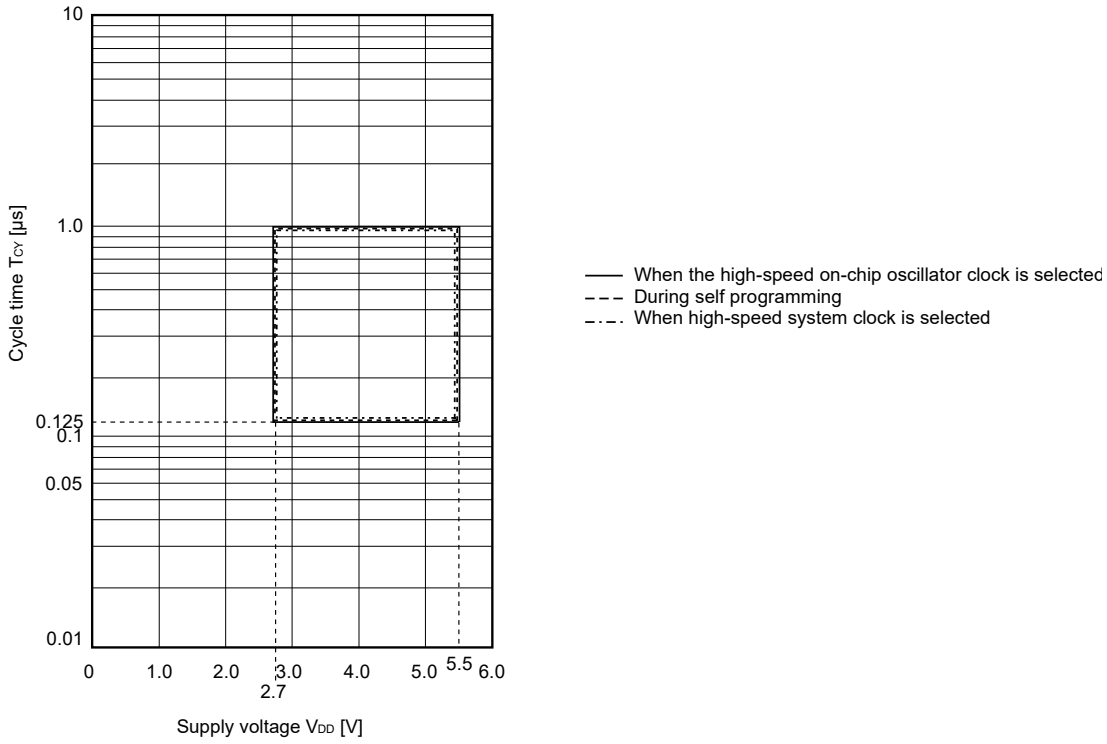
Remark f_{MCK}: Timer array unit operation clock frequency
(Operation clock to be set by the CKS0n bit of timer mode register 0n (TMR0n). n: Channel number (n = 0 to 7))

Minimum Instruction Execution Time during Main System Clock Operation

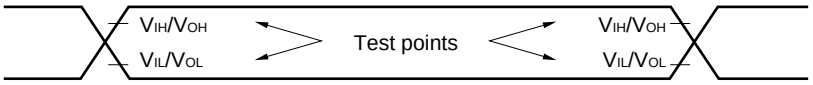
T_{CY} vs V_{DD} (HS (high-speed main) mode)



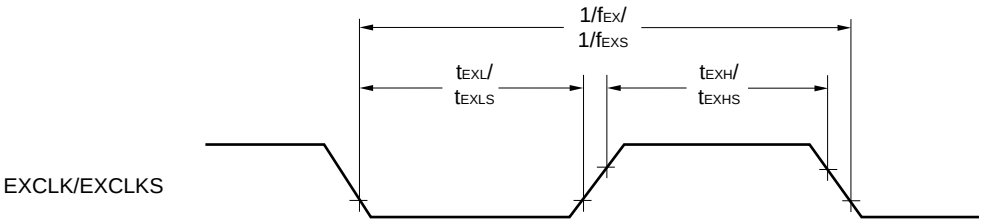
T_{CY} vs V_{DD} (LS (low-speed main) mode)



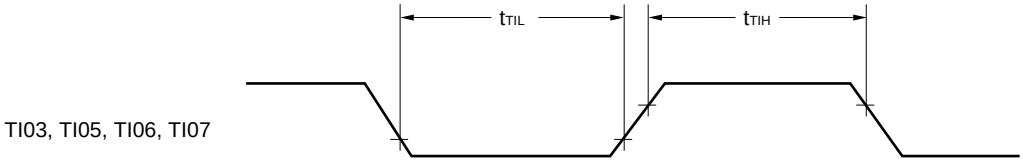
AC Timing Test Points

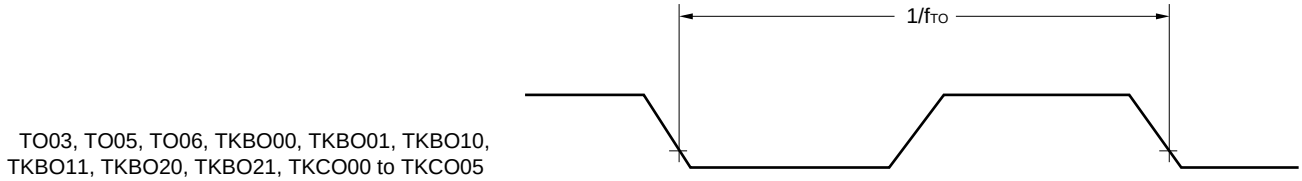


External System Clock Timing

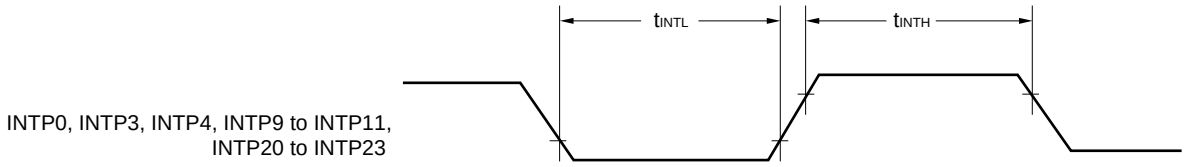


TI/TO Timing

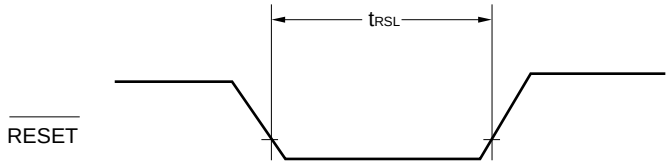




Interrupt Request Input Timing

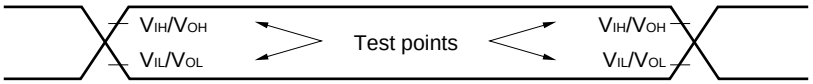


$\overline{\text{RESET}}$ Input Timing



32.5 Peripheral Functions Characteristics

AC Timing Test Points



32.5.1 Serial array unit 0, 4 (UART0, UART1, CSI00, DALI/UART4)

(1) During communication at same potential (UART mode)

(T_A = –40 to +105°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

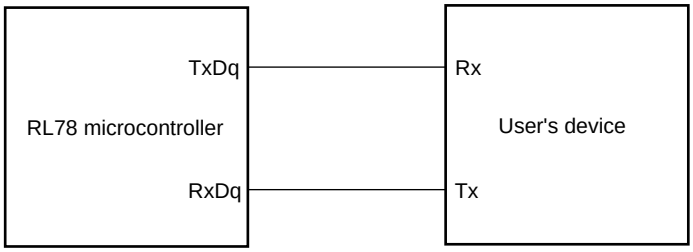
Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
Transfer rate[Note 1]		2.7 V≤ V_{DD} ≤ 5.5 V			f_{MCK}/6		f_{MCK}/6	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK}[Note 2]		5.3		1.3	Mbps

Notes 1. Transfer rate in the SNOOZE mode is 4800 bps only.

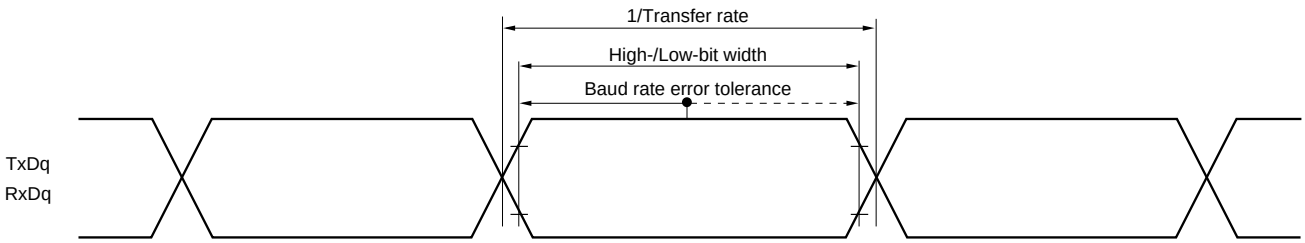
2. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
HS (high-speed main) mode: 32 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)
LS (low-speed main) mode: 8 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V), T_A = –40 to +85°C

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).

UART mode connection diagram (during communication at same potential)



UART mode bit width (during communication at same potential) (reference)



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).

Remarks 1. q: UART number (q = 0, 1), g: PIM and POM number (g = 0, 1)

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00 to 03))

(2) During communication at same potential (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = –40 to +105°C[Note 5], 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq 4/f_{CLK}$	125		500		ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V	$t_{KCY1}/2 - 12$		$t_{KCY1}/2 - 50$		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V	$t_{KCY1}/2 - 18$		$t_{KCY1}/2 - 50$		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V	44		110		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V	44		110		ns
SIp hold time (from SCKp↑) Note 2	t_{KSI1}		19		19		ns
Delay time from SCKp↓ to SOp output[Note 3]	t_{KSO1}	C = 30 pF[Note 4]		25		25	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the SCKp and SOp output lines.

5. Operating conditions of LS (low-speed main) mode is T_A = –40 to +85°C.

Caution Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remarks 1. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM number (g = 1)

2. f_{MCK}: Serial array unit operation clock frequency (Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00))

(3) During communication at same potential (Simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(T_A = –40 to +105°C[Note 6], 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCKp cycle time [Note 5]	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V	20 MHz < f_{MCK}	8/f_{MCK}		–		ns
			f_{MCK} ≤ 20 MHz	6/f_{MCK}		6/f_{MCK}		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V	16 MHz < f_{MCK}	8/f_{MCK}		–		ns
			f_{MCK} ≤ 16 MHz	6/f_{MCK}		6/f_{MCK}		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}			t_{KCY2}/2		t_{KCY2}/2		ns
SIp setup time (to SCKp↑)[Note 1]	t_{SIK2}			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		ns
Delay time from SCKp↓ to SOp output[Note 3]	t_{KSO2}	C = 30 pF[Note 4]			2/f_{MCK}+ 44		2/f_{MCK}+ 110	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the SOp output lines.

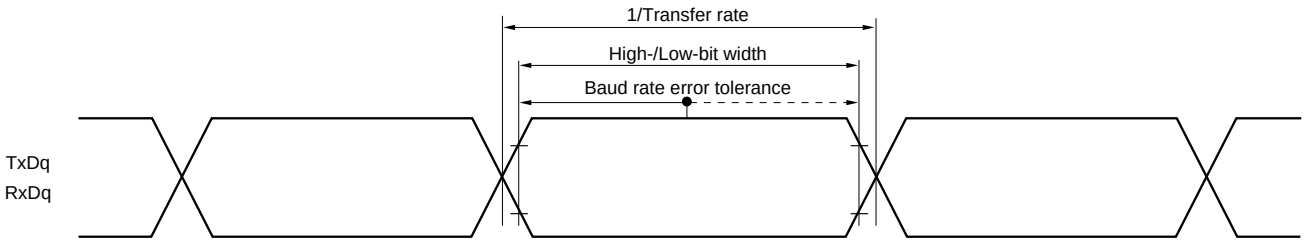
5. Transfer rate in the SNOOZE mode: MAX. 1 Mbps

6. Operating conditions of LS (low-speed main) mode is T_A = –40 to +85°C.

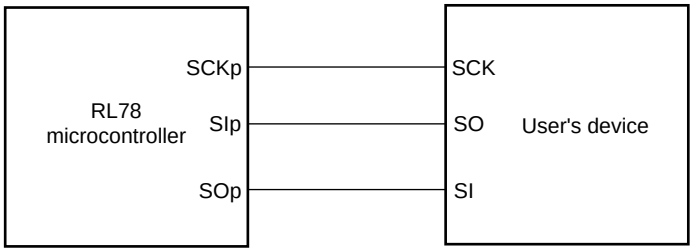
Caution Select the normal input buffer for the SIp pin and SCKp pin and the normal output mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remarks 1. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM number (g = 1)

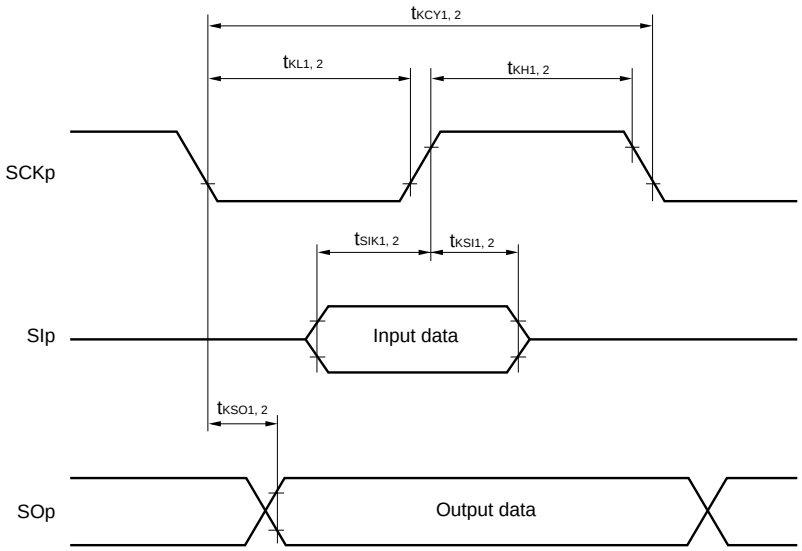
2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00))



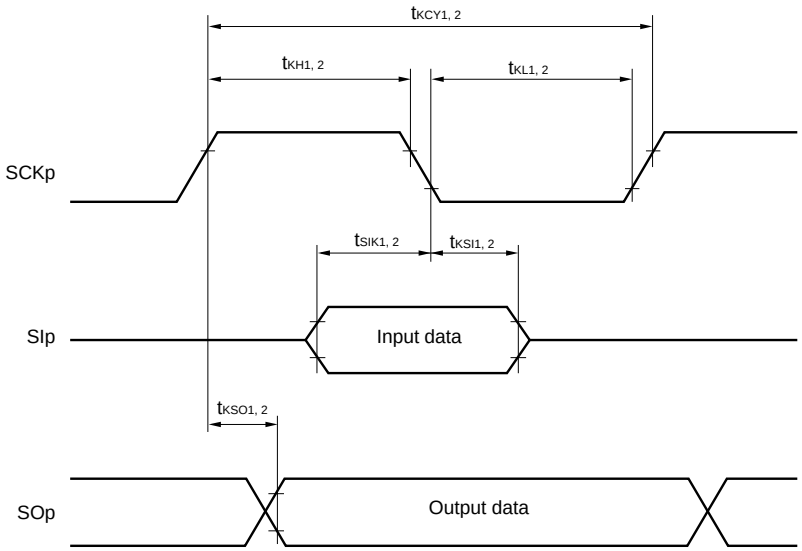
Simplified SPI (CSI) mode connection diagram (during communication at same potential)



Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remarks 1. p: CSI number (p = 00)

2. m: Unit number, n: Channel number (mn = 00)

(4) Communication at different potential (2.5 V, 3 V) (UART mode) (1/2)

(T_A = –40 to +105°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
Transfer rate		Reception	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		f_{MCK}/6[Note 1]		f_{MCK}/6[Note 1]	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK}[Note 2]		5.3		1.3	Mbps
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		f_{MCK}/6[Note 1]		f_{MCK}/6[Note 1]	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK}[Note 2]		5.3		1.3	Mbps

Notes 1. Transfer rate in the SNOOZE mode is 4800 bps only.

2. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:

HS (high-speed main) mode: 32 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)

LS (low-speed main) mode: 8 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V), T_A = –40 to +85°C.

Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Remarks 1. V_b[V]: Communication line voltage

2. q: UART number (q = 0, 1), g: PIM and POM number (g = 0, 1)

3. f_{MCK}: Serial array unit operation clock frequency

(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00 to 03)

(4) Communication at different potential (2.5 V, 3 V) (UART mode) (2/2)

(T_A = –40 to +105°C [Note 5], 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
Transfer rate		Transmission	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		**Note 1**		**Note 1**	bps
			Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 1.4 kΩ, V_b = 2.7 V		2.8 [Note 2]		2.8 [Note 2]	Mbps
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		**Note 3**		**Note 3**	bps
			Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 2.7 kΩ, V_b = 2.3 V		1.2 [Note 4]		1.2 [Note 4]	Mbps

Notes 1. The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 4.0 V ≤ V_{DD} ≤ 5.5 V and 2.7 V ≤ V_b ≤ 4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

2. This value as an example is calculated 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.

3. The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 2.7 V ≤ V_{DD} < 4.0 V and 2.3 V ≤ V_b ≤ 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

4. This value as an example is calculated 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.

5. Operating conditions of LS (low-speed main) mode is T_A = –40 to +85°C.

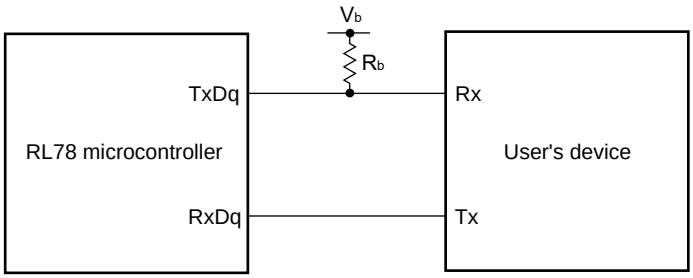
Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Remarks 1. R_b[Ω]: Communication line (TxDq) pull-up resistance,
C_b[F]: Communication line (TxDq) load capacitance, V_b[V]: Communication line voltage

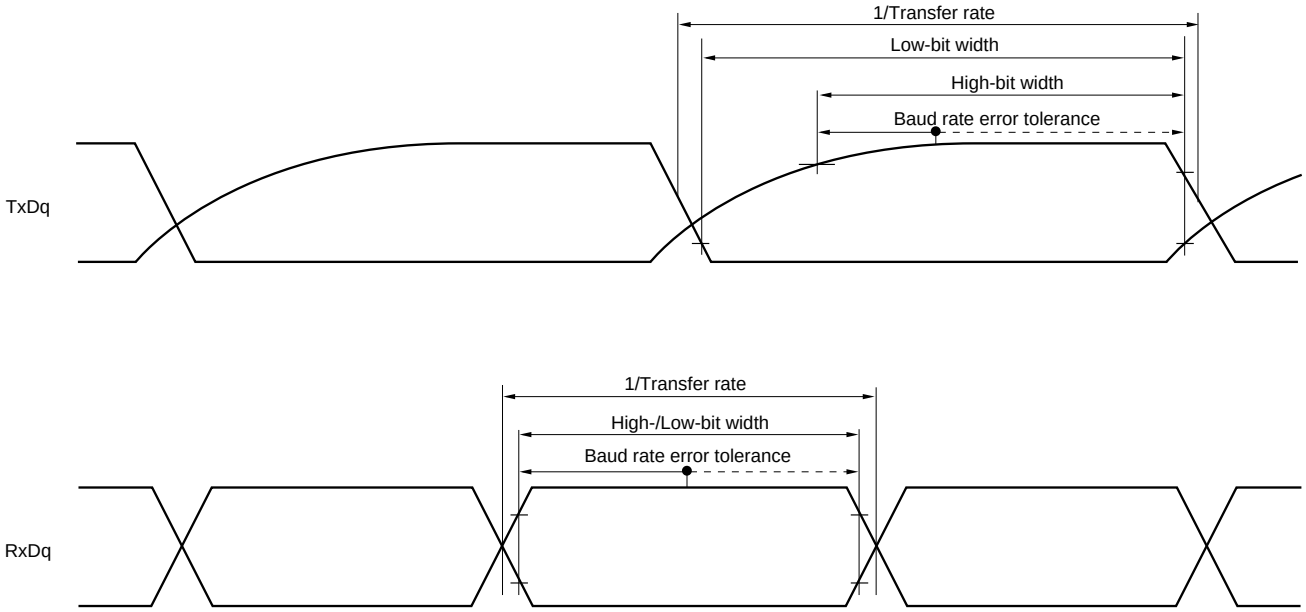
2. q: UART number (q = 0, 1), g: PIM and POM number (g = 0, 1)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00 to 03))

UART mode connection diagram (during communication at different potential)



UART mode bit width (during communication at different potential) (reference)



Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remarks 1. R_b[Ω]: Communication line (TxDq) pull-up resistance, V_b[V]: Communication line voltage

2. q: UART number (q = 0, 1), g: PIM and POM number (g = 0, 1)

(5) Communication at different potential (2.5 V, 3 V) (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = –40 to +105°C [Note 3], 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq 2/f_{CLK}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	200		1150		ns
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	300		1150		ns
SCKp high-level width	t_{KH1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 – 50		t_{KCY1}/2 – 75		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 – 120		t_{KCY1}/2 – 170		ns
SCKp low-level width	t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 – 7		t_{KCY1}/2 – 50		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 – 10		t_{KCY1}/2 – 50		ns
SIp setup time (to SCKp↑)[Note 1]	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		81		479		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		177		479		ns
SIp hold time (from SCKp↑)[Note 1]	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		10		19		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		10		19		ns
Delay time from SCKp↓ to SOp output[Note 1]	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ			60		100	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			130		195	ns
SIp setup time (to SCKp↓)[Note 2]	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		44		110		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		44		110		ns
SIp hold time (from SCKp↓)[Note 2]	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		10		19		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		10		19		ns
Delay time from SCKp↑ to SOp output[Note 2]	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ			10		25	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			10		25	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

2. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. Operating conditions of LS (low-speed main) mode is T_A = –40 to +85°C.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Remarks 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

2. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM number (g = 1)

(6) Communication at different potential (2.5 V, 3 V) (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = –40 to +105°C Note 3, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

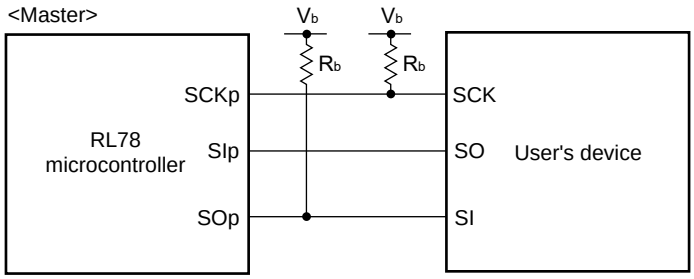
Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq 4/f_{CLK}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	300		1150		ns
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	500		1150		ns
SCKp high-level width	t_{KH1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		$t_{KCY1}/2 - 75$		$t_{KCY1}/2 - 75$		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		$t_{KCY1}/2 - 170$		$t_{KCY1}/2 - 170$		ns
SCKp low-level width	t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		$t_{KCY1}/2 - 12$		$t_{KCY1}/2 - 50$		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		$t_{KCY1}/2 - 18$		$t_{KCY1}/2 - 50$		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		81		479		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		177		479		ns
SIp hold time (from SCKp↑) Note 1	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		19		19		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		19		19		ns
Delay time from SCKp↓ to SOp output Note 1	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ			100		100	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			195		195	ns
SIp setup time (to SCKp↓) Note 2	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		44		110		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		44		110		ns
SIp hold time (from SCKp↓) Note 2	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, Cb = 30 pF, Rb = 1.4 kΩ		19		19		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, Cb = 30 pF, Rb = 2.7 kΩ		19		19		ns
Delay time from SCKp↑ to SOp output Note 2	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, Cb = 30 pF, Rb = 1.4 kΩ			25		25	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, Cb = 30 pF, Rb = 2.7 kΩ			25		25	ns

Notes
1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.
2. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.
3. Operating conditions of LS (low-speed main) mode is T_A = –40 to +85°C.

(**Caution** and **Remarks** are listed on the next page.)

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

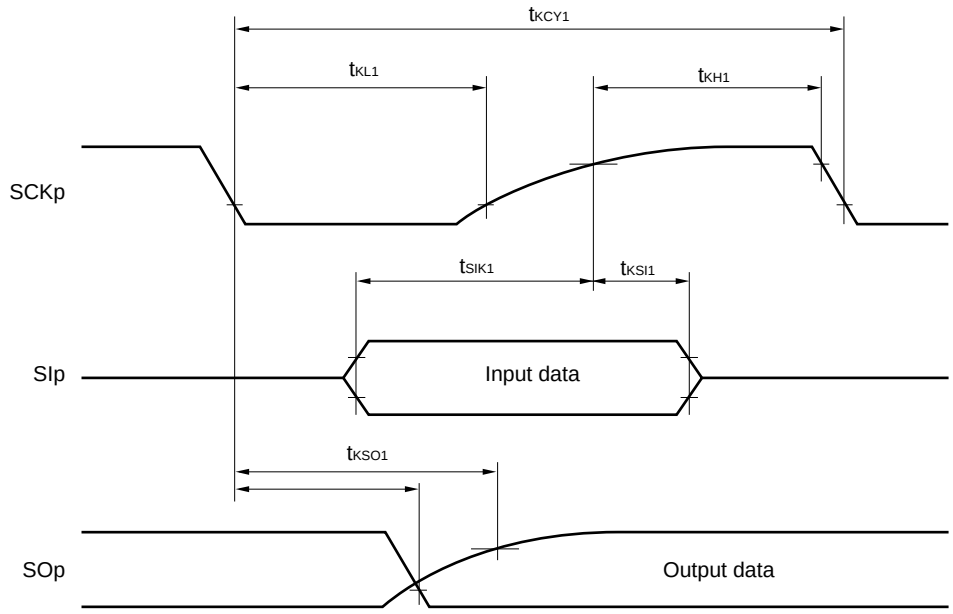
Simplified SPI (CSI) mode connection diagram (during communication at different potential)



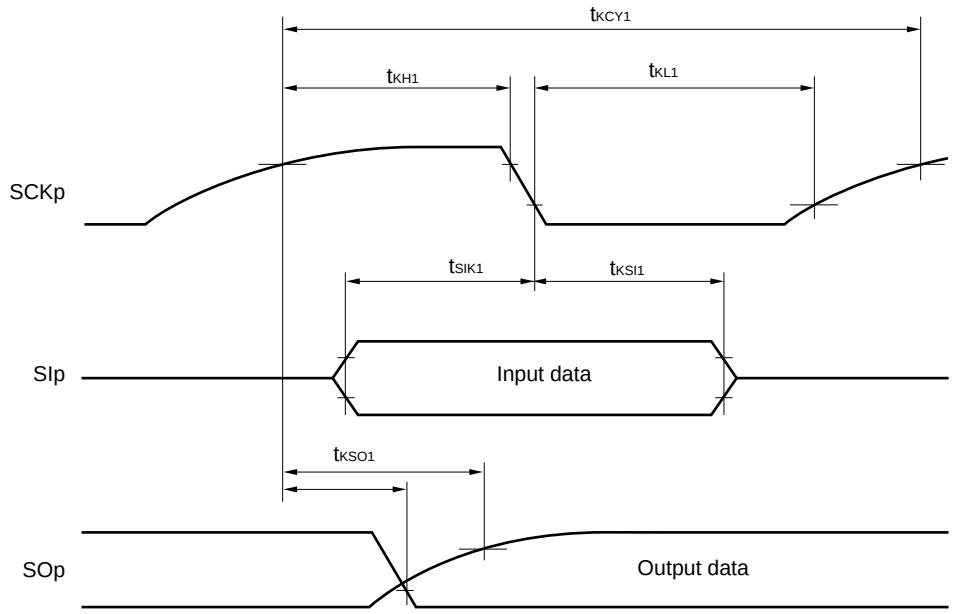
Remarks 1. R_b[Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

2. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM number (g = 1)

Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remark p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM number (g = 1)

(7) DALI/UART4 mode

(T_A = –40 to +105°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
Transfer rate					f_{MCK}/12		f_{MCK}/12	bps
			Maximum transfer rate theoretical value HS: f_{CLK} = 32 MHz, f_{MCK} = f_{CLK} LS: f_{CLK} = 8 MHz, f_{MCK} = f_{CLK}		2.6		0.6	Mbps

Remark f_{MCK}: Operation clock frequency of DALI/UART.
(Operation clock to be set by the serial clock select register mn (SPS4).)

Caution Operating conditions of LS (low-speed main) mode is T_A = –40 to +85°C.

32.5.2 Serial interface IICA

(1) I^2C standard mode

(T_A = –40 to +105°C Note 3, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Standard mode: f_{CLK} ≥ 1 MHz	0	100	0	100	kHz
Setup time of restart condition	$t_{SU:STA}$		4.7		4.7		μs
Hold time Note 1	$t_{HD:STA}$		4.0		4.0		μs
Hold time when SCLA0 = “L”	t_{LOW}		4.7		4.7		μs
Hold time when SCLA0 = “H”	t_{HIGH}		4.0		4.0		μs
Data setup time (reception)	$t_{SU:DAT}$		250		250		ns
Data hold time (transmission) Note 2	$t_{HD:DAT}$		0	3.45	0	3.45	μs
Setup time of stop condition	$t_{SU:STO}$		4.0		4.0		μs
Bus-free time	t_{BUF}		4.7		4.7		μs

Notes 1. The first clock pulse is generated after this period when the start/restart condition is detected.

2. The maximum value (MAX.) of $t_{HD:DAT}$ is during normal transfer and a clock stretch state is inserted in the ACK (acknowledge) timing.

3. Operating conditions of LS (low-speed main) mode is T_A = –40 to +85°C.

Remark The maximum value of Cb (communication line capacitance) and the value of Rb (communication line pull-up resistor) at that time in each mode are as follows.

Standard mode: C_b = 400 pF, R_b = 2.7 kΩ

(2) I^2C fast mode

(T_A = –40 to +105°C Note 3, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	fast mode: f_{CLK} ≥ 3.5 MHz	0	400	0	400	kHz
Setup time of restart condition	$t_{SU:STA}$		0.6		0.6		μs
Hold time Note 1	$t_{HD:STA}$		0.6		0.6		μs
Hold time when SCLA0 = “L”	t_{LOW}		1.3		1.3		μs
Hold time when SCLA0 = “H”	t_{HIGH}		0.6		0.6		μs
Data setup time (reception)	$t_{SU:DAT}$		100		100		ns
Data hold time (transmission) Note 2	$t_{HD:DAT}$		0	0.9	0	0.9	μs
Setup time of stop condition	$t_{SU:STO}$		0.6		0.6		μs
Bus-free time	t_{BUF}		1.3		1.3		μs

Notes 1. The first clock pulse is generated after this period when the start/restart condition is detected.

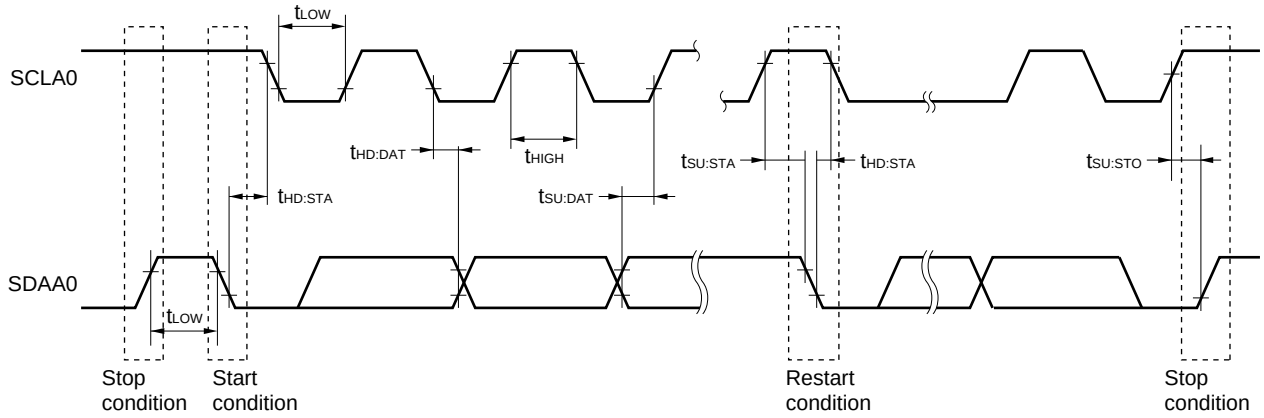
2. The maximum value (MAX.) of $t_{HD:DAT}$ is during normal transfer and a clock stretch state is inserted in the ACK (acknowledge) timing.

3. Operating conditions of LS (low-speed main) mode is T_A = –40 to +85°C.

Remark The maximum value of Cb (communication line capacitance) and the value of Rb (communication line pull-up resistor) at that time in each mode are as follows.

Fast mode: C_b = 320 pF, R_b = 1.1 kΩ

IICA serial transfer timing



32.6 Analog Characteristics

32.6.1 A/D converter characteristics

Classification of A/D converter characteristics

Input channel	Reference Voltage		
	Reference voltage (+) = AV_{REFP} Reference voltage (–) = AV_{REFM}	Reference voltage (+) = V_{DD} Reference voltage (–) = V_{SS}	Reference voltage (+) = V_{BGR} Reference voltage (–) = AV_{REFM}
ANI0 to ANI2, ANI4 to ANI7	See **32.6.1 (1)**.	See **32.6.1 (3)**.	See **32.6.1 (4)**.
ANI16 to ANI19	See **32.6.1 (2)**.		
Internal reference voltage Temperature sensor output voltage	See **32.6.1 (1)**.		–

(1) When reference voltage (+)= AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (–) = AV_{REFM}/ANI1 (ADREFM = 1), target pin: ANI2, ANI4 to ANI7, internal reference voltage, and temperature sensor output voltage

(T_A = –40 to +105°C, 2.7 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = AV_{REFP}, Reference voltage (–) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error[Note 1]	AINL	10-bit resolution AV_{REFP} = V_{DD}[Note 3]			1.2	±3.5	LSB
Conversion time	t_{CONV}	10-bit resolution Target pin: ANI2, ANI4 to ANI7	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	μs
		10-bit resolution Target pin: Internal reference voltage, and temperature sensor output voltage (HS (high-speed main) mode)	3.6 V ≤ V_{DD} ≤ 5.5 V	2.375		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.5625		39	μs
Zero-scale error[Notes 1, 2]	E_{ZS}	10-bit resolution AV_{REFP} = V_{DD}[Note 3]				±0.25	%FSR
Full-scale error[Notes 1, 2]	E_{FS}	10-bit resolution AV_{REFP} = V_{DD}[Note 3]				±0.25	%FSR
Integral linearity error[Note 1]	ILE	10-bit resolution AV_{REFP} = V_{DD}[Note 3]				±2.5	LSB
Differential linearity error[Note 1]	DLE	10-bit resolution AV_{REFP} = V_{DD}[Note 3]				±1.5	LSB
Analog input voltage	V_{AIN}	ANI2, ANI4 to ANI7		0		AV_{REFP}	V
		Internal reference voltage (HS (high-speed main) mode)		V_{BGR}[Note 4]			V
		Temperature sensor output voltage (HS (high-speed main) mode)		V_{TMPS25}[Note 4]			V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. When AV_{REFP} < V_{DD}, the MAX. values are as follows.
Overall error: Add ±1.0 LSB to the MAX. value when AV_{REFP} = V_{DD}.
Zero-scale error/Full-scale error: Add ±0.05%FSR to the MAX. value when AV_{REFP} = V_{DD}.
Integral linearity error/Differential linearity error: Add ±0.5 LSB to the MAX. value when AV_{REFP} = V_{DD}.

4. See **32.6.2 Temperature sensor/internal reference voltage characteristics**.

(2) When reference voltage (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (–) = AV_{REFM}/ANI1 (ADREFM = 1), target pin: ANI16 to ANI19

(T_A = –40 to +105°C, 2.7 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = AV_{REFP}, Reference voltage (–) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error[Note 1]	AINL	10-bit resolution AV_{REFP} = V_{DD}[Notes 3]			1.2	±5.0	LSB
Conversion time	t_{CONV}	10-bit resolution Target ANI pin : ANI16 to ANI19	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	μs
Zero-scale error[Notes 1, 2]	E_{ZS}	10-bit resolution AV_{REFP} = V_{DD}[Notes 3]				±0.35	%FSR
Full-scale error[Notes 1, 2]	E_{FS}	10-bit resolution AV_{REFP} = V_{DD}[Notes 3]				±0.35	%FSR
Integral linearity error[Note 1]	ILE	10-bit resolution AV_{REFP} = V_{DD}[Notes 3]				±3.5	LSB
Differential linearity error[Note 1]	DLE	10-bit resolution AV_{REFP} = V_{DD}[Notes 3]				±2.0	LSB
Analog input voltage	V_{AIN}	ANI16 to ANI19		0		AV_{REFP} and V_{DD}	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. When AV_{REFP} < V_{DD}, the MAX. values are as follows.
Overall error: Add ±1.0 LSB to the MAX. value when AV_{REFP} = V_{DD}.
Zero-scale error/Full-scale error: Add ±0.05%FSR to the MAX. value when AV_{REFP} = V_{DD}.
Integral linearity error/Differential linearity error: Add ±0.5 LSB to the MAX. value when AV_{REFP} = V_{DD}.

(3) When reference voltage (+) = V_{DD} (ADREFP1 = 0, ADREFP0 = 0), reference voltage (–) = V_{SS} (ADREFM = 0), target pin: ANI0 to ANI2, ANI4 to ANI7, ANI16 to ANI19, internal reference voltage, and temperature sensor output voltage

(T_A = –40 to +105°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = V_{DD}, Reference voltage (–) = V_{SS})

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error[Note 1]	AINL	10-bit resolution			1.2	±7.0	LSB
Conversion time	t_{CONV}	10-bit resolution Target pin: ANI0 to ANI2, ANI4 to ANI7, ANI16 to ANI19	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	μs
Conversion time	t_{CONV}	10-bit resolution Target pin: Internal reference voltage, and temperature sensor output voltage (HS (high-speed main) mode)	3.6 V ≤ V_{DD} ≤ 5.5 V	2.375		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.5625		39	μs
Zero-scale error[Notes 1, 2]	E_{ZS}	10-bit resolution				±0.60	%FSR
Full-scale error[Notes 1, 2]	E_{FS}	10-bit resolution				±0.60	%FSR
Integral linearity error[Note 1]	ILE	10-bit resolution				±4.0	LSB
Differential linearity error[Note 1]	DLE	10-bit resolution				±2.0	LSB
Analog input voltage	V_{AIN}	ANI0 to ANI2, ANI4 to ANI7		0		V_{DD}	V
		ANI16 to ANI19		0		V_{DD}	V
		Internal reference voltage (HS (high-speed main) mode)		V_{BGR}[Note 3]			V
		Temperature sensor output voltage (HS (high-speed main) mode)		V_{TMPS25}[Note 3]			V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. See **32.6.2 Temperature sensor/internal reference voltage characteristics**.

(4) When reference voltage (+) = Internal reference voltage (ADREFP1 = 1, ADREFP0 = 0), reference voltage (–) = AV_{REFM}/ANI1 (ADREFM = 1), target pin: ANI0, ANI2, ANI4 to ANI7, ANI16 to ANI19

(T_A = –40 to +105°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = V_{BGR}[Note 3], Reference voltage (–) = AV_{REFM} = 0 V[Note 4], HS (high-speed main) mode)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES		8			bit
Conversion time	t_{CONV}	8-bit resolution	17		39	μs
Zero-scale error[Notes 1, 2]	E_{ZS}	8-bit resolution			±0.60	%FSR
Integral linearity error[Note 1]	ILE	8-bit resolution			±2.0	LSB
Differential linearity error[Note 1]	DLE	8-bit resolution			±1.0	LSB
Analog input voltage	V_{AIN}		0		V_{BGR}[Note 3]	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. See **32.6.2 Temperature sensor/internal reference voltage characteristics**.

4. When reference voltage (–) = V_{SS}, the MAX. values are as follows.
Zero-scale error: Add ±0.35%FSR to the MAX. value when reference voltage (–) = AV_{REFM}.
Integral linearity error: Add ±0.5 LSB to the MAX. value when reference voltage (–) = AV_{REFM}.
Differential linearity error: Add ±0.2 LSB to the MAX. value when reference voltage (–) = AV_{REFM}.

32.6.2 Temperature sensor/internal reference voltage characteristics

(T_A = –40 to +105°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, HS (high-speed main) mode)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Temperature sensor output voltage	V_{TMPS25}	Setting ADS register = 80H, T_A = +25°C		1.05		V
Internal reference voltage	V_{BGRT}	Setting ADS register = 81H	1.38	1.45	1.5	V
Temperature coefficient	F_{VTMPS}	Temperature sensor that depends on the temperature		–3.6		mV/C
Operation stabilization wait time	t_{AMP}		5			μs

32.6.3 Programmable gain amplifier

(T_A = –40 to +105°C, 2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V, V_{SS} = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input offset voltage	V_{IOPGA}					±5	±10	mV
Input voltage range	V_{IPGA}				0		0.9V_{DD}/gain	V
Gain error[Note 1]		4, 8 times					±1	%
		16 times					±1.5	%
		32 times					±2	%
Slew rate[Note 1]	SR_{RPGA}	Rising edge	4.0 V ≤ V_{DD} ≤ 5.5 V	4, 8 times	4			V/μs
				16, 32 times	1.4			V/μs
			2.7 V ≤ V_{DD} < 4.0 V	4, 8 times	1.8			V/μs
				16, 32 times	0.5			V/μs
	SR_{FPGA}	Falling edge	4.0 V ≤ V_{DD} ≤ 5.5 V	4, 8 times	3.2			V/μs
				16, 32 times	1.4			V/μs
			2.7 V ≤ V_{DD} < 4.0 V	4, 8 times	1.2			V/μs
				16, 32 times	0.5			V/μs
Operation stabilization wait time[Note 2]	t_{PGA}	4, 8 times			5			μs
		16, 32 times			10			μs

Notes 1. When V_{IPGA} = 0.1V_{DD}/gain to 0.9V_{DD}/gain.

2. Time required until a state is entered where the DC and AC specifications of the PGA are satisfied after the PGA operation has been enabled (PGAEN = 1).

Remark These characteristics apply when AV_{REFM} is selected as GND of the PGA by using the CVRVS1 bit.

32.6.4 Comparator

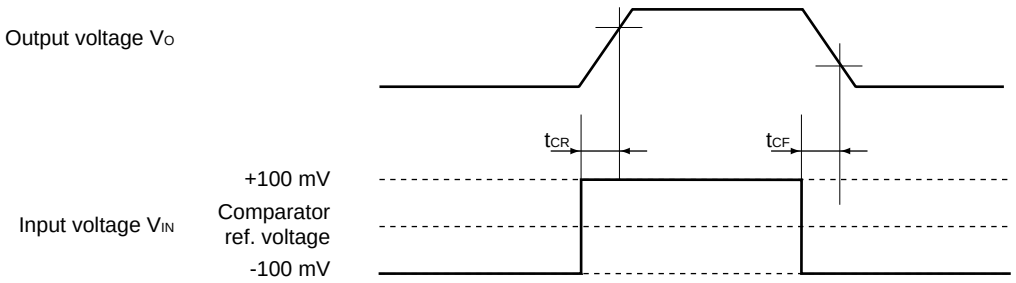
(T_A = –40 to +105°C, 2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V, V_{SS} = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input offset voltage	V_{IOCMP}			±5	±40	mV
Input voltage range	V_{ICMP}	CMP0P to CMP5P	0		V_{DD}	V
		CMPCOM	0.045		0.9V_{DD}	V
Internal reference voltage deviation	ΔV_{IREF}	CmRVM register values: 7FH to 80H (m = 0 to 2)			±2	LSB
		Other than above			±1	LSB
Response time	t_{CR}, t_{CF}	Input amplitude = ±100 mV		70	150	ns
Operation stabilization wait time[Note 1]	t_{CMP}	3.3 V ≤ V_{DD} ≤ 5.5 V	1			μs
		2.7 V ≤ V_{DD} < 3.3 V	3			μs
Reference voltage stabilization wait time	t_{VR}	CVRE: 0 to 1[Note 2]	10			μs

Notes 1. Time required until a state is entered where the DC and AC specifications of the comparator are satisfied after the operation of the comparator has been enabled (CMPnEN bit = 1: n = 0 to 5)

2. Enable comparator output (CnOE bit = 1; n = 0 to 5) after enabling operation of the internal reference voltage generator (by setting the CVREm bit to 1; m = 0 to 2) and waiting for the operation stabilization time to elapse.

Remark These characteristics apply when AV_{REFP} is selected as the power supply source of the internal reference voltage by using the CVRVS0 bit, and when AV_{REFM} is selected as GND of the internal reference voltage by using the CVRVS1 bit.

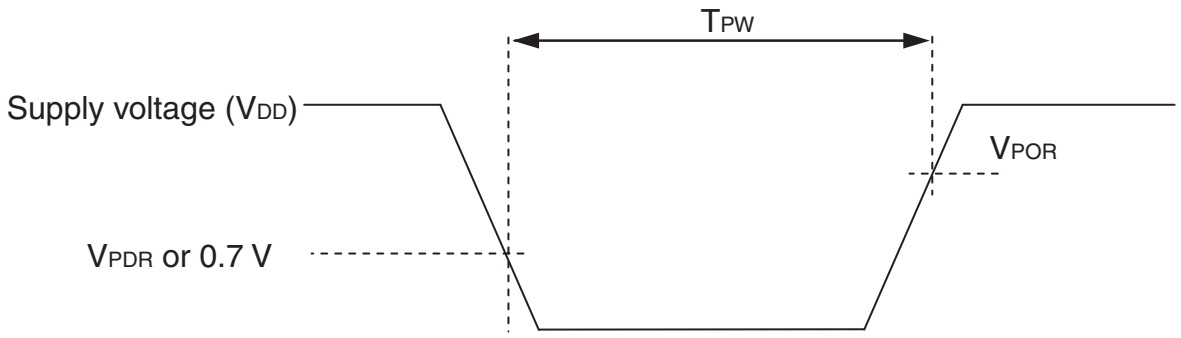


32.6.5 POR circuit characteristics

(T_A = –40 to +105°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}	Power supply rise time	1.45	1.51	1.57	V
	V_{PDR}	Power supply fall time	1.44	1.50	1.56	V
Minimum pulse width[Note]	T_{PW}		300			μs

Note Minimum time required for a POR reset when V_{DD} exceeds below V_{PDR}. This is also the minimum time required for a POR reset from when V_{DD} exceeds below 0.7 V to when V_{DD} exceeds V_{POR} while STOP mode is entered or the main system clock is stopped through setting bit 0 (HIOSTOP) and bit 7 (MSTOP) in the clock operation status control register (CSC).



32.6.6 LVD circuit characteristics

LVD Detection Voltage of Reset Mode and Interrupt Mode

(T_A = –40 to +105°C, $V_{PDR} \leq V_{DD} \leq$ 5.5 V, V_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVD0}	Power supply rise time	3.97	4.06	4.14	V
			Power supply fall time	3.89	3.98	4.06	V
		V_{LVD1}	Power supply rise time	3.67	3.75	3.82	V
			Power supply fall time	3.59	3.67	3.74	V
		V_{LVD2}	Power supply rise time	3.06	3.13	3.19	V
			Power supply fall time	2.99	3.06	3.12	V
		V_{LVD3}	Power supply rise time	2.95	3.02	3.08	V
			Power supply fall time	2.89	2.96	3.02	V
		V_{LVD4}	Power supply rise time	2.85	2.92	2.97	V
			Power supply fall time	2.79	2.86	2.91	V
		V_{LVD5}	Power supply rise time	2.75	2.81	2.87	V
			Power supply fall time	2.70	2.75	2.81	V
Minimum pulse width		t_{LW}		300			μs
Detection delay time						300	μs

LVD Detection Voltage of Interrupt & Reset Mode

(T_A = –40 to +105°C, $V_{PDR} \leq V_{DD} \leq$ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Interrupt and reset mode	V_{LVD0}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 1, 1, falling reset voltage: 2.7 V			2.70	2.75	2.81	V
	V_{LVD1}		LVIS1, LVIS0 = 1, 0	Rising release reset voltage	2.85	2.92	2.97	V
				Falling interrupt voltage	2.79	2.86	2.91	V
	V_{LVD2}		LVIS1, LVIS0 = 0, 1	Rising release reset voltage	2.95	3.02	3.08	V
				Falling interrupt voltage	2.89	2.96	3.02	V
	V_{LVD3}		LVIS1, LVIS0 = 0, 0	Rising release reset voltage	3.97	4.06	4.14	V
				Falling interrupt voltage	3.89	3.98	4.06	V

32.6.7 Supply voltage rise inclination characteristics

(T_A = –40 to +105°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Supply voltage rise	SV_{DD}				54	V/ms

Caution Keep the internal reset status by using the LVD circuit or an external reset signal until V_{DD} rises to within the operating voltage range shown in 32.4 AC Characteristics.

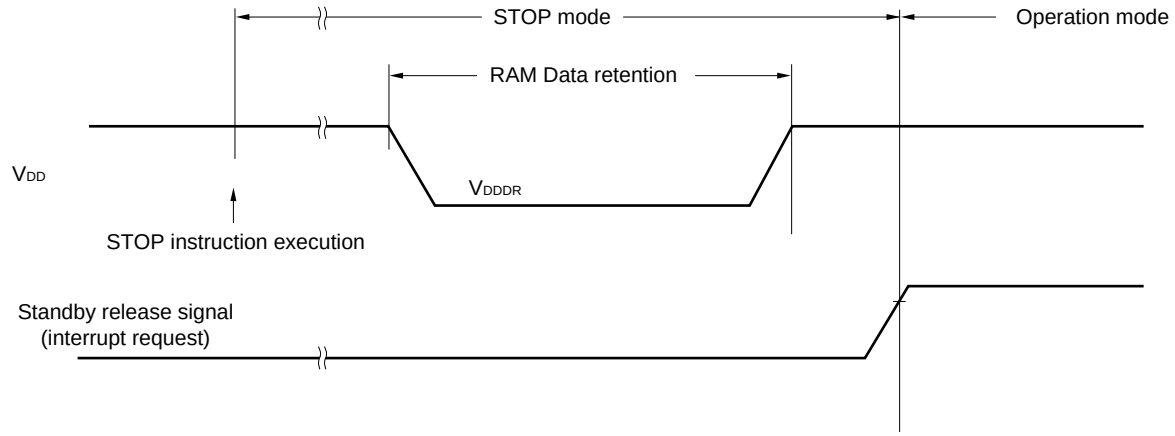
32.7 RAM Data Retention Characteristics

(T_A = –40 to +105°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage[Note 2]	V_{DDDR}		1.44[Note 1]		5.5	V

Note The value depends on the POR detection voltage. When the voltage drops, the RAM data is retained before a POR reset is effected, but RAM data is not retained when a POR reset is effected.

Caution When CPU is operated at the voltage of out of the operation voltage range, RAM data is not retained. Therefore, set STOP mode before the supplied voltage is below the operation voltage range.



32.8 Flash Memory Programming Characteristics

(T_A = –40 to +105°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
CPU/peripheral hardware clock frequency	f_{CLK}	2.7 V ≤ V_{DD} ≤ 5.5 V	1		32	MHz
Number of code flash rewrites Notes 1, 2, 3	C_{erwr}	Retained for 20 years, T_A = 85°C Note 3	1,000			Times
Number of data flash rewrites Notes 1, 2, 3		Retained for 1 year, T_A = 25°C Note 3		1,000,000		
		Retained for 5 years, T_A = 85°C Note 3	100,000			
		Retained for 20 years, T_A = 85°C Note 3	10,000			

Notes 1. 1 erase + 1 write after the erase is regarded as 1 rewrite. The retaining years are until next rewrite after the rewrite.

2. When using flash memory programmer and Renesas Electronics self programming library

3. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

32.9 Dedicated Flash Memory Programmer Communication (UART)

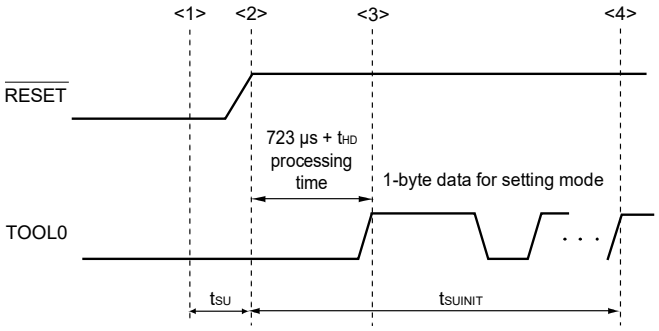
(T_A = –40 to +105°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		During serial programming	115.2 k		1 M	bps

32.10 Timing of Entry to Flash Memory Programming Modes

(T_A = –40 to +105°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
How long from when an external reset ends until the initial communication settings are specified	t_{SUINIT}	POR and LVD reset must end before the external reset ends.			100	ms
How long from when the TOOL0 pin is placed at the low level until an external reset ends	t_{SU}	POR and LVD reset must end before the external reset ends.	10			μs
How long the TOOL0 pin must be kept at the low level after a reset ends (except soft processing time)	t_{HD}	POR and LVD reset must end before the external reset ends.	1			ms



<1> The low level is input to the TOOL0 pin.
<2> The external reset ends (POR and LVD reset must end before the pin reset ends.).
<3> The TOOL0 pin is set to the high level.
<4> Complete the baud rate setting by UART reception.

Remark t_{SUINIT}: The segment shows that it is necessary to finish specifying the initial communication settings within 100 ms from when the resets end.
t_{SU}: How long from when the TOOL0 pin is placed at the low level until an external reset ends
t_{HD}: How long to keep the TOOL0 pin at the low level from when the external and internal resets end (except soft processing time)

CHAPTER 33 ELECTRICAL SPECIFICATIONS (M: Industrial applications, T_A = –40 to +125°C)

In this chapter, shows the electrical specifications of the target products.

Target products (M: Industrial applications): T_A = – 40 to +125°C
R5F107xxMxx

Cautions 1. The RL78/I1A has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

2. The pins mounted depend on the product. See 2.1 Port Function to 2.2.1 Functions for each product.

3. When any of these products are used at 105°C or lower, see CHAPTER 32 ELECTRICAL SPECIFICATIONS (T_A = –40 to +105°C).

33.1 Absolute Maximum Ratings

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}		–0.5 to +6.5	V
REGC pin input voltage	V_{IREGC}	REGC	–0.3 to +2.8 and –0.3 to V_{DD} +0.3 Note 1	V
Input voltage	V_{I1}	P02, P03, P05, P06, P10 to P12, P20 to P22, P24 to P27, P30, P31, P40, P75 to P77, P120 to P124, P137, P147, P200 to P206, EXCLK, EXCLKS, $\overline{\text{RESET}}$	–0.3 to V_{DD} +0.3 Note 2	V
Output voltage	V_{O1}	P02, P03, P05, P06, P10 to P12, P20 to P22, P24 to P27, P30, P31, P40, P75 to P77, P120, P147, P200 to P206	–0.3 to V_{DD} +0.3 Note 2	V
Analog input voltage	V_{AI1}	ANI0 to ANI2, ANI4 to ANI7, ANI16 to ANI19	–0.3 to V_{DD} +0.3 and –0.3 to $AV_{REF(+)}$ +0.3 Notes 2, 3	V

Notes 1. Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

2. Must be 6.5 V or lower.

3. Do not exceed $AV_{REF(+)}$ + 0.3 V in case of A/D conversion target pin.

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

2. $AV_{REF(+)}$: + side reference voltage of the A/D converter.

3. V_{SS} : Reference voltage

Absolute Maximum Ratings (T_A = 25°C) (2/2)

Parameter	Symbols	Conditions		Ratings	Unit
Output current, high	I_{OH1}	Per pin	P02, P03, P05, P06, P10 to P12, P30, P31, P40, P75 to P77, P120, P147, P200 to P206	–40	mA
		Total of all pins –170 mA	P02, P03, P40, P120	–70	mA
			P05, P06, P10 to P12, P30, P31, P75 to P77, P147, P200 to P206	–100	mA
	I_{OH2}	Per pin	P20 to P22, P24 to P27	–0.5	mA
		Total of all pins		–2	mA
Output current, low	I_{OL1}	Per pin	P02, P03, P05, P06, P10 to P12, P30, P31, P40, P75 to P77, P120, P147, P200 to P206	40	mA
		Total of all pins 170 mA	P02, P03, P40, P120	70	mA
			P05, P06, P10 to P12, P30, P31, P75 to P77, P147, P200 to P206	100	mA
	I_{OL2}	Per pin	P20 to P22, P24 to P27	1	mA
		Total of all pins		5	mA
Operating ambient temperature	T_A	In normal operation mode		–40 to +125	°C
		In flash memory programming mode		–40 to +105	
Storage temperature	T_{stg}			–65 to +150	°C

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

33.2 Oscillator Characteristics

33.2.1 X1, XT1 oscillator characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Resonator	Conditions	MIN.	TYP.	MAX.	Unit
X1 clock frequency (f_X)Note	Ceramic resonator/ crystal resonator		1.0		20.0	MHz
XT1 clock frequency (f_{XT})Note	Crystal resonator		32	32.768	35	kHz

Note Indicates only permissible oscillator frequency ranges. See **AC Characteristics** for instruction execution time. Request evaluation by the manufacturer of the oscillator circuit mounted on a board to check the oscillator characteristics.

Caution Since the CPU is started by the high-speed on-chip oscillator clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and the oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.

Remark When using the X1 oscillator and XT1 oscillator, see **5.4 System Clock Oscillator**.

33.2.2 On-chip oscillator characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Oscillators	Parameters	Conditions	MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency[Note 1]	f_{IH}		1		32	MHz
High-speed on-chip oscillator clock frequency accuracy[Note 2]		T_A = –20 to 85°C	–1		+1	%
		T_A = –40 to 105°C	–1.5		+1.5	%
		T_A = –40 to 125°C When 16 MHz selected	–2		+2	%
Low-speed on-chip oscillator clock frequency	f_{IL}			15		kHz
Low-speed on-chip oscillator clock frequency accuracy			–15		+15	%

Notes 1. Frequency can be selected in a high-speed on-chip oscillator. Selected by bits 0 to 3 of option byte (000C2H/010C2H).

2. This indicates the oscillator characteristics only. See **AC Characteristics** for instruction execution time.

Remark When using the device at an ambient temperature that exceeds T_A = 105°C, the selectable oscillation frequency is 16 MHz max.

33.2.3 PLL characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
PLL input clock frequency[Note]	f_{PLLIN}	High-speed system clock is selected (f_{MX} = 4 MHz)	3.92	4.00	4.08	MHz
		High-speed on-chip oscillator clock is selected (f_{IH} = 4 MHz)	3.92	4.00	4.08	MHz
PLL output clock frequency[Note]	f_{PLL}		f_{PLLIN} × 16			MHz

Note This only indicates the oscillator characteristics. See **AC Characteristics** for instruction execution time.

Remark When using the device at an ambient temperature that exceeds T_A = 105°C, only 16 MHz (f_{PLL} × 1/4) can be selected as the CPU operating frequency.

33.3 DC Characteristics

33.3.1 Pin characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high[Note 1]	I_{OH1}	Per pin for P02, P03, P05, P06, P10 to P12, P30, P31, P40, P75 to P77, P120, P147, P200 to P206	4.0 V ≤ V_{DD} ≤ 5.5 V			–3.0[Note 2]	mA
			2.7 V ≤ V_{DD} < 4.0 V			–1.0	mA
		Total of P02, P03, P40, P120 (When duty ≤ 70%[Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			–9.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–3.0	mA
		Total of P05, P06, P10 to P12, P30, P31, P75 to P77, P147, P200 to P206 (When duty ≤ 70%[Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			–21.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–6.0	mA
		Total of all pins (When duty ≤ 70%[Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			–21.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–9.0	mA
	I_{OH2}	Per pin for P20 to P22, P24 to P27	2.7 V ≤ V_{DD} ≤ 5.5 V			–0.1[Note 2]	mA
		Total of all pins (When duty ≤ 70%[Note 3])	2.7 V ≤ V_{DD} ≤ 5.5 V			–0.4	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from the V_{DD} pin to an output pin.

2. However, do not exceed the total current value.

3. Specification under conditions where the duty factor ≤ 70%.
The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).

- Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OH} = –10.0 mA

Total output current of pins = (–10.0 × 0.7)/(80 × 0.01) ≅ –8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Caution P02, P10 to P12 do not output high level in N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low[Note 1]	I_{OL1}	Per pin for P02, P03, P05, P06, P10 to P12, P30, P31, P40, P75 to P77, P120, P147, P200 to P206	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 P02, P03, P40, P120 (When duty ≤ 70%[Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			20.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			5.0	mA
		Total of P05, P06, P10 to P12, P30, P31, P75 to P77, P147, P200 to P206 (When duty ≤ 70%[Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			20.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			10.0	mA
		Total of all pins (When duty ≤ 70%[Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			40.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			15.0	mA
	I_{OL2}	Per pin for P20 to P22, P24 to P27	2.7 V ≤ V_{DD} ≤ 5.5 V			0.4[Note 2]	mA
		Total of all pins (When duty ≤ 70%[Note 3])	2.7 V ≤ V_{DD} ≤ 5.5 V			1.6	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to the V_{SS} pin.

2. However, do not exceed the total current value.

3. Specification under conditions where the duty factor ≤ 70%.
The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).

- Total output current of pins = (I_{OL} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OL} = –10.0 mA

Total output current of pins = (–10.0 × 0.7)/(80 × 0.01) ≅ –8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P02, P03, P05, P06, P10 to P12, P20 to P22, P24 to P27, P30, P31, P40, P75 to P77, P120 to P124, P137, P147, P200 to P206, EXCLK, EXCLKS, $\overline{\text{RESET}}$	Normal input buffer	0.8V_{DD}		V_{DD}	V
	V_{IH2}	P03, P10, P11	TTL input buffer 4.0 V ≤ V_{DD} ≤ 5.5 V	2.1		V_{DD}	V
			TTL input buffer 3.3 V ≤ V_{DD} < 4.0 V	2.0		V_{DD}	V
			TTL input buffer 2.7 V ≤ V_{DD} < 3.3 V	1.5		V_{DD}	V
Input voltage, low	V_{IL1}	P02, P03, P05, P06, P10 to P12, P20 to P22, P24 to P27, P30, P31, P40, P75 to P77, P120 to P124, P137, P147, P200 to P206, EXCLK, EXCLKS, $\overline{\text{RESET}}$	Normal input buffer	0		0.2V_{DD}	V
	V_{IL2}	P03, P10, P11	TTL input buffer 4.0 V ≤ V_{DD} ≤ 5.5 V	0		0.8	V
			TTL input buffer 3.3 V ≤ V_{DD} < 4.0 V	0		0.5	V
			TTL input buffer 2.7 V ≤ V_{DD} < 3.3 V	0		0.32	V

Caution The maximum value of V_{IH} of pins P02, P10 to P12 is V_{DD}, even in the N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH1}	P02, P03, P05, P06, P10 to P12, P30, P31, P40, P75 to P77, P120, P147, P200 to P206	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –3.0 mA	V_{DD} – 0.7			V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –1.0 mA	V_{DD} – 0.5			V
	V_{OH2}	P20 to P22, P24 to P27	2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OH2} = –100 μA	V_{DD} – 0.5			V
Output voltage, low	V_{OL1}	P02, P03, P05, P06, P10 to P12, P30, P31, P40, P75 to P77, P120, P147, P200 to P206	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 8.5 mA			0.7	V
			4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 4.0 mA			0.4	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 1.5 mA			0.4	V
	V_{OL2}	P20 to P22, P24 to P27	2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL2} = 400 μA			0.4	V

Caution P02, P10 to P12 do not output high level in N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P02, P03, P05, P06, P10 to P12, P20 to P22, P24 to P27, P30, P31, P40, P75 to P77, P120, P137, P147, P200 to P206, $\overline{\text{RESET}}$	$V_I = V_{DD}$				1	μA
	I_{LIH2}	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	$V_I = V_{DD}$	In input port or external clock input			1	μA
				In resonator connection			10	μA
Input leakage current, low	I_{LIL1}	P02, P03, P05, P06, P10 to P12, P20 to P22, P24 to P27, P30, P31, P40, P75 to P77, P120, P137, P147, P200 to P206, $\overline{\text{RESET}}$	$V_I = V_{SS}$				–1	μA
	I_{LIL2}	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	$V_I = V_{SS}$	In input port or external clock input			–1	μA
				In resonator connection			–10	μA
On-chip pull-up resistance	R_U	P02, P03, P05, P06, P10 to P12, P30, P31, P40, P75 to P77, P120, P147, P200 to P206	$V_I = V_{SS}$, In input port		10	20	100	kΩ

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

33.3.2 Supply current characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (1/2)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD1}	Operating mode	HS (high-speed main) mode Note 5	f_{IH} = 16 MHz Note 3	V_{DD} = 5.0 V		2.9	4.8	mA
					V_{DD} = 3.0 V		2.9	4.8	mA
			HS (high-speed main) mode Note 5	f_{MX} = 20 MHz Note 2, V_{DD} = 5.0 V	Square wave input		3.2	5.6	mA
					Resonator connection		3.3	5.7	mA
				f_{MX} = 20 MHz Note 2, V_{DD} = 3.0 V	Square wave input		3.2	5.6	mA
					Resonator connection		3.3	5.7	mA
				f_{MX} = 10 MHz Note 2, V_{DD} = 5.0 V	Square wave input		2.0	3.3	mA
					Resonator connection		2.0	3.3	mA
				f_{MX} = 10 MHz Note 2, V_{DD} = 3.0 V	Square wave input		2.0	3.3	mA
					Resonator connection		2.0	3.3	mA
			HS (high-speed main) mode Note 5	f_{IH} = 4 MHz Note 3 f_{PLL} = 64 MHz, f_{CLK} = 16 MHz	V_{DD} = 5.0 V		3.3	6.5	mA
					V_{DD} = 3.0 V		3.3	6.5	mA
			Subsystem clock operation	f_{SUB} = 32.768 kHz Note 4 T_A = –40°C	Square wave input		4.2	6.0	μA
					Resonator connection		4.4	6.2	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +25°C	Square wave input		4.2	6.0	μA
					Resonator connection		4.4	6.2	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +50°C	Square wave input		4.3	7.2	μA
					Resonator connection		4.5	7.4	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +70°C	Square wave input		4.4	8.1	μA
					Resonator connection		4.6	8.3	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +85°C	Square wave input		5.2	11.4	μA
					Resonator connection		5.4	11.6	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +105°C	Square wave input		6.9	20.8	μA
					Resonator connection		7.1	21.0	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +125°C	Square wave input		11.1	51.2	μA
					Resonator connection		11.3	51.4	μA

(**Notes** and **Remarks** are listed on the next page.)

Notes 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The following points apply in the HS (high-speed main) modes.

- The currents in the “TYP.” column do not include the operating currents of the peripheral modules.
- The currents in the “MAX.” column include the operating currents of the peripheral modules, except for those flowing into the A/D converter, comparator, programmable gain amplifier, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors, and those flowing while the data flash memory is being rewritten.

In the subsystem clock operation, the currents in both the “TYP.” and “MAX.” columns do not include the operating currents of the peripheral modules. However, in HALT mode, including the current flowing into the RTC.

2. When high-speed on-chip oscillator and subsystem clock are stopped.

3. When high-speed system clock and subsystem clock are stopped.

4. When high-speed on-chip oscillator and high-speed system clock are stopped. When AMPHS1 = 1 (Ultra-low power consumption oscillation).

5. Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.

HS (high-speed main) mode: 2.7 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 20 MHz

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: High-speed on-chip oscillator clock frequency

3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

4. Except subsystem clock operation, temperature condition of the TYP. value is T_A = 25°C

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (2/2)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD2} Note 2	HALT mode	HS (high-speed main) mode Note 6	f_{IH} = 16 MHz Note 4	V_{DD} = 5.0 V		0.50	2.0	mA
					V_{DD} = 3.0 V		0.50	2.0	mA
			HS (high-speed main) mode Note 6	f_{MX} = 20 MHz Note 3, V_{DD} = 5.0 V	Square wave input		0.40	2.2	mA
					Resonator connection		0.50	2.3	mA
				f_{MX} = 20 MHz Note 3, V_{DD} = 3.0 V	Square wave input		0.40	2.2	mA
					Resonator connection		0.50	2.3	mA
				f_{MX} = 10 MHz Note 3, V_{DD} = 5.0 V	Square wave input		0.24	1.22	mA
					Resonator connection		0.30	1.28	mA
				f_{MX} = 10 MHz Note 3, V_{DD} = 3.0 V	Square wave input		0.24	1.22	mA
					Resonator connection		0.30	1.28	mA
			HS (high-speed main) mode Note 6	f_{IH} = 4 MHz Note 4 f_{PLL} = 64 MHz, f_{CLK} = 16 MHz	V_{DD} = 5.0 V		0.95	3.7	mA
					V_{DD} = 3.0 V		0.95	3.7	mA
			Subsystem clock operation	f_{SUB} = 32.768 kHz Note 5 T_A = –40°C	Square wave input		0.28	0.70	μA
					Resonator connection		0.47	0.89	μA
				f_{SUB} = 32.768 kHz Note 5 T_A = +25°C	Square wave input		0.33	0.70	μA
					Resonator connection		0.52	0.89	μA
				f_{SUB} = 32.768 kHz Note 5 T_A = +50°C	Square wave input		0.41	1.90	μA
					Resonator connection		0.60	2.09	μA
				f_{SUB} = 32.768 kHz Note 5 T_A = +70°C	Square wave input		0.54	2.80	μA
					Resonator connection		0.73	2.99	μA
				f_{SUB} = 32.768 kHz Note 5 T_A = +85°C	Square wave input		1.27	6.10	μA
					Resonator connection		1.46	6.29	μA
				f_{SUB} = 32.768 kHz Note 5 T_A = +105°C	Square wave input		3.04	15.5	μA
					Resonator connection		3.23	15.7	μA
				f_{SUB} = 32.768 kHz Note 5 T_A = +125°C	Square wave input		7.20	45.2	μA
					Resonator connection		7.53	45.5	μA
	I_{DD3}	STOP mode Note 7	T_A = –40°C				0.18	0.50	μA
			T_A = +25°C				0.23	0.50	μA
			T_A = +50°C				0.27	1.70	μA
			T_A = +70°C				0.44	2.60	μA
			T_A = +85°C				1.17	5.90	μA
			T_A = +105°C				2.94	15.3	μA
			T_A = +125°C				7.14	45.1	μA

(**Notes** and **Remarks** are listed on the next page.)

Notes 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The following points apply in the HS (high-speed main) modes.

- The currents in the “TYP.” column do not include the operating currents of the peripheral modules.
- The currents in the “MAX.” column include the operating currents of the peripheral modules, except for those flowing into the A/D converter, comparator, programmable gain amplifier, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors, and those flowing while the data flash memory is being rewritten.

In the subsystem clock operation, the currents in both the “TYP.” and “MAX.” columns do not include the operating currents of the peripheral modules. However, in HALT mode, including the current flowing into the RTC.

In the STOP mode, the currents in both the “TYP.” and “MAX.” columns do not include the operating currents of the peripheral modules.

2. During HALT instruction execution by flash memory.

3. When high-speed on-chip oscillator and subsystem clock are stopped.

4. When high-speed system clock and subsystem clock are stopped.

5. When high-speed on-chip oscillator and high-speed system clock are stopped. When RTCLPC = 1 and setting ultra-low current consumption (AMPHS1 = 1).

6. Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.
HS (high-speed main) mode: 2.7 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 20 MHz

7. Regarding the value for current operate the subsystem clock in STOP mode, refer to that in HALT mode.

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: High-speed on-chip oscillator clock frequency

3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

4. Except subsystem clock operation and STOP mode, temperature condition of the TYP. value is T_A = 25°C

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Low-speed on-chip oscillator operating current	I_{FIL} Note 1				0.20		μA
RTC operating current	I_{RTC} Notes 1, 2, 3				0.02		μA
12-bit interval timer operating current	I_{IT} Notes 1, 2, 4				0.02		μA
Watchdog timer operating current	I_{WDT} Notes 1, 2, 5	f_{IL} = 15 kHz			0.22		μA
A/D converter operating current	I_{ADC} Notes 1, 6	When conversion at maximum speed	Normal mode, AV_{REFP} = V_{DD} = 5.0 V		1.3	1.7	mA
A/D converter reference voltage current	I_{ADREF} Note 1				75.0		μA
Temperature sensor operating current	I_{TMPS} Note 1				75.0		μA
LVD operating current	I_{LVD} Notes 1, 7				0.08		μA
Self-programming operating current	I_{FSP} Notes 1, 8				2.5	12.2	mA
Programmable gain amplifier operating current	I_{PGA} Note 9		AV_{REFP} = V_{DD} = 5.0 V		0.21	0.37	mA
			AV_{REFP} = V_{DD} = 3.0 V		0.18	0.35	mA
Comparator operating current	I_{CMP} Note 10	When one comparator channel is operating	AV_{REFP} = V_{DD} = 5.0 V		41.4	74	μA
			AV_{REFP} = V_{DD} = 3.0 V		37.2	71	μA
	I_{VREF}	When one internal reference voltage circuit is operating	AV_{REFP} = V_{DD} = 5.0 V		14.8	31	μA
			AV_{REFP} = V_{DD} = 3.0 V		8.9	24	μA
Programmable gain amplifier/ comparator reference current source	I_{IREF} Note 11		AV_{REFP} = V_{DD} = 5.0 V		3.2	6.1	μA
			AV_{REFP} = V_{DD} = 3.0 V		2.9	4.9	μA
BGO operating current	I_{BGO} Note 12				2.50	12.2	mA
SNOOZE operating current	I_{SNOZ} Note 1	A/D converter operation	The mode is performed Note 13		0.50	1.10	mA
			The A/D conversion operations are performed, Normal mode, AV_{REFP} = V_{DD} = 5.0 V		1.20	2.17	mA
		Simplified SPI (CSI)/UART operation			0.70	1.27	mA

(**Notes** and **Remarks** are listed on the next page.)

Notes 1. Current flowing to the V_{DD}.

2. When the high-speed on-chip oscillator and high-speed system clock are stopped.

3. Current flowing only to the real-time clock (RTC) (excluding the operating current of the low-speed on-chip oscillator and the XT1 oscillator). The supply current of the RL78 microcontrollers is the sum of the values of either I_{DD1} or I_{DD2}, and I_{RTC}, when the real-time clock is operating in operating mode or in HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be added. I_{DD2} subsystem clock operation includes the operational current of the real-time clock.

4. Current flowing only to the 12-bit interval timer (excluding the operating current of the XT1 oscillator and f_{IL} operating current). The current of the RL78 microcontrollers is the sum of the values of either I_{DD1} or I_{DD2}, and I_{IT}, when the 12-bit interval timer operates in operation mode or HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be added.

5. Current flowing only to the watchdog timer (including the operating current of the low-speed on-chip oscillator). The supply current value of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3}, and I_{WDT}, when the watchdog timer is operating.

6. Current flowing only to the A/D converter. The supply current value of the RL78 microcontrollers is the sum of I_{DD1} or I_{DD2} and I_{ADC}, when the A/D converter is operating in operating mode or in HALT mode.

7. Current flowing only 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 in operation.

8. Current flowing during self-programming operation.

9. Current flowing only to the programmable gain amplifier. The supply current value of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3}, and I_{PGA}, when the programmable gain amplifier is operating in operating mode or in HALT mode.

10. Current flowing only to the comparator. The supply current value of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3}, and I_{CMP}, when the comparator is operating.

11. This is the current required to flow to V_{DD} pin of the current circuit that is used as the programmable gain amplifier and the comparator.

12. Current flowing only during data flash rewrite.

13. See **21.3.3 SNOOZE mode** for shift time to the SNOOZE mode.

Remarks 1. f_{IL}: Low-speed on-chip oscillator clock frequency

2. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

3. f_{CLK}: CPU/peripheral hardware clock frequency

4. Temperature condition of the TYP. value is T_A = 25°C

5. Example of calculating current value when using programmable gain amplifier and comparator.

Examples 1) TYP. operating current value when three comparator channels, one internal reference voltage generator, and PGA are operating (when AV_{REFP} = V_{DD} = 5.0 V)

$I_{CMP} \times 3 + I_{VREF} + I_{PGA} + I_{IREF}$
= 41.4 [μA] × 3 + 14.8 [μA] × 1 + 210 [μA] + 3.2 [μA]
= 352.2 [μ A]

Examples 2) TYP. operating current value when using two comparator channels, without using internal reference voltage generator (when AV_{REFP} = V_{DD} = 5.0 V)

$I_{CMP} \times 2 + I_{IREF}$
= 41.4 [μA] × 2 + 3.2 [μA]
= 86.0 [μA]

33.4 AC Characteristics

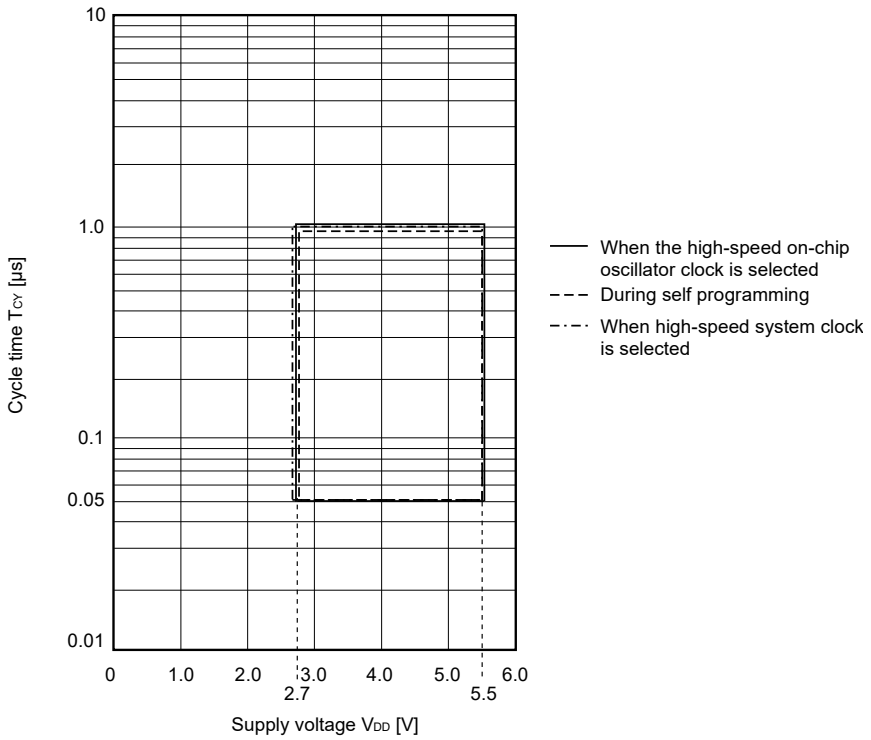
(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	HS (high-speed main) mode		0.05		1	μs
		Subsystem clock (f_{SUB}) operation			28.5	30.5	31.3	μs
		In the self programming mode	HS (high-speed main) mode	T_A = –40 to +105°C	0.05		1	μs
External system clock frequency	f_{EX}				1.0		20.0	MHz
	f_{EXS}				32		35	kHz
External system clock input high-level width, low-level width	t_{EXH}, t_{EXL}				24			ns
	t_{EXHS}, t_{EXLS}				13.7			μs
TI03, TI05, TI06, TI07 input high-level width, low-level width	t_{TIH}, t_{TIL}				2/f_{MCK}+10			ns
TO03, TO05, TO06, TKBO00, TKBO01, TKBO10, TKBO11, TKBO20, TKBO21, TKCO00 to TKCO05 output frequency (When duty = 50%)	f_{TO}	HS (high-speed main) mode		4.0 V ≤ V_{DD} ≤ 5.5 V			5	MHz
				2.7 V ≤ V_{DD} < 4.0 V			4	MHz
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}	INTP0, INTP3, INTP4, INTP9 to INTP11, INTP20 to INTP23		2.7 V ≤ V_{DD} ≤ 5.5 V	1			μs
$\overline{\text{RESET}}$ low-level width	t_{RSL}				10			μs

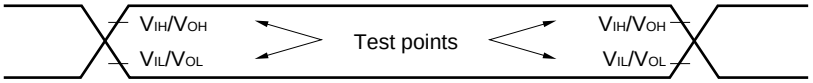
Remark f_{MCK}: Timer array unit operation clock frequency
(Operation clock to be set by the CKS0n bit of timer mode register 0n (TMR0n). n: Channel number (n = 0 to 7))

Minimum Instruction Execution Time during Main System Clock Operation

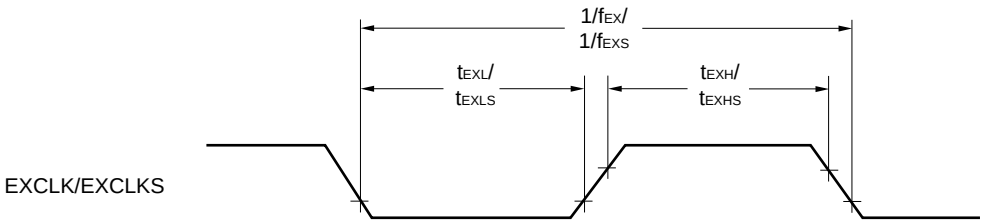
T_{CY} vs V_{DD} (HS (high-speed main) mode)



AC Timing Test Points

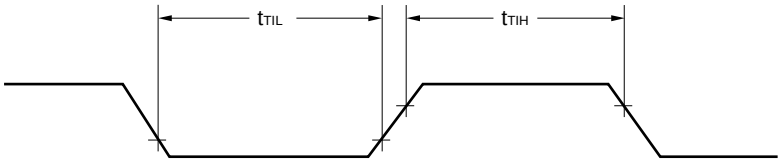


External System Clock Timing

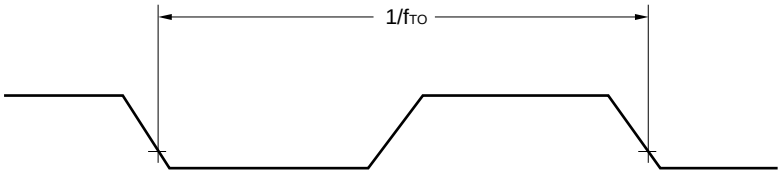


TI/TO Timing

TI03, TI05, TI06, TI07

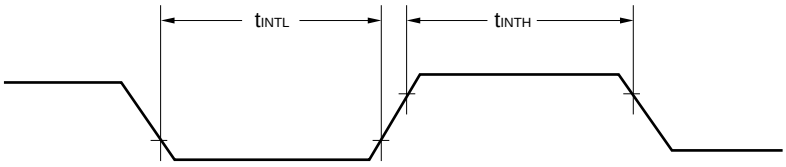


TO03, TO05, TO06, TKBO00, TKBO01, TKBO10, TKBO11, TKBO20, TKBO21, TKCO00 to TKCO05



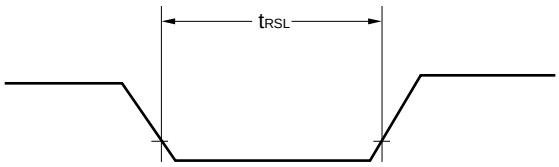
Interrupt Request Input Timing

INTP0, INTP3, INTP4, INTP9 to INTP11, INTP20 to INTP23



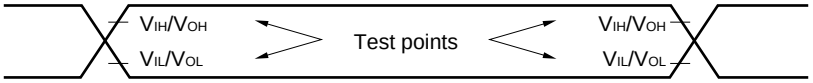
$\overline{\text{RESET}}$ Input Timing

$\overline{\text{RESET}}$



33.5 Peripheral Functions Characteristics

AC Timing Test Points



33.5.1 Serial array unit 0, 4 (UART0, UART1, CSI00, DALI/UART4)

(1) During communication at same potential (UART mode)

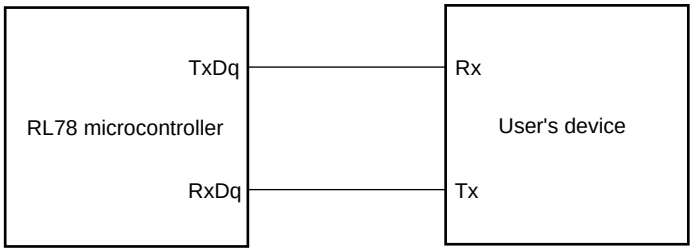
(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
Transfer rate[Note 1]				f_{MCK}/6	bps
		Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK}[Note 2]		3.3	Mbps

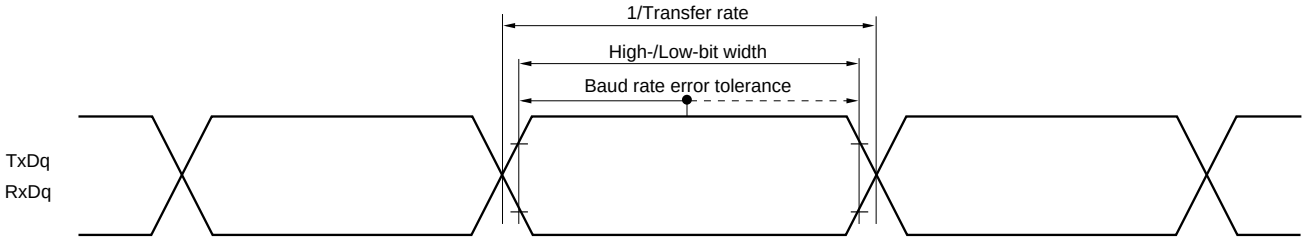
Notes 1. Transfer rate in the SNOOZE mode is 4800 bps only.

2. The operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
HS (high-speed main) mode: 20 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)

UART mode connection diagram (during communication at same potential)



UART mode bit width (during communication at same potential) (reference)



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).

Remarks 1. q: UART number (q = 0, 1), g: PIM and POM number (g = 0, 1)

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00 to 03))

(2) During communication at same potential (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq 4/f_{CLK}$	4.0 V ≤ V_{DD} ≤ 5.5 V	250		ns
			2.7 V ≤ V_{DD} ≤ 5.5 V	500		ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V		$t_{KCY1}/2 - 20$		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		$t_{KCY1}/2 - 40$		ns
SIp setup time (to SCKp↑)Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V		80		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		80		ns
SIp hold time (from SCKp↑)Note 2	t_{KSI1}			40		ns
Delay time from SCKp↓ to SOp outputNote 3	t_{KSO1}	C = 30 pFNote 4			80	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the SCKp and SOp output lines.

Caution Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remarks 1. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM number (g = 1)

2. f_{MCK}: Serial array unit operation clock frequency (Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00))

(3) During communication at same potential (Simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
SCKp cycle time[Note 5]	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V	f_{MCK} ≤ 20 MHz	6/f_{MCK}		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V	16 MHz < f_{MCK}	8/f_{MCK}		ns
			f_{MCK} ≤ 16 MHz	6/f_{MCK}		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}			t_{KCY2}/2		ns
SIp setup time (to SCKp↑)[Note 1]	t_{SIK2}			1/f_{MCK}+40		ns
SIp hold time (from SCKp↑)[Note 2]	t_{KSI2}			1/f_{MCK}+60		ns
Delay time from SCKp↓ to SOp output[Note 3]	t_{KSO2}	C = 30 pF[Note 4]			2/f_{MCK}+80	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

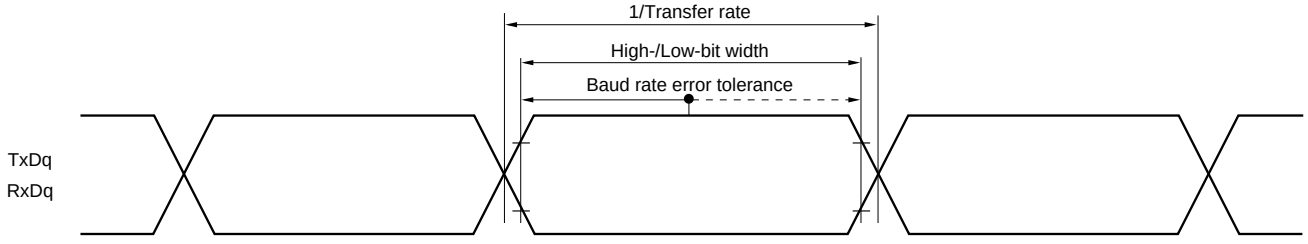
4. C is the load capacitance of the SOp output lines.

5. Transfer rate in the SNOOZE mode: MAX. 1 Mbps

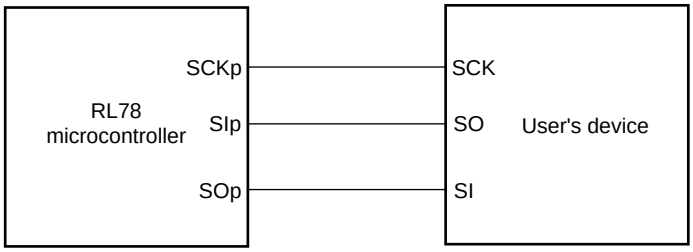
Caution Select the normal input buffer for the SIp pin and SCKp pin and the normal output mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remarks 1. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM number (g = 1)

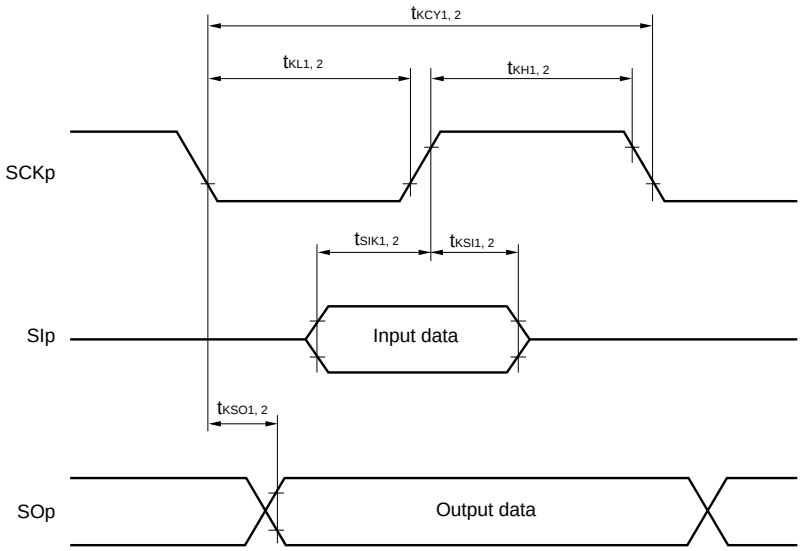
2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00))



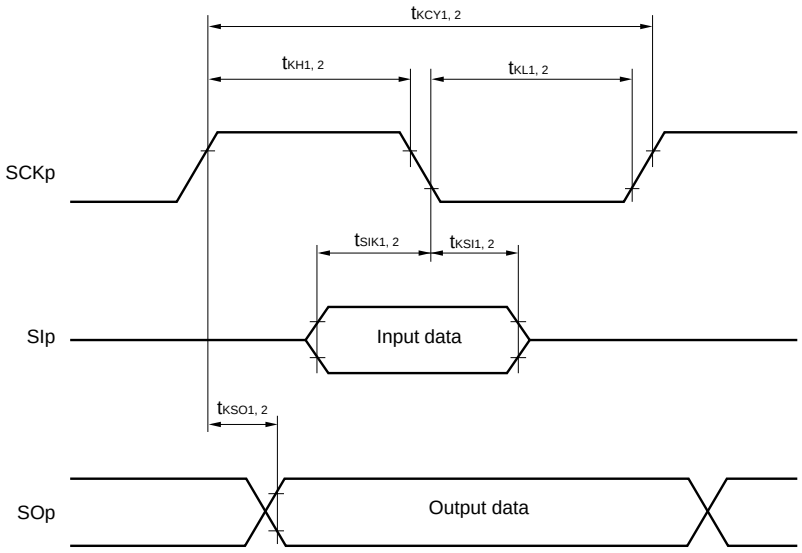
Simplified SPI (CSI) mode connection diagram (during communication at same potential)



Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remarks 1. p: CSI number (p = 00)

2. m: Unit number, n: Channel number (mn = 00)

(4) Communication at different potential (2.5 V, 3 V) (UART mode) (1/2)

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions			HS (high-speed main) Mode		Unit
					MIN.	MAX.	
Transfer rate		Reception	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V			f_{MCK}/6 Note 1	bps
				Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 2		3.3	Mbps
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V			f_{MCK}/6 Note 1	bps
				Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 2		3.3	Mbps

Notes 1. Transfer rate in the SNOOZE mode is 4800 bps only.

2. The operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
HS (high-speed main) mode: 20 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)

Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remarks 1. V_b[V]: Communication line voltage

2. q: UART number (q = 0, 1), g: PIM and POM number (g = 0, 1)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00 to 03)

(4) Communication at different potential (2.5 V, 3 V) (UART mode) (2/2)

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions			HS (high-speed main) Mode		Unit
					MIN.	MAX.	
Transfer rate		Transmission	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V			**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	Mbps
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V			**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	Mbps

Notes 1. The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 4.0 V ≤ V_{DD} ≤ 5.5 V and 2.7 V ≤ V_b ≤ 4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

2. This value as an example is calculated 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.

3. The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 2.7 V ≤ V_{DD} < 4.0 V and 2.3 V ≤ V_b ≤ 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

4. This value as an example is calculated 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.

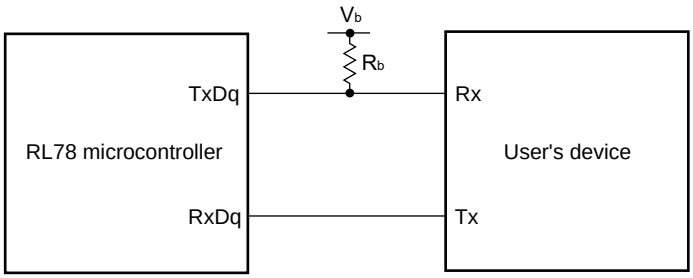
Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remarks 1. R_b[Ω]: Communication line (TxDq) pull-up resistance,
C_b[F]: Communication line (TxDq) load capacitance, V_b[V]: Communication line voltage

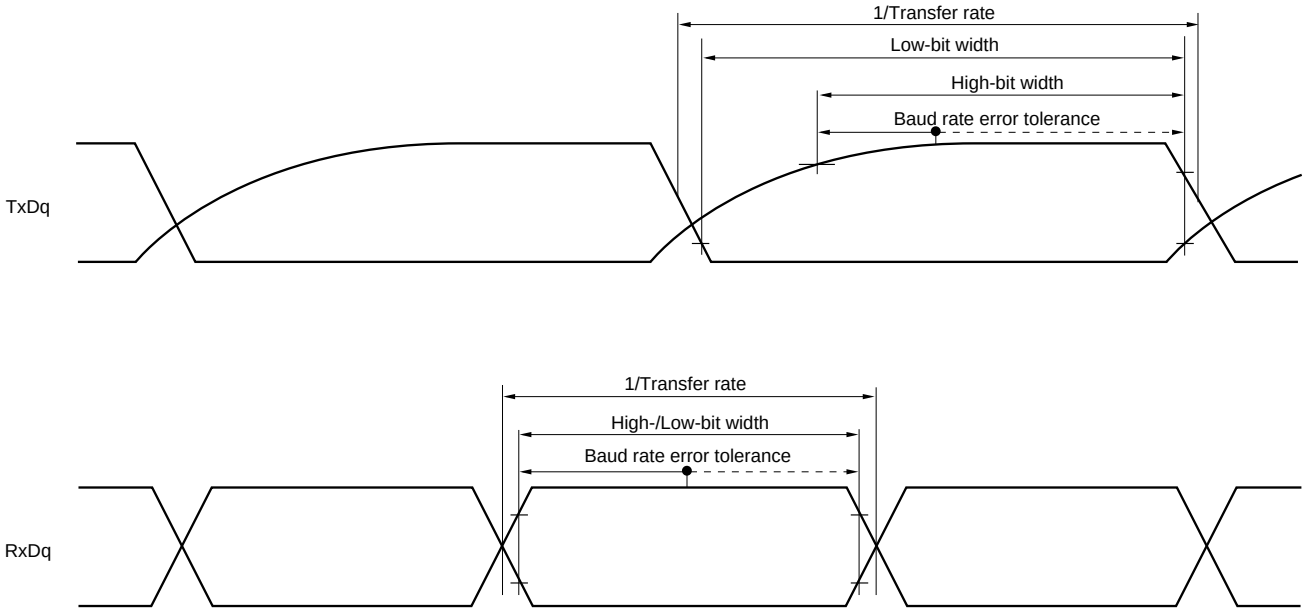
2. q: UART number (q = 0, 1), g: PIM and POM number (g = 0, 1)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00 to 03))

UART mode connection diagram (during communication at different potential)



UART mode bit width (during communication at different potential) (reference)



Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remarks 1. R_b[Ω]: Communication line (TxDq) pull-up resistance, V_b[V]: Communication line voltage

2. q: UART number (q = 0, 1), g: PIM and POM number (g = 0, 1)

(5) Communication at different potential (2.5 V, 3 V) (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq 4/f_{CLK}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	600		ns
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	1000		ns
SCKp high-level width	t_{KH1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 – 80		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 – 170		ns
SCKp low-level width	t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 – 28		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 – 40		ns
SIp setup time (to SCKp↑)[Note 1]	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		160		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		250		ns
SIp hold time (from SCKp↑)[Note 1]	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		40		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		40		ns
Delay time from SCKp↓ to SOp output[Note 1]	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ			160	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			250	ns
SIp setup time (to SCKp↓)[Note 2]	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		80		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		80		ns
SIp hold time (from SCKp↓)[Note 2]	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		40		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		40		ns
Delay time from SCKp↑ to SOp output[Note 2]	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ			80	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			80	ns

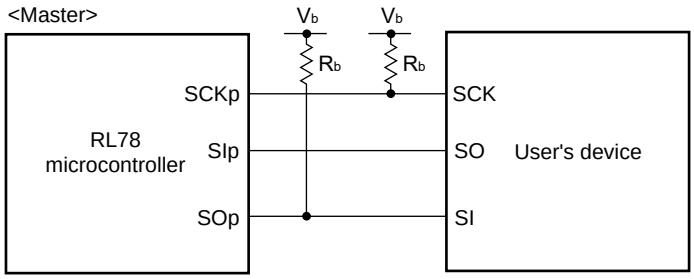
Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

2. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

(**Caution** and **Remarks** are listed on the next page.)

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

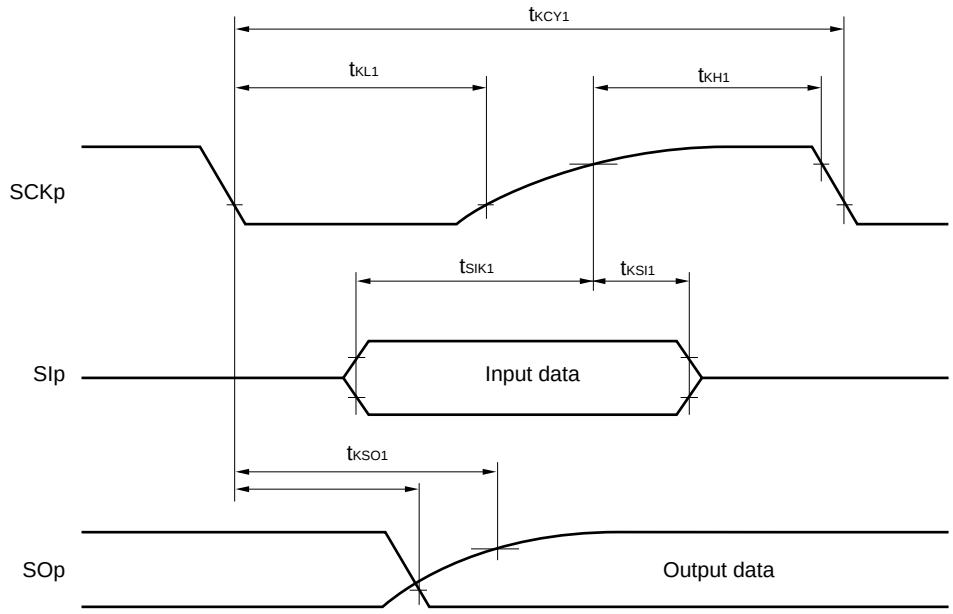
Simplified SPI (CSI) mode connection diagram (during communication at different potential)



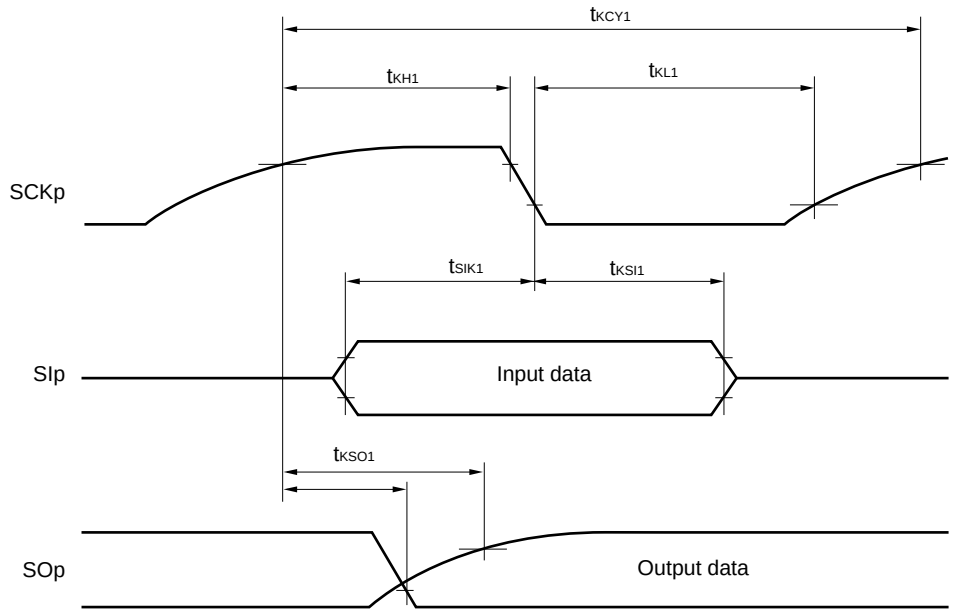
Remarks 1. R_b[Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

2. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM number (g = 1)

Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remark p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM number (g = 1)

(6) DALI/UART4 mode

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
Transfer rate				f_{MCK}/12	bps
		Maximum transfer rate theoretical value f_{CLK} = 20 MHz, f_{MCK} = f_{CLK}		1.6	Mbps

Remark f_{MCK}: Operation clock frequency of DALI/UART.
(Operation clock to be set by the serial clock select register 4 (SPS4).)

33.5.2 Serial interface IICA

(1) I^2C standard mode

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
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 = “L”	t_{LOW}		4.7		μs
Hold time when SCLA0 = “H”	t_{HIGH}		4.0		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	$t_{SU:STO}$		4.0		μs
Bus-free time	t_{BUF}		4.7		μs

Notes 1. The first clock pulse is generated after this period when the start/restart condition is detected.

2. The maximum value (MAX.) of $t_{HD:DAT}$ is during normal transfer and a clock stretch state is inserted in the ACK (acknowledge) timing.

Remark The maximum value of Cb (communication line capacitance) and the value of Rb (communication line pull-up resistor) at that time in each mode are as follows.

Standard mode: C_b = 400 pF, R_b = 2.7 kΩ

(2) I^2C fast mode

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
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 = “L”	t_{LOW}		1.3		μs
Hold time when SCLA0 = “H”	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

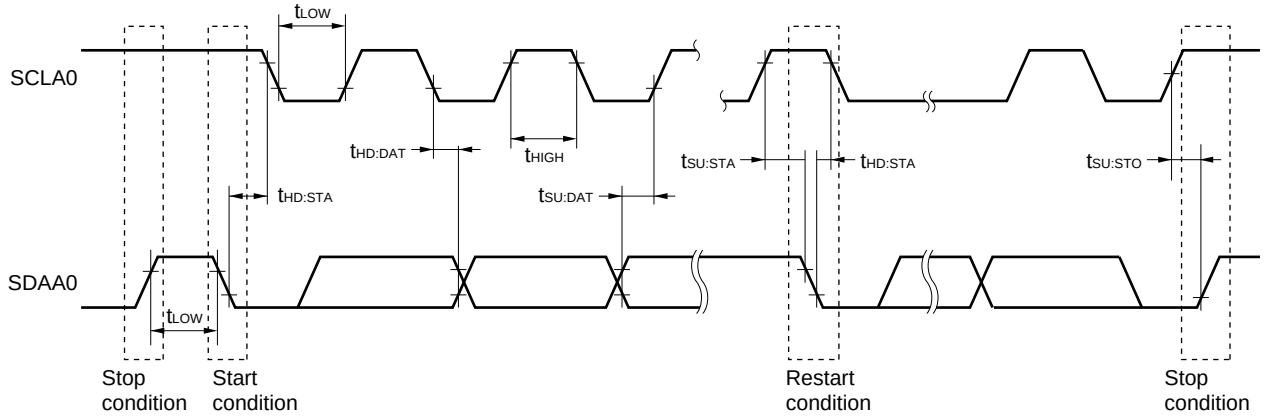
Notes 1. The first clock pulse is generated after this period when the start/restart condition is detected.

2. The maximum value (MAX.) of $t_{HD:DAT}$ is during normal transfer and a clock stretch state is inserted in the ACK (acknowledge) timing.

Remark The maximum value of Cb (communication line capacitance) and the value of Rb (communication line pull-up resistor) at that time in each mode are as follows.

fast mode: C_b = 320 pF, R_b = 1.1 kΩ

IICA serial transfer timing



33.6 Analog Characteristics

33.6.1 A/D converter characteristics

Classification of A/D converter characteristics

Input channel	Reference Voltage		
	Reference voltage (+) = AV_{REFP} Reference voltage (–) = AV_{REFM}	Reference voltage (+) = V_{DD} Reference voltage (–) = V_{SS}	Reference voltage (+) = V_{BGR} Reference voltage (–) = AV_{REFM}
ANI0 to ANI2, ANI4 to ANI7	See **33.6.1 (1)**.	See **33.6.1 (3)**.	See **33.6.1 (4)**.
ANI16 to ANI19	See **33.6.1 (2)**.		
Internal reference voltage Temperature sensor output voltage	See **33.6.1 (1)**.		–

(1) When reference voltage (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (–) = AV_{REFM}/ANI1 (ADREFM = 1), target ANI pin: ANI2, ANI4 to ANI7, internal reference voltage, and temperature sensor output voltage

(T_A = –40 to +125°C, 2.7 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = AV_{REFP}, Reference voltage (–) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error[Note 1]	AINL	10-bit resolution AV_{REFP} = V_{DD}[Note 3]			1.2	±3.5	LSB
Conversion time	t_{CONV}	10-bit resolution Target pin: ANI2, ANI4 to ANI7	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.4		39	μs
		10-bit resolution Target pin: Internal reference voltage, and temperature sensor output voltage (HS (high-speed main) mode)	3.6 V ≤ V_{DD} ≤ 5.5 V	2.375		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.8		39	μs
Zero-scale error[Notes 1, 2]	E_{ZS}	10-bit resolution AV_{REFP} = V_{DD}[Note 3]				±0.25	%FSR
Full-scale error[Notes 1, 2]	E_{FS}	10-bit resolution AV_{REFP} = V_{DD}[Note 3]				±0.25	%FSR
Integral linearity error[Note 1]	ILE	10-bit resolution AV_{REFP} = V_{DD}[Note 3]				±2.5	LSB
Differential linearity error[Note 1]	DLE	10-bit resolution AV_{REFP} = V_{DD}[Note 3]				±1.5	LSB
Analog input voltage	V_{AIN}	ANI2, ANI4 to ANI7		0		AV_{REFP}	V
		Internal reference voltage (HS (high-speed main) mode)		V_{BGR}[Note 4]			V
		Temperature sensor output voltage (HS (high-speed main) mode)		V_{TMPS25}[Note 4]			V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. When AV_{REFP} < V_{DD}, the MAX. values are as follows.
Overall error: Add ±1.0 LSB to the MAX. value when AV_{REFP} = V_{DD}.
Zero-scale error/Full-scale error: Add ±0.05%FSR to the MAX. value when AV_{REFP} = V_{DD}.
Integral linearity error/Differential linearity error: Add ±0.5 LSB to the MAX. value when AV_{REFP} = V_{DD}.

4. See **33.6.2 Temperature sensor/internal reference voltage characteristics**.

(2) When reference voltage (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (–) = AV_{REFM}/ANI1 (ADREFM = 1), target pin: ANI16 to ANI19

(T_A = –40 to +125°C, 2.7 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = AV_{REFP}, Reference voltage (–) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error[Note 1]	AINL	10-bit resolution AV_{REFP} = V_{DD}[Note 3]			1.2	±5.0	LSB
Conversion time	t_{CONV}	10-bit resolution Target ANI pin : ANI16 to ANI19	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} < 5.5 V	3.4		39	μs
Zero-scale error[Notes 1, 2]	E_{ZS}	10-bit resolution AV_{REFP} = V_{DD}[Note 3]				±0.35	%FSR
Full-scale error[Notes 1, 2]	E_{FS}	10-bit resolution AV_{REFP} = V_{DD}[Note 3]				±0.35	%FSR
Integral linearity error[Note 1]	ILE	10-bit resolution AV_{REFP} = V_{DD}[Note 3]				±3.5	LSB
Differential linearity error[Note 1]	DLE	10-bit resolution AV_{REFP} = V_{DD}[Note 3]				±2.0	LSB
Analog input voltage	V_{AIN}	ANI16 to ANI19		0		AV_{REFP} and V_{DD}	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. When AV_{REFP} < V_{DD}, the MAX. values are as follows.
Overall error: Add ±4.0 LSB to the MAX. value when AV_{REFP} = V_{DD}.
Zero-scale error/Full-scale error: Add ±0.2%FSR to the MAX. value when AV_{REFP} = V_{DD}.
Integral linearity error/Differential linearity error: Add ±2.0 LSB to the MAX. value when AV_{REFP} = V_{DD}.

(3) When reference voltage (+) = V_{DD} (ADREFP1 = 0, ADREFP0 = 0), reference voltage (–) = V_{SS} (ADREFM = 0), target pin: ANI0 to ANI2, ANI4 to ANI7, ANI16 to ANI19, internal reference voltage, and temperature sensor output voltage

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = V_{DD}, Reference voltage (–) = V_{SS})

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error[Note 1]	AINL	10-bit resolution			1.2	±7.0	LSB
Conversion time	t_{CONV}	10-bit resolution Target pin: ANI0 to ANI2, ANI4 to ANI7, ANI16 to ANI19	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.4		39	μs
Conversion time	t_{CONV}	10-bit resolution Target pin: Internal reference voltage, and temperature sensor output voltage (HS (high-speed main) mode)	3.6 V ≤ V_{DD} ≤ 5.5 V	2.375		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.8		39	μs
Zero-scale error[Notes 1, 2]	E_{ZS}	10-bit resolution				±0.60	%FSR
Full-scale error[Notes 1, 2]	E_{FS}	10-bit resolution				±0.60	%FSR
Integral linearity error[Note 1]	ILE	10-bit resolution				±4.0	LSB
Differential linearity error[Note 1]	DLE	10-bit resolution				±2.0	LSB
Analog input voltage	V_{AIN}	ANI0 to ANI2, ANI4 to ANI7		0		V_{DD}	V
		ANI16 to ANI19		0		V_{DD}	V
		Internal reference voltage (HS (high-speed main) mode)		V_{BGR} [Note 3]			V
		Temperature sensor output voltage (HS (high-speed main) mode)		V_{TMPS25} [Note 3]			V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. See **33.6.2 Temperature sensor/internal reference voltage characteristics**.

(4) When reference voltage (+) = Internal reference voltage (ADREFP1 = 1, ADREFP0 = 0), reference voltage (–) = AV_{REFM}/ANI1 (ADREFM = 1), target pin: ANI0, ANI2, ANI4 to ANI7, ANI16 to ANI19

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = V_{BGR} Note 3, Reference voltage (–) = AV_{REFM} Note 4 = 0 V, HS (high-speed main) mode)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES			8		bit
Conversion time	t_{CONV}	8-bit resolution	17		39	μs
Zero-scale error Notes 1, 2	E_{ZS}	8-bit resolution			±0.60	%FSR
Integral linearity error Note 1	ILE	8-bit resolution			±2.0	LSB
Differential linearity error Note 1	DLE	8-bit resolution			±1.0	LSB
Analog input voltage	V_{AIN}		0		V_{BGR} Note 3	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. See **33.6.2 Temperature sensor/internal reference voltage characteristics**.

4. When reference voltage (–) = V_{SS}, the MAX. values are as follows.
Zero-scale error: Add ±0.35%FSR to the MAX. value when reference voltage (–) = AV_{REFM}.
Integral linearity error: Add ±0.5 LSB to the MAX. value when reference voltage (–) = AV_{REFM}.
Differential linearity error: Add ±0.2 LSB to the MAX. value when reference voltage (–) = AV_{REFM}.

33.6.2 Temperature sensor/internal reference voltage characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, HS (high-speed main) mode)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Temperature sensor output voltage	V_{TMPS25}	Setting ADS register = 80H, T_A = +25°C		1.05		V
Internal reference voltage	V_{BGR}	Setting ADS register = 81H	1.38	1.45	1.5	V
Temperature coefficient	F_{VTMPS}	Temperature sensor that depends on the temperature		–3.6		mV/C
Operation stabilization wait time	t_{AMP}		5			μs

33.6.3 Programmable gain amplifier

(T_A = –40 to +125°C, 2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V, V_{SS} = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input offset voltage	V_{IOPGA}					±5	±10	mV
Input voltage range	V_{IPGA}				0		$0.9V_{DD}$/gain	V
Gain error[Note 1]		4, 8 times					±1	%
		16 times					±1.5	%
		32 times					±2	%
Slew rate[Note 1]	SR_{RPGA}	Rising edge	4.0 V ≤ V_{DD} ≤ 5.5 V	4, 8 times	4			V/μs
				16, 32 times	1.4			V/μs
			2.7 V ≤ V_{DD} < 4.0 V	4, 8 times	1.8			V/μs
				16, 32 times	0.5			V/μs
	SR_{FPGA}	Falling edge	4.0 V ≤ V_{DD} ≤ 5.5 V	4, 8 times	3.2			V/μs
				16, 32 times	1.4			V/μs
			2.7 V ≤ V_{DD} < 4.0 V	4, 8 times	1.2			V/μs
				16, 32 times	0.5			V/μs
Operation stabilization wait time[Note 2]	t_{PGA}	4, 8 times			5			μs
		16, 32 times			10			μs

Notes 1. When V_{IPGA} = $0.1V_{DD}$/gain to $0.9V_{DD}$/gain.

2. Time required until a state is entered where the DC and AC specifications of the PGA are satisfied after the PGA operation has been enabled (PGAEN = 1).

Remark These characteristics apply when AV_{REFM} is selected as GND of the PGA by using the CVRVS1 bit.

33.6.4 Comparator

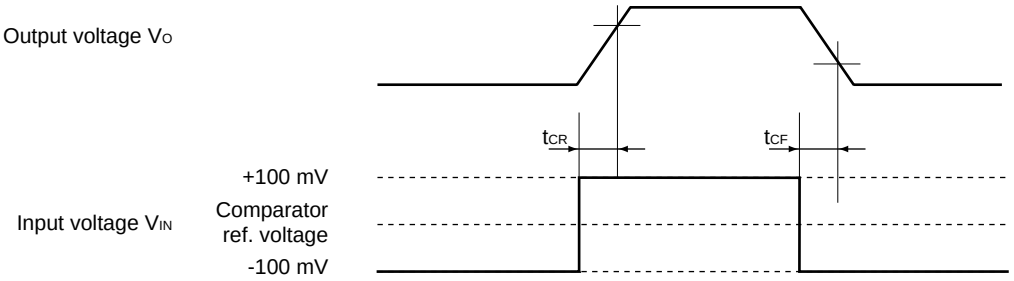
(T_A = –40 to +125°C, 2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V, V_{SS} = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input offset voltage	V_{IOCMP}			±5	±40	mV
Input voltage range	V_{ICMP}	CMP0P to CMP5P	0		V_{DD}	V
		CMPCOM	0.045		0.9V_{DD}	V
Internal reference voltage deviation	ΔV_{IREF}	CmRVM register values: 7FH to 80H (m = 0 to 2)			±2	LSB
		Other than above			±1	LSB
Response time	t_{CR}, t_{CF}	Input amplitude = ±100 mV		70	150	ns
Operation stabilization wait time[Note 1]	t_{CMP}	3.3 V ≤ V_{DD} ≤ 5.5 V	1			μs
		2.7 V ≤ V_{DD} < 3.3 V	3			μs
Reference voltage stabilization wait time	t_{VR}	CVRE: 0 to 1[Note 2]	10			μs

Notes 1. Time required until a state is entered where the DC and AC specifications of the comparator are satisfied after the operation of the comparator has been enabled (CMPnEN bit = 1: n = 0 to 5)

2. Enable comparator output (CnOE bit = 1; n = 0 to 5) after enabling operation of the internal reference voltage generator (by setting the CVREm bit to 1; m = 0 to 2) and waiting for the operation stabilization time to elapse.

Remark These characteristics apply when AV_{REFP} is selected as the power supply source of the internal reference voltage by using the CVRVS0 bit, and when AV_{REFM} is selected as GND of the internal reference voltage by using the CVRVS1 bit.

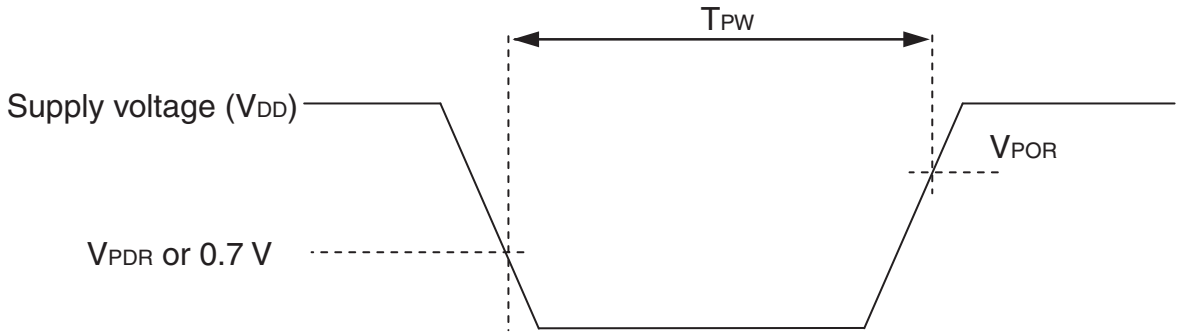


33.6.5 POR circuit characteristics

(T_A = –40 to +125°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}	Power supply rise time	1.45	1.51	1.62	V
	V_{PDR}	Power supply fall time	1.44	1.50	1.61	V
Minimum pulse width[Note]	T_{PW}		300			μs

Note Minimum time required for a POR reset when V_{DD} exceeds below V_{PDR}. This is also the minimum time required for a POR reset from when V_{DD} exceeds below 0.7 V to when V_{DD} exceeds V_{POR} while STOP mode is entered or the main system clock is stopped through setting bit 0 (HIOSTOP) and bit 7 (MSTOP) in the clock operation status control register (CSC).



33.6.6 LVD circuit characteristics

LVD Detection Voltage of Reset Mode and Interrupt Mode

(T_A = –40 to +125°C, $V_{PDR} \leq V_{DD} \leq$ 5.5 V, V_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVD0}	Power supply rise time	3.97	4.06	4.25	V
			Power supply fall time	3.89	3.98	4.15	V
		V_{LVD1}	Power supply rise time	3.67	3.75	3.93	V
			Power supply fall time	3.59	3.67	3.83	V
		V_{LVD2}	Power supply rise time	3.06	3.13	3.28	V
			Power supply fall time	2.99	3.06	3.20	V
		V_{LVD3}	Power supply rise time	2.95	3.02	3.17	V
			Power supply fall time	2.89	2.96	3.09	V
		V_{LVD4}	Power supply rise time	2.85	2.92	3.07	V
			Power supply fall time	2.79	2.86	2.99	V
		V_{LVD5}	Power supply rise time	2.75	2.81	2.95	V
			Power supply fall time	2.70	2.75	2.88	V
Minimum pulse width		t_{LW}		300			μs
Detection delay time						300	μs

LVD Detection Voltage of Interrupt & Reset Mode

(T_A = –40 to +125°C, $V_{PDR} \leq V_{DD} \leq 5.5$ V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Interrupt and reset mode	V_{LVD0}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 1, 1, falling reset voltage: 2.7 V		2.70	2.75	2.88	V
	V_{LVD1}	LVIS1, LVIS0 = 1, 0	Rising release reset voltage	2.85	2.92	3.07	V
			Falling interrupt voltage	2.79	2.86	2.99	V
	V_{LVD2}	LVIS1, LVIS0 = 0, 1	Rising release reset voltage	2.95	3.02	3.17	V
			Falling interrupt voltage	2.89	2.96	3.09	V
	V_{LVD3}	LVIS1, LVIS0 = 0, 0	Rising release reset voltage	3.97	4.06	4.25	V
			Falling interrupt voltage	3.89	3.98	4.15	V

33.6.7 Supply voltage rise inclination characteristics

(T_A = –40 to +125°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Supply voltage rise	SV_{DD}				54	V/ms

Caution Keep the internal reset status by using the LVD circuit or an external reset signal until V_{DD} rises to within the operating voltage range shown in 33.4 AC Characteristics.

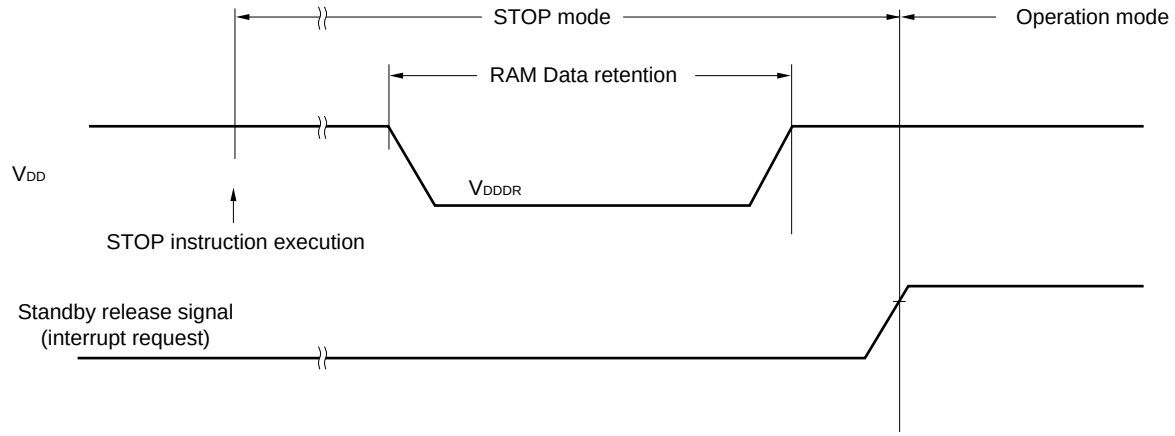
33.7 RAM Data Retention Characteristics

(T_A = –40 to +125°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage[Note 2]	V_{DDDR}		1.47[Note 1]		5.5	V

Note The value depends on the POR detection voltage. When the voltage drops, the RAM data is retained before a POR reset is effected, but RAM data is not retained when a POR reset is effected.

Caution When CPU is operated at the voltage of out of the operation voltage range, RAM data is not retained. Therefore, set STOP mode before the supplied voltage is below the operation voltage range.



33.8 Flash Memory Programming Characteristics

(T_A = –40 to +105°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
CPU/peripheral hardware clock frequency	f_{CLK}	2.7 V ≤ V_{DD} ≤ 5.5 V	1		32	MHz
Number of code flash rewrites Notes 1, 2, 3	C_{erwr}	Retained for 20 years, T_A = 85°C Note 3, 4	1,000			Times
Number of data flash rewrites Notes 1, 2, 3		Retained for 1 year, T_A = 25°C Note 3, 4		1,000,000		
		Retained for 5 years, T_A = 85°C Note 3, 4	100,000			
		Retained for 20 years, T_A = 85°C Note 3, 4	10,000			

Notes 1. 1 erase + 1 write after the erase is regarded as 1 rewrite. The retaining years are until next rewrite after the rewrite.

2. When using flash memory programmer and Renesas Electronics self programming library

3. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

4. These are the average temperature of during the retainment.

33.9 Dedicated Flash Memory Programmer Communication (UART)

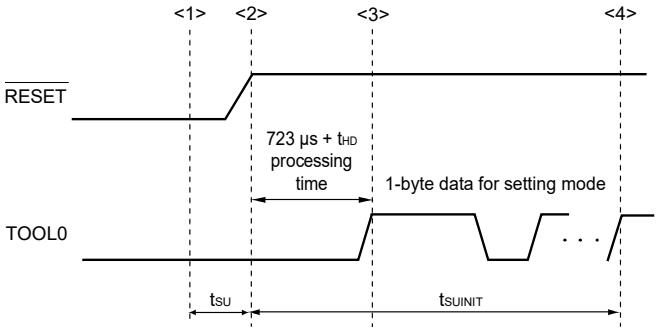
(T_A = –40 to +105°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		During serial programming	115.2 k		1 M	bps

33.10 Timing of Entry to Flash Memory Programming Modes

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
How long from when an external reset ends until the initial communication settings are specified	t_{SUINIT}	POR and LVD reset must end before the external reset ends.			100	ms
How long from when the TOOL0 pin is placed at the low level until an external reset ends	t_{SU}	POR and LVD reset must end before the external reset ends.	10			μs
How long the TOOL0 pin must be kept at the low level after a reset ends (except soft processing time)	t_{HD}	POR and LVD reset must end before the external reset ends.	1			ms



<1> The low level is input to the TOOL0 pin.
<2> The external reset ends (POR and LVD reset must end before the pin reset ends.).
<3> The TOOL0 pin is set to the high level.
<4> Complete the baud rate setting by UART reception.

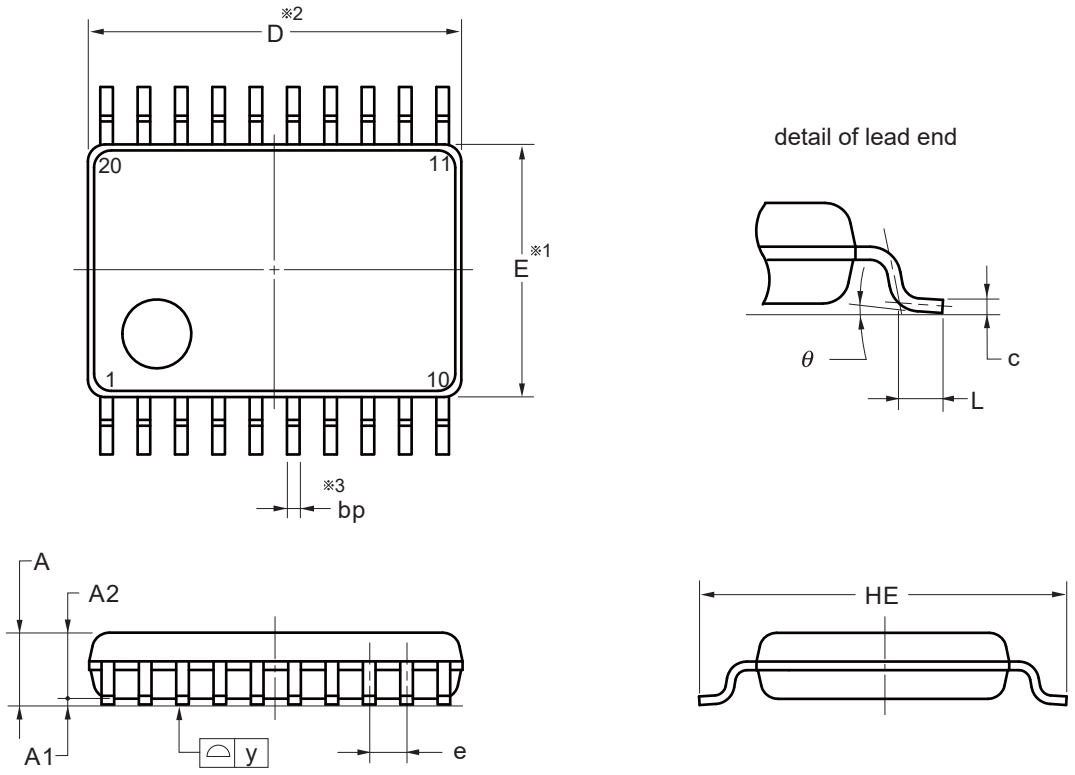
Remark t_{SUINIT}: The segment shows that it is necessary to finish specifying the initial communication settings within 100 ms from when the resets end.
t_{SU}: How long from when the TOOL0 pin is placed at the low level until an external reset ends
t_{HD}: How long to keep the TOOL0 pin at the low level from when the external and internal resets end (except soft processing time)

CHAPTER 34 PACKAGE DRAWINGS

34.1 20-pin Products

R5F1076CGSP#V0, R5F1076CGSP#X0, R5F1076CMSP#V0, R5F1076CMSP#X0

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



(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°

NOTE

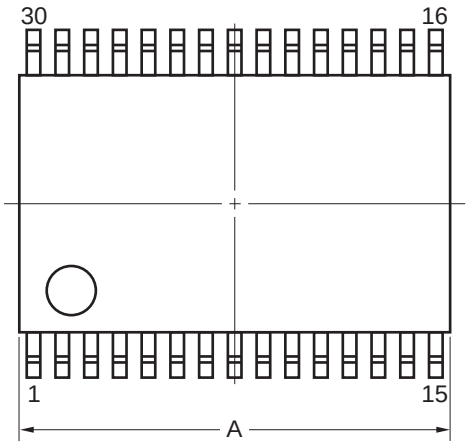
1.Dimensions "※1" and "※2" do not include mold flash.

2.Dimension "※3" does not include trim offset.

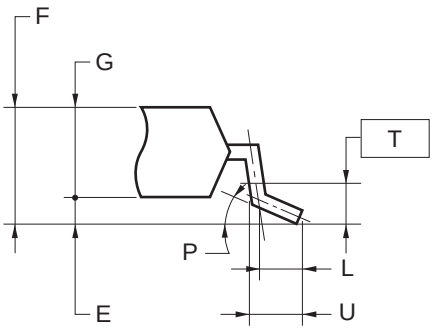
34.2 30-pin Products

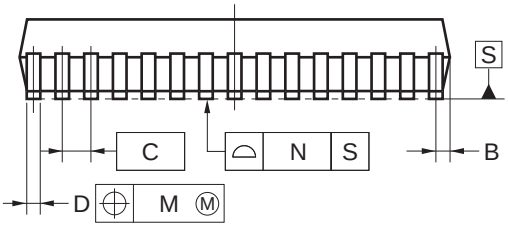
R5F107ACGSP#V0, R5F107AEGSP#V0, R5F107ACGSP#X0, R5F107AEGSP#X0, R5F107ACMSP#V0, R5F107AEMSP#V0, R5F107ACMSP#X0, R5F107AEMSP#X0

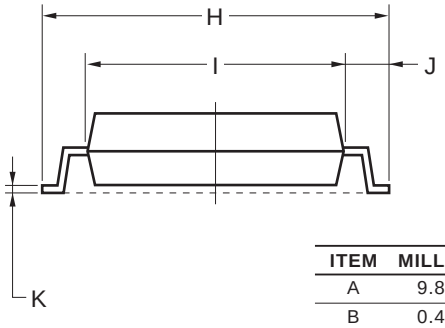
JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LSSOP30-0300-0.65	PLSP0030JB-B	S30MC-65-5A4-3	0.18



detail of lead end







NOTE

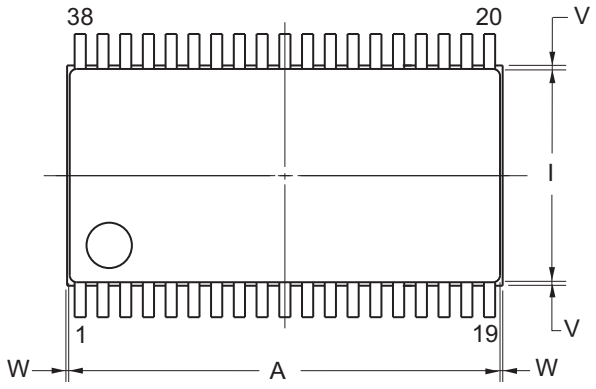
Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

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

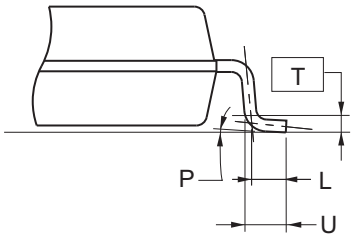
34.3 38-pin Products

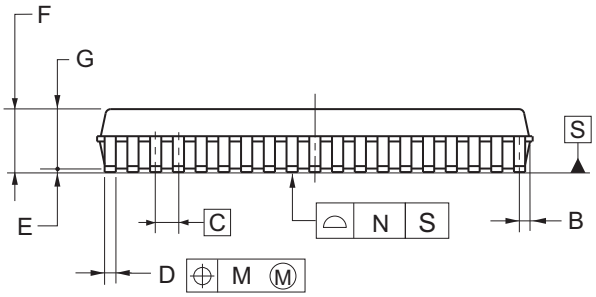
R5F107DEGSP#V0, R5F107DEGSP#X0, R5F107DEMSP#V0, R5F107DEMSP#X0

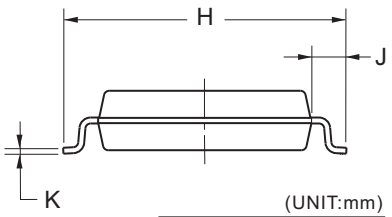
JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-SSOP38-0300-0.65	PRSP0038JA-A	P38MC-65-2A4-2	0.3



detail of lead end







(UNIT:mm)

ITEM	DIMENSIONS
A	12.30±0.10
B	0.30
C	0.65 (T.P.)
D	$0.32^{+0.08}_{-0.07}$
E	0.125±0.075
F	2.00 MAX.
G	1.70±0.10
H	8.10±0.20
I	6.10±0.10
J	1.00±0.20
K	$0.17^{+0.08}_{-0.07}$
L	0.50
M	0.10
N	0.10
P	$3^{\circ}{}^{+7^{\circ}}_{-3^{\circ}}$
T	0.25(T.P.)
U	0.60±0.15
V	0.25 MAX.
W	0.15 MAX.

NOTE

Each lead centerline is located within 0.10 mm of its true position (T.P.) at maximum material condition.

CHAPTER 35 CAUTIONS FOR WAIT

35.1 Cautions for Wait

During access to the registers assigned to the addresses between F0500H and F06FFH in the extended special function registers (2nd SFR) area, the CPU does not start the next instruction processing but becomes wait state. If this happens, the number of execution clocks of an instruction increases by the number of wait clocks.

35.2 Peripheral Hardware That Generates Wait

Table 35-1 lists the registers that issue a wait request when accessed by the CPU, and the number of CPU wait clocks.

Table 35-1. Registers That Generate Wait and Number of CPU Wait Clocks (1/2)

Peripheral Hardware	Register	Access	Number of Wait Clocks
Clock generator	PLLCTL, PER1, PER2 registers	Read	1 clock
		Write	1 clock
Port function	P20, PM20, PU20, POM20 registers	Read	1 clock
		Write	1 clock
16-bit timer KB0 to KB2	TPS2, TKBCRn0 to TKBCRn3, TKBTGCRn, TKBCTLn0, TKBCTLn1, TKBIOCn0, TKBIOCn1, TKBFLGn, TKBTRGn, TKBCLRn, TKBCRLDn0, TKBCRLDn1,TKBDNRn0, TKBDNRn1, TKBSIRn0, TKBSIRn1, TKBSSRn0, TKBSSRn1, TKBMFRn, TKBPACTLn0, TKBPACTLn1, TKBPACTLn2, TKBPAFLGn, TKBPAHFSn, TKBPAHFTn registers (n = 0 to 2)	Read	1 clock
			2 clocks (Timer KB0 to KB2: PLLCTL =0DH (PLL using, CPU = 32 MHz operation))
		Write	1 clock

Table 35-1. Registers That Generate Wait and Number of CPU Wait Clocks (2/2)

Peripheral Hardware	Register	Access	Number of Wait Clocks
16-bit timer KC0	TKCCR0, TKCDUTY00 to TKCDUTY05, TKCCTL0, TKCIOC00, TKCIOC01, TKCTOF0, TKCFLG0, TKCTRG0 registers	Read	1 clock
			2 clocks (Timer KC0: PLLCTL =0DH (PLL using, CPU = 32 MHz operation))
		Write	1 clock
Programmable gain amplifier (PGA)	PGACTL, PGAINS registers	Read	1 clock
			2 clocks (PGA: PLLCTL =0DH (PLL using, CPU = 32 MHz operation))
		Write	1 clock
Comparator	CnCTL, CVRCTL, CmRVM, CMPEGP0, CMPEGN0, CMPMON, CMPWDC, CMPSEL registers (n = 0 to 5, m = 0 to 2)	Read	1 clock
			2 clocks (Comparator: PLLCTL =0DH (PLL using, CPU = 32 MHz operation))
		Write	1 clock
Serial array unit 4 (DALI/UART4)	SDTL4, SDTH4, SDCL4, SDCH4, SDR40, SDR41, SSR40, SSR41, SIR40, SIR41, SMR40, SMR41, SCR40, SCR41, SE4, SS4, ST4, SPS4, SO4, SOE4, SOL4, SSC4, SOC4, NFEN3, SUCTL registers	Read	1 clock
		Write	1 clock
Interrupt functions	EGP2, EGN2, INTPCTL, INTMK0, INTMF0 registers	Read	1 clock
		Write	1 clock
Safety functions	IAWCTL1 register	Read	1 clock
		Write	1 clock
Other function	PIOR1, PFSEL0 registers	Read	1 clock
		Write	1 clock

APPENDIX A REVISION HISTORY

A.1 Major Revisions in This Edition

Page	Description	Classification
CHAPTER 11 WATCHDOG TIMER		
p. 443	Addition of note in **Table 11-3. Setting of Overflow Time of Watchdog Timer**	(c)
CHAPTER 14 COMPARATOR		
p. 533	Modification of **Figure 14-16. Operation Setting Flow Chart 2 of Comparator (CMP)**	(a)
CHAPTER 25 SAFETY FUNCTIONS		
p. 962	Modification of **25.1 Overview of Safety Functions**	(c)
p. 966	Modification of **25.3.2 CRC operation function (general-purpose CRC)**	(c)
p. 969	Modification of **25.3.4 RAM guard function**	(c)
p. 970	Modification of **25.3.5 SFR guard function**	(c)
CHAPTER 28 FLASH MEMORY		
p. 1011	Addition of cautions 4 in **28.7.3 Procedure for accessing data flash memory**	(c)

Remark “Classification” in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note, (d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

A.2 Revision History of Preceding Editions

Here is the revision history of the preceding editions. Chapter indicates the chapter of each edition.

(1/20)

Edition	Description	Chapter
Ver.0.02	Change of **1.1 Features**	**CHAPTER 1 OUTLINE**
	Change of **1.2 Ordering Information**	
	Change of **Figure 2-1. Pin I/O Circuit List**	**CHAPTER 2 PIN FUNCTIONS**
	Change of description in **3.1.2 Mirror area**	**CHAPTER 3 CPU ARCHITECTURE**
	Change of **Figure 3-4. Correspondence Between Data Memory and Addressing (R5F1076C, R5F107AC, R5F107BC)**	
	Change of **Figure 3-15. Outline of Table Indirect Addressing**	
	Addition of **4.2.1 Port 0** to **4.2.10 Port 20**	**CHAPTER 4 PORT FUNCTIONS**
	Change of and deletion of caution 2 of **Figure 4-23. Format of Port Mode Register (38-pin products)**	
	Change of description in **5.1 (2) Subsystem clock**	**CHAPTER 5 CLOCK GENERATOR**
	Change of description in **5.3 (2) System clock control register (CKC)**	
	Change of **Figure 5-10. Format of Peripheral Enable Register 2 (PER2)**	
	Change of description and deletion of note 4 in **5.3 (7) Peripheral enable registers 0 to 2 (PER0 to PER2)**	
	Deletion of note 4 in **Figure 5-7. Format of Peripheral Enable Register 0 (PER0)**	
	Change of description and deletion of caution in **5.3 (8) Operation speed mode control register (OSMC)**	
	Change of **5.3 (9) High-speed on-chip oscillator trimming register (HIOTRM)**	
	Change of **5.6.1 Example of setting high-speed on-chip oscillator**	
	Change of description in **5.6.2 Example of setting X1 oscillation clock**	
	Change of **6.2 (1) Timer count register mn (TCRmn)**	**CHAPTER 6 TIMER ARRAY UNIT**
	Addition of note to **Table 6-4. Interval Times Available for Operation Clock CKSm2 or CKSm3**	
	Change of caution in **6.3 (3) Timer mode register mn (TMRmn)**	
	Change of **Figure 6-8. Format of Timer Mode Register mn (TMRmn)**	
	Change of **Figure 6-25. Format of Port Mode Registers 0, 3 (PM0, PM3) (38-pin products)**	
	Change of description in **6.4 Basic Rules of Simultaneous Channel Operation Function**	
	Change of note in **Figure 6-35, Figure 6-37, Figure 6-41, Figure 6-45, Figure 6-49, Figure 6-51, Figure 6-53, Figure 6-56, Figure 6-63, Figure 6-68**	
	Change of operation clock (f_{MCK}) selection in **Figure 6-37, Figure 6-41, Figure 6-45, Figure 6-49, Figure 6-53**	
	Addition of note to **Figure 6-43, Figure 6-47**	
	Change of description in **6.7.5 Operation as delay counter**	

(2/20)

Edition	Description	Chapter
Ver.0.02	Change of all	**CHAPTER 7 16-BIT TIMERS KB0, KB1, AND KB2**
	Addition of 16-bit timer KC output pin control register (TOETKC0)	**CHAPTER 8 16-BIT TIMER KC0**
	Change of description in **8.1 Functions of 16-bit Timer KC0**	
	Change of **Figure 8-1. Block Diagram of 16-bit Timer KC0**	
	Change of **Figure 8-4. Format of Peripheral Enable Register 2 (PER2)**	
	Change of description in **8.3 (5) 16-bit timer KC output control register 01 (TKCIOC01)**	
	Addition of remark to **8.3 (9) 16-bit timer KC trigger register 0 (TKCTRG0)**	
	Change of **Figure 8-13. Format of Port Mode Registers 1, 20 (PM1, PM20) (38-Pin Products)**	
	Change of **8.4 Operation of 16-bit Timer KC0**	
	Change of figure in **Figure 9-1. Block Diagram of Real-time Clock**	**CHAPTER 9 REAL-TIME CLOCK**
	Deletion of caution of **9.3 (1) Peripheral enable register 0 (PER0)**	
	Change of **Figure 9-5. Format of Real-time Clock Control Register 1 (RTCC1) (2/2)**	
	Addition of caution 2 to **Figure 9-20. Procedure for Writing Real-time Clock**	
	Change of caution 4 and deletion of caution 5 in **11.4.1 Controlling operation of watchdog timer** Deletion of caution of **Table 11-3. Setting of Overflow Time of Watchdog Timer**	**CHAPTER 11 WATCHDOG TIMER**
	Deletion of caution 1 and change of remark in **Table 11-4. Setting Window Open Period of Watchdog Timer**	
	Change of the internal reference voltage	**CHAPTER 12 A/D CONVERTER**
	Change of **Figure 12-1. Block Diagram of A/D Converter**	
	Change of description in **12.2 Configuration of A/D Converter**	
	Addition of note to **Table 12-1. Settings of ADCS and ADCE Bits** Change of **Table 12-2. Setting and Clearing Conditions for ADCS Bit**	
	Change of **Table 12-3. A/D Conversion Time Selection**	
	Change of **Figure 12-7. Format of A/D Converter Mode Register 2 (ADM2)**	
	Change of **Figure 12-8. ADRCK Bit Interrupt Signal Generation Range**	
	Change of and deletion of note of **Figure 12-17. Formats of Port Mode Registers 0, 2, 12, and 14 (PM0, PM2, PM12, PM14)**	
	Change of **12.7.4 Setup when using temperature sensor (example for software trigger mode and one-shot conversion mode)**	
	Addition of Comparator internal reference voltage control register (CVRCTL)	**CHAPTER 13 PROGRAMMABLE GAIN AMPLIFIER**
	Change of **Table 13-1. Configuration of Programmable Gain Amplifier**	
	Change of caution in **Figure 13-3. Format of Programmable Gain Amplifier Control Register (PGACTL)**	
	Addition of caution 2 to **Figure 13-5. Format of Comparator Input Switch Control Register (CMPSEL)**	
	Change of note in **Figure 13-6. Format of A/D Port Configuration Register (ADPC)**	
	Change of caution in **13.3 (7) Port mode registers 0, 2, 14 (PM0, PM2, PM14)** Change of and deletion of note of **Figure 13-8. Format of Port Mode Registers 0, 2, 14 (PM0, PM2, PM14) (38-Pin Products)**	
	Change of description in **13.4 Operation of Programmable Gain Amplifier**	
	Addition of **13.5 Setting Procedure of Programmable Gain Amplifier**	

(3/20)

Edition	Description	Chapter
Ver.0.02	Addition of description of **CHAPTER 14 COMPARATOR**	**CHAPTER 14 COMPARATOR**
	Change of **Table 14-1. Configuration of Comparator**	
	Change of caution 2 in **Figure 14-4. Format of Comparator Internal Reference Voltage Control Register (CVRCTL)**	
	Change of note in **Figure 14-11. Format of A/D Port Configuration Register (ADPC)**	
	Change of **Figure 14-14. Format of Port Mode Registers 0, 2, 14 (PM0, PM2, PM14)**	
	Addition of **14.4 Setting Procedure of Comparator**	
	Change of caution 2 in **Figure 15-4. Format of Peripheral Enable Register 0 (PER0)**	**CHAPTER 15 SERIAL ARRAY UNIT 0**
	Change of **Figure 15-5. Format of Serial Clock Select Register m (SPSm)**	
	Change of caution 2 in **Figure 15-11. Format of Serial Channel Start Register m (SSm)**	
	Change of description in **15.3 (13) Serial output level register m (SOLm)**	
	Addition of caution to **15.3 (14) Serial standby control register 0 (SSC0)**	
	Change of **Figure 15-22. Format of Port Mode Registers 0 and 1 (PM0 and PM1)**	
	Addition of peripheral I/O redirection register (PIOR1)	**CHAPTER 16 SERIAL ARRAY UNIT 4 (DALI/UART4)**
	Change of description in **16.1 (1) Asynchronous serial communication (UART) mode**	
	Change of **Figure 16-8. Format of Serial Communication Operation Setting Register 4n (SCR4n)**	
	Addition of note 3 in **Figure 16-9. Format of Serial Data Register 4n (SDR4n)**	
	Change of **Figure 16-23. Format of Port Mode Register 1, 20 (PM1, PM20) (38-pin products)**	
	Change of caution 1 and addition of note 3 to **16.7 SNOOZE Mode Function (Only DALI/UART4 Reception)**	
	Addition of remark 2 in **16.9 Calculating Baud Rate**	
	Change of **Figure 17-7. Format of IICA Status Register 0 (IICS0)**	**CHAPTER 17 SERIAL INTERFACE IICA**
	Change of **Figure 17-12. Format of Port Mode Register 1 (PM1)**	
	Change of description in **Figure 17-33. Example of Slave to Master Communication (8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (1/3)**	
	Change of **Figure 18-6. Timing Diagram of Multiplication (Unsigned) Operation (2 × 3 = 6)**	**CHAPTER 18 MULTIPLIER AND DIVIDER/MULTIPLY-ACCUMULATOR**
	Change of description in **18.4.3 Multiply-accumulation (unsigned) operation**	
	Change of **Figure 18-8. Timing Diagram of Multiply-Accumulation (Unsigned) Operation (2 × 3 + 3 = 9 → 32767 × 2 + 4294901762 = 0 (over flow generated))**	
	Change of description in **18.4.4 Multiply-accumulation (signed) operation**	
	Change of **Figure 18-9. Timing Diagram of Multiply-Accumulation (signed) Operation (2 × 3 + (–4) = 2 → 32767 × (–1) + (–2147483647) = –2147450882 (overflow occurs.))**	
	Change of (4) and addition of (7) to **19.6 Cautions on Using DMA Controller**	**CHAPTER 19 DMA CONTROLLER**
	Change of **Table 20-1. Interrupt Source List**	**CHAPTER 20 INTERRUPT FUNCTION**

(4/20)

Edition	Description	Chapter
Ver.0.02	Change of description and deletion caution 3 in **CHAPTER 22 RESET FUNCTION**	**CHAPTER 22 SAFETY FUNCTIONS**
	Change of **Table 22-2. Hardware Statuses After Reset Acknowledgment (1/4)** and change of note 2	
	Change of values of LVIM, LVIS of note 2 in **Table 22-2. Hardware Statuses After Reset Acknowledgment (4/4)**	
	Change of **Figure 23-3. Example of Software Processing After Reset Release**	**CHAPTER 23 POWER-ON-RESET CIRCUIT**
	Change of description in **24.1 Functions of Voltage Detector**	**CHAPTER 24 VOLTAGE DETECTOR**
	Change of note 2 and addition of notes 3, 4 to **Figure 24-2. Format of Voltage Detection Register (LVIM)**	
	Change of **Figure 24-3. Format of Voltage Detection Level Select Register (LVIS)**	
	Change of **Table 24-1. LVD Operation Mode and Detection Voltage Settings for User Option Byte (000C1H/010C1H)**	
	Change of description in **24.4.1 When used as reset mode**	
	Change of **Figure 24-4. Timing of Voltage Detector Internal Reset Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 1)**	
	Change of description in **24.4.2 When used as interrupt mode**	
	Change of **Figure 24-5. Timing of Voltage Detector Internal Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 0, 1)**	
	Change of description in **24.4.3 When used as interrupt and reset mode**	
	Change of **Figure 24-6. Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0)**	
	Change of **Figure 24-8. Delay from the time LVD reset source is generated until the time LVD reset has been generated or released**	
	Change of all	**CHAPTER 25 SAFETY FUNCTIONS**
	Change of **26.1 Regulator Overview** and **Table 26-1. Regulator Output Voltage Conditions**	**CHAPTER 26 REGULATOR**
	Change of description in **27.1.1 User option byte (000C0H to 000C2H/010C0H to 010C2H)**	**CHAPTER 27 OPTION BYTE**
	Change of caution in **Figure 27-1. Format of User Option Byte (000C0H/010C0H)**	
	Change of **Figure 27-2. Format of User Option Byte (000C1H/010C1H)**	
	Change of description and note in **Table 28-1. Wiring Between RL78/I1A and Dedicated Flash Memory Programmer**	**CHAPTER 28 FLASH MEMORY**
	Change of **28.1.2 Communication Mode** Change of **Figure 28-2. Communication with Dedicated Flash Memory Programmer**	
	Change of description and note in **Table 28-2. Pin Connection**	
	Change of **28.2.1 P40/TOOL0 pin**	
	Change of description in **28.3.1 Data flash overview**	
	Change of description in **28.4.2 Flash memory programming mode**	
	Change of **Table 28-8. Setting Security in Each Programming Mode**	
	Change of **Table 28-9. Relationship between Flash Shield Window Function Setting/Change Methods and Commands**	
	Change of **Figure 29-2. Memory Spaces Where Debug Monitor Programs Are Allocated**	**CHAPTER 29 ON-CHIP DEBUG FUNCTION**

(5/20)

Edition	Description	Chapter
Ver.0.02	Change of **34.2 Absolute Maximum Ratings**	**CHAPTER 32 ELECTRICAL SPECIFICATIONS**
	Addition of note to **32.3.4 Recommended Oscillator Constants**	
	Addition of **32.6.2 Serial array unit 4 (DALI)**	
	Change of **32.10 Timing Specs for Switching Modes**	
	Change of **33.1 20-pin products**	**CHAPTER 33 PACKAGE DRAWINGS**
	Change of **33.3 32-pin products**	
Ver.1.00	Renamed interval timer (unit) to 12-bit interval timer	Though out
	Renamed V_{LVI}, V_{LVIH}, V_{LVIL} to V_{LVD}, V_{LVDH}, V_{LVDL} (LVD detection voltage)	
	Renamed interrupt source of RAM parity error (RAMTOP) to RPE	
	Renamed f_{EXS} to f_{EXT}	
	Addition of **Figure 1-1. Part Number, Memory Size, and Package of RL78/I1A**	**CHAPTER 1 OUTLINE**
	Change of **1.6 Outline of Functions**	
	Addition of **caution** and **remark** to **2.2.1 With functions for each product**	**CHAPTER 2 PIN FUNCTIONS**
	Addition of **Remark to 2.3 Pin I/O Circuits and Recommended Connection of Unused Pins**	
	Addition of **caution** to **Figure 3-1. Memory Map (R5F1076C, R5F107AC, R5F107BC)**	**CHAPTER 3 CPU ARCHITECTURE**
	Change of **note** and addition of **caution** in **Figure 3-2. Memory Map (R5F107AE, R5F107DE)**	
	Change of **caution 2** in **3.1.3 Internal data memory space**	
	Addition of **caution** to **Figure 3-4. Correspondence Between Data Memory and Addressing (R5F1076C, R5F107AC, R5F107BC)**	
	Change of **note** and addition of **caution** in **Figure 3-5. Correspondence Between Data Memory and Addressing (R5F107AE, R5F107DE)**	
	Change of **caution 3** in **3.2.1 (3) Stack pointer (SP)**	
	Change of **caution 2** in **3.2.2 General-purpose registers**	
	Addition of **note** to **Table 3-6. Extended SFR (2nd SFR) List (7/12)**	
	Change of **4.2 Port Configuration** Addition of register setting table when using each port Change of block diagrams of each port	**CHAPTER 4 PORT FUNCTIONS**
	Change of description in **4.2.3 Port 2**	
	Addition of **caution** to **4.3 Registers Controlling Port Function**	
	Change of **Figure 4-24. Format of Port Register (38-pin products)**	
	Change of description and addition of **caution** in **4.3 (3) Pull-up resistor option registers (PUxx)**	
	Change of **4.3 (5) Port output mode registers (POMxx)**	
	Addition of **cautions 1** and **2** to **Figure 4-28. Format of Port Mode Control Register**	
	Change of description in **4.3 (8) Peripheral I/O redirection register (PIOR1)**	
	Change of **4.4.4 Connecting to external device with different potential (2.5 V, 3 V)**	
	Change of **4.5 Settings of Port Mode Register, and Output Latch When Using Alternate Function**	
	Addition of **4.6.2 Notes on specifying the pin settings**	

(6/20)

Edition	Description	Chapter
Ver.1.00	Change of **Figure 5-1. Block Diagram of Clock Generator**	**CHAPTER 5 CLOCK GENERATOR**
	Change of **caution** in **Figure 5-2. Format of Clock Operation Mode Control Register (CMC)**	
	Change of **note 3** in **Figure 5-16. Clock Generator Operation When Power Supply Voltage Is Turned On**	
	Change of **Figure 5-17. CPU Clock Status Transition Diagram**	
	Table 5-3. CPU Clock Transition and SFR Register Setting Examples Change of **(2) CPU operating with high-speed system clock (C) after reset release (A)**	
	Change of description in **Table 5-4. Changing CPU Clock**	
	Change of **Table 5-6. Maximum Number of Clocks Required for $f_{IH} \leftrightarrow f_{MX}$**	
	Change of **Table 5-7. Maximum Number of Clocks Required for $f_{MAIN} \leftrightarrow f_{SUB}$**	
	Change of **6.1.1 (6) Delay counter**	**CHAPTER 6 TIMER ARRAY UNIT**
	Change of **Figure 6-5. Format of Timer Data Register 0n (TDR0n) (n = 1, 3)**	
	Change of **Figure 6-7. Format of Timer Clock Select register 0 (TPS0)**	
	Change of **Figure 6-8. Format of Timer Mode Register 0n (TMR0n)**	
	Addition of **caution** to **Figure 6-13. Format of Timer Input Select register 0 (TIS0)**	
	Change of description in in **Figure 6-14. Format of Timer Output Enable register 0 (TOE0)**	
	Addition of **6.5 Operation of Counter**	
	Change of **Figures 6-30** to **6-33**	
	Change of description in **Figures 6-38, 6-42, 6-46, 6-50, 6-54, 6-59 Example of Set Contents of Registers**	
	Change of **Figures 6-40, 6-44, 6-48, 6-52, 6-56 Block Diagram**	
	Change of **Figures 6-47, 6-51, 6-60 Operation Procedure**	
	Change of **7.1 Functions of 16-bit Timers KB0, KB1, and KB2**	**CHAPTER 7 16-BIT TIMERS KB0, KB1, AND KB2**
	Addition of **Figures 7-21** to **7-23**	
	Addition of **Tables 7-3** to **7-5**	
	Addition of **7.8 Operation of Forced Output Stop Function 2**	
	Change of **Figure 8-1. Block Diagram of 16-bit Timer KC0**	**CHAPTER 8 16-BIT TIMERS KC0**
	Addition of **Figure 8-14. Timer KC operation setting example (operation start flow)**	
	Addition of **Figure 8-15. Timer KC operation setting example (operation stop flow)**	
	Addition of **Figure 8-16. Timer KC operation setting example (Compare register batch overwrite flow)**	
	Change of description in **9.3 (5)** to **9.3 (11)**	**CHAPTER 9 REAL-TIME CLOCK**
	Change of **9.4.2 Shifting to HALT/STOP mode after starting operation**	
	Addition of **caution 3** to **Figure 10-4. Format of Interval Timer Control Register (ITMC)**	**CHAPTER 10 12-BIT INTERVAL TIMER**
	Change of **Figure 10-5. 12-bit Interval Timer Operation Timing (ITMCMP11 to ITMCMP0 = 0FFH, count clock: f_{SUB} = 32.768 kHz)**	
	Change of description in **11.1 Functions of Watchdog Timer**, **11.4.3 Setting window open period of watchdog timer**, **11.4.4 Setting watchdog timer interval interrupt**	**CHAPTER 11 WATCHDOG TIMER**

(7/20)

Edition	Description	Chapter
Ver.1.00	Change of **Figure 12-1. Block Diagram of A/D Converter**	**CHAPTER 12 A/D CONVERTER**
	Deletion of **note 2** and addition of **cautions 1** and **2** in **Figure 12-3. Format of A/D Converter Mode Register 0 (ADM0)**	
	Change of description and addition of **note 2** and **caution 4** in **Figure 12-4. Timing Chart When A/D Voltage Comparator Is Used**	
	Change of **Table 12-3. A/D Conversion Time Selection**	
	Change of **cautions 2** and addition of **cautions 3** in **Figure 12-6. Format of A/D Converter Mode Register 1 (ADM1)**	
	Change of **Figure 12-7. Format of A/D Converter Mode Register 2 (ADM2)**	
	Addition of **note** to **12.3 (5) 10-bit A/D conversion result register (ADCR)**, and **12.3 (6) 8-bit A/D conversion result register (ADCRH)**	
	Addition of **note** and **cautions 10, 11** to **Figure 12-11. Format of Analog Input Channel Specification Register (ADS)**	
	Addition of **caution 3** to **12. 3 (11) A/D port configuration register (ADPC)**	
	Addition of **caution** to **12.3 (12) Port mode control registers 0, 12, 14 (PMC0, PMC12, PMC14)**	
	Addition of **Caution 2** to **12. 3 (13) Port mode register 0, 2, 12, 14 (PM0, PM2, PM12, PM14)**	
	Addition of **note 1** to **12.4 A/D Converter Conversion Operations**	
	Change of **Figures 12-32** to **12-36**	
	Change of description in **12.8 SNOOZE Mode Function**	
	Addition of **caution** to **12.10 (2) Input range of ANI0 to ANI2, ANI4 to ANI7, and ANI16 to ANI19 pins**	
	Change of description in **12.10 (5) Analog input (ANIn) pins**	
	Change of **Table 12-6. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)**	
	Change of **Figure 14-1. Block Diagram of Comparator**	**CHAPTER 14 COMPARATOR**
	Change of all	**CHAPTER 15 SERIAL ARRAY UNIT 0**
	Change of **Figure 16-18. Format of Serial Standby Control Register 4 (SSC4)**	**CHAPTER 16 SERIAL ARRAY UNIT 4 (DALI/UART4)**
	Change of **16.1.3 (18) Port output mode register 1, 20 (POM1, POM20)**	
	Change of **Figure 16-29. Procedure for Stopping UART Transmission**	
	Change of **Figure 16-30. Procedure for Resuming UART Transmission**	
	Change of **Figure 16-32. Flowchart of UART Transmission (in Single-Transmission Mode)**	
	Change of **Figure 16-33. Timing Chart of UART Transmission (in Continuous Transmission Mode)**	
	Change of **Figure 16-34. Flowchart of UART Transmission (in Continuous Transmission Mode)**	
	Change of **Figure 16-37. Procedure for Stopping UART Reception**	
	Change of **Figure 16-38. Procedure for Resuming UART Reception**	
	Change of **Figure 16-39. Timing Chart of UART Reception**	
	Change of **Figure 16-40. Flowchart of UART Reception**	
	Addition of **16.11 DMX512 Communication Operation**	

(8/20)

Edition	Description	Chapter
Ver.1.00	Addition of **caution** to **Figure 17-9. Format of IICA Control Register 01 (IICCTL01) (2/2)**	**CHAPTER 17 SERIAL INTERFACE IICA**
	Change of description in **17.5.13 Wakeup function**	
	Change of **Figure 17-29, 17-30, 17-31**	
	Change of **17.5.17 (2) (d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop**	
	Change of **17.5.17 (3) (d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop**	
	Change of **Figure 18-1. Block Diagram of Multiplier and Divider/Multiply-Accumulator**	**CHAPTER 18 MULTIPLIER AND DIVIDER/MULTIPLY-ACCUMULATOR**
	Change of **caution 1** in **18. 2 (2) Multiplication/division data register B (MDBL, MDBH)**	
	Change of **caution 2** in **18. 2 (3) Multiplication/division data register C (MDCL, MDCH)**	
	Change of description in **Figure 18-5. Format of Multiplication/Division Control Register (MDUC)**	
	Change of value in **Figure 18-6. Timing Diagram of Multiplication (Unsigned) Operation (2 × 3 = 6)**	
	Change of value in **Figure 18-7. Timing Diagram of Multiplication (Signed) Operation (–2 × 32767 = –65534)**	
	Change of value in **Figure 18-8. Timing Diagram of Multiply-Accumulation (Unsigned) Operation (2 × 3 + 3 = 9 → 32767 × 2 + 4294901762 = 0 (over flow generated))**	
	Change of value in **Figure 18-9. Timing Diagram of Multiply-Accumulation (signed) Operation (2 × 3+(–4) = 2 → 32767 × (–1) + (–2147483647) = –2147450882 (overflow occurs.))**	
	Change of description in **18.4.5 Division operation**	
	Change of **Figure 18-10. Timing Diagram of Division Operation (Example: 35 ÷ 6 = 5, Remainder 5)**	
	Addition of **Table 19-2. Response Time of DMA Transfer**	**CHAPTER 19 DMA CONTROLLER**
	Addition of description	**CHAPTER 20 INTERRUPT FUNCTION**
	Change of **Table 20-1. Interrupt Source List (3/3)**	
	Addition of **note** to **Figure 20-8. Interrupt Monitor Flag Register 0 (INTMF0) (38-pin)**	
	Change of description and addition of **note** in **Table 20-4. Time from Generation of Maskable Interrupt Until Servicing**	
	Change of **Figure 20-11. Interrupt Request Acknowledgment Timing (Minimum Time)** and **Figure 20-12. Interrupt Request Acknowledgment Timing (Maximum Time)**	
	Change of **Table 20-5. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing**	
	Change of **20.4.4 Interrupt request hold**	

(9/20)

Edition	Description	Chapter
Ver.1.00	Change of **Figure 21-2. Format of Oscillation Stabilization Time Select Register (OSTS)**	**CHAPTER 21 STANDBY FUNCTION**
	Change of **Table 21-1. Operating Statuses in HALT Mode**	
	Change of **note** in **Figure 21-3. HALT Mode Release by Interrupt Request Generation**	
	Change of **Figure 21-4. HALT Mode Release by Reset**	
	Change of description in **Table 21-2. Operating Statuses in STOP Mode**	
	Change of **note** in **Figure 21-5. STOP Mode Release by Interrupt Request Generation**	
	Change of **note** in **Figure 21-6. STOP Mode Release by Reset**	
	Change of description in **21.2.3 (1) SNOOZE mode setting and operating statuses**	
	Change of **Figures 22-2** to **22-4**	**CHAPTER 22 RESET FUNCTION**
	Change of **Table 22-1. Operation Statuses During Reset Period**	
	Change of **note 2** in **Table 22-2. Hardware Statuses After Reset Acknowledgment**	
	Addition of **cautions 2** and **3** to **Figure 22-5. Format of Reset Control Flag Register (RESF)**	
	Change of description and **notes** in **Figure 23-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector**	**CHAPTER 23 POWER-ON-RESET CIRCUIT**
	Change of **Figure 24-1. Block Diagram of Voltage Detector**	**CHAPTER 24 VOLTAGE DETECTOR**
	Addition of figure to **Table 24-1. LVD Operation Mode and Detection Voltage Settings for User Option Byte (000C1H)**	
	Change of **Figure 24-4. Timing of Voltage Detector Internal Reset Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 1)**	
	Change of **Figure 24-5. Timing of Voltage Detector Internal Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 0, 1)**	
	Change of **24.4.3 When used as interrupt and reset mode**	
	Change of **remark** in **25.1 Overview of Safety Functions**	**CHAPTER 25 SAFETY FUNCTIONS**
	Addition of description and **caution** to **25.3.1 Flash memory CRC operation function (high-speed CRC)**	
	Addition of description and **caution** to **25.3.2 CRC operation function (general-purpose CRC)**	
	Change of **Figure 25-6. CRC Operation Function (General-Purpose CRC)**	
	Change of **caution** in **Figure 25-7. Format of RAM Parity Error Control Register (RPECTL)**	
	Addition of **remark** to **Figure 25-12. Format of Invalid Memory Access Detection Control Register 0 (IAWCTL0)**	
	Addition of description to **25.3.8 A/D test function**	
	Change of **Figure 25-16. Format of A/D Test Register (ADTES)**	
	Addition of **note** to **Figure 25-17. Format of Analog Input Channel Specification Register (ADS)**	
	Change of description in **Figure 27-1. Format of User Option Byte (000C0H/010C0H)**	**CHAPTER 27 OPTION BYTE**
	Change of **Figure 27-2. Format of User Option Byte (000C1H/010C1H)**	

(10/20)

Edition	Description	Chapter
Ver.1.00	Change of **note** in **Table 28-1. Wiring Between RL78/I1A and Dedicated Flash Memory Programmer**	**CHAPTER 28 FLASH MEMORY**
	Change of description in **28.1.1 Programming Environment**	
	Change of **note** in **Figure 28-2. Communication with Dedicated Flash Memory Programmer**	
	Change of **note** in **Table 28-2. Pin Connection**	
	Addition of **remark** to **28.2 Connection of Pins on Board**	
	Change of description in **28.3.1 Data flash overview**	
	Addition of **28.4.5 Description of signature data**	
	Addition of description and **caution** to **28.5 Security Settings**	
	Addition of **caution 3** to **28.6 Flash Memory Programming by Self-Programming**	
	Change of all	**CHAPTER 31 INSTRUCTION SET**
	Change of all	**CHAPTER 32 ELECTRICAL SPECIFICATIONS**
	Change of **33.2 30-pin products** and **33.4 38-pin products**	**CHAPTER 33 PACKAGE DRAWINGS**
Ver.2.10	Deletion of 32-pin product	**Though out**
	Renamed f_{EXT} to f_{EXS}	
	Modification of **1.1 Features**	**CHAPTER 1 OUTLINE**
	Modification of **1.2 Ordering Information**	
	Modification of **Figure 1-1. Part Number, Memory Size, and Package of RL78/I1A**	
	Modification of **1.4 Pin Identification**	
	Modification of **1.6 Outline of Functions**	
	Modification of **2.2.1 With functions for each product**	**CHAPTER 2 PIN FUNCTIONS**
	Modification of **3.1 Memory Space**	**CHAPTER 3 CPU ARCHITECTURE**
	Modification of **caution** to **Figure 3-1** and **Figure 3-2**	
	Deletion of **caution 2** to **Figure 3-3. Format of Configuration of Processor Mode Control Register (PMC)**	
	Modification of **3.1.3 Internal data memory space**	
	Modification of **3.1.6 Data memory addressing**	
	Modification of **caution** to **Figure 3-4** and **Figure 3-5**	
	Modification of **3.2.1 Control registers**	
	Modification of **3.2.2 General-purpose registers**	
	Modification of **3.2.3 ES and CS registers**, and addition of **Figure 3-12. Extension of Data Area Which Can Be Accessed**	
	Modification of **3.2.4 Special function registers (SFRs)**	
	Modification of **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**	
	Modification of **Table 3-6. Extended SFR (2nd SFR) List**	

(11/20)

Edition	Description	Chapter
Ver.2.10	Modification of **3.4.1 Implied addressing**	**CHAPTER 3 CPU ARCHITECTURE**
	Modification of **3.4.3 Direct addressing**	
	Modification of **3.4.5 SFR addressing**	
	Modification of **Figure 3-24** to **Figure 3-34**	
	Modification of **3.4.7 Based addressing**	
	Modification of **3.4.9 Stack addressing**	
	Addition of **Figure 3-35** to **Figure 3-40**	
	Modification of **caution** to **4.3 Registers Controlling Port Function**	**CHAPTER 4 PORT FUNCTIONS**
	Modification of **Figure 5-1. Block Diagram of Clock Generator**	**CHAPTER 5 CLOCK GENERATOR**
	Modification of **caution 7** to **Figure 5-2. Format of Clock Operation Mode Control Register (CMC)**	
	Modification of **cautions 1** to **3** to **Figure 5-12. Format of High-speed On-chip Oscillator Frequency Select Register (HOCODIV)**	
	Modification of **5.4.2 XT1 oscillator**	
	Modification of **caution** to **Figure 5-15. Example of External Circuit of XT1 Oscillator**	
	Modification of **Figure 5-17. Clock Generator Operation When Power Supply Voltage Is Turned On**	
	Modification of **5.6.1 Example of setting high-speed on-chip oscillator**	
	Addition of **5.7 Resonator and Oscillator Constants**	
	Modification of **6.2.2 Timer data register 0n (TDR0n)**	**CHAPTER 6 TIMER ARRAY UNIT**
	Modification of **Figure 6-8. Figure 6-8. Format of Timer Mode Register 0n (TMR0n)**	
	Modification of **Figure 6-25. Operation Timing (In Capture Mode: Input Pulse Interval Measurement)**	
	Modification of **6.8.1 Operation as one-shot pulse output function**	
	Modification of **Figure 7-6. Format of 16-bit Timer KB Operation Control Register n0 (TKBCTLn0)**	**CHAPTER 7 16-BIT TIMERS KB0, KB1, AND KB2**
	Modification of **remark** to **Table 7-5. External Trigger Assignment List of Forced Output Stop Function 2**	
	Modification of **Figure 9-23. Correction Operation When (DEV, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 0, 1, 1, 1, 0)**	**CHAPTER 9 REAL-TIME CLOCK**
	Modification of **Figure 10-1. Block Diagram of 12-bit Interval Timer**	**CHAPTER 10 12-BIT INTERVAL TIMER**
	Modification of **Figure 10-2. Format of Peripheral Enable Register 0 (PER0)**	
	Modification of **10.3.2 Operation speed mode control register (OSMC)**	
	Modification of **Figure 10-5. 12-bit Interval Timer Operation Timing**	
	Modification of **Figure 12-1. Block Diagram of A/D Converter**	**CHAPTER 12 A/D CONVERTER**
	Modification of **12.2 (10) AV_{REFP} pin**	
	Modification of **caution 1** to **Figure 12-2. Format of Peripheral Enable Register 0 (PER0)**	
	Modification of **cautions 1** and **2** to **Figure 12-3. Format of A/D Converter Mode Register 0 (ADM0)**	

(12/20)

Edition	Description	Chapter
Ver.2.10	Modification of **Table 12-1. Settings of ADCS and ADCE Bits**	**CHAPTER 12 A/D CONVERTER**
	Modification of **Table 12-3. A/D Conversion Time Selection**	
	Modification of **caution 1** to **Figure 12-6. Format of A/D Converter Mode Register 1 (ADM1)**	
	Modification of **caution** to **Figure 12-7. Format of A/D Converter Mode Register 2 (ADM2)**	
	Modification of **caution 5** to **Figure 12-11. Format of Analog Input Channel Specification Register (ADS)**	
	Modification of **Figure 12-22**, **Figure 12-23**, **Figure 12-26**, **Figure 12-27**, **Figure 12-30**, and **Figure 12-31**	
	Modification of **12.7.4 Setup when temperature sensor output/internal reference voltage output is selected (example for software trigger mode and one-shot conversion mode)**	
	Modification of **Figure 12-48. Internal Equivalent Circuit of ANIn Pin** and **Figure 12-6. Format of A/D Converter Mode Register 1 (ADM1)**	
	Addition of **caution** to **Figure 14-12. Format of Peripheral Function Switch Register 0 (PFSEL0)**	**CHAPTER 14 COMPARATOR**
	Modification of **Figure 14-15. Operation Setting Flow Chart 1 of Comparator (CMP)** **(Using Timer Forced Output Stop Request Signal by INTCMPn, CMPn)**	
	Modification of **15.6.4 (2) Baud rate error during transmission**	**CHAPTER 15 SERIAL ARRAY UNIT 0**
	Addition of **caution** to **16.1 (1) Asynchronous serial communication (UART) mode**	**CHAPTER 16 SERIAL ARRAY UNIT 4 (DALI/UART4)**
	Modification of **16.1 (2) DALI mode**	
	Modification of **16.2.3 DALI transmit data registers H4, L4 (SDTH4, SDTL4)**	
	Modification of **16.2.4 DALI receive data registers H4, L4 (SDCH4, SDCL4)**	
	Modification of **Figure 16-8. Format of Serial Communication Operation Setting Register 4n (SCR4n)**	
	Modification of **16.3.14 Serial standby control register 4 (SSC4)**	
	Modification of **Figure 16-27. Example of Contents of Registers for UART Transmission**	
	Modification of **16.6 DALI Mode**	
	Modification of **16.6.1 DALI transmission**	
	Modification of **Figure 16-46. Example of Contents of Registers for DALI Transmission**	
	Modification of **16.6.2 DALI reception**	
	Modification of **Figure 16-52. Example of Contents of Registers for DALI Reception**	
	Modification of **16.7 STOP & HALT Mode Function (Only DALI/UART4 Reception)**	
	Modification of **16.11 DMX512 Communication Operation**	

(13/20)

Edition	Description	Chapter
Ver.2.10	Modification of **Figure 18-5. Format of Multiplication/Division Control Register (MDUC)**	**CHAPTER 18 MULTIPLIER AND DIVIDER/MULTIPLY-ACCUMULATOR**
	Modification of **Figure 18-9. Timing Diagram of Multiply-Accumulation (signed) Operation (2 × 3 + (–4) = 2 → 32767 × (–1) + (–2147483647) = –2147450882 (Overflow Occurs.))**	
	Addition of **note 3** to **Table 20-1. Interrupt Source List**	**CHAPTER 20 INTERRUPT FUNCTIONS**
	Modification of **caution** to **21.1 Standby function**	**CHAPTER 21 STANDBY FUNCTION**
	Modification of **Table 21-1. Operating Statuses in HALT Mode**	
	Modification of **note** to **Figure 21-4. HALT Mode Release by Reset**	
	Modification of **caution 1** to **21.3.2 (1)STOP mode setting and operating statuses**	
	Modification of **note 2** and addition of **remark 2** to **Figure 21-5. STOP Mode Release by Interrupt Request Generation**	
	Addition of **note** to **Figure 21-6. STOP Mode Release by Reset**	
	Modification of **21.3.3 (1) SNOOZE mode setting and operating statuses**	
	Modification of **note 3** to **CHAPTER 22 RESET FUNCTION**	**CHAPTER 22 RESET FUNCTION**
	Modification of and addition of **note** to **Figure 22-2** and **Figure 22-3**	
	Modification of **note** to **Figure 22-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input**	
	Modification of **Table 22-1. Operation Statuses During Reset Period**	
	Modification of **Table 22-2. Hardware Statuses After Reset Acknowledgment**	
	Modification of **caution 2** to **Figure 22-5. Format of Reset Control Flag Register (RESF)**	
	Modification of **Figure 23-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector**	**CHAPTER 23 POWER-ON-RESET CIRCUIT**
	Modification of **25.1 (2) RAM parity error detection function**	**CHAPTER 25 SAFETY FUNCTIONS**
	Modification of **caution** to **Figure 25-7. Format of RAM Parity Error Control Register (RPECTL)**	
	Modification of **27.1 Functions of Option Bytes**	**CHAPTER 27 OPTION BYTE**
	Modification of **caution** to **Figure 27-1. Format of User Option Byte (000C0H/010C0H)**	
	Modification of **caution** to **Figure 27-3. Format of Option Byte (000C2H/010C2H)**	
	Modification of **28.2.1 P40/TOOL0 pin**	**CHAPTER 28 FLASH MEMORY**
	Modification of description and **caution** to **28.3.1 Data flash overview**	
	Modification of **28.4.2 Flash memory programming mode**	
	Modification of **caution 4** to **28.6 Flash Memory Programming by Self-Programming**	
	Modification of **error correction** to **Table 31-6. Operation List (14/17)**	**CHAPTER 31 INSTRUCTION SET**

(14/20)

Edition	Description	Chapter
Ver.2.10	Addition of **note** to **32.1 Absolute Maximum Ratings**	**CHAPTER 32 ELECTRICAL SPECIFICATIONS (T_A = –40 to +105°C)**
	Modification of **32.3.2 Supply current characteristics**	
	Deletion of **note 2** to **32.5.1 (7) DALI/UART4 mode**	
	Deletion of **note 2** to **32.6.1 A/D converter characteristics (1), (2), (3) Overall error**	
	Renamed to **32.6.2 Temperature sensor/internal reference voltage characteristics**	
	Modification of description to **32.6.2 Temperature sensor/internal reference voltage characteristics**	
	Modification of **32.6.3 Programmable gain amplifier** and **32.6.4 Comparator**	
	Addition of **32.6.7 Supply voltage rise inclination characteristics**	
	Modification of **32.7 Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics** and **32.8 Flash Memory Programming Characteristics**	
	Modification of description to **32.9 Timing Specs for Switching Flash Memory Programming Modes**	
	Addition of this chapter	**CHAPTER 33 ELECTRICAL SPECIFICATIONS (T_A = –40 to +125°C)**
	Modification of **34.1 20-pin Products**	**CHAPTER 34 PACKAGE DRAWINGS**
	Modification of **Table 35-1. Registers That Generate Wait and Number of CPU Wait Clocks**	**CHAPTER 35 CAUTIONS FOR WAIT**
Ver.3.10	Modification of descriptions	**CHAPTER 1 OUTLINE**
	Modification of description in **1.3.1 20-pin products**	
	Modification of description in **1.3.2 30-pin products**	
	Modification of description in **1.6 Outline of Functions**	
	Modification of descriptions	**CHAPTER 2 PORT FUNCTIONS**
	Modification of description in **2.2.2 Description of functions**	
	Modification of descriptions	**CHAPTER 4 PIN FUNCTIONS**
	Modification of description in **4.2.3 Port 2**	
	Modification of **Table 4-5. Setting Examples of Registers and Output Latches When Using Pin Function**	
	Modification of **Table 4-5. Setting Examples of Registers and Output Latches When Using Pin Function**	
	Modification of descriptions	**CHAPTER 5 CLOCK GENERATOR**
	Addition of description to **Figure 5-4. Format of Clock Operation Status Control Register (CSC)**	
	Modification of description in **5.1 Functions of Clock Generator**	
	Addition of **caution** to **5.6.2 Example of setting X1 oscillation clock**	

(15/20)

Edition	Description	Chapter
Ver.3.10	Modification of descriptions	**CHAPTER 6 TIMER ARRAY UNIT**
	Modification of **6.1.2 (2) PWM (Pulse Width Modulation) output**	
	Modification of description in **6.3.2 Timer clock select register 0 (TPS0)**	
	Modification of **Figure 6-10. Format of Timer Mode Register 0n (TMR0n) (1/4)**	
	Modification of **6.4.1 Basic rules of simultaneous channel operation function**	
	Modification of description in **6.4.2 8-bit timer operation function overview (only channels 1 and 3)**	
	Modification of **Figure 6-42. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (1/2)**	
	Modification of descriptions	**CHAPTER 7 16-BIT TIMERS KB0, KB1, AND KB2**
	Modification of description in **7.2.2 16-bit timer KB trigger compare register n (TKBTGCRn)**	
	Modification of description in **7.3.2 Timer clock select register 2 (TPS2)**	
	Modification of **remark 1** in **Table 7-2. 16-bit Timer KB Dithering Count Register np (TKBDNRnp) Setting**	
	Modification of description in **7.4.1 (2) Clear operation**	
	Modification of description in **7.4.1 (3) (a) Basic timing**	
	Modification of description in **7.4.3 Stop/restart operation**	
	Modification of description in **7.4.3 (2) Count operation stop timing**	
	Modification of **Figure 7-41. Configuration of Standalone Mode (Period Controlled by External Trigger Input)**	
	Modification of description in **7.5 Option Function of 16-bit Timers KB0, KB1 and KB2**	
	Modification of **Figure 7-41. Configuration of Standalone Mode (Period Controlled by External Trigger Input)**	
	Modification of **remark 1** in **Figure 7-61. Figure of Relationship Between TKBDNRnp and the Period which Extends Active Period for 1 Count Clock**	
	Modification of description in **7.5.4 (1) Operation mode available for PMW output gate function**	
	Modification of **Table 7-6. Configuration of Forced Output Stop Function**	
	Modification of description in **7.6.3 Registers controlling forced output stop function**	
	Renamed to **7.6.3.6 Forced output stop function1 cancel trigger register n (TKBPAHFTn)**	
	Renamed to **Figure 7-79. Format of Forced Output Stop Function1 Cancel Trigger Register n (TKBPAHFTn)**	
	Modification of descriptions	**CHAPTER 8 16-BIT TIMER KC0**
	Modification of description in **8.3.2 Timer clock select register 2 (TPS2)**	
	Modification of **Figure 8-13. Format of Port Mode Registers 1, 20 (PM1, PM20) (38-pin Products)**	

(16/20)

Edition	Description	Chapter
Ver.3.10	Modification of descriptions	**CHAPTER 9 REAL-TIME CLOCK**
	Addition of **notes** to **Figure 9-5. Format of Real-time Clock Control Register 1 (RTCC1) (2/2)**	
	Modification of descriptions	**CHAPTER 10 12-BIT INTERVAL TIMER**
	Modification of descriptions	**CHAPTER 11 WATCHDOG TIMER**
	Modification of descriptions	**CHAPTER 12 A/D CONVERTER**
	Modification of description in **12.2(10) AVREFP pin**	
	Modification of **Figure 12-11. Format of Analog Input Channel Specification Register (ADS)**	
	Modification of **Figure 12-27. Example of Hardware Trigger Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing**	
	Modification of **Figure 12-37. Flowchart for Setting up SNOOZE Mode**	
	Modification of descriptions	**CHAPTER 13 PROGRAMMABLE GAIN AMPLIFIER**
	Modification of descriptions	**CHAPTER 14 COMPARATOR**
	Modification of **Figure 14-1. Block Diagram of Comparator**	
	Modification of **Table 14-1. Configuration of Comparator**	
	Modification of description in **14.3 Registers Controlling Comparator**	
	Modification of descriptions	**CHAPTER 15 SERIAL ARRAY UNIT 0**
	Modification of description in **15.1.2 UART (UART0, UART1)**	
	Modification of **caution 1** in **Figure 15-4. Format of Peripheral Enable Register 0 (PER0)**	
	Correction of **Note3** and **Caution** to **Figure 15-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (1/2)**	
	Correction of **Caution** to **Figure 15-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (2/2)**	
	Modification of **Figure 15-18. Format of Serial Standby Control Register 0 (SSC0)**	
	Modification of description and **Remark 2** in **Figure 15-24. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI00)**	
	Modification of **remark 2** in **Figure 15-40. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O**	
	Modification of description in **15.5.7 SNOOZE mode function**, and modification of **Figure 15-70. Timing Chart of SNOOZE Mode Operation (Once Startup) (Type 1: DAPmn = 0, CKPmn = 0)**	
	Modification of **remark 2** in **Figure 15-75. Example of Contents of Registers for UART Transmission of UART (UART0, UART1) (2/2)**	

(17/20)

Edition	Description	Chapter
Ver.3.10	Modification of **Figure 15-80. Flowchart of UART Transmission (in Single-Transmission Mode)**	**CHAPTER 15 SERIAL ARRAY UNIT 0**
	Addition of **Caution 5** to**15.6.3 SNOOZE mode function**	
	Modification of **Figure 15-89. Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)** to **Figure 15-92. Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)**	
	Modification of descriptions	**CHAPTER 16 SERIAL ARRAY UNIT 4 (DALI/UART4)**
	Modification of **Figure 16-31. Timing Chart of UART Transmission (in Single-Transmission Mode)**	
	Addition of **Caution 5** to **16.5.3 SNOOZE mode function**	
	Modification of **Figure 16-41. Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)**	
	Modification of **Figure 16-42. Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)**	
	Modification of **Figure 16-44. Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)**	
	Modification of description and **remark 1** in **16.6.1 DALI transmission**	
	Modification of **Figure 16-55. Timing Chart of DALI Transmission**	
	Modification of description in **Figure 16-57. Example of Contents of Registers for DALI Reception (1/2)**	
	Modification of **Figure 16-61. Timing Chart of DALI Reception**	
	Modification of **Figure 16-63. Timing Chart of Returning from STOP Mode**	
	Modification of **Figure 16-64. Flowchart of Returning from STOP Mode for Reception**	
	Modification of description in **16.11 DMX512 Communication Operation**	
	Modification of descriptions	**CHAPTER 17 SERIAL INTERFACE IICA**
	Modification of description in **the meanings of <8> to <19> in (3) Data ~ data ~ stop condition in Figure 17-34**	
	Modification of descriptions	**CHAPTER 18 MULTIPLIER AND DIVIDER/MULTIPLY-ACCUMULATOR**
	Modification of **Figure 18-3. Format of Multiplication/Division Data Register B (MDBH, MDBL)**	
	Modification of descriptions	**CHAPTER 19 DMA CONTROLLER**
	Modification of descriptions	**CHAPTER 20 INTERRUPT FUNCTIONS**
	Change of **note1** in **Table 20-2. Flags Corresponding to Interrupt Request Sources (2/3)**	
	Modification of **Figure 20-3. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (38-pin) (1/2)**	
	Modification of description in **20.3.5 Comparator rising edge enable register 0 (CMPEGP0), comparator falling edge enable register 0 (CMPEGN0)**	

(18/20)

Edition	Description	Chapter
Ver.3.10	Modification of descriptions	**CHAPTER 21 STANDBY FUNCTION**
	Modification of descriptions	**CHAPTER 22 RESET FUNCTION**
	Renamed to **Figure 22-3. Timing of Reset Due to Execution of Illegal Instruction, Watchdog Timer, RAM Parity Error, or Illegal Memory Access**	
	Addition of description in **Figure 22-5. Procedure for Checking Reset Source**	
	Modification of descriptions	**CHAPTER 23 POWER-ON-RESET CIRCUIT**
	Modification of **Figure 23-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (1/3)**	
	Modification of descriptions	**CHAPTER 24 VOLTAGE DETECTOR**
	Modification of descriptions	**CHAPTER 25 SAFETY FUNCTIONS**
	Modification of description in **25.3.7 Frequency detection function**	
	Modification of descriptions	**CHAPTER 27 OPTION BYTE**
	Modification of description in **27.1.1 User option byte (000C0H to 000C2H/010C0H to 010C2H)**	
	Modification of **Figure 27-3. Format of Option Byte (000C2H/010C2H)**	
	Modification of descriptions	**CHAPTER 28 FLASH MEMORY**
	Modification of Table 28-1. Wiring Between RL78/I1A and Dedicated Flash Memory Programmer	
	Modification of Figure 28-2. Communication with Dedicated Flash Memory Programmer	
	Modification of Table 28-2. Pin Connection	
	Modification of **Table 29-1. On-chip Debug Security ID**	**CHAPTER 29 ON-CHIP DEBUG FUNCTION**
	Modification of descriptions	**CHAPTER 31 INSTRUCTION SET**
	Modification of descriptions	**CHAPTER 32 ELECTRICAL SPECIFICATIONS**
	Modification of description of the target products	
	Modification of symbols in **32.6.6 LVD circuit characteristics**	
	Modification of **32.8 Flash Memory Programming Characteristics**	
	Modification of descriptions	**CHAPTER 33 ELECTRICAL SPECIFICATIONS**
	Modification of description of the target products	
	Modification of symbols in **33.6.6 LVD circuit characteristics**	
	Addition of Note 4 to **33.8 Flash Memory Programming Characteristics**	
	Modification of **33.10 Timing of Entry to Flash Memory Programming Modes**	

(19/20)

Edition	Description	Chapter
Ver. 3.20	Modification of **caution** in **Figure 2-7 Pin Block Diagram for Pin Type 7-1-2**	**CHAPTER 2 PIN FUNCTIONS**
	Modification of **caution** in **Figure 2-8 Pin Block Diagram for Pin Type 7-1-6**	
	Modification of **caution** in **Figure 2-10 Pin Block Diagram for Pin Type 7-3-2**	
	Modification of **caution 1** in **Figure 2-12 Pin Block Diagram for Pin Type 8-1-2**	
	Modification of the title and description in **3.3.4 Register indirect addressing**	**CHAPTER 3 CPU ARCHITECTURE**
	Modification of the title in **Figure 3-16 Outline of Register Indirect Addressing**	
	Modification of **Figure 3-36 Example of CALL, CALLT**	
	Modification of **caution** in **Figure 4-1 Format of Port Mode Register**	**CHAPTER 4 PORT FUNCTIONS**
	Modification of description in **5.4.4 Low-speed on-chip oscillator**	**CHAPTER 5 CLOCK GENERATOR**
	Modification of description in **7.6.1 (2) Start/cancel of Forced Output Stop Function 1 & 2**	**CHAPTER 7 16-BIT TIMERS KB0, KB1, AND KB2**
	Addition of **caution 2** in **Figure 9-4 Format of Real-time Clock Control Register 0 (RTCC0)**	**CHAPTER 9 REAL-TIME CLOCK**
	Addition of **note** in **Table 11-4 Setting Window Open Period of Watchdog Timer**	**CHAPTER 11 WATCHDOG TIMER**
	Addition of description in **15.3.17 Registers controlling port functions of serial I/O pins**	**CHAPTER 15 SERIAL ARRAY UNIT 0**
	Modification of **Figure 15-43 Procedure for Resuming Master Transmission/Reception**	
	Modification of **Figure 15-97 Master Transmission Operation of LIN**	
	Modification of **Figure 15-98 Flowchart for LIN Transmission**	
	Modification of **Figure 15-99 Reception Operation of LIN**	
	$\overline{\text{ACK}}$ has been changed to ACK.	**CHAPTER 17 SERIAL INTERFACE IICA**
	Addition of description in **17.3.6 IICA low-level width setting register 0 (IICWL0)**	
	Deletion of **note 3** in **Table 20-1 Interrupt Source List (3/3)**	**CHAPTER 20 INTERRUPT FUNCTIONS**
	Modification of **Figure 21-6 When the Interrupt Request Signal is not Generated in the SNOOZE Mode**	**CHAPTER 21 STANDBY FUNCTION**
	Addition of **note 3** in **Figure 27-1 Format of User Option Byte (000C0H/010C0H)**	**CHAPTER 27 OPTION BYTE**
	Modification of **Figure 28-5 Setting of Flash Memory Programming Mode**	**CHAPTER 28 FLASH MEMORY**
	$\overline{\text{ACK}}$ has been changed to ACK.	**CHAPTER 32 ELECTRICAL SPECIFICATIONS (G: Industrial applications, T_A = -40 to +105ºC)**
	Modification of the figure in **32.10 Timing of Entry to Flash Memory Programming Modes**	
	$\overline{\text{ACK}}$ has been changed to ACK.	**CHAPTER 33 ELECTRICAL SPECIFICATIONS (M: Industrial applications, T_A = -40 to +125ºC)**
	Modification of the figure in **33.10 Timing of Entry to Flash Memory Programming Modes**	

(20/20)

Edition	Description	Chapter
Ver. 3.20	Modification of figure in **34.1 20-pin Products**	**CHAPTER 34 PACKAGE DRAWINGS**
Ver. 3.30	The module name for 3-wire serial I/O and 3-wire serial were changed to simplified SPI.	**All**
	The module name for CSI was changed to simplified SPI.	
	“Wait” was modified to “clock stretch”.	
	Addition of Note in **1.1 Features**	**CHAPTER 1 OUTLINE**
	Addition of Note in **4.4.5 Handling different potential (2.5 V, 3 V) by using I/O buffers**	**CHAPTER 4 PORT FUNCTIONS**
	Modification of **Figure 9-5. Format of Real-time Clock Control Register 1 (RTCC1) (2/2)**	**CHAPTER 9 REAL-TIME CLOCK**
	Modification of Caution in **Figure 9-21. Procedure for Reading Real-time Clock**	
	Modification of Cautions 1 in **Figure 9-22. Procedure for Writing Real-time Clock**	
	Modification of **CHAPTER 13 PROGRAMMABLE GAIN AMPLIFIER**	**CHAPTER 13 PROGRAMMABLE GAIN AMPLIFIER**
	Modification of **13.1 Functions of Programmable Gain Amplifier**	
	Modification of **13.2 Configuration of Programmable Gain Amplifier**	
	Addition of Note in **CHAPTER 15 SERIAL ARRAY UNIT 0**	**CHAPTER 15 SERIAL ARRAY UNIT 0**
	Modification of **Figure 16-22. Format of Port Output Mode Registers 1 and 20 (POM1 and POM20)**	**CHAPTER 16 SERIAL ARRAY UNIT 4 (DALI/UART4)**
	Modification of title in **Figure 17-21. Clock stretching (1/2)**	**CHAPTER 17 SERIAL INTERFACE IICA**
	Modification of title in **Figure 17-21. Clock stretching (2/2)**	
	Addition of Note in **19.5.1 Simplified SPI (CSI) consecutive transmission**	**CHAPTER 19 DMA CONTROLLER**
	Modification of Notes 1 and 4 in **32.3.2 Supply current characteristics**	**CHAPTER 32 ELECTRICAL SPECIFICATIONS (G: Industrial applications, TA = −40 to +105°C)**
	Modification of Notes 1 and 5 and delete Notes 6 in **32.3.2 Supply current characteristics**	
	Modification of Notes 1 and 4 in **33.3.2 Supply current characteristics**	**CHAPTER 33 ELECTRICAL SPECIFICATIONS (M: Industrial applications, TA = −40 to +125°C)**
	Modification of Notes 1 and 5 and delete Notes 6 in **33.3.2 Supply current characteristics**	

RL78/I1A User's Manual: Hardware

Publication Date: Rev.0.01 Mar 31, 2011
Rev.3.40 Mar 29, 2024

Published by: Renesas Electronics Corporation

RL78/I1A

Renesas Electronics Corporation

R01UH0169EJ0340