User's Manual

16

RL78/L1C

User's Manual: Hardware

16-Bit Single-Chip Microcontrollers

Renesas Electronics
www.renesas.com

Rev.2.40 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/L1C and design and develop application systems and programs for these devices.
The target products are as follows.

- 80-pin: R5F110Mx, R5F111Mx (x = E, F, G, H, J)
- 85-pin: R5F110Nx, R5F111Nx (x = E, F, G, H, J)
- 100-pin: R5F110Px, R5F111Px (x = E, F, G, H, J)

Purpose

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

Organization

The RL78/L1C manual is separated into two parts: this manual and the software edition (common to the RL78 family).

RL78/L1C 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/L1C 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/L1C User's Manual: Hardware	This manual
RL78 Family User's Manual: Software	R01US0015E

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

Document Name	Document No.
PG-FP6 Flash Memory Programmer User's Manual	R20UT4025E
E1, E20 Emulator User's Manual	R20UT0398E
E2 Emulator User's Manual	R20UT3538E
E2 Lite Emulator User's Manual	R20UT3240E
Renesas Flash Programmer Flash Memory Programming Software User's Manual	R20UT4066E
Renesas Flash Development Toolkit User's Manual	R20UT0508E

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

Other Documents

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

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

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

CONTENTS

CHAPTER 1 OUTLINE

1.1 Features

○ Ultra-low power consumption technology
- V_{DD} = single power supply voltage of 1.6 to 3.6 V
- HALT mode
- STOP mode
- SNOOZE mode

○ RL78 CPU core
- CISC architecture with 3-stage pipeline
- Minimum instruction execution time: Can be changed from high speed (0.04167 μs: @ 24 MHz operation with high-speed on-chip oscillator clock or PLL clock) to ultra-low speed (30.5 μs: @ 32.768 kHz operation with subsystem clock)
- Multiply/divide and multiply/accumulate instructions are supported.
- Address space: 1 Mbyte
- General-purpose registers: (8-bit register × 8) × 4 banks
- On-chip RAM: 8 to 16 KB

○ Code flash memory
- Code flash memory: 64 to 256 KB
- Block size: 1 KB
- Prohibition of block erase and rewriting (security function)
- On-chip debug function
- Self-programming (with boot swap function/flash shield window function)

○ Data flash memory
- Data flash memory: 8 KB
- Background operation (BGO): Instructions can be executed from the program memory while rewriting the data flash memory.
- Number of rewrites: 1,000,000 times (TYP.)
- Voltage of rewrites: V_{DD} = 1.8 to 3.6 V

○ High-speed on-chip oscillator
- Select from 48 MHz, 24 MHz, 16 MHz, 12 MHz, 8 MHz, 6 MHz, 4 MHz, 3 MHz, 2 MHz, and 1 MHz
- High accuracy: ±1.0% (V_{DD} = 1.8 to 3.6 V, T_A = -20 to +85°C)

○ Operating ambient temperature
- T_A = -40 to +85°C (A: Consumer applications)
- T_A = -40 to +105°C (G: Industrial applications)

○ Power management and reset function
- On-chip power-on-reset (POR) circuit
- On-chip voltage detector (LVD) (Select interrupt and reset from 12 levels)

○ Data transfer controller (DTC)
- Transfer modes: Normal transfer mode, repeat transfer mode, block transfer mode
- Activation sources: Activated by interrupt sources (30 to 33 sources).
- Chain transfer function

○ Event link controller (ELC)
- Event signals of 30 or 31 types can be linked to the specified peripheral function.

○ Serial interfaces
- Simplified SPI (CSI[Note 1]): 4 channels
- UART/UART (LIN-bus supported): 4 channels
- I^2C/simplified I^2C: 5 channels

○ Timers
- 16-bit timer: 11 channels
- 12-bit interval timer: 1 channel
- Real-time clock 2: 1 channel (calendar for 99 years, alarm function, and clock correction function)
- Watchdog timer: 1 channel (operable with the dedicated low-speed on-chip oscillator)

○ LCD controller/driver
- Internal voltage boosting method, capacitor split method, and external resistance division method are switchable.
- Segment signal output: 44 (40) [Note 2] to 56 (52) [Note 2]
- Common signal output: 4 (8) [Note 2]

○ USB Note 3
- USB version 2.0 (function controller)
- Full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps) are supported
- Compliant to Battery Charging Specification Revision 1.2

○ A/D converter
- 8/10-bit resolution A/D converter (V_{DD} = 1.6 to 3.6 V)
- 12-bit resolution A/D converter (V_{DD} = 2.4 to 3.6 V)
- Analog input: 9 to 13 channels
- Internal reference voltage (TYP. 1.45 V) and temperature sensor Note 3

○ D/A converter
- 8-bit resolution D/A converter (V_{DD} = 1.6 to 3.6 V)
- Analog output: 2 channels
- Output voltage: 0 V to V_{DD}
- Real-time output function

○ Comparator
- 2 channels
- Operating modes: Comparator high-speed mode, comparator low-speed mode, window mode
- The external reference voltage or internal reference voltage can be selected as the reference voltage.

○ I/O ports
- I/O ports: 59 to 77 (N-ch open drain I/O [withstand voltage of 6 V]: 2)
- Can be set to N-ch open drain, TTL input buffer, and on-chip pull-up resistor
- On-chip key interrupt function
- On-chip clock output/buzzer output controller

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

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

Note 2. The number in parentheses indicates the number of signal outputs when 8 coms are used.

Note 3. Selectable only in HS (high-speed main) mode.

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

○ ROM, RAM capacities

Products with USB

Flash ROM	Data Flash	RAM	RL78/L1C		
			80 pins	85 pins	100 pins
256 KB	8 KB	16 KB Note	R5F110MJ	R5F110NJ	R5F110PJ
192 KB	8 KB	16 KB Note	R5F110MH	R5F110NH	R5F110PH
128 KB	8 KB	12 KB	R5F110MG	R5F110NG	R5F110PG
96 KB	8 KB	10 KB	R5F110MF	R5F110NF	R5F110PF
64 KB	8 KB	8 KB	R5F110ME	R5F110NE	R5F110PE

Products without USB

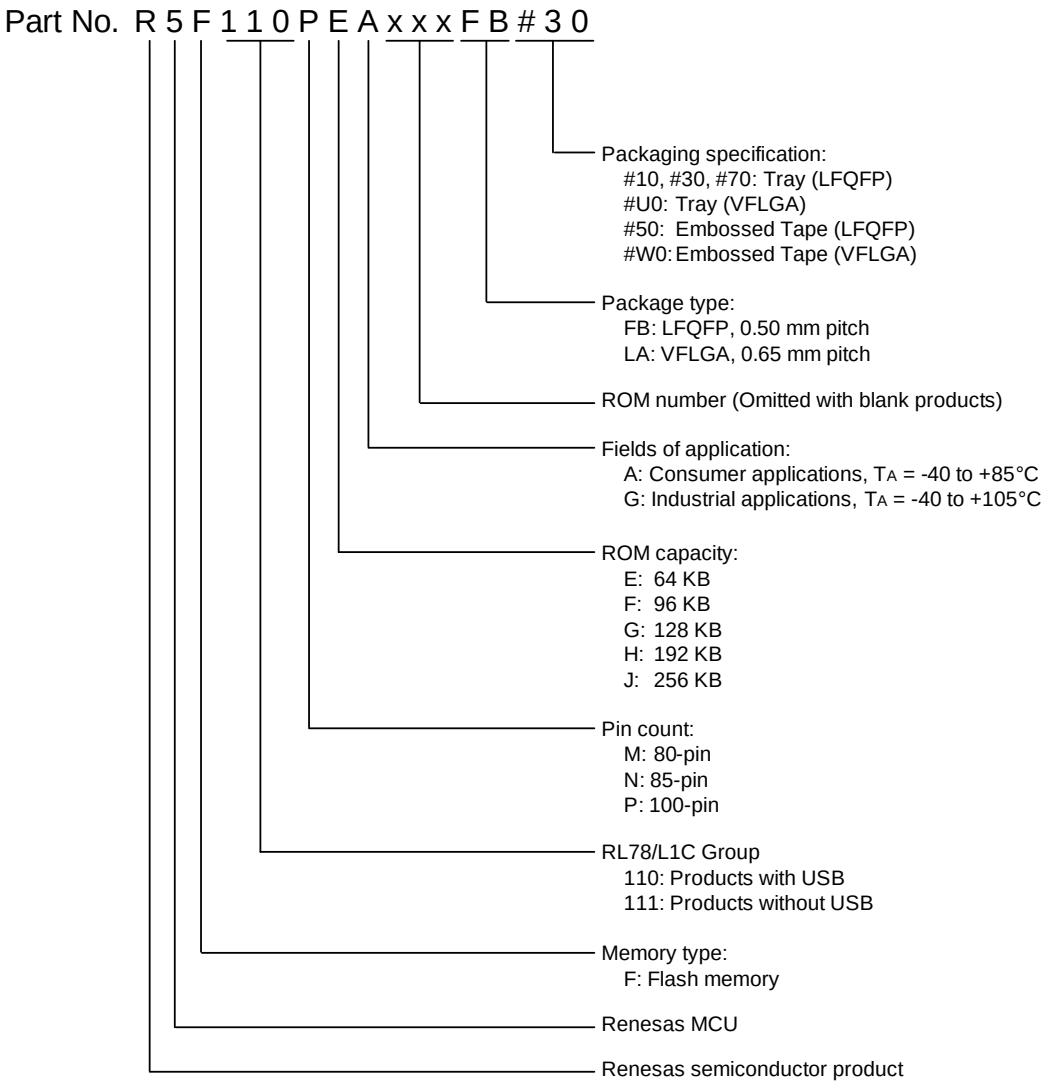
Flash ROM	Data Flash	RAM	RL78/L1C		
			80 pins	85 pins	100 pins
256 KB	8 KB	16 KB Note	R5F111MJ	R5F111NJ	R5F111PJ
192 KB	8 KB	16 KB Note	R5F111MH	R5F111NH	R5F111PH
128 KB	8 KB	12 KB	R5F111MG	R5F111NG	R5F111PG
96 KB	8 KB	10 KB	R5F111MF	R5F111NF	R5F111PF
64 KB	8 KB	8 KB	R5F111ME	R5F111NE	R5F111PE

Note This is about 15 KB when the self-programming function and data flash function are used (For details, see **CHAPTER 3**).

1.2 Ordering Information

Figure 1 - 1 Part Number, Memory Size, and Package of RL78/L1C

<R>



Caution Orderable part numbers are current as of when this manual was published.
Please make sure to refer to the relevant product page on the Renesas website for the latest part numbers.

<R>

Table 1 - 1 List of Ordering Part Numbers

Products with USB

Pin Count	Package	Fields of Application	Orderable Part Number		RENESAS Code
			Product Name	Packaging Specifications	
80 pins	80-pin plastic LFQFP (12 × 12 mm, 0.5 mm pitch)	A	R5F110MEAFB, R5F110MFAFB, R5F110MGAFB, R5F110MHAFB, R5F110MJAFB	#10,#50,#70	PLQP0080KB-B PLQP0080KJ-A
				#30	PLQP0080KB-B
		G	R5F110MEGFB, R5F110MFGFB, R5F110MGGFB, R5F110MHGFB, R5F110MJGFB	#10,#50,#70	PLQP0080KB-B PLQP0080KJ-A
				#30	PLQP0080KB-B
85 pins	85-pin plastic VFLGA (7 × 7 mm, 0.65 mm pitch)	A	R5F110NEALA, R5F110NFALA, R5F110NGALA, R5F110NHALA, R5F110NJALA	#U0,#W0	PVLG0085JA-A
		G	R5F110NEGLA, R5F110NFGLA, R5F110NGGLA, R5F110NHGLA, R5F110NJGLA		
100 pins	100-pin plastic LFQFP (14 × 14 mm, 0.5 mm pitch)	A	R5F110PEAFB, R5F110PFAFB, R5F110PGAFB, R5F110PHAFB, R5F110PJAFB	#10,#50,#70	PLQP0100KB-B PLQP0100KP-A
				#30	PLQP0100KB-B
		G	R5F110PEGFB, R5F110PFGFB, R5F110PGGFB, R5F110PHGFB, R5F110PJGFB	#10,#50,#70	PLQP0100KB-B PLQP0100KP-A
				#30	PLQP0100KB-B

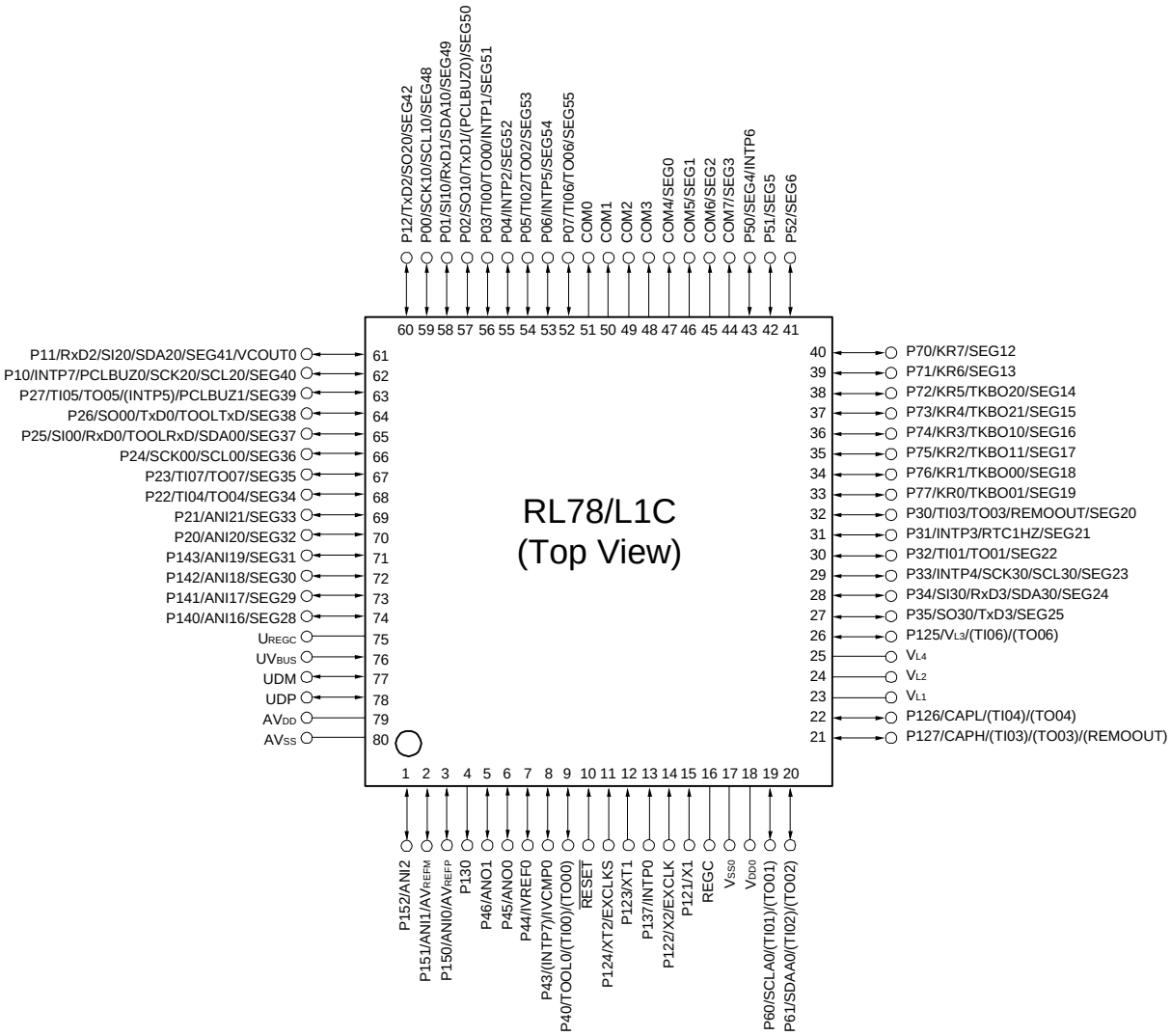
Products without USB

Pin Count	Package	Fields of Application	Orderable Part Number		RENESAS Code
			Product Name	Packaging Specifications	
80 pins	80-pin plastic LFQFP (12 × 12 mm, 0.5 mm pitch)	A	R5F111MEAFB, R5F111MFAFB, R5F111MGAFB, R5F111MHAFB, R5F111MJAFB	#10,#50,#70	PLQP0080KB-B PLQP0080KJ-A
				#30	PLQP0080KB-B
		G	R5F111MEGFB, R5F111MFGFB, R5F111MGGFB, R5F111MHGFB, R5F111MJGFB	#10,#50,#70	PLQP0080KB-B PLQP0080KJ-A
				#30	PLQP0080KB-B
85 pins	85-pin plastic VFLGA (7 × 7 mm, 0.65 mm pitch)	A	R5F111NEALA, R5F111NFALA, R5F111NGALA, R5F111NHALA, R5F111NJALA	#U0,#W0	PVLG0085JA-A
		G	R5F111NEGLA, R5F111NFGLA, R5F111NGGLA, R5F111NHGLA, R5F111NJGLA		
100 pins	100-pin plastic LFQFP (14 × 14 mm, 0.5 mm pitch)	A	R5F111PEAFB, R5F111PFAFB, R5F111PGAFB, R5F111PHAFB, R5F111PJAFB	#10,#50,#70	PLQP0100KB-B PLQP0100KP-A
				#30	PLQP0100KB-B
		G	R5F111PEGFB, R5F111PFGFB, R5F111PGGFB, R5F111PHGFB, R5F111PJGFB	#10,#50,#70	PLQP0100KB-B PLQP0100KP-A
				#30	PLQP0100KB-B

1.3 Pin Configuration (Top View)

1.3.1 80-pin products (with USB)

• 80-pin plastic LFQFP (12 × 12 mm, 0.5 mm pitch)



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

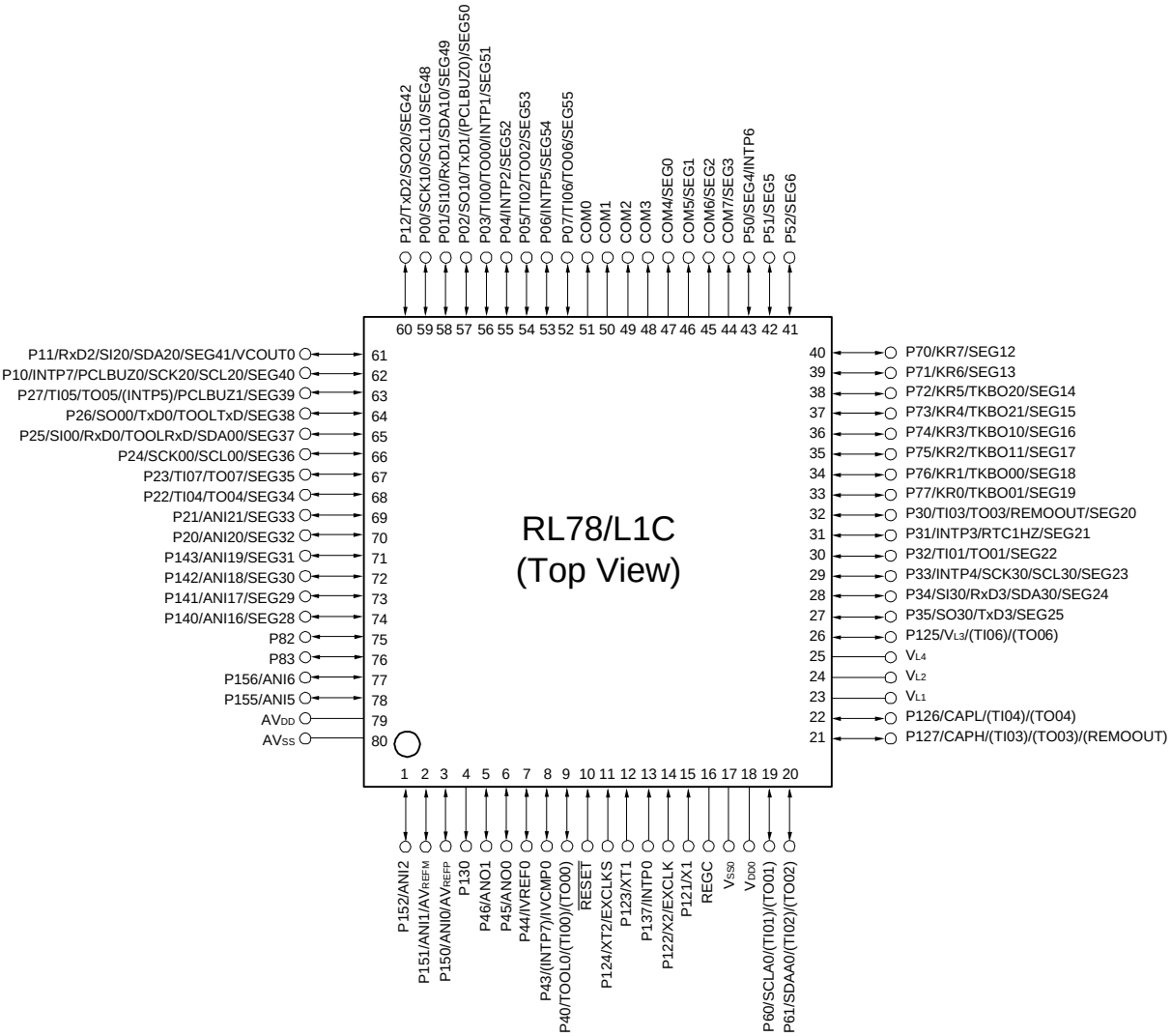
Caution 2. Connect the UREGC pin to Vss pin via a capacitor (0.33 µF).

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

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

1.3.2 80-pin products (without USB)

• 80-pin plastic LFQFP (fine pitch) (12 × 12 mm, 0.5 mm pitch)

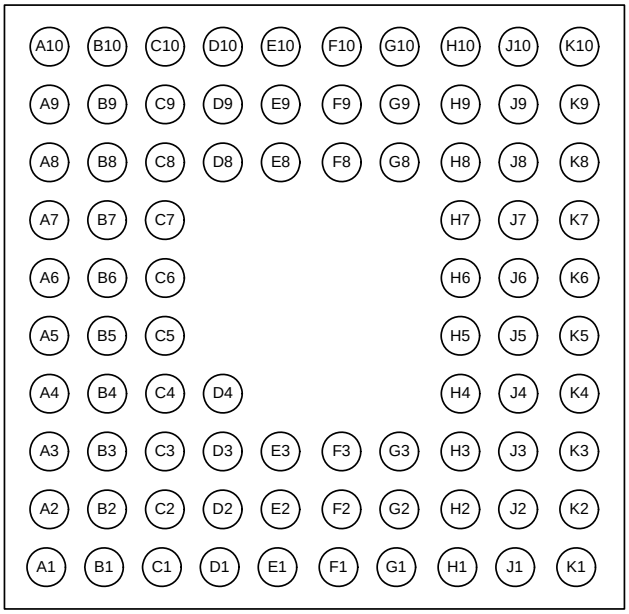


Caution Connect the REGC pin to Vss pin via a capacitor (0.47 to 1 µF).

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

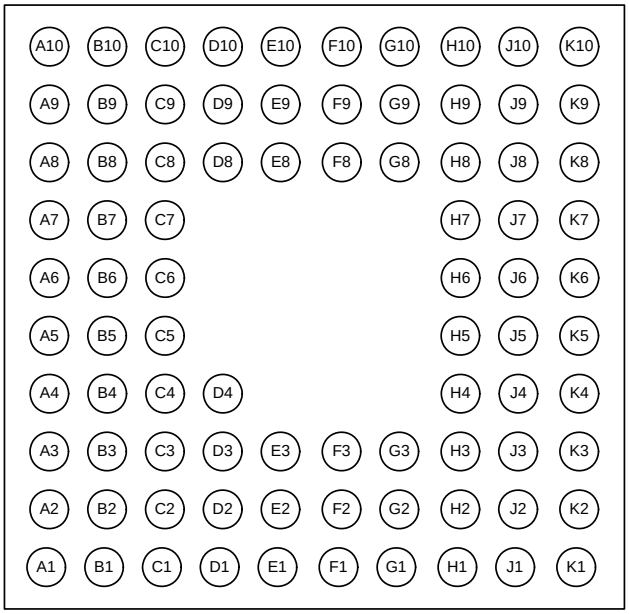
Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

1.3.3 85-pin products (with USB)



Pin	Name	Pin	Name	Pin	Name	Pin	Name	Pin	Name
A1	COM7/SEG3	C1	COM2	E1	P04/INTP2/SEG52	G1	P00/SCK10/SCL10/ SEG48	J1	V_{SS0}
A2	P51/SEG5	C2	COM5/SEG1	E2	P05/TI02/TO02/SEG53	G2	V_{SS0}	J2	P11/RxD2/SI20/SDA20/ SEG41/VCOUT0
A3	P70/KR7/SEG12	C3	COM6/SEG2	E3	P06/INTP5/SEG54	G3	P12/TxD2/SO20/SEG42/ VCOUT1	J3	P26/SO00/TxD0/ TOOLTxD/SEG38
A4	P73/KR4/TKBO21/SEG15	C4	P71/KR6/SEG13	E4	—	G4	—	J4	P23/TI07/TO07/SEG35
A5	P74/KR3/TKBO10/SEG16	C5	P76/KR1/TKBO00/SEG18	E5	—	G5	—	J5	P20/ANI20/SEG32
A6	P31/INTP3/RTC1HZ/ SEG21	C6	P77/KR0/TKBO01/SEG19	E6	—	G6	—	J6	P141/ANI17/SEG29
A7	P33/INTP4/SCK30/SCL30/ SEG23	C7	P34/SI30/RxD3/SDA30/ SEG24	E7	—	G7	—	J7	U_{REGC}
A8	P35/SO30/TxD3/SEG25	C8	V_{L1}	E8	P40/TOOL0/(TI00)/(TO00)	G8	P44/(SCK10)/(SCL10)/ IVREF0	J8	UV_{BUS}
A9	V_{L4}	C9	P61/SDAA0/(TI02)/(TO02)	E9	P137/INTP0	G9	P45/ANO0	J9	AV_{DD}
A10	P126/CAPL/(TI04)/(TO04)	C10	V_{DD0}	E10	P122/X2/EXCLK	G10	P123/XT1	J10	P150/ANI0/AV_{REFP}
B1	COM4/SEG0	D1	COM0	F1	P03/TI00/TO00/INTP1/ SEG51	H1	V_{SS0}	K1	V_{SS0}
B2	P50/SEG4/INTP6	D2	COM1	F2	P02/SO10/TxD1/ (PCLBUZ0)/SEG50	H2	V_{SS0}	K2	P27/TI05/TO05/(INTP5)/ PCLBUZ1/SEG39
B3	P52/SEG6	D3	P07/TI06/TO06/SEG55	F3	P01/SI10/RxD1/SDA10/ SEG49	H3	P10/INTP7/PCLBUZ0/ SCK20/SCL20/SEG40	K3	P25/SI00/RxD0/ TOOLRxD/SDA00/SEG37
B4	P72/KR5/TKBO20/SEG14	D4	COM3	F4	—	H4	P24/SCK00/SCL00/ SEG36	K4	P22/TI04/TO04/SEG34
B5	P75/KR2/TKBO11/SEG17	D5	—	F5	—	H5	P21/ANI21/SEG33	K5	P143/ANI19/SEG31
B6	P30/TI03/TO03/ REMOOUT/SEG20	D6	—	F6	—	H6	P140/ANI16/SEG28	K6	P142/ANI18/SEG30
B7	P32/TI01/TO01/SEG22	D7	—	F7	—	H7	P152/ANI2	K7	UDM
B8	P125/V_{L3}/(TI06)/(TO06)	D8	P60/SCLA0/(TI01)/(TO01)	F8	P43/(INTP7)/(SI10)/ (RxD1)/(SDA10)/IVCMP0	H8	P46/ANO1	K8	UDP
B9	V_{L2}	D9	REGC	F9	$\overline{\text{RESET}}$	H9	P130	K9	AV_{SS}
B10	P127/CAPH/(TI03)/ (TO03)/(REMOOUT)	D10	P121/X1	F10	V_{SS0}	H10	P124/XT2/EXCLKS	K10	P151/ANI1/AV_{REFM}

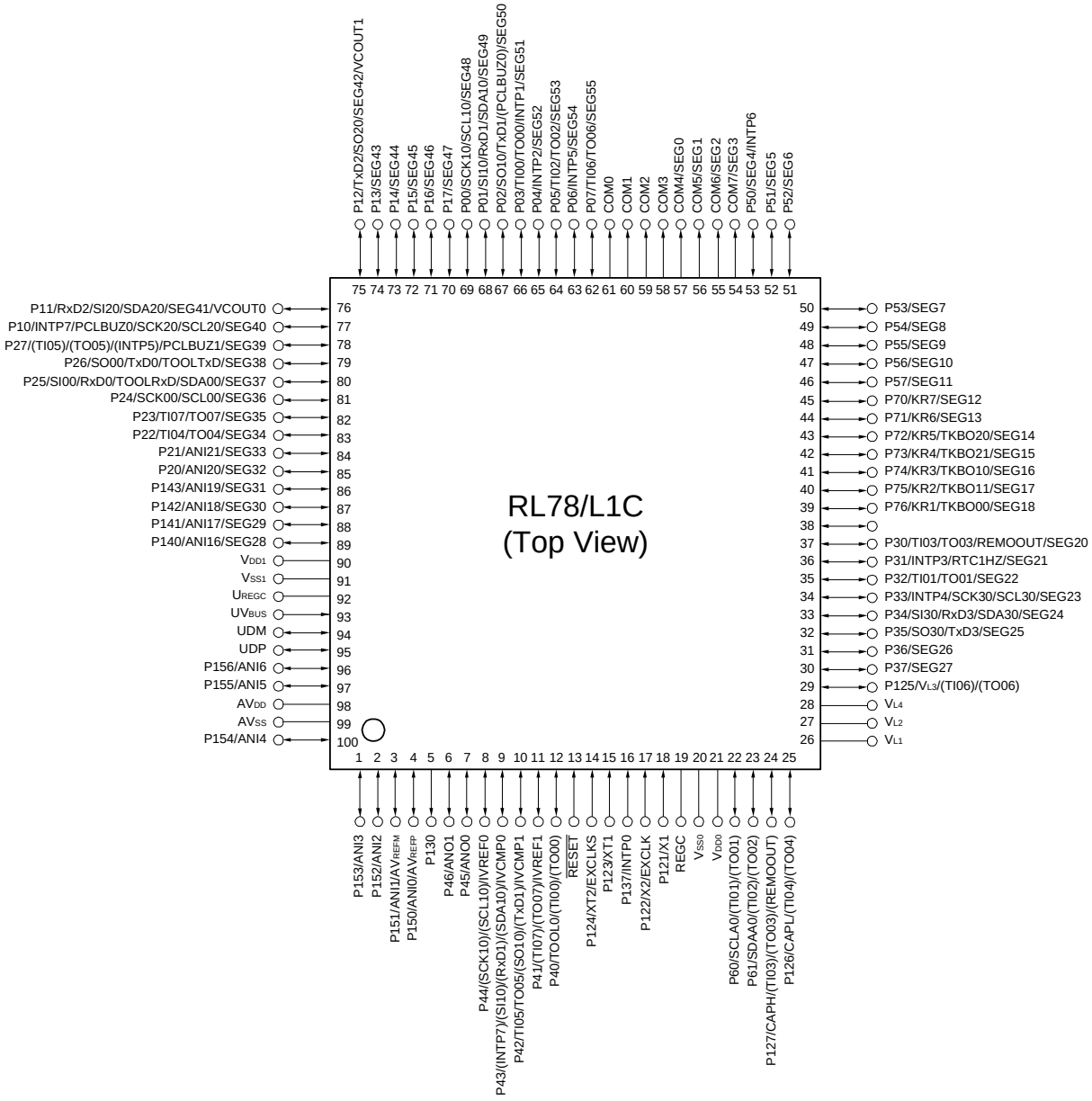
1.3.4 85-pin products (without USB)



Pin	Name	Pin	Name	Pin	Name	Pin	Name	Pin	Name
A1	COM7/SEG3	C1	COM2	E1	P04/INTP2/SEG52	G1	P00/SCK10/SCL10/ SEG48	J1	V_{SS0}
A2	P51/SEG5	C2	COM5/SEG1	E2	P05/TI02/TO02/SEG53	G2	V_{SS0}	J2	P11/RxD2/SI20/SDA20/ SEG41/VCOUT0
A3	P70/KR7/SEG12	C3	COM6/SEG2	E3	P06/INTP5/SEG54	G3	P12/TxD2/SO20/SEG42/ VCOUT1	J3	P26/SO00/TxD0/ TOOLTxD/SEG38
A4	P73/KR4/TKBO21/SEG15	C4	P71/KR6/SEG13	E4	—	G4	—	J4	P23/TI07/TO07/SEG35
A5	P74/KR3/TKBO10/SEG16	C5	P76/KR1/TKBO00/SEG18	E5	—	G5	—	J5	P20/ANI20/SEG32
A6	P31/INTP3/RTC1HZ/ SEG21	C6	P77/KR0/TKBO01/ SEG19	E6	—	G6	—	J6	P141/ANI17/SEG29
A7	P33/INTP4/SCK30/ SCL30/SEG23	C7	P34/SI30/RxD3/SDA30/ SEG24	E7	—	G7	—	J7	P82
A8	P35/SO30/TxD3/SEG25	C8	V_{L1}	E8	P40/TOOL0/(TI00)/(TO00)	G8	P44/(SCK10)/(SCL10)/ IVREF0	J8	P83
A9	V_{L4}	C9	P61/SDAA0/(TI02)/(TO02)	E9	P137/INTP0	G9	P45/ANO0	J9	AV_{DD}
A10	P126/CAPL/(TI04)/(TO04)	C10	V_{DD0}	E10	P122/X2/EXCLK	G10	P123/XT1	J10	P150/ANI0/AV_{REFP}
B1	COM4/SEG0	D1	COM0	F1	P03/TI00/TO00/INTP1/ SEG51	H1	V_{SS0}	K1	V_{SS0}
B2	P50/SEG4/INTP6	D2	COM1	F2	P02/SO10/TxD1/ (PCLBUZ0)/SEG50	H2	V_{SS0}	K2	P27/TI05/TO05/(INTP5)/ PCLBUZ1/SEG39
B3	P52/SEG6	D3	P07/TI06/TO06/SEG55	F3	P01/SI10/RxD1/SDA10/ SEG49	H3	P10/INTP7/PCLBUZ0/ SCK20/SCL20/SEG40	K3	P25/SI00/RxD0/ TOOLRxD/SDA00/SEG37
B4	P72/KR5/TKBO20/SEG14	D4	COM3	F4	—	H4	P24/SCK00/SCL00/ SEG36	K4	P22/TI04/TO04/SEG34
B5	P75/KR2/TKBO11/SEG17	D5	—	F5	—	H5	P21/ANI21/SEG33	K5	P143/ANI19/SEG31
B6	P30/TI03/TO03/ REMOOUT/SEG20	D6	—	F6	—	H6	P140/ANI16/SEG28	K6	P142/ANI18/SEG30
B7	P32/TI01/TO01/SEG22	D7	—	F7	—	H7	P152/ANI2	K7	P156/ANI6
B8	P125/V_{L3}/(TI06)/(TO06)	D8	P60/SCLA0/(TI01)/(TO01)	F8	P43/(INTP7)/(SI10)/ (RxD1)/(SDA10)/IVCMP0	H8	P46/ANO1	K8	P155/ANI5
B9	V_{L2}	D9	REGC	F9	$\overline{\text{RESET}}$	H9	P130	K9	AV_{SS}
B10	P127/CAPH/(TI03)/ (TO03)/(REMOOUT)	D10	P121/X1	F10	V_{SS0}	H10	P124/XT2/EXCLKS	K10	P151/ANI1/AV_{REFM}

1.3.5 100-pin products (with USB)

• 100-pin plastic LFQFP (fine pitch) (14 × 14 mm, 0.5 mm pitch)



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

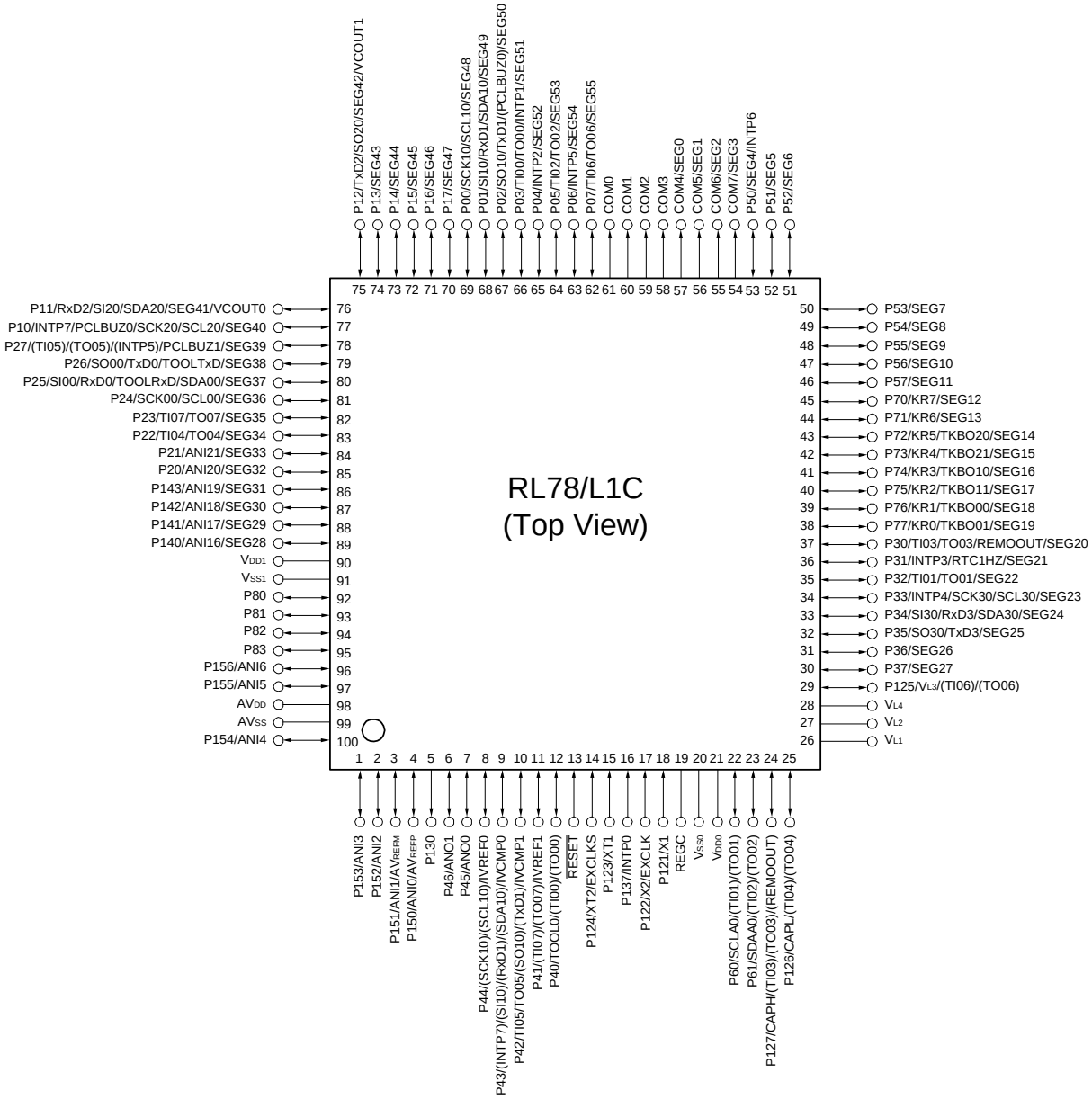
Caution 2. Connect the UREGC pin to Vss pin via a capacitor (0.33 µF).

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

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

1.3.6 100-pin products (without USB)

• 100-pin plastic LFQFP (fine pitch) (14 × 14 mm, 0.5 mm pitch)



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

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

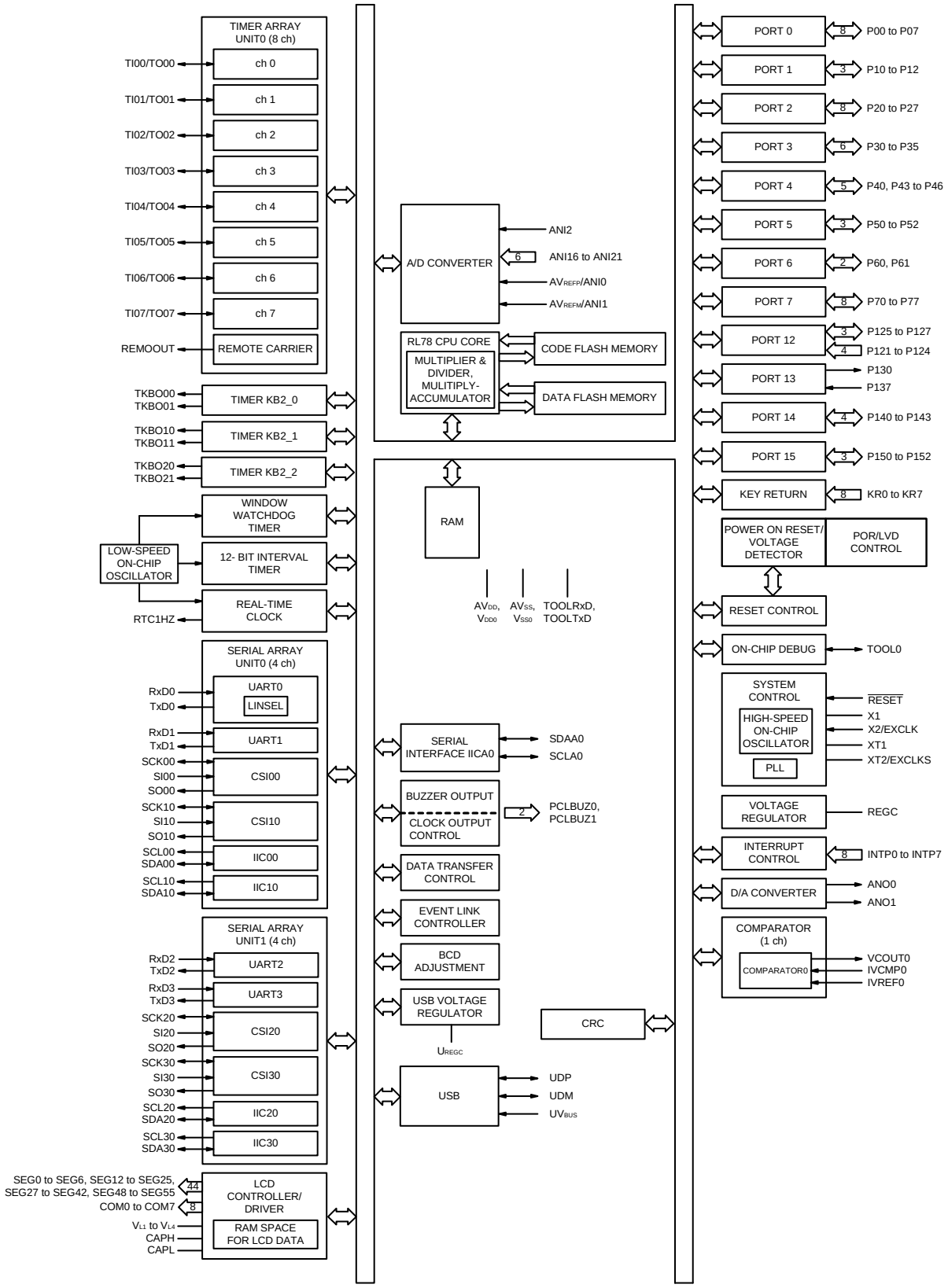
Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

1.4 Pin Identification

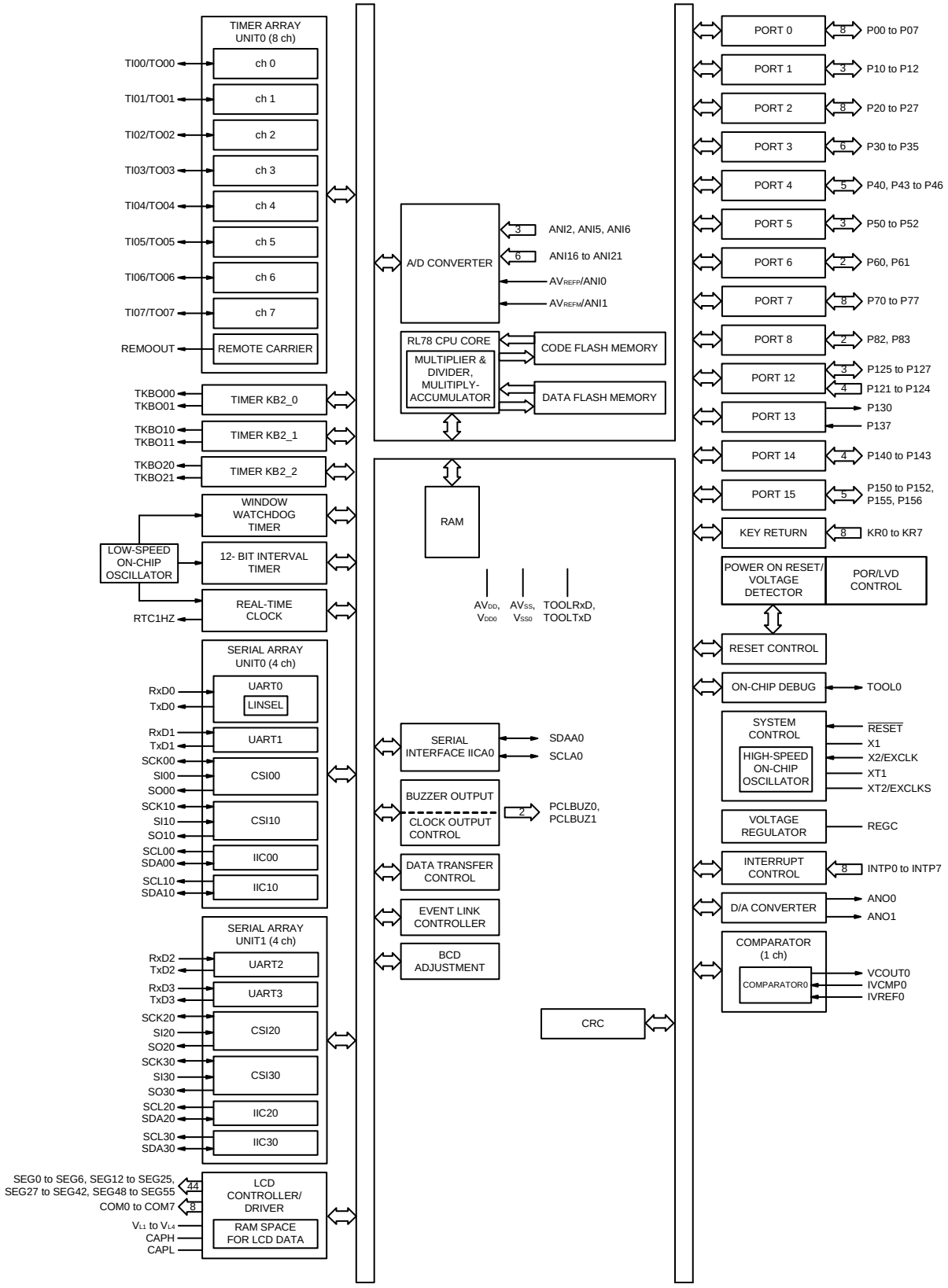
ANI0 to ANI6, ANI16 to ANI21 : Analog Input
ANO0, ANO1 : Analog Output
AVDD : Analog Power Supply
AVREFM : Analog Reference Voltage Minus
AVREFP : Analog Reference Voltage Plus
AVSS : Analog Ground
CAPH, CAPL : Capacitor for LCD
COM0 to COM7 : LCD Common Output
EXCLK : External Clock Input (Main System Clock)
EXCLKS : External Clock Input (Subsystem Clock)
INTP0 to INTP7 : External Interrupt Input
IVCMP0, IVCMP1 : Comparator Input
IVREF0, IVREF1 : Comparator Reference Input
KR0 to KR7 : Key Return
P00 to P07 : Port 0
P10 to P17 : Port 1
P20 to P27 : Port 2
P30 to P37 : Port 3
P40 to P46 : Port 4
P50 to P57 : Port 5
P60 to P62 : Port 6
P70 to P77 : Port 7
P80 to P83 : Port 8
P121 to P127 : Port 12
P130, P137 : Port 13
P140 to P143 : Port 14
P150 to P156 : Port 15
PCLBUZ0, PCLBUZ1 : Programmable Clock Output/ Buzzer Output
REGC : Regulator Capacitance
REMOOUT : Remote Control Output
$\overline{\text{RESET}}$: Reset
RTC1HZ : Real-time Clock Correction Clock (1 Hz) Output
RxD0 to RxD3 : Receive Data
SCK00, SCK10, SCK20, SCK30 : Serial Clock Input/Output
SCLA0 : Serial Clock Input/Output
SCL00, SCL10, SCL20, SCL30 : Serial Clock Output
SDAA0, SDA00, SDA10, SDA20, SDA30 : Serial Data Input/Output
SEG0 to SEG55 : LCD Segment Output
SI00, SI10, SI20, SI30 : Serial Data Input
SO00, SO10, SO20, SO30 : Serial Data Output
TI00 to TI07 : Timer Input
TO00 to TO07 : Timer Output
TKBO00, TKBO01, TKBO10, TKBO11, TKBO20, TKBO21
TOOL0 : Data Input/Output for Tool
TOOLRxD, TOOLTxD : Data Input/Output for External Device
UDM, UDP : USB Input/Output
UREGC : USB Regulator Capacitance
UVBUS : USB Input/USB Power Supply
TxD0 to TxD3 : Transmit Data
VCOUT0, VCOUT1 : Comparator Output
VDD0, VDD1 : Power Supply
VL1 to VL4 : LCD Power Supply
VSS0, VSS1 : Ground
X1, X2 : Crystal Oscillator (Main System Clock)
XT1, XT2 : Crystal Oscillator (Subsystem Clock)

1.5 Block Diagram

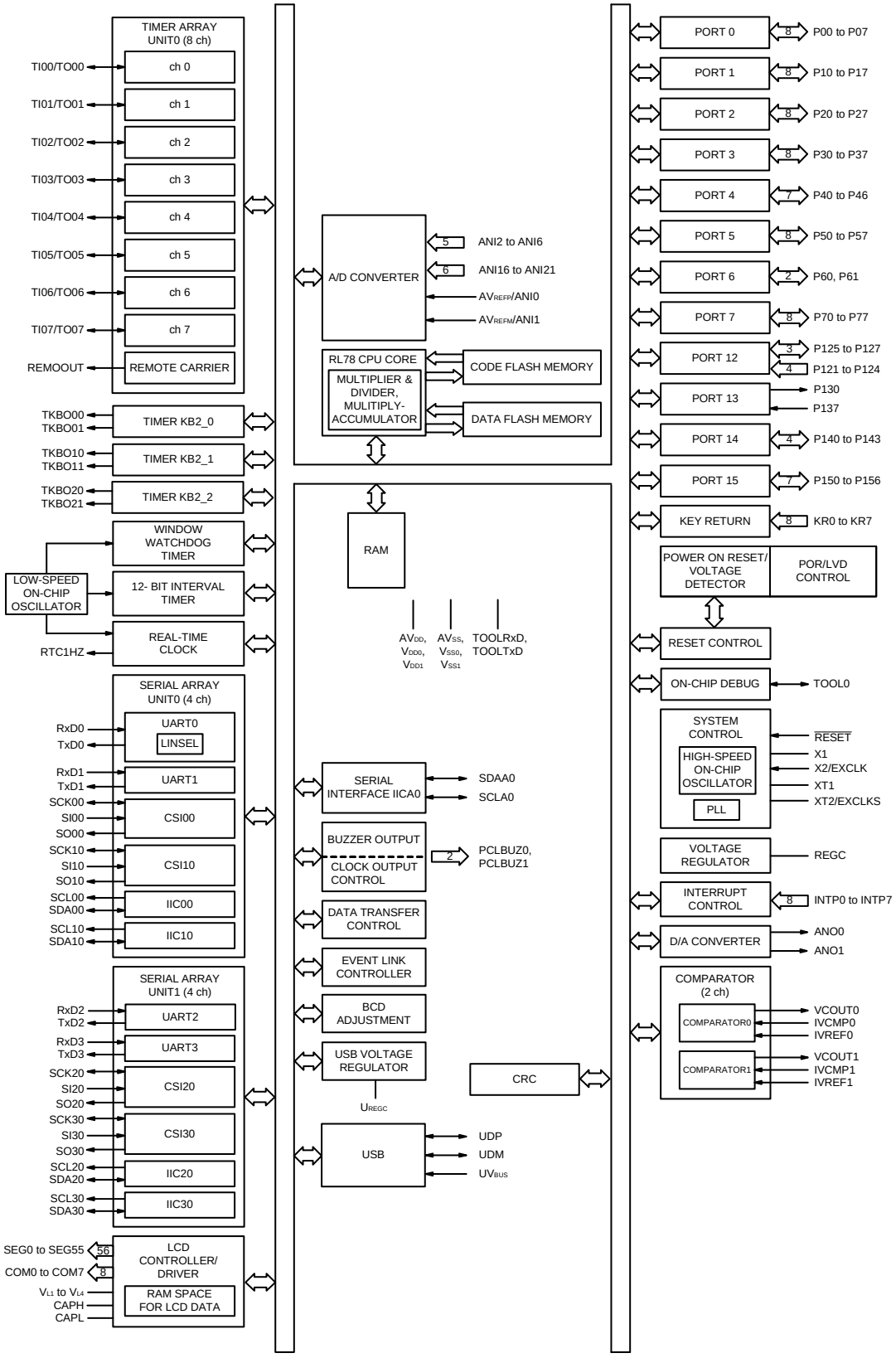
1.5.1 80/85-pin products (with USB)



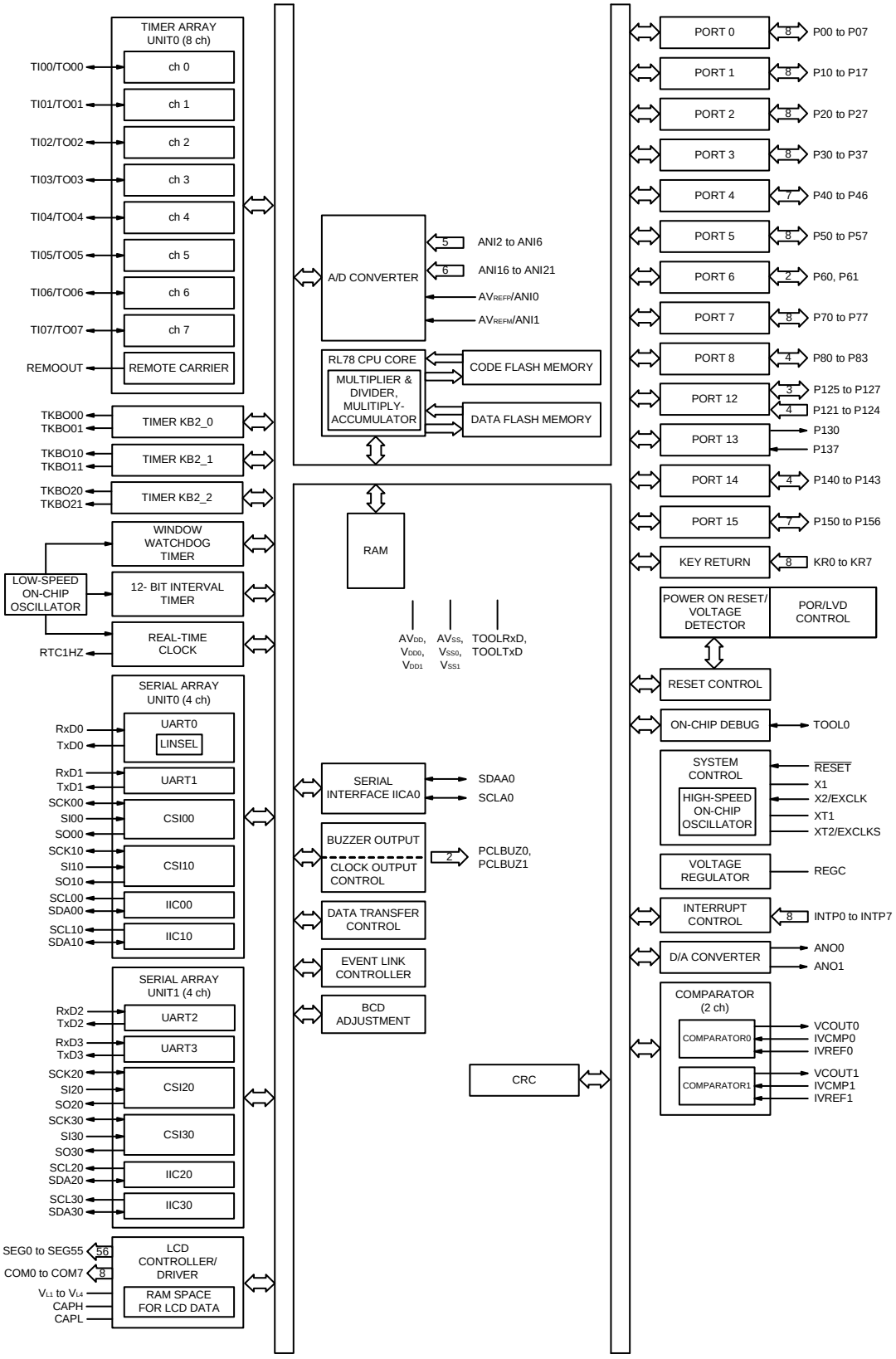
1.5.2 80/85-pin products (without USB)



1.5.3 100-pin products (with USB)



1.5.4 100-pin products (without USB)



1.6 Outline of Functions

[80/85-pin, 100-pin products (with USB)] **(1/2)**

Item		80/85-pin	100-pin
		R5F110Mx/R5F110Nx (x = E to H, J)	R5F110Px (x = E to H, J)
Code flash memory (KB)		64 to 256	64 to 256
Data flash memory (KB)		8	8
RAM (KB)		8 to 16 Note 1	8 to 16 Note 1
Memory space		1 MB	
Main system clock	High-speed system clock	X1 (crystal/ceramic) oscillation, external main system clock input (EXCLK) 1 to 20 MHz: VDD = 2.7 to 3.6 V, 1 to 8 MHz: VDD = 1.8 to 2.7 V, 1 to 4 MHz: VDD = 1.6 to 1.8 V	
	High-speed on-chip oscillator clock	HS (high-speed main) operation mode: 1 to 24 MHz (VDD = 2.7 to 3.6 V), HS (high-speed main) operation mode: 1 to 16 MHz (VDD = 2.4 to 3.6 V), LS (low-speed main) operation mode: 1 to 8 MHz (VDD = 1.8 to 3.6 V), LV (low-voltage main) operation mode: 1 to 4 MHz (VDD = 1.6 to 3.6 V)	
	PLL clock	6, 12, 24 MHz Note 2: VDD = 2.4 to 3.6 V	
Subsystem clock		XT1 (crystal) oscillation, external subsystem clock input (EXCLKS) 32.768 kHz (TYP.): VDD = 1.6 to 3.6 V	
Low-speed on-chip oscillator clock		15 kHz (TYP.): VDD = 1.6 to 3.6 V	
General-purpose register		8 bits × 32 registers (8 bits × 8 registers × 4 banks)	
Minimum instruction execution time		0.04167 μs (High-speed on-chip oscillator clock: fHOCO = fIH = 24 MHz operation)	
		0.04167 μs (PLL clock: fPLL = 48 MHz/fIH = 24 MHz Note 2 operation)	
		0.05 μs (High-speed system clock: fMX = 20 MHz operation)	
		30.5 μs (Subsystem clock: fSUB = 32.768 kHz operation)	
Instruction set		• Data transfer (8/16 bits) • Adder and subtractor/logical operation (8/16 bits) • Multiplication (8 bits × 8 bits, 16 bits × 16 bits), Division (16 bits ÷ 16 bits, 32 bits ÷ 32 bits) • Multiplication and Accumulation (16 bits × 16 bits + 32 bits) • Rotate, barrel shift, and bit manipulation (Set, reset, test, and Boolean operation), etc.	
I/O port	Total	59	77
	CMOS I/O	51	69
	CMOS input	5	5
	CMOS output	1	1
	N-ch open-drain I/O (6 V tolerance)	2	2
Timer	16-bit timer TAU	8 channels (with 1 channel remote control output function) (Timer outputs: 8, PWM outputs: 7 Note 3)	
	16-bit timer KB2	3 channels (PWM outputs: 6)	
	Watchdog timer	1 channel	
	12-bit interval timer	1 channel	
	Real-time clock 2	1 channel	
	RTC output	1 1 Hz (subsystem clock: fSUB = 32.768 kHz)	

Note 1. In the case of the 16 KB, this is about 15 KB when the self-programming function and data flash function are used (For details, see **CHAPTER 3**).

Note 2. In the PLL clock 48 MHz operation, the system clock is 2/4/8 dividing ratio.

Note 3. The number of outputs varies, depending on the setting of channels in use and the number of the master.

(2/2)

Item			80/85-pin	100-pin
			R5F110Mx/R5F110Nx (x = E to H, J)	R5F110Px (x = E to H, J)
Clock output/buzzer output			2	2
			• 2.44 kHz, 4.88 kHz, 9.76 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz (Main system clock: f_{MAIN} = 20 MHz operation) • 256 Hz, 512 Hz, 1.024 kHz, 2.048 kHz, 4.096 kHz, 8.192 kHz, 16.384 kHz, 32.768 kHz (Subsystem clock: f_{SUB} = 32.768 kHz operation)	
8/12-bit resolution A/D converter			9 channels	13 channels
D/A converter			2 channels	2 channels
Comparator			1 channel	2 channels
Serial interface			• Simplified SPI (CSI): 1 channel/UART (UART supporting LIN-bus): 1 channel/simplified I^2C: 1 channel • Simplified SPI (CSI): 1 channel/UART: 1 channel/simplified I^2C: 1 channel • Simplified SPI (CSI): 1 channel/UART: 1 channel/simplified I^2C: 1 channel • Simplified SPI (CSI): 1 channel/UART: 1 channel/simplified I^2C: 1 channel	
	I^2C bus		1 channel	1 channel
USB	Function		1 channel	
LCD controller/driver			Internal voltage boosting method, capacitor split method, and external resistance division method are switchable.	
	Segment signal output		44 (40) Note 1	56 (52) Note 1
	Common signal output		4 (8) Note 1	
Data transfer controller (DTC)			32 sources	33 sources
Event link controller (ELC)			Event input: 30, Event trigger output: 22	Event input: 31, Event trigger output: 22
Vectored interrupt sources	Internal		36	37
	External		9	9
Key interrupt			8	8
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 2 • Internal reset by RAM parity error • Internal reset by illegal-memory access	
Power-on-reset circuit			• Power-on-reset: 1.51 ± 0.03 V • Power-down-reset: 1.50 ± 0.03 V	
Voltage detector			• Rising edge: 1.67 V to 3.13 V (12 stages) • Falling edge: 1.63 V to 3.06 V (12 stages)	
On-chip debug function			Provided	
Power supply voltage			V_{DD} = 1.6 to 3.6 V (T_A = -40 to +85°C) V_{DD} = 2.4 to 3.6 V (T_A = -40 to +105°C)	
Operating ambient temperature			T_A = -40 to +85°C (A: Consumer applications), T_A = -40 to +105°C (G: Industrial applications)	

Note 1. The number in parentheses indicates the number of signal outputs when 8 coms are used.

Note 2. The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution not is issued by emulation with the in-circuit emulator or on-chip debug emulator.

[80/85-pin, 100-pin products (without USB)] **(1/2)**

Item		80/85-pin	100-pin
		R5F111Mx/R5F111Nx (x = E to H, J)	R5F111Px (x = E to H, J)
Code flash memory (KB)		64 to 256	64 to 256
Data flash memory (KB)		8	8
RAM (KB)		8 to 16 Note 1	8 to 16 Note 1
Memory space		1 MB	
Main system clock	High-speed system clock	X1 (crystal/ceramic) oscillation, external main system clock input (EXCLK) 1 to 20 MHz: VDD = 2.7 to 3.6 V, 1 to 8 MHz: VDD = 1.8 to 2.7 V, 1 to 4 MHz: VDD = 1.6 to 1.8 V	
	High-speed on-chip oscillator clock	HS (high-speed main) operation mode: 1 to 24 MHz (VDD = 2.7 to 3.6 V), HS (high-speed main) operation mode: 1 to 16 MHz (VDD = 2.4 to 3.6 V, LS (low-speed main) operation mode: 1 to 8 MHz (VDD = 1.8 to 3.6 V), LV (low-voltage main) operation mode: 1 to 4 MHz (VDD = 1.6 to 3.6 V)	
Subsystem clock		XT1 (crystal) oscillation, external subsystem clock input (EXCLKS) 32.768 kHz (TYP.): VDD = 1.6 to 3.6 V	
Low-speed on-chip oscillator clock		15 kHz (TYP.): VDD = 1.6 to 3.6 V	
General-purpose register		8 bits × 32 registers (8 bits × 8 registers × 4 banks)	
Minimum instruction execution time		0.04167 μs (High-speed on-chip oscillator clock: fHOCO = fIH = 24 MHz operation)	
		0.05 μs (High-speed system clock: fMX = 20 MHz operation)	
		30.5 μs (Subsystem clock: fSUB = 32.768 kHz operation)	
Instruction set		• Data transfer (8/16 bits) • Adder and subtractor/logical operation (8/16 bits) • Multiplication (8 bits × 8 bits, 16 bits × 16 bits), Division (16 bits ÷ 16 bits, 32 bits ÷ 32 bits) • Multiplication and Accumulation (16 bits × 16 bits + 32 bits) • Rotate, barrel shift, and bit manipulation (Set, reset, test, and Boolean operation), etc.	
I/O port	Total	63	81
	CMOS I/O	55	73
	CMOS input	5	5
	CMOS output	1	1
	N-ch open-drain I/O (6 V tolerance)	2	2
Timer	16-bit timer TAU	8 channels (with 1 channel remote control output function) (Timer outputs: 8, PWM outputs: 7 Note 2)	
	16-bit timer KB2	3 channels (PWM outputs: 6)	
	Watchdog timer	1 channel	
	12-bit interval timer	1 channel	
	Real-time clock 2	1 channel	
	RTC output	1 1 Hz (subsystem clock: fSUB = 32.768 kHz)	

Note 1. In the case of the 16 KB, this is about 15 KB when the self-programming function and data flash function are used (For details, see **CHAPTER 3**).

Note 2. The number of outputs varies, depending on the setting of channels in use and the number of the master.

(2/2)

Item			80/85-pin	100-pin
			R5F111Mx/R5F111Nx (x = E to H, J)	R5F111Px (x = E to H, J)
Clock output/buzzer output			2	2
			• 2.44 kHz, 4.88 kHz, 9.76 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz (Main system clock: f_{MAIN} = 20 MHz operation) • 256 Hz, 512 Hz, 1.024 kHz, 2.048 kHz, 4.096 kHz, 8.192 kHz, 16.384 kHz, 32.768 kHz (Subsystem clock: f_{SUB} = 32.768 kHz operation)	
8/12-bit resolution A/D converter			11 channels	13 channels
D/A converter			2 channels	2 channels
Comparator			1 channel	2 channels
Serial interface			• Simplified SPI (CSI): 1 channel/UART (UART supporting LIN-bus): 1 channel/simplified I^2C: 1 channel • Simplified SPI (CSI): 1 channel/UART: 1 channel/simplified I^2C: 1 channel • Simplified SPI (CSI): 1 channel/UART: 1 channel/simplified I^2C: 1 channel • Simplified SPI (CSI): 1 channel/UART: 1 channel/simplified I^2C: 1 channel	
	I^2C bus		1 channel	1 channel
LCD controller/driver			Internal voltage boosting method, capacitor split method, and external resistance division method are switchable.	
	Segment signal output		44 (40) Note 1	56 (52) Note 1
	Common signal output		4 (8) Note 1	
Data transfer controller (DTC)			30 sources	31 sources
Event link controller (ELC)			Event input: 30, Event trigger output: 22	Event input: 31, Event trigger output: 22
Vectored interrupt sources		Internal	32	33
		External	9	9
Key interrupt			8	8
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 2 • Internal reset by RAM parity error • Internal reset by illegal-memory access	
Power-on-reset circuit			• Power-on-reset: 1.51 ± 0.03 V • Power-down-reset: 1.50 ± 0.03 V	
Voltage detector			• Rising edge: 1.67 V to 3.13 V (12 stages) • Falling edge: 1.63 V to 3.06 V (12 stages)	
On-chip debug function			Provided	
Power supply voltage			V_{DD} = 1.6 to 3.6 V (T_A = -40 to +85°C) V_{DD} = 2.4 to 3.6 V (T_A = -40 to +105°C)	
Operating ambient temperature			T_A = -40 to +85°C (A: Consumer applications), T_A = -40 to +105°C (G: Industrial applications)	

Note 1. The number in parentheses indicates the number of signal outputs when 8 coms are used.

Note 2. The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution not is issued by emulation with the in-circuit emulator or on-chip debug emulator.

CHAPTER 2 PIN FUNCTIONS

2.1 Port Function

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

Table 2 - 1 Pin I/O Buffer Power Supplies

(1) 80/85-pin products (with USB)

Power Supply	Corresponding Pins
V_{DD0}	• Port pins other than P150 to P152 • Pins other than port pins
AV_{DD}	• P150 to P152
UV_{BUS}	• UDP, UDM, U_{REGC}

(2) 80/85-pin products (without USB)

Power Supply	Corresponding Pins
V_{DD0}	• Port pins other than P150 to P152, P155, and P156 • Pins other than port pins
AV_{DD}	• P150 to P152, P155, and P156

(3) 100-pin products (with USB)

Power Supply	Corresponding Pins
V_{DD0}, V_{DD1}	• Port pins other than P150 to P156 • Pins other than port pins
AV_{DD}	• P150 to P156
UV_{BUS}	• UDP, UDM, U_{REGC}

(4) 100-pin products (without USB)

Power Supply	Corresponding Pins
V_{DD0}, V_{DD1}	• Port pins other than P150 to P156 • Pins other than port pins
AV_{DD}	• P150 to P156

Caution Set V_{DD0} and V_{DD1} to the same potential.

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

2.1.1 80/85-pin products (with USB)

(1/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	8-5-10	I/O	Digital input invalid Note 1	SCK10/SCL10/SEG48	Port 0. 8-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. Input of P00 and P01 can be set to TTL input buffer. Output of P00 to P02 can be set to N-ch open-drain output (V_{DD} tolerance).
P01				SI10/RxD1/SDA10/ SEG49	
P02	7-5-10			SO10/TxD1/ (PCLBUZ0)/SEG50	
P03	7-5-4			TI00/TO00/INTP1/ SEG51	
P04				INTP2/SEG52	
P05				TI02/TO02/SEG53	
P06				INTP5/SEG54	
P07				TI06/TO06/SEG55	
P10	8-5-10	I/O	Digital input invalid Note 1	INTP7/PCLBUZ0/ SCK20/SCL20/SEG40	Port 1. 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. 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).
P11				RxD2/SI20/SDA20/ SEG41/VCOUT0	
P12	7-5-10			TxD2/SO20/SEG42	
P20	7-10-3	I/O	Analog input	ANI20/SEG32	Port 2. 8-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. P20 and P21 can be set to analog input Note 2. Input of P24 and P25 can be set to TTL input buffer. Output of P24 to P26 can be set to N-ch open-drain output (V_{DD} tolerance).
P21				ANI21/SEG33	
P22	7-5-4		Digital input invalid Note 1	TI04/TO04/SEG34	
P23				TI07/TO07/SEG35	
P24	8-5-10			SCK00/SCL00/SEG36	
P25				SI00/RxD0/TOOLRxD/ SDA00/SEG37	
P26	7-5-10			SO00/TxD0/TOOLTxD/ SEG38	
P27	7-5-4			TI05/TO05/(INTP5)/ PCLBUZ1/SEG39	

Note 1. "Digital input invalid" refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

Note 2. Each pin can be specified as either digital or analog by setting port mode control register x (PMCx) (Can be specified in 1-bit units).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).
For details, see **Figure 4 - 8 Format of Peripheral I/O redirection register (PIOR)**.

(2/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P30	7-5-4	I/O	Digital input invalid Note 1	TI03/TO03/REMOOUT/ SEG20	Port 3. 6-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. Input of P33 and P34 can be set to TTL input buffer. Output of P33 to P35 can be set to N-ch open-drain output (V_{DD} tolerance).
P31				INTP3/RTC1HZ/SEG21	
P32				TI01/TO01/SEG22	
P33	8-5-10			INTP4/SCK30/SCL30/SEG23	
P34				SI30/RxD3/SDA30/ SEG24	
P35	7-5-10			SO30/TxD3/SEG25	
P40	7-1-3	I/O	Input port	TOOL0/(TI00)/(TO00)	Port 4. 5-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. P43 to P46 can be set to analog input Note 2.
P43	8-3-4			(INTP7)/IVCMP0	
P44				IVREF0	
P45	7-4-1			ANO0	
P46				ANO1	
P50	7-5-4	I/O	Digital input invalid Note 1	SEG4/INTP6	Port 5. 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.
P51				SEG5	
P52				SEG6	
<R> P60	12-1-3	I/O	Input port	SCLA0/(TI01)/(TO01)	Port 6. 2-bit I/O port. Input/output can be specified in 1-bit units. Output of P60 and P61 can be set to N-ch open-drain output (6 V tolerance).
P61				SDAA0/(TI02)/(TO02)	
P70	7-5-4	I/O	Digital input invalid Note 1	KR7/SEG12	Port 7. 8-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.
P71				KR6/SEG13	
P72				KR5/TKBO20/SEG14	
P73				KR4/TKBO21/SEG15	
P74				KR3/TKBO10/SEG16	
P75				KR2/TKBO11/SEG17	
P76				KR1/TKBO00/SEG18	
P77				KR0/TKBO01/SEG19	

Note 1. "Digital input invalid" refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

Note 2. Each pin can be specified as either digital or analog by setting port mode control register x (PMCx) (Can be specified in 1-bit units).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).
For details, see **Figure 4 - 8 Format of Peripheral I/O redirection register (PIOR)**.

(3/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P121	2-2-1	I/O	Input port	X1	Port 12. 3-bit I/O port and 4-bit input only port. For only P125 to P127, input/output can be specified. For only P125 to P127, use of an on-chip pull-up resistor can be specified by a software setting at input port.
P122				X2/EXCLK	
P123				XT1	
P124				XT2/EXCLKS	
P125	7-5-6	I/O	Digital input invalid Note 1	VL3/(TI06)/(TO06)	
P126	7-5-5			CAPL/(TI04)/(TO04)	
P127				CAPH/(TI03)/(TO03)/(REMOOUT)	
P130	1-1-2	Output	Output port	—	Port 13. 1-bit output port and 1-bit input port.
P137	2-1-2	Input	Input port	INTP0	
P140	7-10-3	I/O	Analog input	ANI16/SEG28	Port 14. 4-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. P140 to P143 can be set to analog input Note 2.
P141				ANI17/SEG29	
P142				ANI18/SEG30	
P143				ANI19/SEG31	
P150	4-3-3	I/O	Analog input	ANI0/AVREFP	Port 15. 3-bit I/O port. Input/output can be specified in 1-bit units. P150 to P152 can be set to analog input Note 3.
P151				ANI1/AVREFM	
P152				ANI2	
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for external reset. Connect to VDD directly or via a resistor when external reset is not used.
COM0 to COM3	18-5-1	Output	Output port	—	COM-only pin.
COM4				SEG0	
COM5				SEG1	
COM6				SEG2	
COM7				SEG3	
UVBUS (products with USB)	17-11-1	Input	—	—	USB cable connection monitor pin and positive power supply pin for the USB transceiver. Connect to the VBUS pin of the USB bus.
UDM (products with USB)	18-11-1	I/O	—	—	D- I/O pin of the USB port. Connect to the D- pin of the USB bus.
UDP (products with USB)				—	D+ I/O pin of the USB port. Connect to the D+ pin of the USB bus.

Note 1. "Digital input invalid" refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

Note 2. Each pin can be specified as either digital or analog by setting port mode control register x (PMCx) (Can be specified in 1-bit units).

Note 3. 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 (PIOR).
For details, see **Figure 4 - 8 Format of Peripheral I/O redirection register (PIOR)**.

2.1.2 80/85-pin products (without USB)

(1/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	8-5-10	I/O	Digital input invalid Note 1	SCK10/SCL10/SEG48	Port 0. 8-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. Input of P00 and P01 can be set to TTL input buffer. Output of P00 to P02 can be set to N-ch open-drain output (V_{DD} tolerance).
P01				SI10/RxD1/SDA10/ SEG49	
P02	7-5-10			SO10/TxD1/ (PCLBUZ0)/SEG50	
P03	7-5-4			TI00/TO00/INTP1/ SEG51	
P04				INTP2/SEG52	
P05				TI02/TO02/SEG53	
P06				INTP5/SEG54	
P07				TI06/TO06/SEG55	
P10	8-5-10	I/O	Digital input invalid Note 1	INTP7/PCLBUZ0/ SCK20/SCL20/SEG40	Port 1. 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. 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).
P11				RxD2/SI20/SDA20/ SEG41/VCOUT0	
P12	7-5-10			TxD2/SO20/SEG42	
P20	7-10-3	I/O	Analog input	ANI20/SEG32	Port 2. 8-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. P20 and P21 can be set to analog input Note 2. Input of P24 and P25 can be set to TTL input buffer. Output of P24 to P26 can be set to N-ch open-drain output (V_{DD} tolerance).
P21				ANI21/SEG33	
P22	7-5-4		Digital input invalid Note 1	TI04/TO04/SEG34	
P23				TI07/TO07/SEG35	
P24	8-5-10			SCK00/SCL00/SEG36	
P25				SI00/RxD0/TOOLRxD/ SDA00/SEG37	
P26	7-5-10			SO00/TxD0/TOOLTxD/ SEG38	
P27	7-5-4			TI05/TO05/(INTP5)/ PCLBUZ1/SEG39	

Note 1. "Digital input invalid" refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

Note 2. Each pin can be specified as either digital or analog by setting port mode control register x (PMCx) (Can be specified in 1-bit units).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).
For details, see **Figure 4 - 8 Format of Peripheral I/O redirection register (PIOR)**.

(2/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P30	7-5-4	I/O	Digital input invalid Note 1	TI03/TO03/REMOOUT/ SEG20	Port 3. 6-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. Input of P33 and P34 can be set to TTL input buffer. Output of P33 to P35 can be set to N-ch open-drain output (V_{DD} tolerance).
P31				INTP3/RTC1HZ/SEG21	
P32				TI01/TO01/SEG22	
P33	8-5-10			INTP4/SCK30/SCL30/SEG23	
P34				SI30/RxD3/SDA30/ SEG24	
P35	7-5-10			SO30/TxD3/SEG25	
P40	7-1-3	I/O	Input port	TOOL0/(TI00)/(TO00)	Port 4. 5-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. P43 to P46 can be set to analog input Note 2.
P43	8-3-4			(INTP7)/IVCMP0	
P44				IVREF0	
P45	7-4-1			ANO0	
P46				ANO1	
P50	7-5-4	I/O	Digital input invalid Note 1	SEG4/INTP6	Port 5. 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.
P51				SEG5	
P52				SEG6	
<R> P60	12-1-3	I/O	Input port	SCLA0/(TI01)/(TO01)	Port 6. 2-bit I/O port. Input/output can be specified in 1-bit units. Output of P60 and P61 can be set to N-ch open-drain output (6 V tolerance).
P61				SDAA0/(TI02)/(TO02)	
P70	7-5-4	I/O	Digital input invalid Note 1	KR7/SEG12	Port 7. 8-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.
P71				KR6/SEG13	
P72				KR5/TKBO20/SEG14	
P73				KR4/TKBO21/SEG15	
P74				KR3/TKBO10/SEG16	
P75				KR2/TKBO11/SEG17	
P76				KR1/TKBO00/SEG18	
P77				KR0/TKBO01/SEG19	
P82	7-1-3	I/O		—	Port 8. 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.
P83				—	

Note 1. "Digital input invalid" refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

Note 2. Each pin can be specified as either digital or analog by setting port mode control register x (PMCx) (Can be specified in 1-bit units).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).
For details, see **Figure 4 - 8 Format of Peripheral I/O redirection register (PIOR)**.

(3/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P121	2-2-1	I/O	Input port	X1	Port 12. 3-bit I/O port and 4-bit input only port. For only P125 to P127, input/output can be specified. For only P125 to P127, use of an on-chip pull-up resistor can be specified by a software setting at input port.
P122				X2/EXCLK	
P123				XT1	
P124				XT2/EXCLKS	
P125	7-5-6	I/O	Digital input invalid Note 1	VL3/(TI06)/(TO06)	
P126	7-5-5			CAPL/(TI04)/(TO04)	
P127				CAPH/(TI03)/(TO03)/(REMOOUT)	
P130	1-1-2	Output	Output port	—	Port 13. 1-bit output port and 1-bit input port.
P137	2-1-2	Input	Input port	INTP0	
P140	7-10-3	I/O	Analog input	ANI16/SEG28	Port 14. 4-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. P140 to P143 can be set to analog input Note 2.
P141				ANI17/SEG29	
P142				ANI18/SEG30	
P143				ANI19/SEG31	
P150	4-3-3	I/O	Analog input	ANI0/AVREFP	Port 15. 5-bit I/O port. Input/output can be specified in 1-bit units. P150 to P152, P155, and P156 can be set to analog input Note 3.
P151				ANI1/AVREFM	
P152				ANI2	
P155				ANI5	
P156				ANI6	
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for external reset. Connect to VDD directly or via a resistor when external reset is not used.
COM0 to COM3	18-5-1	Output	Output port	—	COM-only pin.
COM4				SEG0	
COM5				SEG1	
COM6				SEG2	
COM7				SEG3	

Note 1. “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

Note 2. Each pin can be specified as either digital or analog by setting port mode control register x (PMCx) (Can be specified in 1-bit units).

Note 3. 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 (PIOR).
For details, see **Figure 4 - 8 Format of Peripheral I/O redirection register (PIOR)**.

2.1.3 100-pin products (with USB)

(1/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	8-5-10	I/O	Digital input invalid Note 1	SCK10/SCL10/SEG48	Port 0. 8-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. Input of P00 and P01 can be set to TTL input buffer. Output of P00 to P02 can be set to N-ch open-drain output (V_{DD} tolerance).
P01				SI10/RxD1/SDA10/ SEG49	
P02	7-5-10			SO10/TxD1/ (PCLBUZ0)/SEG50	
P03	7-5-4			TI00/TO00/INTP1/ SEG51	
P04				INTP2/SEG52	
P05				TI02/TO02/SEG53	
P06				INTP5/SEG54	
P07				TI06/TO06/SEG55	
P10	8-5-10	I/O	Digital input invalid Note 1	INTP7/PCLBUZ0/ SCK20/SCL20/SEG40	Port 1. 8-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. 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).
P11				RxD2/SI20/SDA20/ SEG41/VCOUT0	
P12	7-5-10			TxD2/SO20/SEG42	
P13	7-5-4			SEG43	
P14				SEG44	
P15				SEG45	
P16				SEG46	
P17				SEG47	
P20	7-10-3	I/O	Analog input	ANI20/SEG32	Port 2. 8-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. P20 and P21 can be set to analog input Note 2. Input of P24 and P25 can be set to TTL input buffer. Output of P24 to P26 can be set to N-ch open-drain output (V_{DD} tolerance).
P21				ANI21/SEG33	
P22	7-5-4		Digital input invalid Note 1	TI04/TO04/SEG34	
P23				TI07/TO07/SEG35	
P24	8-5-10			SCK00/SCL00/SEG36	
P25				SI00/RxD0/TOOLRxD/ SDA00/SEG37	
P26	7-5-10			SO00/TxD0/TOOLTxD/ SEG38	
P27	7-5-4			TI05/TO05/(INTP5)/ PCLBUZ1/SEG39	

Note 1. "Digital input invalid" refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

Note 2. Each pin can be specified as either digital or analog by setting port mode control register x (PMCx) (Can be specified in 1-bit units).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).
For details, see **Figure 4 - 8 Format of Peripheral I/O redirection register (PIOR)**.

(2/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P30	7-5-4	I/O	Digital input invalid Note 1	P30/TI03/TO03/ REMOOUT/SEG20	Port 3. 8-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. Input of P33 and P34 can be set to TTL input buffer. Output of P33 to P35 can be set to N-ch open-drain output (V_{DD} tolerance).
P31				INTP3/RTC1HZ/SEG21	
P32				TI01/TO01/SEG22	
P33	8-5-10			INTP4/SCK30/SCL30/ SEG23	
P34				SI30/RxD3/SDA30/ SEG24	
P35	7-5-10			SO30/TxD3/SEG25	
P36	7-5-4			SEG26	
P37				SEG27	
P40	7-1-3	I/O	Input port	TOOL0/(TI00)/(TO00)	Port 4. 7-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. P41 to P46 can be set to analog input Note 2. Input of P43 and P44 can be set to TTL input buffer. Output of P42 to P44 can be set to N-ch open-drain output (V_{DD} tolerance).
P41	7-4-1			(TI07)/(TO07)/IVREF1	
P42	7-3-4			TI05/TO05/(SO10)/ (TxD1)/IVCMP1	
P43	8-3-4			(INTP7)/(SI10)/(RxD1)/ (SDA10)/IVCMP0	
P44				(SCK10)/(SCL10)/ IVREF0	
P45	7-4-1			ANO0	
P46				ANO1	
P50	7-5-4	I/O	Digital input invalid Note 1	SEG4/INTP6	Port 5. 8-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.
P51				SEG5	
P52				SEG6	
P53				SEG7	
P54				SEG8	
P55				SEG9	
P56				SEG10	
P57				SEG11	
<R> P60	12-1-3	I/O	Input port	SCLA0/(TI01)/(TO01)	Port 6. 2-bit I/O port. Input/output can be specified in 1-bit units. Output of P60 and P61 is N-ch open-drain output (6 V tolerance).
P61				SDAA0/(TI02)/(TO02)	

Note 1. "Digital input invalid" refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

Note 2. Each pin can be specified as either digital or analog by setting port mode control register x (PMCx) (Can be specified in 1-bit units).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).
For details, see **Figure 4 - 8 Format of Peripheral I/O redirection register (PIOR)**.

(3/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P70	7-5-4	I/O	Digital input invalid Note 1	KR7/SEG12	Port 7. 8-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.
P71				KR6/SEG13	
P72				KR5/TKBO20/SEG14	
P73				KR4/TKBO21/SEG15	
P74				KR3/TKBO10/SEG16	
P75				KR2/TKBO11/SEG17	
P76				KR1/TKBO00/SEG18	
P77				KR0/TKBO01/SEG19	
P121	2-2-1	Input	Input port	X1	Port 12. 3-bit I/O port and 4-bit input only port. For only P125 to P127, input/output can be specified. For only P125 to P127, use of an on-chip pull-up resistor can be specified by a software setting at input port.
P122				X2/EXCLK	
P123				XT1	
P124				XT2/EXCLKS	
P125	7-5-6	I/O	Digital input invalid Note 1	VL3/(TI06)/(TO06)	
P126	7-5-5			CAPL/(TI04)/(TO04)	
P127				CAPH/(TI03)/(TO03)/(REMOOUT)	
P130	1-1-2	Output	Output port	—	Port 13. 1-bit output port and 1-bit input port.
P137	2-1-2	Input	Input port	INTP0	
P140	7-10-3	I/O	Analog input	ANI16/SEG28	Port 14. 4-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. P140 to P143 can be set to analog input Note 2.
P141				ANI17/SEG29	
P142				ANI18/SEG30	
P143				ANI19/SEG31	
P150	4-3-3	I/O	Analog input	ANI0/AVREFP	Port 15. 7-bit I/O port. Input/output can be specified in 1-bit units. P150 to P156 can be set to analog input Note 3.
P151				ANI1/AVREFM	
P152				ANI2	
P153				ANI3	
P154				ANI4	
P155				ANI5	
P156				ANI6	
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for external reset. Connect to VDD directly or via a resistor when external reset is not used.
COM0 to COM3	18-5-1	Output	Output port	—	COM-only pin.
COM4				SEG0	
COM5				SEG1	
COM6				SEG2	
COM7				SEG3	
UVBUS (products with USB)	17-11-1	Input	—	—	USB cable connection monitor pin and positive power supply pin for the USB transceiver. Connect to the VBUS pin of the USB bus.
UDM (products with USB)	18-11-1	I/O	—	—	D- I/O pin of the USB port. Connect to the D- pin of the USB bus.
UDP (products with USB)				—	D+ I/O pin of the USB port. Connect to the D+ pin of the USB bus.

Note 1. "Digital input invalid" refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

Note 2. Each pin can be specified as either digital or analog by setting port mode control register x (PMCx) (Can be specified in 1-bit units).

Note 3. 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 (PIOR).
For details, see **Figure 4 - 8 Format of Peripheral I/O redirection register (PIOR)**.

2.1.4 100-pin products (without USB)

(1/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	8-5-10	I/O	Digital input invalid Note 1	SCK10/SCL10/SEG48	Port 0. 8-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. Input of P00 and P01 can be set to TTL input buffer. Output of P00 to P02 can be set to N-ch open-drain output (V_{DD} tolerance).
P01				SI10/RxD1/SDA10/ SEG49	
P02	7-5-10			SO10/TxD1/ (PCLBUZ0)/SEG50	
P03	7-5-4			TI00/TO00/INTP1/ SEG51	
P04				INTP2/SEG52	
P05				TI02/TO02/SEG53	
P06				INTP5/SEG54	
P07				TI06/TO06/SEG55	
P10	8-5-10	I/O	Digital input invalid Note 1	INTP7/PCLBUZ0/ SCK20/SCL20/SEG40	Port 1. 8-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. 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).
P11				RxD2/SI20/SDA20/ SEG41/VCOUT0	
P12	7-5-10			TxD2/SO20/SEG42	
P13	7-5-4			SEG43	
P14				SEG44	
P15				SEG45	
P16				SEG46	
P17				SEG47	
P20	7-10-3	I/O	Analog input	ANI20/SEG32	Port 2. 8-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. P20 and P21 can be set to analog input Note 2. Input of P24 and P25 can be set to TTL input buffer. Output of P24 to P26 can be set to N-ch open-drain output (V_{DD} tolerance).
P21				ANI21/SEG33	
P22	7-5-4		Digital input invalid Note 1	TI04/TO04/SEG34	
P23				TI07/TO07/SEG35	
P24	8-5-10			SCK00/SCL00/SEG36	
P25				SI00/RxD0/TOOLRxD/ SDA00/SEG37	
P26	7-5-10			SO00/TxD0/TOOLTxD/ SEG38	
P27	7-5-4			TI05/TO05/(INTP5)/ PCLBUZ1/SEG39	

Note 1. "Digital input invalid" refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

Note 2. Each pin can be specified as either digital or analog by setting port mode control register x (PMCx) (Can be specified in 1-bit units).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).
For details, see **Figure 4 - 8 Format of Peripheral I/O redirection register (PIOR)**.

(2/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P30	7-5-4	I/O	Digital input invalid Note 1	P30/TI03/TO03/ REMOOUT/SEG20	Port 3. 8-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. Input of P33 and P34 can be set to TTL input buffer. Output of P33 to P35 can be set to N-ch open-drain output (V_{DD} tolerance).
P31				INTP3/RTC1HZ/SEG21	
P32				TI01/TO01/SEG22	
P33	8-5-10			INTP4/SCK30/SCL30/ SEG23	
P34				SI30/RxD3/SDA30/ SEG24	
P35	7-5-10			SO30/TxD3/SEG25	
P36	7-5-4			SEG26	
P37				SEG27	
P40	7-1-3	I/O	Input port	TOOL0/(TI00)/(TO00)	Port 4. 7-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. P41 to P46 can be set to analog input Note 2. Input of P43 and P44 can be set to TTL input buffer. Output of P42 to P44 can be set to N-ch open-drain output (V_{DD} tolerance).
P41	7-4-1			(TI07)/(TO07)/IVREF1	
P42	7-3-4			TI05/TO05/(SO10)/ (TxD1)/IVCMP1	
P43	8-3-4			(INTP7)/(SI10)/(RxD1)/ (SDA10)/IVCMP0	
P44				(SCK10)/(SCL10)/ IVREF0	
P45	7-4-1			ANO0	
P46				ANO1	
P50	7-5-4	I/O	Digital input invalid Note 1	SEG4/INTP6	Port 5. 8-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.
P51				SEG5	
P52				SEG6	
P53				SEG7	
P54				SEG8	
P55				SEG9	
P56				SEG10	
P57				SEG11	
<R> P60	12-1-3	I/O	Input port	SCLA0/(TI01)/(TO01)	Port 6. 2-bit I/O port. Input/output can be specified in 1-bit units. Output of P60 and P61 is N-ch open-drain output (6 V tolerance).
P61				SDAA0/(TI02)/(TO02)	

Note 1. "Digital input invalid" refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

Note 2. Each pin can be specified as either digital or analog by setting port mode control register x (PMCx) (Can be specified in 1-bit units).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).
For details, see **Figure 4 - 8 Format of Peripheral I/O redirection register (PIOR)**.

(3/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P70	7-5-4	I/O	Digital input invalid Note 1	KR7/SEG12	Port 7. 8-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.
P71				KR6/SEG13	
P72				KR5/TKBO20/SEG14	
P73				KR4/TKBO21/SEG15	
P74				KR3/TKBO10/SEG16	
P75				KR2/TKBO11/SEG17	
P76				KR1/TKBO00/SEG18	
P77				KR0/TKBO01/SEG19	
P80	7-1-3	I/O	Input port	—	Port 8. 4-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.
P81				—	
P82				—	
P83				—	
P121	2-2-1	Input	Input port	X1	Port 12. 3-bit I/O port and 4-bit input only port. For only P125 to P127, input/output can be specified. For only P125 to P127, use of an on-chip pull-up resistor can be specified by a software setting at input port.
P122				X2/EXCLK	
P123				XT1	
P124				XT2/EXCLKS	
P125	7-5-6	I/O	Digital input invalid Note 1	VL3/(TI06)/(TO06)	
P126	7-5-5			CAPL/(TI04)/(TO04)	
P127				CAPH/(TI03)/(TO03)/(REMOOUT)	
P130	1-1-2	Output	Output port	—	Port 13. 1-bit output port and 1-bit input port.
P137	2-1-2	Input	Input port	INTP0	
P140	7-10-3	I/O	Analog input	ANI16/SEG28	Port 14. 4-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. P140 to P143 can be set to analog input Note 2.
P141				ANI17/SEG29	
P142				ANI18/SEG30	
P143				ANI19/SEG31	
P150	4-3-3	I/O	Analog input	ANI0/AVREFP	Port 15. 7-bit I/O port. Input/output can be specified in 1-bit units. P150 to P156 can be set to analog input Note 3.
P151				ANI1/AVREFM	
P152				ANI2	
P153				ANI3	
P154				ANI4	
P155				ANI5	
P156				ANI6	
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for external reset. Connect to VDD directly or via a resistor when external reset is not used.
COM0 to COM3	18-5-1	Output	Output port	—	COM-only pin.
COM4				SEG0	
COM5				SEG1	
COM6				SEG2	
COM7				SEG3	

Note 1. "Digital input invalid" refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

Note 2. Each pin can be specified as either digital or analog by setting port mode control register x (PMCx) (Can be specified in 1-bit units).

Note 3. 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 (PIOR).
For details, see **Figure 4 - 8 Format of Peripheral I/O redirection register (PIOR)**.

2.2 Functions other than port pins

2.2.1 With functions for each product

(1/5)

Function Name	Products with USB		Products without USB	
	100-pin	80/85-pin	100-pin	80/85-pin
ANI0	√	√	√	√
ANI1	√	√	√	√
ANI2	√	√	√	√
ANI3	√	—	√	—
ANI4	√	—	√	—
ANI5	√	—	√	√
ANI6	√	—	√	√
ANI16	√	√	√	√
ANI17	√	√	√	√
ANI18	√	√	√	√
ANI19	√	√	√	√
ANI20	√	√	√	√
ANI21	√	√	√	√
ANO0	√	√	√	√
ANO1	√	√	√	√
INTP0	√	√	√	√
INTP1	√	√	√	√
INTP2	√	√	√	√
INTP3	√	√	√	√
INTP4	√	√	√	√
INTP5	√	√	√	√
INTP6	√	√	√	√
INTP7	√	√	√	√
IVCMP0	√	√	√	√
IVCMP1	√	—	√	—
IVREF0	√	√	√	√
IVREF1	√	—	√	—
KR0	√	√	√	√
KR1	√	√	√	√
KR2	√	√	√	√
KR3	√	√	√	√
KR4	√	√	√	√
KR5	√	√	√	√
KR6	√	√	√	√
KR7	√	√	√	√
PCLBUZ0	√	√	√	√
PCLBUZ1	√	√	√	√
REGC	√	√	√	√
RTC1HZ	√	√	√	√

(2/5)

Function Name	Products with USB		Products without USB	
	100-pin	80/85-pin	100-pin	80/85-pin
REMOOUT	√	√	√	√
$\overline{\text{RESET}}$	√	√	√	√
RxD0	√	√	√	√
RxD1	√	√	√	√
RxD2	√	√	√	√
RxD3	√	√	√	√
SCK00	√	√	√	√
SCK10	√	√	√	√
SCK20	√	√	√	√
SCK30	√	√	√	√
SCLA0	√	√	√	√
SCL00	√	√	√	√
SCL10	√	√	√	√
SCL20	√	√	√	√
SCL30	√	√	√	√
SDAA0	√	√	√	√
SDA00	√	√	√	√
SDA10	√	√	√	√
SDA20	√	√	√	√
SDA30	√	√	√	√
SI00	√	√	√	√
SI10	√	√	√	√
SI20	√	√	√	√
SI30	√	√	√	√
SO00	√	√	√	√
SO10	√	√	√	√
SO20	√	√	√	√
SO30	√	√	√	√
TI00	√	√	√	√
TI01	√	√	√	√
TI02	√	√	√	√
TI03	√	√	√	√
TI04	√	√	√	√
TI05	√	√	√	√
TI06	√	√	√	√
TI07	√	√	√	√
TKBO00	√	√	√	√
TKBO01	√	√	√	√
TKBO10	√	√	√	√
TKBO11	√	√	√	√
TKBO20	√	√	√	√
TKBO21	√	√	√	√
TO00	√	√	√	√

(3/5)

Function Name	Products with USB		Products without USB	
	100-pin	80/85-pin	100-pin	80/85-pin
TO01	√	√	√	√
TO02	√	√	√	√
TO03	√	√	√	√
TO04	√	√	√	√
TO05	√	√	√	√
TO06	√	√	√	√
TO07	√	√	√	√
TxD0	√	√	√	√
TxD1	√	√	√	√
TxD2	√	√	√	√
TxD3	√	√	√	√
VCOUT0	√	√	√	√
VCOUT1	√	—	√	—
V_{L1}	√	√	√	√
V_{L2}	√	√	√	√
V_{L3}	√	√	√	√
V_{L4}	√	√	√	√
CAPH	√	√	√	√
CAPL	√	√	√	√
X1	√	√	√	√
X2	√	√	√	√
EXCLK	√	√	√	√
EXCLKS	√	√	√	√
XT1	√	√	√	√
XT2	√	√	√	√
V_{DD0}	√	√	√	√
V_{DD1}	√	—	√	—
AV_{DD}	√	√	√	√
AV_{REFP}	√	√	√	√
AV_{REFM}	√	√	√	√
V_{SS0}	√	√	√	√
V_{SS1}	√	—	√	—
AV_{SS}	√	√	√	√
TOOLRxD	√	√	√	√
TOOLTxD	√	√	√	√
TOOL0	√	√	√	√
UDP	√	√	—	—
UDM	√	√	—	—
UV_{BUS}	√	√	—	—
U_{REGC}	√	√	—	—
COM0	√	√	√	√
COM1	√	√	√	√
COM2	√	√	√	√

(4/5)

Function Name	Products with USB		Products without USB	
	100-pin	80/85-pin	100-pin	80/85-pin
COM3	√	√	√	√
COM4	√	√	√	√
COM5	√	√	√	√
COM6	√	√	√	√
COM7	√	√	√	√
SEG0	√	√	√	√
SEG1	√	√	√	√
SEG2	√	√	√	√
SEG3	√	√	√	√
SEG4	√	√	√	√
SEG5	√	√	√	√
SEG6	√	√	√	√
SEG7	√	—	√	—
SEG8	√	—	√	—
SEG9	√	—	√	—
SEG10	√	—	√	—
SEG11	√	—	√	—
SEG12	√	√	√	√
SEG13	√	√	√	√
SEG14	√	√	√	√
SEG15	√	√	√	√
SEG16	√	√	√	√
SEG17	√	√	√	√
SEG18	√	√	√	√
SEG19	√	√	√	√
SEG20	√	√	√	√
SEG21	√	√	√	√
SEG22	√	√	√	√
SEG23	√	√	√	√
SEG24	√	√	√	√
SEG25	√	√	√	√
SEG26	√	—	√	—
SEG27	√	—	√	—
SEG28	√	√	√	√
SEG29	√	√	√	√
SEG30	√	√	√	√
SEG31	√	√	√	√
SEG32	√	√	√	√
SEG33	√	√	√	√
SEG34	√	√	√	√
SEG35	√	√	√	√
SEG36	√	√	√	√
SEG37	√	√	√	√

(5/5)

Function Name	Products with USB		Products without USB	
	100-pin	80/85-pin	100-pin	80/85-pin
SEG38	√	√	√	√
SEG39	√	√	√	√
SEG40	√	√	√	√
SEG41	√	√	√	√
SEG42	√	√	√	√
SEG43	√	—	√	—
SEG44	√	—	√	—
SEG45	√	—	√	—
SEG46	√	—	√	—
SEG47	√	—	√	—
SEG48	√	√	√	√
SEG49	√	√	√	√
SEG50	√	√	√	√
SEG51	√	√	√	√
SEG52	√	√	√	√
SEG53	√	√	√	√
SEG54	√	√	√	√
SEG55	√	√	√	√

2.2.2 Description of Functions

(1/2)

Function Name	I/O	Function
ANI0 to ANI6, ANI16 to ANI21	Input	A/D converter analog input (See **Figure 12 - 45 Analog Input Pin Connection**.)
ANO0, ANO1	Output	D/A converter output
INTP0 to INTP7	Input	External interrupt request input Specified the valid edge: Rising edge, falling edge, or both rising and falling edges
IVCMP0, IVCMP1	Input	Comparator analog voltage input
IVREF0, IVREF1	Input	Comparator reference voltage input
VCOUT0, VCOUT1	Output	Comparator output
KR0 to KR7	Input	Key interrupt input
PCLBUZ0, PCLBUZ1	Output	Clock output/buzzer output
REGC	—	Pin for connecting regulator output stabilization capacitance for internal operation. Connect this pin to V_{SS} via a capacitor (0.47 to 1 μF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.
REMOOUT	Output	Remote controller output
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 to RxD3	Input	Serial data input pins of serial interfaces UART0, UART1, UART2, and UART3
TxD0 to TxD3	Output	Serial data output pins of serial interfaces UART0, UART1, UART2, and UART3
SCK00, SCK10, SCK20, SCK30	I/O	Serial clock I/O pins of serial interfaces CSI00, CSI10, CSI20, and CSI30
SI00, SI10, SI20, SI30	Input	Serial data input pins of serial interfaces CSI00, CSI10, CSI20, and CSI30
SO00, SO10, SO20, SO30	Output	Serial data output pins of serial interfaces CSI00, CSI10, CSI20, and CSI30
SCL00, SCL10, SCL20, SCL30	Output	Serial clock output pins of serial interfaces IIC00, IIC10, IIC20, and IIC30
SDA00, SDA10, SDA20, SDA30	I/O	Serial data I/O pins of serial interfaces IIC00, IIC10, IIC20, and IIC30
SCLA0	I/O	Serial clock I/O pins of serial interface IICA0
SDAA0	I/O	Serial data I/O pins of serial interface IICA0
TI00 to TI07	Input	The pins for inputting an external count clock/capture trigger to 16-bit timers 00 to 07
TO00 to TO07	Output	Timer output pins of 16-bit timers 00 to 07
TKBO00, TKBO01, TKBO10, TKBO11, TKBO20, TKBO21	Output	Timer output pins of 16-bit timer KB2
V_{L1} to V_{L4}	—	LCD drive voltage
CAPH, CAPL	—	Connecting a capacitor for LCD controller/driver
X1, X2	—	Resonator connection for main system clock
EXCLK	Input	External clock input for main system clock
XT1, XT2	—	Resonator connection for subsystem clock

(2/2)

Function Name	I/O	Function
EXCLKS	Input	External clock input for subsystem clock
V_{DD0}, V_{DD1}	—	Positive power supply for all pins
AV_{DD}	—	Positive power supply for A/D converter
AV_{REFP}	Input	A/D converter reference potential (+ side) input
AV_{REFM}	Input	A/D converter reference potential (- side) input
V_{SS0}, V_{SS1}	—	Ground potential for all pins
AV_{SS}	—	Ground potential for A/D converter
TOOLRxD	Input	UART reception pin for the external device connection used during flash memory programming
TOOLTxD	Output	UART transmission pin for the external device connection used during flash memory programming
TOOL0	I/O	Data I/O for flash memory programmer/debugger
COM0 to COM7	Output	LCD controller/driver common signal output
SEG0-SEG55	Output	LCD controller/driver segment signal output
UDP	I/O	USB data input/output (+ side)
UDM	I/O	USB data input/output (- side)
U_{REGC}	—	Pin for connecting USB regulator capacitance Connect this pin to V_{SS} via a capacitor (0.33 μF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.
UV_{BUS}	—	Positive power supply for the USB (VBUS connection)

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

Table 2 - 2 Relationship Between P40/TOOL0 and Operating Mode After Reset Release

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

For details, see 30.4 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} lines.

2.3 Connection of Unused Pins

Table 2 - 3 shows the connections of unused pins.

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

Table 2 - 3 Connection of Unused Pins

Pin Name	I/O	Recommended Connection of Unused Pins
P00 to P07	I/O	Digital input invalid: Independently connect to V_{DD0}, V_{DD1} or V_{SS0}, V_{SS1} via a resistor. Digital input: Independently connect to V_{DD0}, V_{DD1} or V_{SS0}, V_{SS1} via a resistor. Digital output: Leave open. Segment output: Leave open.
P10 to P17		
P20 to P27		
P30 to P37		
P40/TOOL0		Input: Independently connect to V_{DD0} via a resistor, or leave open. Output: Leave open.
P41 to P46		Input: Independently connect to V_{DD0}, V_{DD1} or V_{SS0}, V_{SS1} via a resistor. Output: Leave open.
P50 to P57		Digital input invalid: Independently connect to V_{DD0}, V_{DD1} or V_{SS0}, V_{SS1} via a resistor. Digital input: Independently connect to V_{DD0}, V_{DD1} or V_{SS0}, V_{SS1} via a resistor. Digital output: Leave open. Segment output: Leave open.
P60, P61		Input: Independently connect to V_{DD0}, V_{DD1} or V_{SS0}, V_{SS1} via a resistor. Output: Leave open.
P70 to P77		Digital input invalid: Independently connect to V_{DD0}, V_{DD1} or V_{SS0}, V_{SS1} via a resistor. Digital input: Independently connect to V_{DD0}, V_{DD1} or V_{SS0}, V_{SS1} via a resistor. Digital output: Leave open. Segment output: Leave open.
P80 to P83		Input: Independently connect to V_{DD0}, V_{DD1} or V_{SS0}, V_{SS1} via a resistor. Output: Leave open.
P121 to P124	Input	Independently connect to V_{DD0}, V_{DD1} or V_{SS0}, V_{SS1} via a resistor.
P125 to P127	I/O	Input: Independently connect to V_{DD0}, V_{DD1} or V_{SS0}, V_{SS1} via a resistor. Output: Leave open.
P130	Output	Leave open.
P137	Input	Independently connect to V_{DD0}, V_{DD1} or V_{SS0}, V_{SS1} via a resistor.
P140 to P143	I/O	Digital input invalid: Independently connect to V_{DD0}, V_{DD1} or V_{SS0}, V_{SS1} via a resistor. Digital input: Independently connect to V_{DD0}, V_{DD1} or V_{SS0}, V_{SS1} via a resistor. Digital output: Leave open. Segment output: Leave open.
P150 to P156		Input: Independently connect to AV_{DD} or AV_{SS} via a resistor. Output: Leave open.
$\overline{\text{RESET}}$	Input	Connect to V_{DD} directly or via a resistor.
REGC	—	Connect to V_{SS} via a capacitor (0.47 to 1 μF).
COM0 to COM7	Output	Leave open.
V_{L1}, V_{L2}, V_{L4}	—	Leave open.
U_{REGC} (products with USB)	—	Connect to V_{SS} via a capacitor (0.33 μF).
U_{VBUS} (products with USB)	Input	Independently connect to V_{SS0}, V_{SS1} via a resistor.
<R> UDM (products with USB)	I/O	Leave open.
UDP (products with USB)		

2.4 Pin Block Diagrams

For the pin types listed in **2.1.1 80/85-pin products (with USB)** to **2.1.4 100-pin products (without USB)**, pin block diagrams are shown in Figures 2 - 1 to 2 - 19.

Figure 2 - 1 Pin Block Diagram of Pin Type 1-1-2

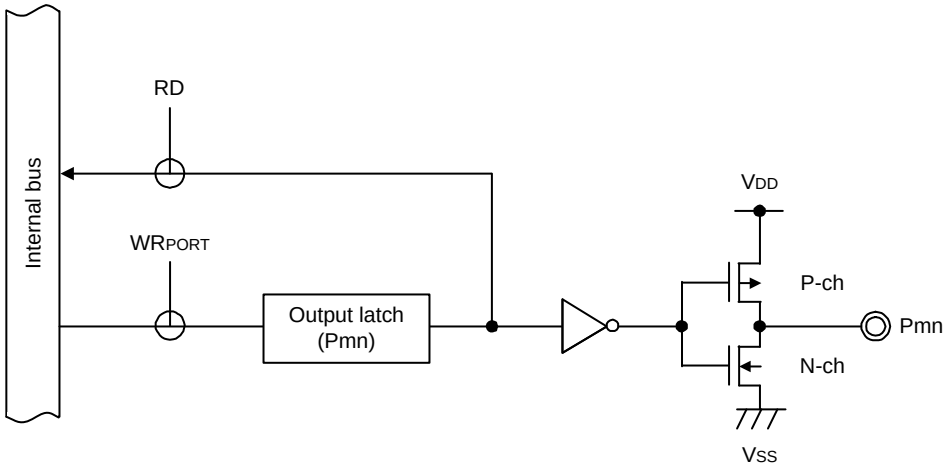


Figure 2 - 2 Pin Block Diagram of Pin Type 2-1-1

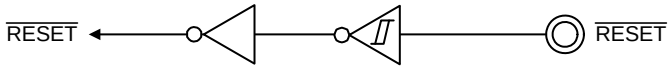


Figure 2 - 3 Pin Block Diagram of Pin Type 2-1-2

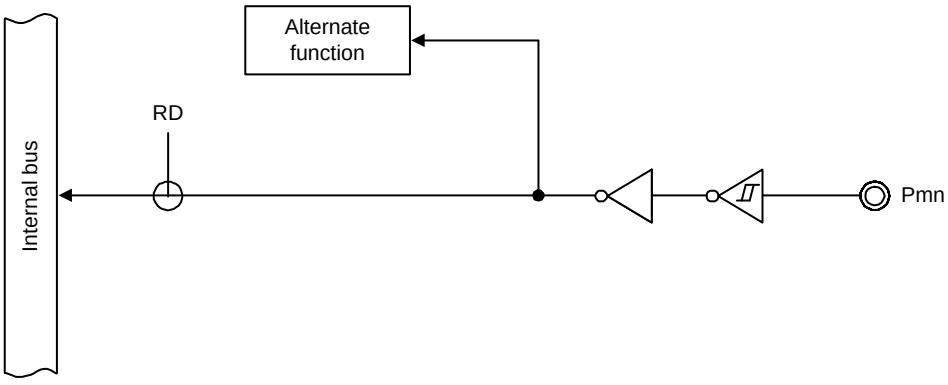
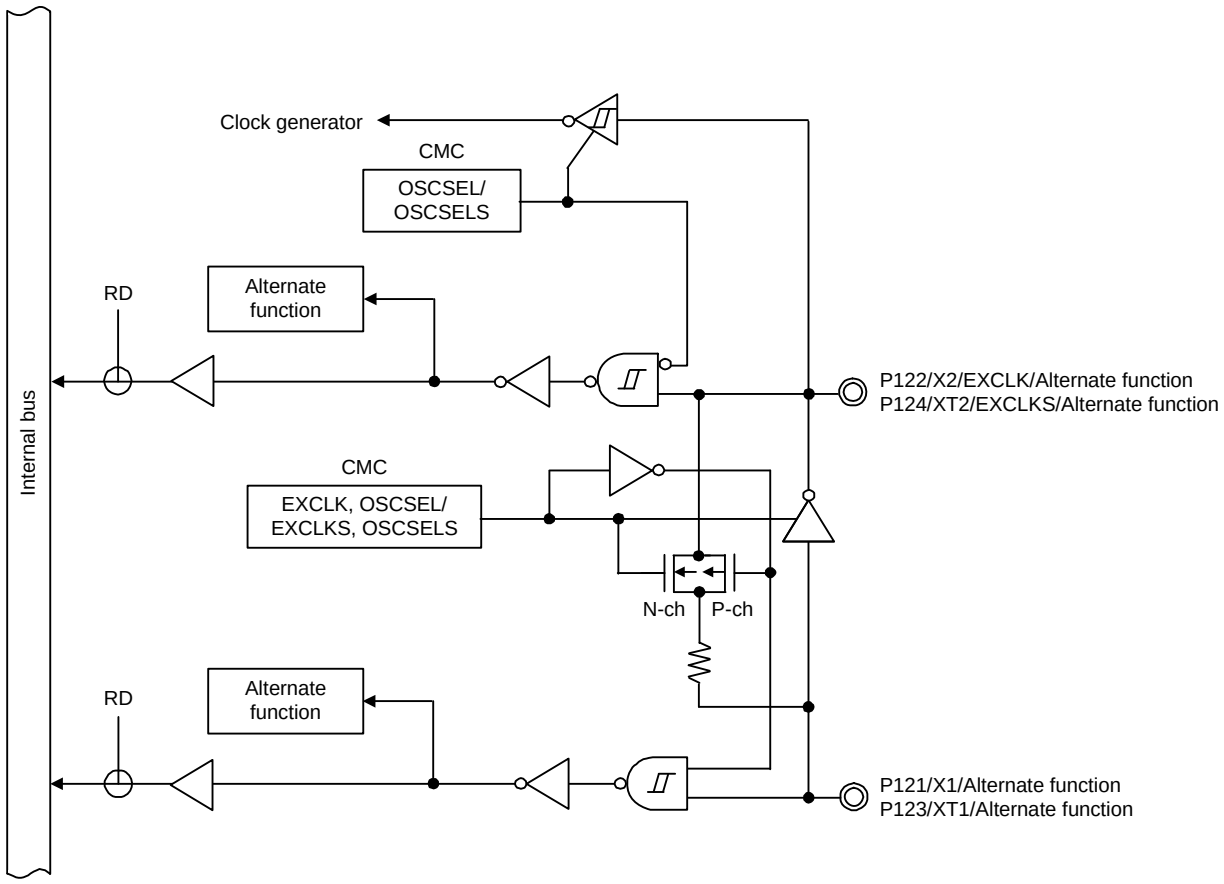


Figure 2 - 4 Pin Block Diagram of Pin Type 2-2-1



Remark Refer to **2.1 Port Function** for alternate functions.

Figure 2 - 5 Pin Block Diagram of Pin Type 4-3-3

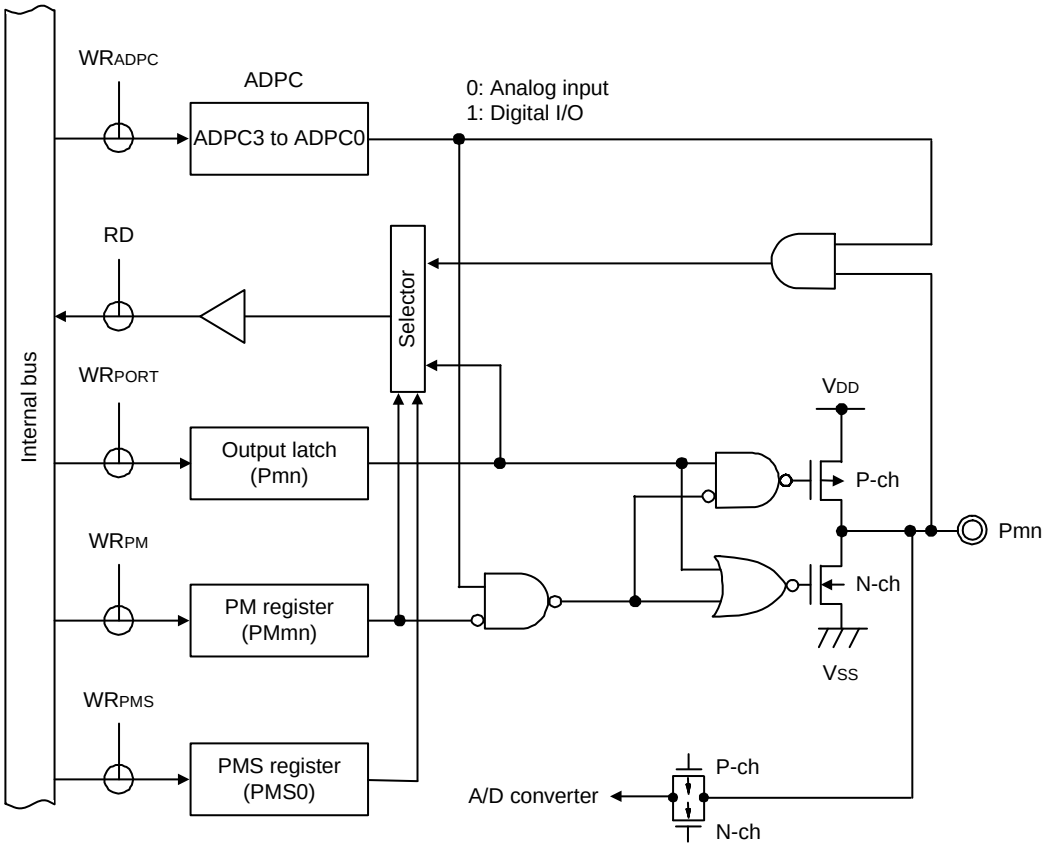
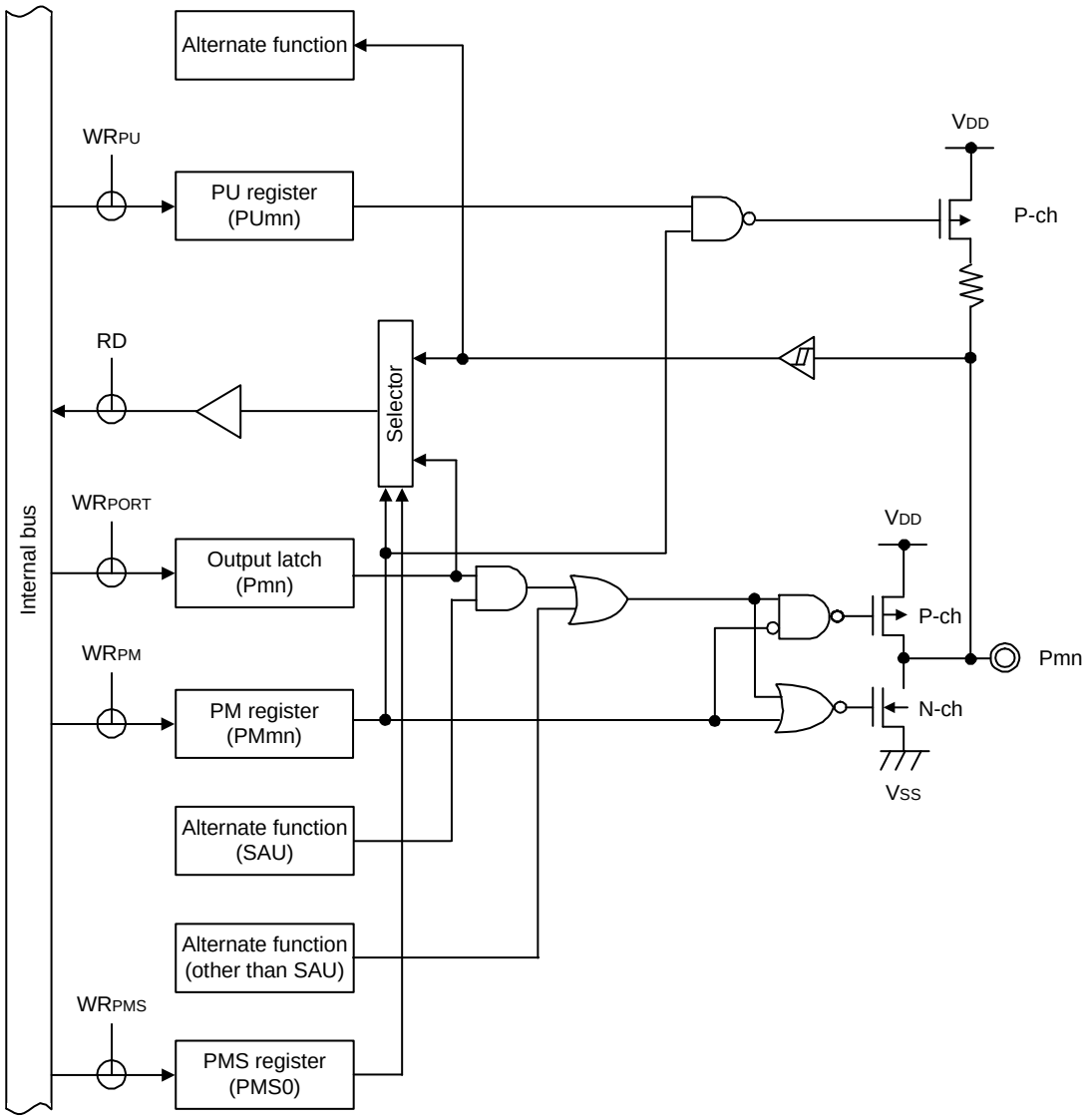


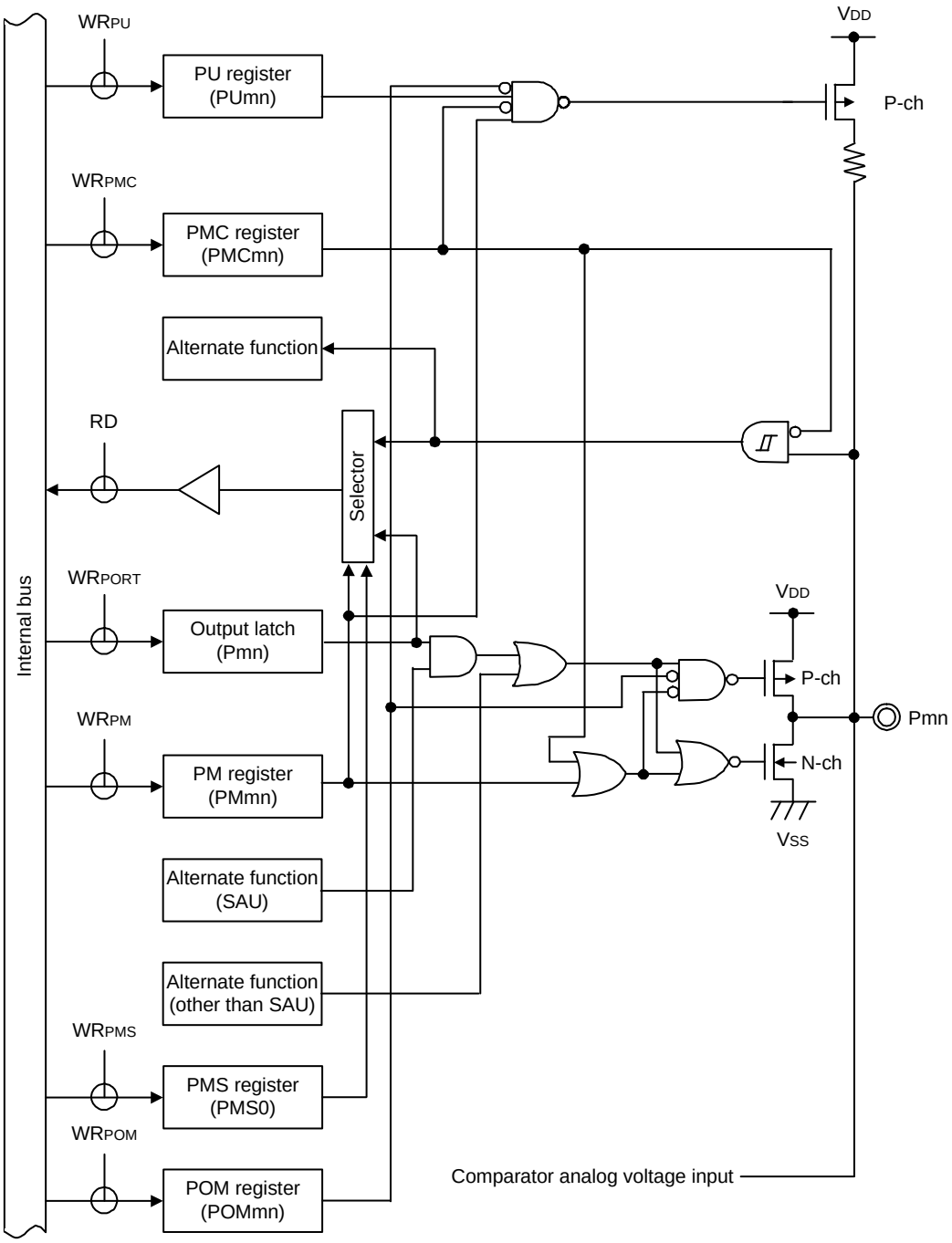
Figure 2 - 6 Pin Block Diagram of Pin Type 7-1-3



Remark 1. Refer to **2.1 Port Function** for alternate functions.

Remark 2. SAU: Serial array unit

Figure 2 - 7 Pin Block Diagram of Pin Type 7-3-4

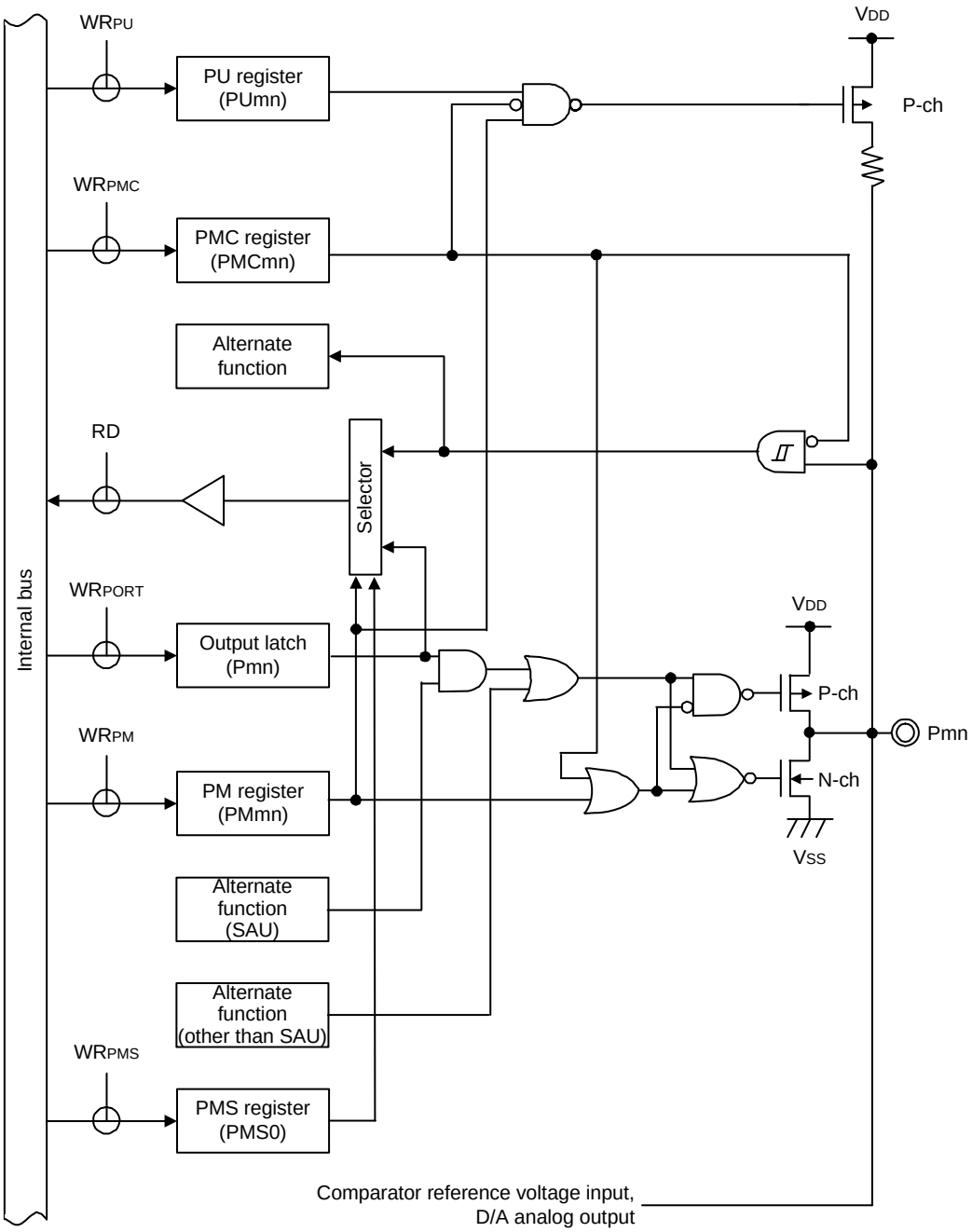


Caution A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).

Remark 1. Refer to **2.1 Port Function** for alternate functions.

Remark 2. SAU: Serial array unit

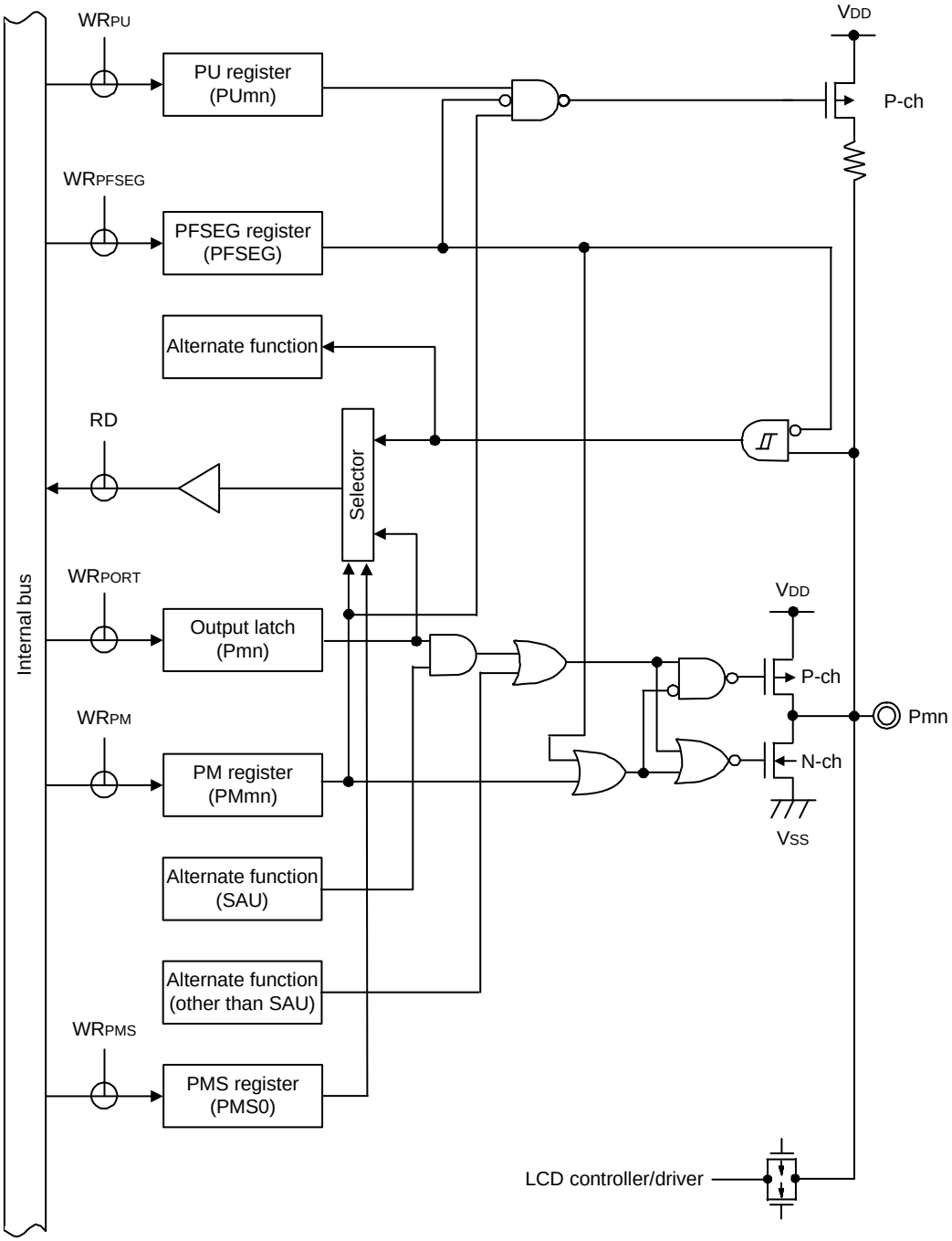
Figure 2 - 8 Pin Block Diagram of Pin Type 7-4-1



Remark 1. Refer to **2.1 Port Function** for alternate functions.

Remark 2. SAU: Serial array unit

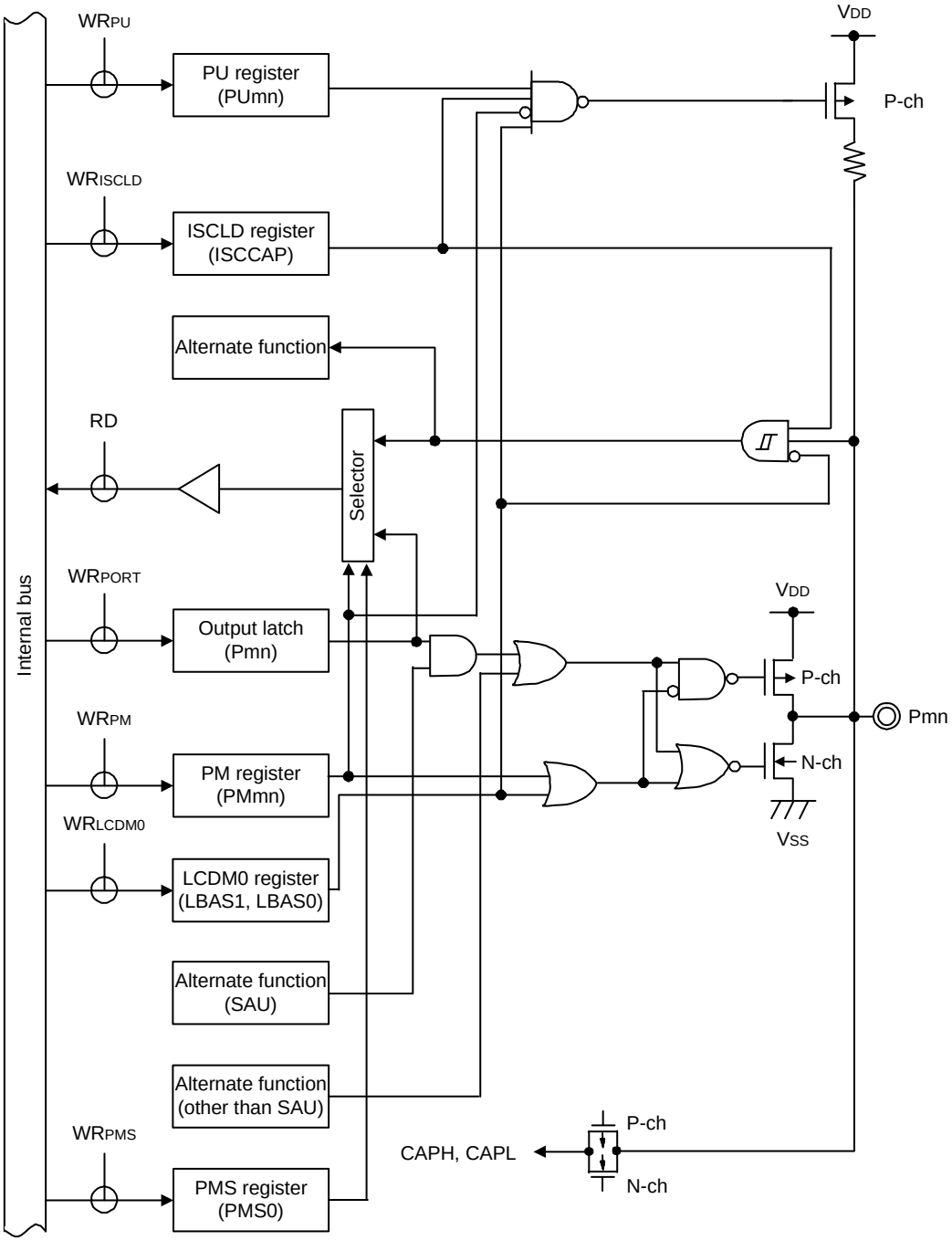
Figure 2 - 9 Pin Block Diagram of Pin Type 7-5-4



Remark 1. Refer to **2.1 Port Function** for alternate functions.

Remark 2. SAU: Serial array unit

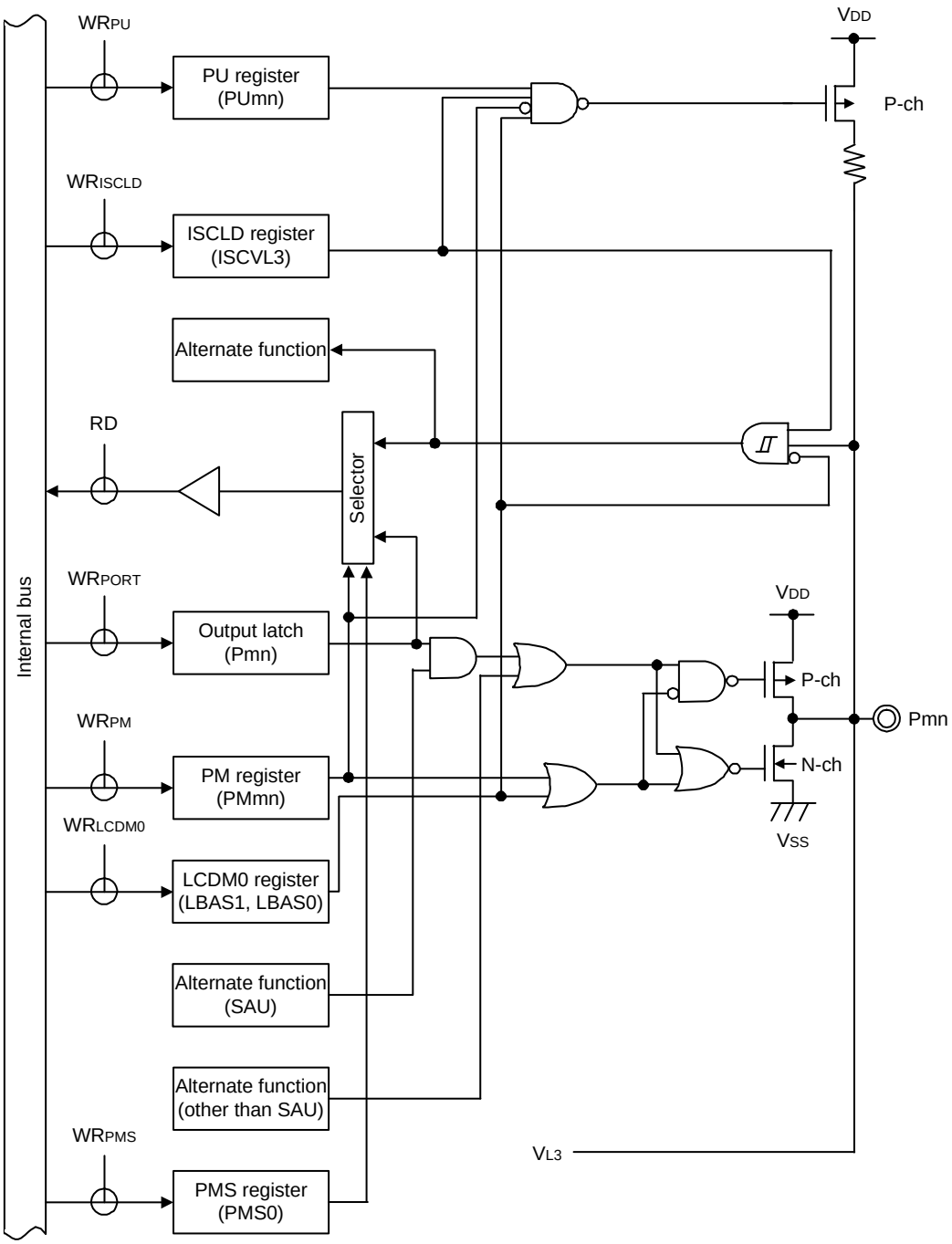
Figure 2 - 10 Pin Block Diagram of Pin Type 7-5-5



Remark 1. Refer to **2.1 Port Function** for alternate functions.

Remark 2. SAU: Serial array unit

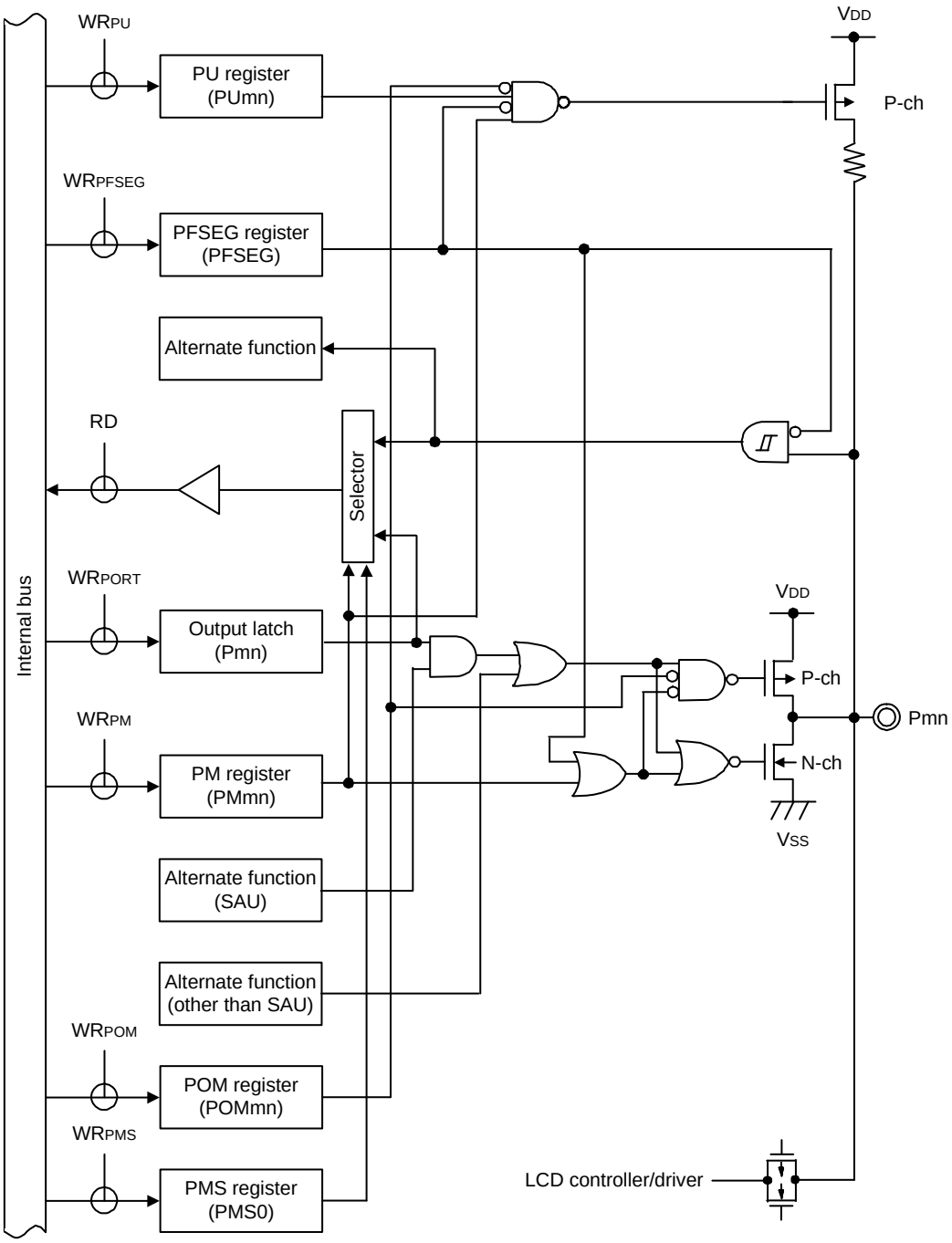
Figure 2 - 11 Pin Block Diagram of Pin Type 7-5-6



Remark 1. Refer to **2.1 Port Function** for alternate functions.

Remark 2. SAU: Serial array unit

Figure 2 - 12 Pin Block Diagram of Pin Type 7-5-10

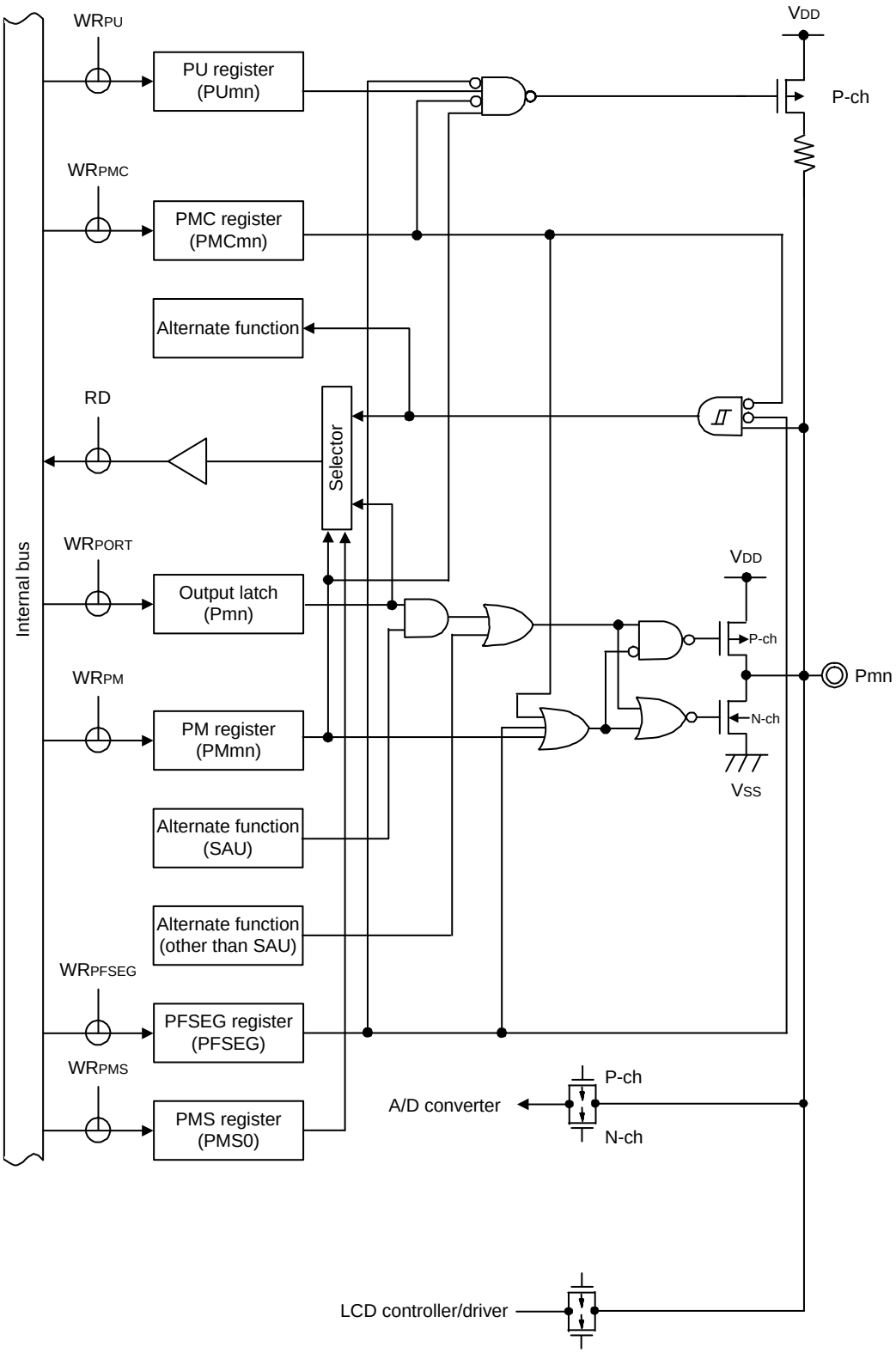


Caution A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).

Remark 1. Refer to **2.1 Port Function** for alternate functions.

Remark 2. SAU: Serial array unit

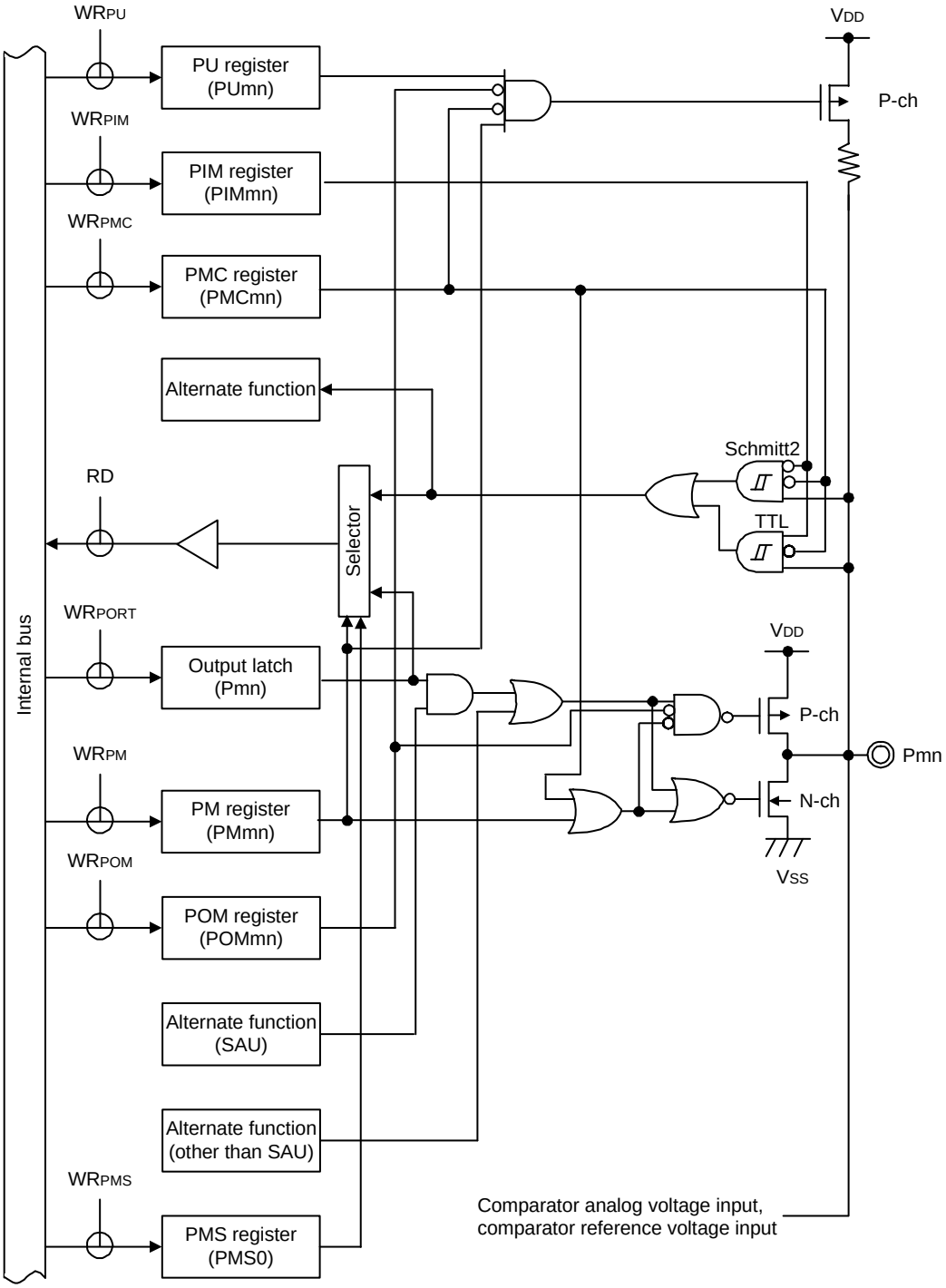
Figure 2 - 13 Pin Block Diagram of Pin Type 7-10-3



Remark 1. Refer to **2.1 Port Function** for alternate functions.

Remark 2. SAU: Serial array unit

Figure 2 - 14 Pin Block Diagram of Pin Type 8-3-4



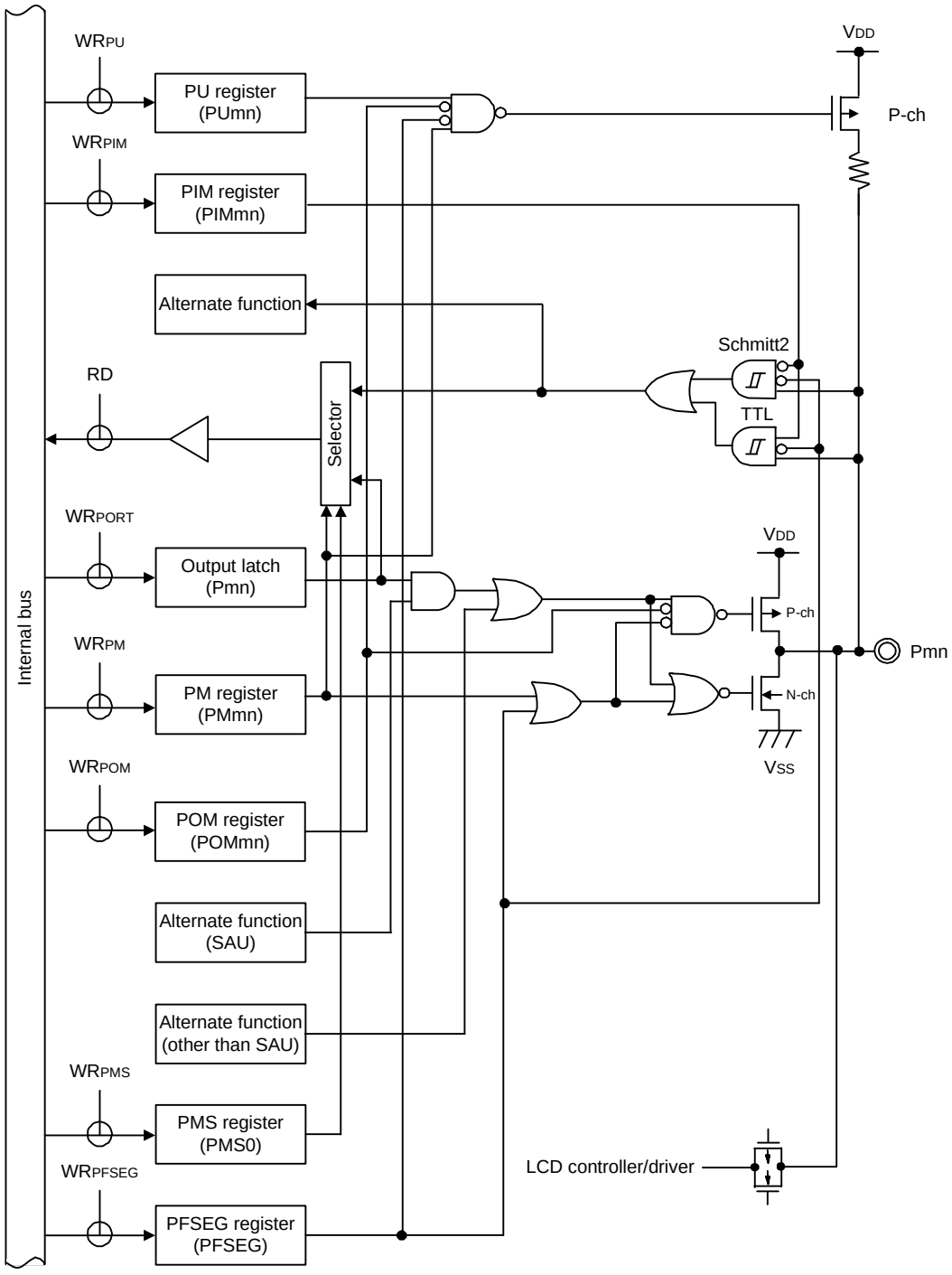
Caution 1. A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).

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

Remark 1. Refer to **2.1 Port Function** for alternate functions.

Remark 2. SAU: Serial array unit

Figure 2 - 15 Pin Block Diagram of Pin Type 8-5-10



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

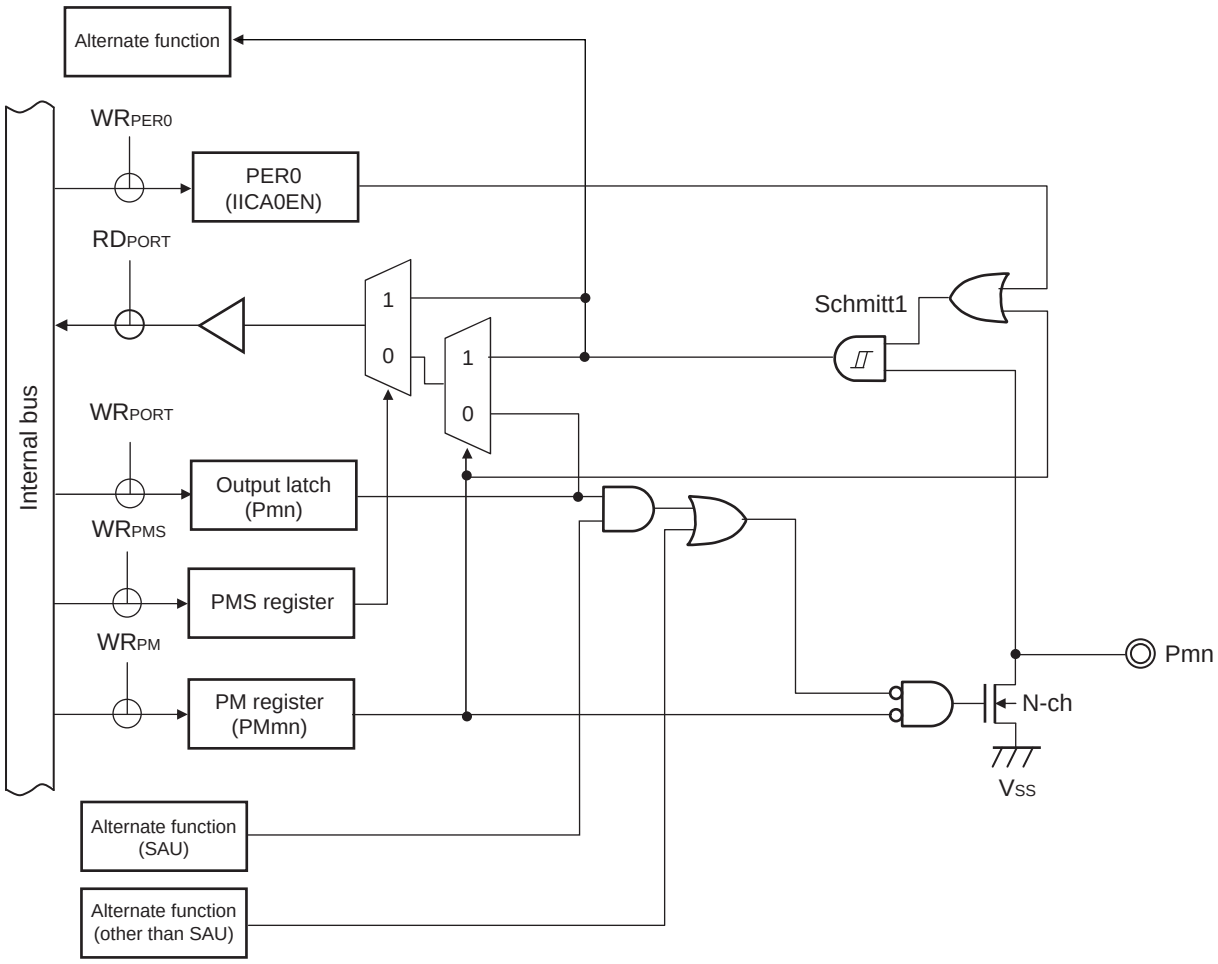
Caution 2. Because of TTL input buffer structure, if the port input mode register (PIMx) is set in TTL input buffer, a through current may flow through in the case of high level input. It is recommended to input a low level to prevent a through current.

Remark 1. Refer to **2.1 Port Function** for alternate functions.

Remark 2. SAU: Serial array unit

<R>

Figure 2 - 16 Pin Block Diagram of Pin Type 12-1-3



Remark 1. Refer to **2.1 Port Function** for alternate functions.

Remark 2. SAU: Serial array unit

Figure 2 - 17 Pin Block Diagram of Pin Type 17-11-1

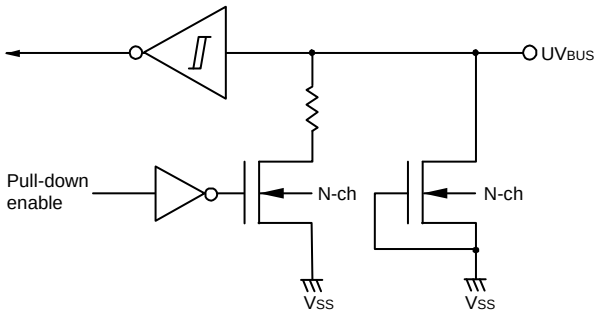


Figure 2 - 18 Pin Block Diagram of Pin Type 18-5-1

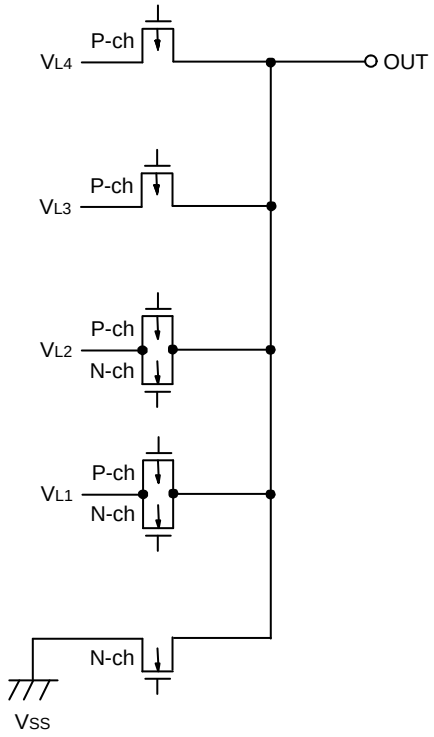
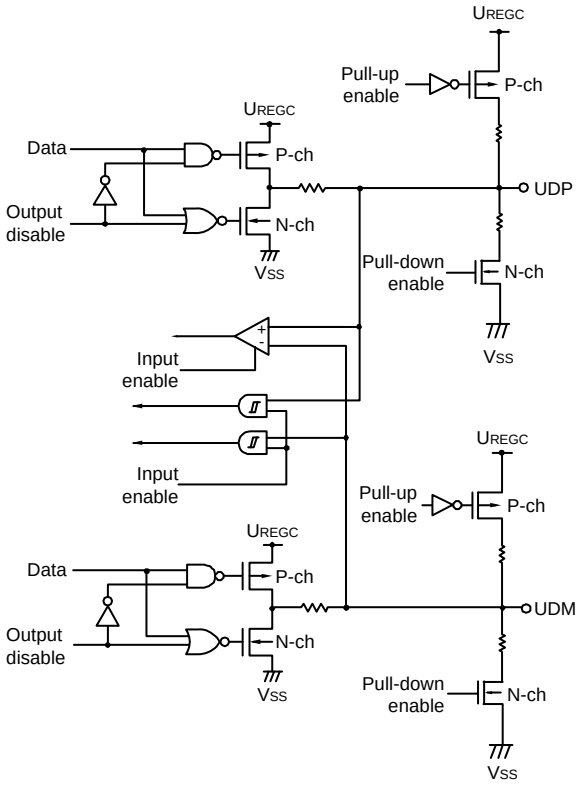


Figure 2 - 19 Pin Block Diagram of Pin Type 18-11-1

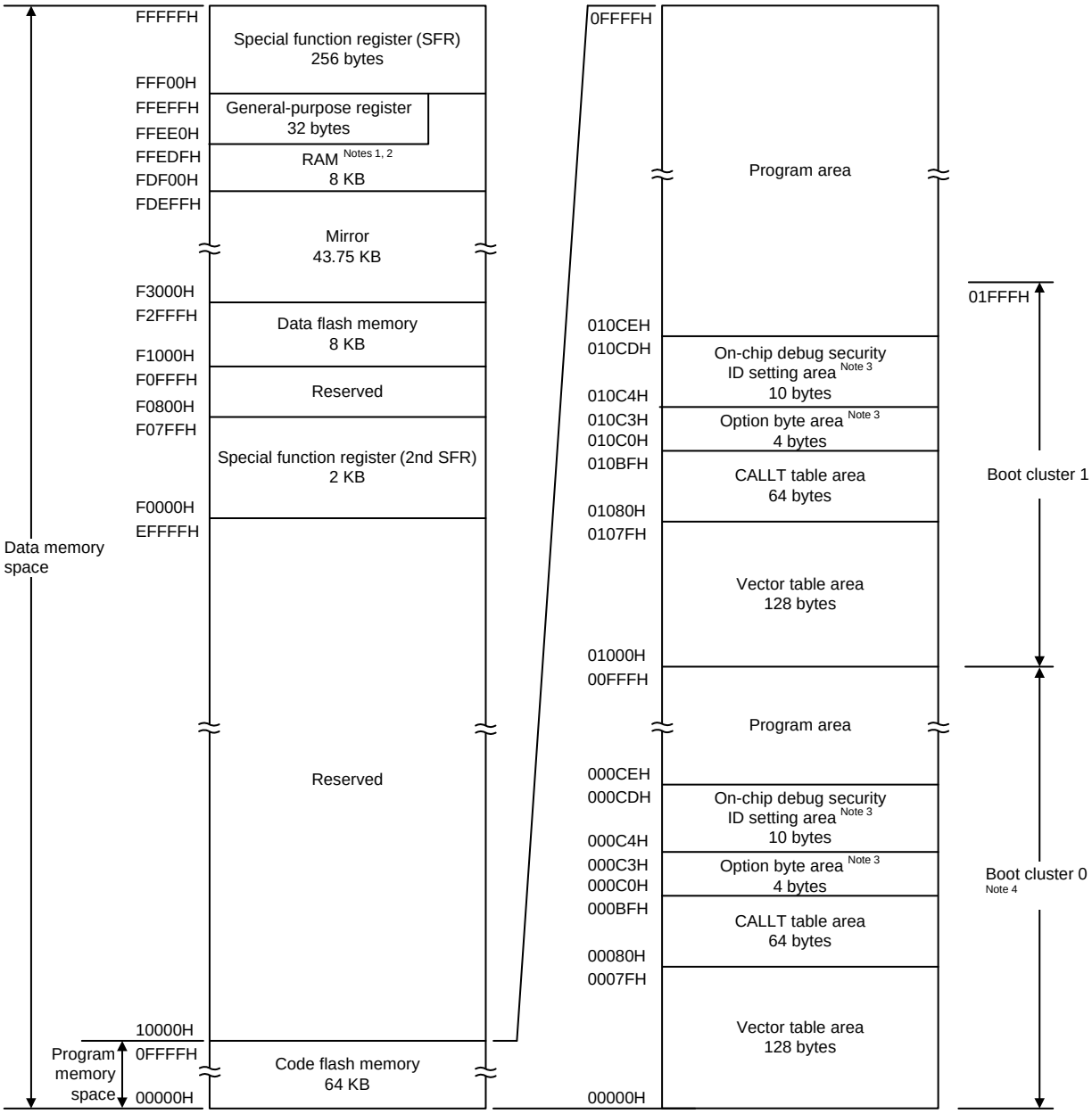


CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Products in the RL78/L1C can access a 1 MB memory space. Figures 3 - 1 to 3 - 5 show the memory maps.

Figure 3 - 1 Memory Map (R5F110xE, R5F111xE (x = M, N, P))



Note 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 DTC transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

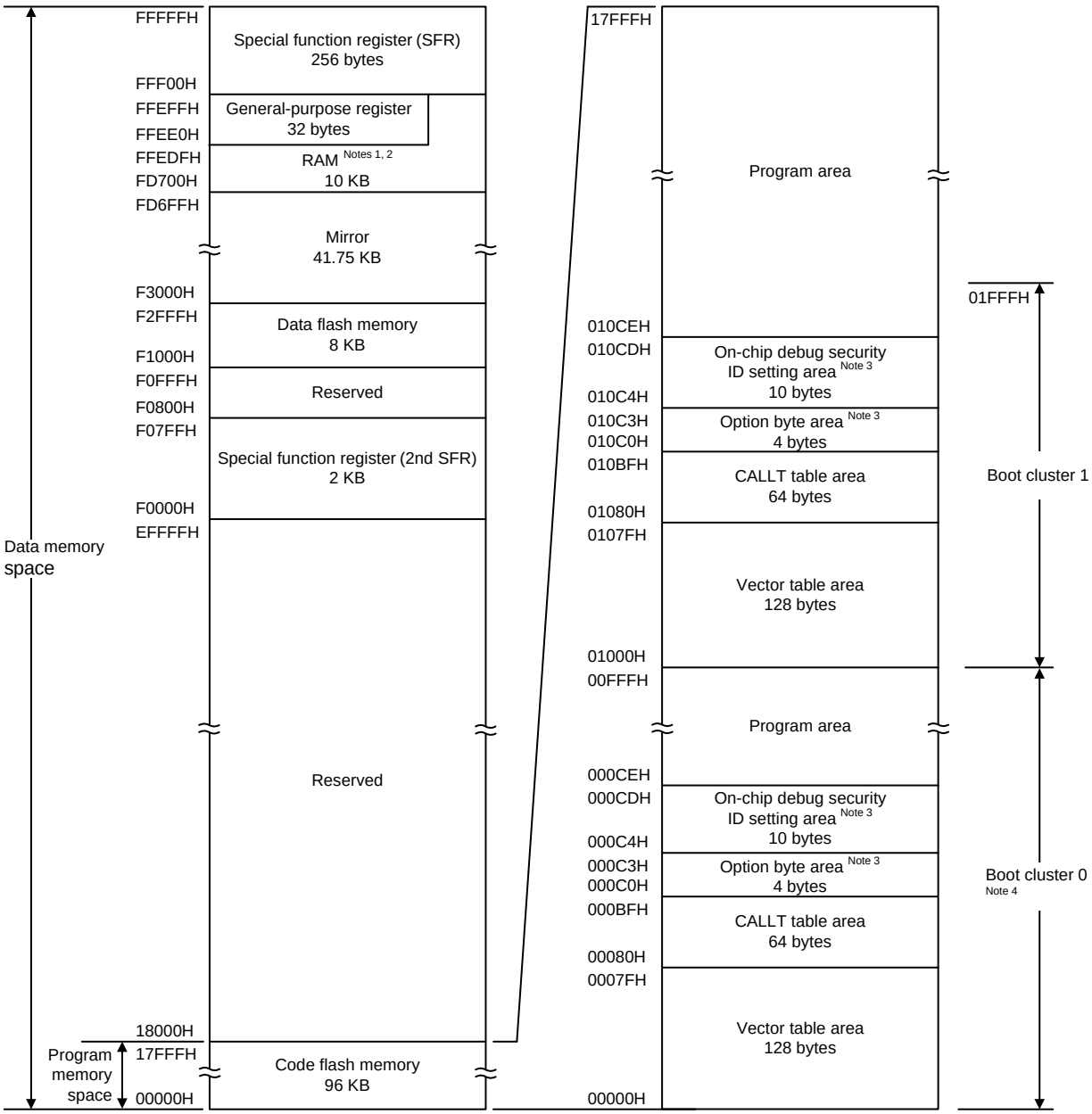
Note 2. Instructions can be executed from the RAM area excluding the general-purpose register area.

Note 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.

Note 4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.7 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 27.3.3 RAM parity error detection function.

Figure 3 - 2 Memory Map (R5F110xF, R5F111xF (x = M, N, P))



Note 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 DTC transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

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

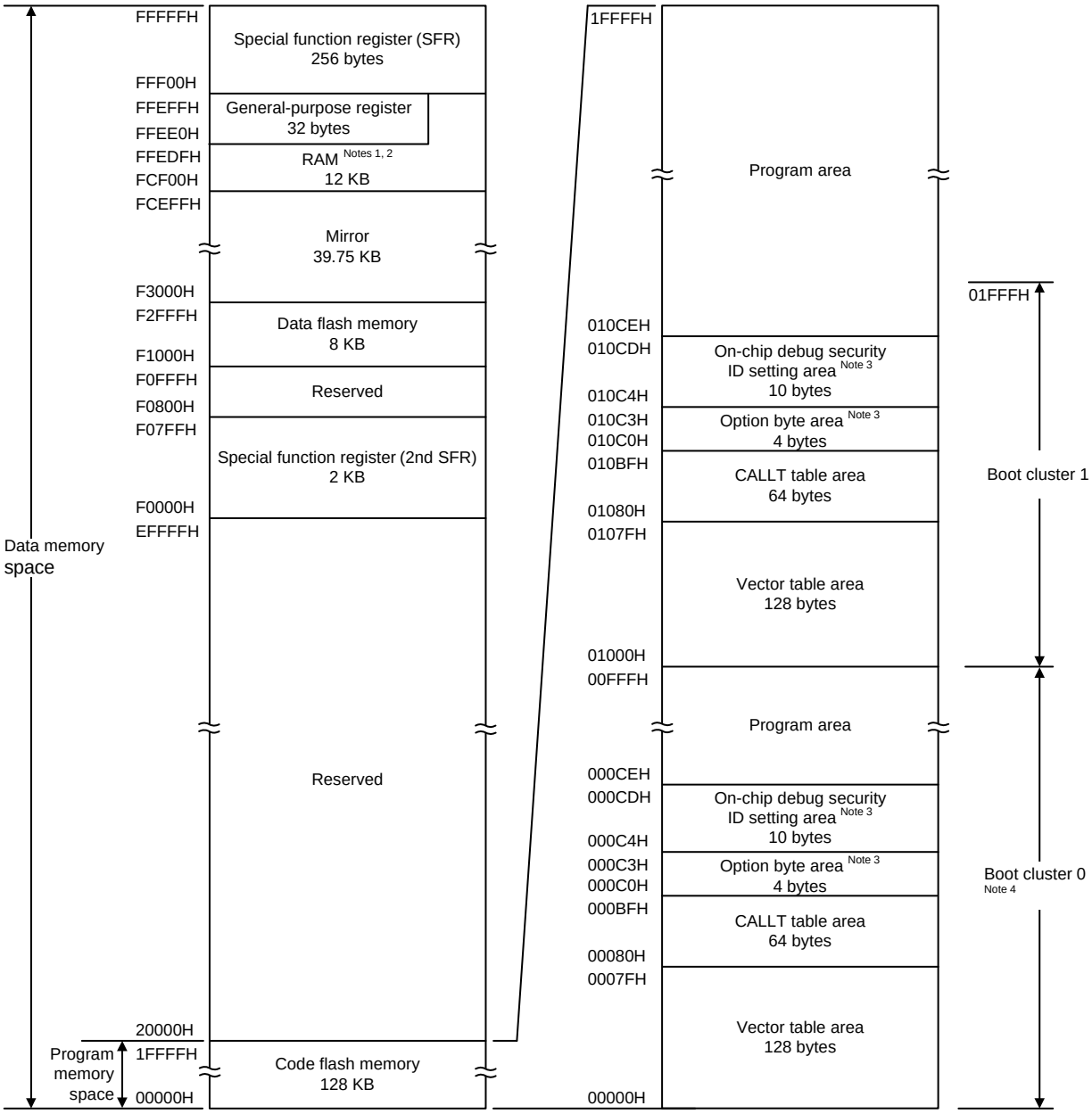
Note 3. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

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.

Note 4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.7 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 27.3.3 RAM parity error detection function.

Figure 3 - 3 Memory Map(R5F110xG, R5F111xG (x = M, N, P))



Note 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 DTC transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

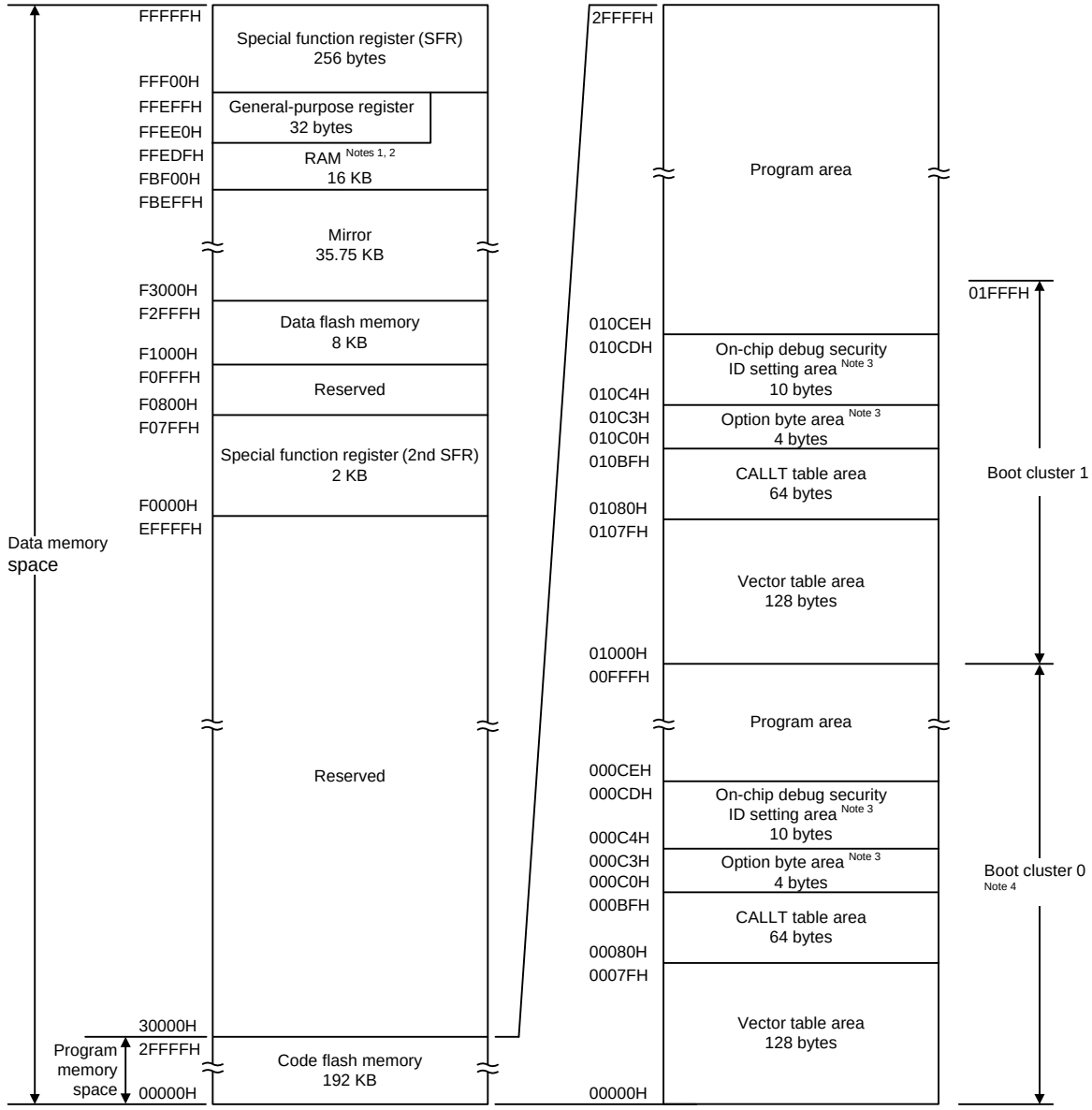
Note 2. Instructions can be executed from the RAM area excluding the general-purpose register area.

Note 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.

Note 4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.7 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 27.3.3 RAM parity error detection function.

Figure 3 - 4 Memory Map (R5F110xH, R5F111xH (x = M, N, P))



Note 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 DTC transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory. Since the area FBF00H to FC309H is used for each library, this area cannot be used.

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

Note 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.

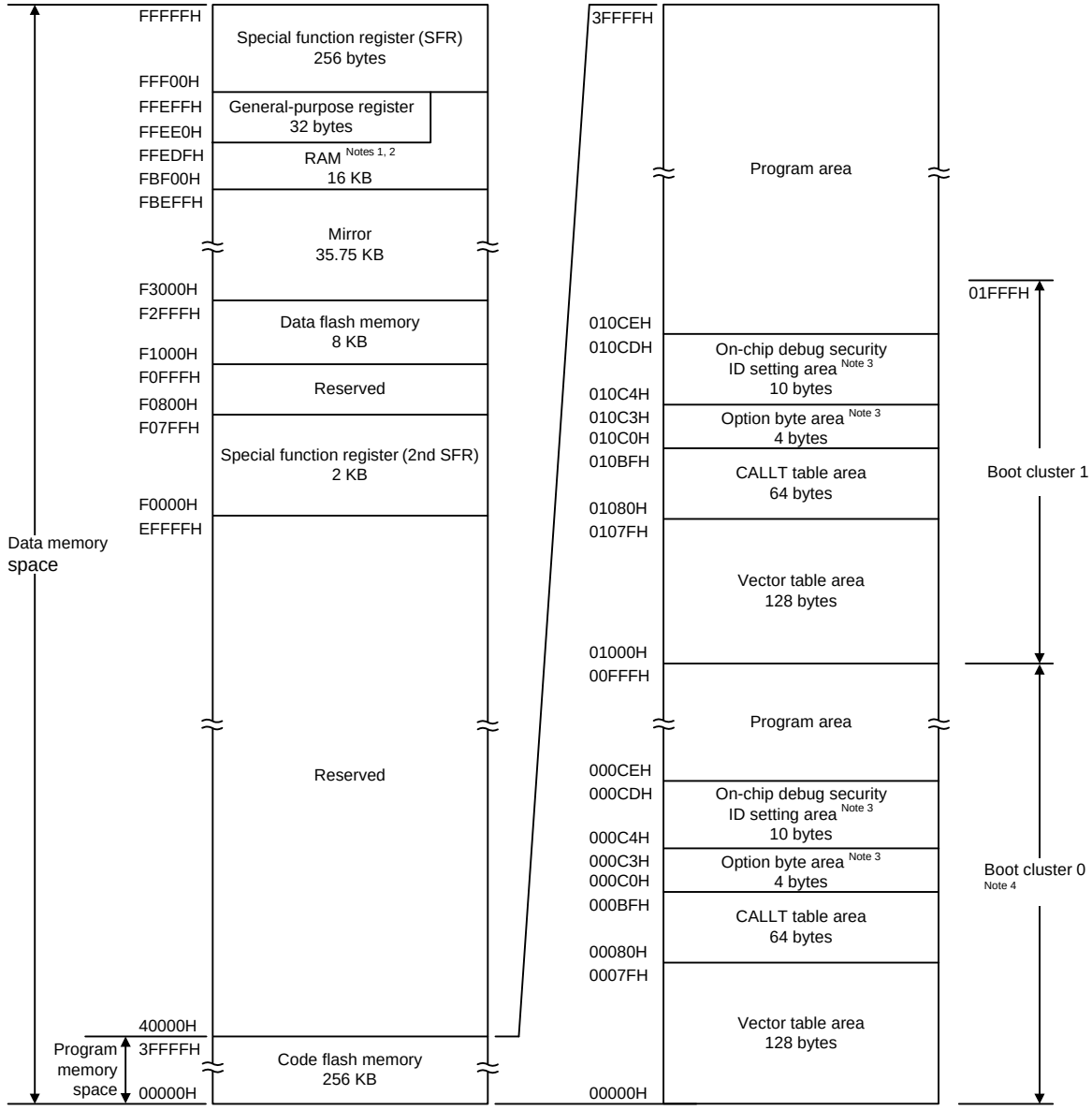
When boot swap is used: Set the option bytes to 000C0H to 000C3H and 010C0H to 010C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 010C4H to 010CDH.

Note 4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.7 Security Settings**).

Caution 1. 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 27.3.3 RAM parity error detection function.

Caution 2. The internal RAM area in the following products cannot be used as the stack memory when using the on-chip debugging trace function.
R5F110xH, R5F111xH (x = M, N, P): FC300H to FC6FFH

Figure 3 - 5 Memory Map (R5F110xJ, R5F111xJ (x = M, N, P))



Note 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 DTC transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory. Since the area FBF00H to FC309H is used for each library, this area cannot be used.

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

Note 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.

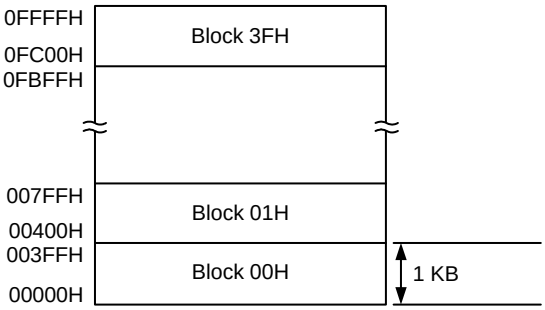
When boot swap is used: Set the option bytes to 000C0H to 000C3H and 010C0H to 010C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 010C4H to 010CDH.

Note 4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.7 Security Settings**).

Caution 1. 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 27.3.3 RAM parity error detection function.

Caution 2. The internal RAM area in the following products cannot be used as the stack memory when using the on-chip debugging trace function.
R5F110xJ, R5F111xJ (x = M, N, P): FC300H to FC6FFH

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Tables 3 - 1** and **3 - 2 Correspondence Between Address Values and Block Numbers in Flash Memory**.



(R5F110xE, R5F111xE (x = M, N, P))

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 (1/2)

Address Value	Block Number	Address Value	Block Number	Address Value	Block Number	Address Value	Block Number
00000H to 003FFH	00H	08000H to 083FFH	20H	10000H to 103FFH	40H	18000H to 183FFH	60H
00400H to 007FFH	01H	08400H to 087FFH	21H	10400H to 107FFH	41H	18400H to 187FFH	61H
00800H to 00BFFH	02H	08800H to 08BFFH	22H	10800H to 10BFFH	42H	18800H to 18BFFH	62H
00C00H to 00FFFH	03H	08C00H to 08FFFH	23H	10C00H to 10FFFH	43H	18C00H to 18FFFH	63H
01000H to 013FFH	04H	09000H to 093FFH	24H	11000H to 113FFH	44H	19000H to 193FFH	64H
01400H to 017FFH	05H	09400H to 097FFH	25H	11400H to 117FFH	45H	19400H to 197FFH	65H
01800H to 01BFFH	06H	09800H to 09BFFH	26H	11800H to 11BFFH	46H	19800H to 19BFFH	66H
01C00H to 01FFFH	07H	09C00H to 09FFFH	27H	11C00H to 11FFFH	47H	19C00H to 19FFFH	67H
02000H to 023FFH	08H	0A000H to 0A3FFH	28H	12000H to 123FFH	48H	1A000H to 1A3FFH	68H
02400H to 027FFH	09H	0A400H to 0A7FFH	29H	12400H to 127FFH	49H	1A400H to 1A7FFH	69H
02800H to 02BFFH	0AH	0A800H to 0ABFFH	2AH	12800H to 12BFFH	4AH	1A800H to 1ABFFH	6AH
02C00H to 02FFFH	0BH	0AC00H to 0AFFFH	2BH	12C00H to 12FFFH	4BH	1AC00H to 1AFFFH	6BH
03000H to 033FFH	0CH	0B000H to 0B3FFH	2CH	13000H to 133FFH	4CH	1B000H to 1B3FFH	6CH
03400H to 037FFH	0DH	0B400H to 0B7FFH	2DH	13400H to 137FFH	4DH	1B400H to 1B7FFH	6DH
03800H to 03BFFH	0EH	0B800H to 0BBFFH	2EH	13800H to 13BFFH	4EH	1B800H to 1BBFFH	6EH
03C00H to 03FFFH	0FH	0BC00H to 0BFFFH	2FH	13C00H to 13FFFH	4FH	1BC00H to 1BFFFH	6FH
04000H to 043FFH	10H	0C000H to 0C3FFH	30H	14000H to 143FFH	50H	1C000H to 1C3FFH	70H
04400H to 047FFH	11H	0C400H to 0C7FFH	31H	14400H to 147FFH	51H	1C400H to 1C7FFH	71H
04800H to 04BFFH	12H	0C800H to 0CBFFH	32H	14800H to 14BFFH	52H	1C800H to 1CBFFH	72H
04C00H to 04FFFH	13H	0CC00H to 0CFFFH	33H	14C00H to 14FFFH	53H	1CC00H to 1CFFFH	73H
05000H to 053FFH	14H	0D000H to 0D3FFH	34H	15000H to 153FFH	54H	1D000H to 1D3FFH	74H
05400H to 057FFH	15H	0D400H to 0D7FFH	35H	15400H to 157FFH	55H	1D400H to 1D7FFH	75H
05800H to 05BFFH	16H	0D800H to 0DBFFH	36H	15800H to 15BFFH	56H	1D800H to 1DBFFH	76H
05C00H to 05FFFH	17H	0DC00H to 0DFFFH	37H	15C00H to 15FFFH	57H	1DC00H to 1DFFFH	77H
06000H to 063FFH	18H	0E000H to 0E3FFH	38H	16000H to 163FFH	58H	1E000H to 1E3FFH	78H
06400H to 067FFH	19H	0E400H to 0E7FFH	39H	16400H to 167FFH	59H	1E400H to 1E7FFH	79H
06800H to 06BFFH	1AH	0E800H to 0EBFFH	3AH	16800H to 16BFFH	5AH	1E800H to 1EBFFH	7AH
06C00H to 06FFFH	1BH	0EC00H to 0EFFFH	3BH	16C00H to 16FFFH	5BH	1EC00H to 1EFFFH	7BH
07000H to 073FFH	1CH	0F000H to 0F3FFH	3CH	17000H to 173FFH	5CH	1F000H to 1F3FFH	7CH
07400H to 077FFH	1DH	0F400H to 0F7FFH	3DH	17400H to 177FFH	5DH	1F400H to 1F7FFH	7DH
07800H to 07BFFH	1EH	0F800H to 0FBFFH	3EH	17800H to 17BFFH	5EH	1F800H to 1FBFFH	7EH
07C00H to 07FFFH	1FH	0FC00H to 0FFFFH	3FH	17C00H to 17FFFH	5FH	1FC00H to 1FFFFH	7FH

Remark R5F110xE, R5F111xE (x = M, N, P): Block numbers 00H to 3FH
R5F110xF, R5F111xF (x = M, N, P): Block numbers 00H to 5FH
R5F110xG, R5F111xG (x = M, N, P): Block numbers 00H to 7FH

Table 3 - 2 Correspondence Between Address Values and Block Numbers in Flash Memory (2/2)

Address Value	Block Number	Address Value	Block Number	Address Value	Block Number	Address Value	Block Number
20000H to 203FFH	80H	28000H to 283FFH	A0H	30000H to 303FFH	C0H	38000H to 383FFH	E0H
20400H to 207FFH	81H	28400H to 287FFH	A1H	30400H to 307FFH	C1H	38400H to 387FFH	E1H
20800H to 20BFFH	82H	28800H to 28BFFH	A2H	30800H to 30BFFH	C2H	38800H to 38BFFH	E2H
20C00H to 20FFFH	83H	28C00H to 28FFFH	A3H	30C00H to 30FFFH	C3H	38C00H to 38FFFH	E3H
21000H to 213FFH	84H	29000H to 293FFH	A4H	31000H to 313FFH	C4H	39000H to 393FFH	E4H
21400H to 217FFH	85H	29400H to 297FFH	A5H	31400H to 317FFH	C5H	39400H to 397FFH	E5H
21800H to 21BFFH	86H	29800H to 29BFFH	A6H	31800H to 31BFFH	C6H	39800H to 39BFFH	E6H
21C00H to 21FFFH	87H	29C00H to 29FFFH	A7H	31C00H to 31FFFH	C7H	39C00H to 39FFFH	E7H
22000H to 223FFH	88H	2A000H to 2A3FFH	A8H	32000H to 323FFH	C8H	3A000H to 3A3FFH	E8H
22400H to 227FFH	89H	2A400H to 2A7FFH	A9H	32400H to 327FFH	C9H	3A400H to 3A7FFH	E9H
22800H to 22BFFH	8AH	2A800H to 2ABFFH	AAH	32800H to 32BFFH	CAH	3A800H to 3ABFFH	EAH
22C00H to 22FFFH	8BH	2AC00H to 2AFFFH	ABH	32C00H to 32FFFH	CBH	3AC00H to 3AFFFH	EBH
23000H to 233FFH	8CH	2B000H to 2B3FFH	ACH	33000H to 333FFH	CCH	3B000H to 3B3FFH	ECH
23400H to 237FFH	8DH	2B400H to 2B7FFH	ADH	33400H to 337FFH	CDH	3B400H to 3B7FFH	EDH
23800H to 23BFFH	8EH	2B800H to 2BBFFH	AEH	33800H to 33BFFH	CEH	3B800H to 3BBFFH	EEH
23C00H to 23FFFH	8FH	2BC00H to 2BFFFH	AFH	33C00H to 33FFFH	CFH	3BC00H to 3BFFFH	EFH
24000H to 243FFH	90H	2C000H to 2C3FFH	B0H	34000H to 343FFH	D0H	3C000H to 3C3FFH	F0H
24400H to 247FFH	91H	2C400H to 2C7FFH	B1H	34400H to 347FFH	D1H	3C400H to 3C7FFH	F1H
24800H to 24BFFH	92H	2C800H to 2CBFFH	B2H	34800H to 34BFFH	D2H	3C800H to 3CBFFH	F2H
24C00H to 24FFFH	93H	2CC00H to 2CFFFH	B3H	34C00H to 34FFFH	D3H	3CC00H to 3CFFFH	F3H
25000H to 253FFH	94H	2D000H to 2D3FFH	B4H	35000H to 353FFH	D4H	3D000H to 3D3FFH	F4H
25400H to 257FFH	95H	2D400H to 2D7FFH	B5H	35400H to 357FFH	D5H	3D400H to 3D7FFH	F5H
25800H to 25BFFH	96H	2D800H to 2DBFFH	B6H	35800H to 35BFFH	D6H	3D800H to 3DBFFH	F6H
25C00H to 25FFFH	97H	2DC00H to 2DFFFH	B7H	35C00H to 35FFFH	D7H	3DC00H to 3DFFFH	F7H
26000H to 263FFH	98H	2E000H to 2E3FFH	B8H	36000H to 363FFH	D8H	3E000H to 3E3FFH	F8H
26400H to 267FFH	99H	2E400H to 2E7FFH	B9H	36400H to 367FFH	D9H	3E400H to 3E7FFH	F9H
26800H to 26BFFH	9AH	2E800H to 2EBFFH	BAH	36800H to 36BFFH	DAH	3E800H to 3EBFFH	FAH
26C00H to 26FFFH	9BH	2EC00H to 2EFFFH	BBH	36C00H to 36FFFH	DBH	3EC00H to 3EFFFH	FBH
27000H to 273FFH	9CH	2F000H to 2F3FFH	BCH	37000H to 373FFH	DCH	3F000H to 3F3FFH	FCH
27400H to 277FFH	9DH	2F400H to 2F7FFH	BDH	37400H to 377FFH	DDH	3F400H to 3F7FFH	FDH
27800H to 27BFFH	9EH	2F800H to 2FBFFH	BEH	37800H to 37BFFH	DEH	3F800H to 3FBFFH	FEH
27C00H to 27FFFH	9FH	2FC00H to 2FFFFH	BFH	37C00H to 37FFFH	DFH	3FC00H to 3FFFFH	FFH

Remark R5F110xH, R5F111xH (x = M, N, P): Block numbers 00H to BFH
R5F110xJ, R5F111xJ (x = M, N, P): Block numbers 00H to FFH

3.1.1 Internal program memory space

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

The RL78/L1C products incorporate internal ROM (flash memory), as shown below.

Table 3 - 3 Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
R5F110xE, R5F111xE (x = M, N, P)	Flash memory	65536 × 8 bits (00000H to 0FFFFH)
R5F110xF, R5F111xF (x = M, N, P)		98304 × 8 bits (00000H to 17FFFH)
R5F110xG, R5F111xG (x = M, N, P)		131072 × 8 bits (00000H to 1FFFFH)
R5F110xH, R5F111xH (x = M, N, P)		196608 × 8 bits (00000H to 2FFFFH)
R5F110xJ, R5F111xJ (x = M, N, P)		262144 × 8 bits (00000H to 3FFFFH)

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 - 4 Vector Table (1/2)

Vector Table Address	Interrupt Source	With USB		Without USB	
		100-pin	80/85-pin	100-pin	80/85-pin
00000H	RESET, POR, LVD, WDT, TRAP, IAW, RPE	√	√	√	√
00004H	INTWDTI	√	√	√	√
00006H	INTLVI	√	√	√	√
00008H	INTP0	√	√	√	√
0000AH	INTP1	√	√	√	√
0000CH	INTP2	√	√	√	√
0000EH	INTP3	√	√	√	√
00010H	INTP4	√	√	√	√
00012H	INTP5	√	√	√	√
00014H	INTST2/INTCSI20/INTIIC20	√	√	√	√
00016H	INTSR2	√	√	√	√
00018H	INTSRE2	√	√	√	√
0001EH	INTST0/INTCSI00/INTIIC00	√	√	√	√
00020H	INTTM00	√	√	√	√
00022H	INTSR0	√	√	√	√
00024H	INTSRE0	√	√	√	√
	INTTM01H	√	√	√	√
00026H	INTST1/INTCSI10/INTIIC10	√	√	√	√
00028H	INTSR1	√	√	√	√
0002AH	INTSRE1	√	√	√	√
	INTTM03H	√	√	√	√
0002CH	INTIICA0	√	√	√	√
0002EH	INTRTIT	√	√	√	√
00032H	INTTM01	√	√	√	√
00034H	INTTM02	√	√	√	√
00036H	INTTM03	√	√	√	√
00038H	INTAD	√	√	√	√
0003AH	INTRTC	√	√	√	√
0003CH	INTIT	√	√	√	√
0003EH	INTKR	√	√	√	√
00040H	INTST3/INTCSI30/INTIIC30	√	√	√	√
00042H	INTSR3	√	√	√	√
00046H	INTTM04	√	√	√	√
00048H	INTTM05	√	√	√	√
0004AH	INTP6	√	√	√	√
0004CH	INTP7	√	√	√	√
00050H	INTCMP0	√	√	√	√
00052H	INTCMP1	√	—	√	—
00054H	INTTM06	√	√	√	√
00056H	INTTM07	√	√	√	√
00058H	INTUSB	√	√	—	—

Table 3 - 5 Vector Table (2/2)

Vector Table Address	Interrupt Source	With USB		Without USB	
		100-pin	80/85-pin	100-pin	80/85-pin
0005AH	INTRSUM	√	√	—	—
0005CH	INTSRE3	√	√	√	√
0005EH	INTTKB2_0	√	√	√	√
00060H	INTTKB2_1	√	√	√	√
00062H	INTFL	√	√	√	√
00064H	INTTKB2_2	√	√	√	√
00066H	DTC0FIFO	√	√	—	—
00068H	DTC1FIFO	√	√	—	—
0007EH	BRK	√	√	√	√

(2) CALLT instruction table area

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

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 29 OPTION BYTE**.

(4) On-chip debug security ID setting area

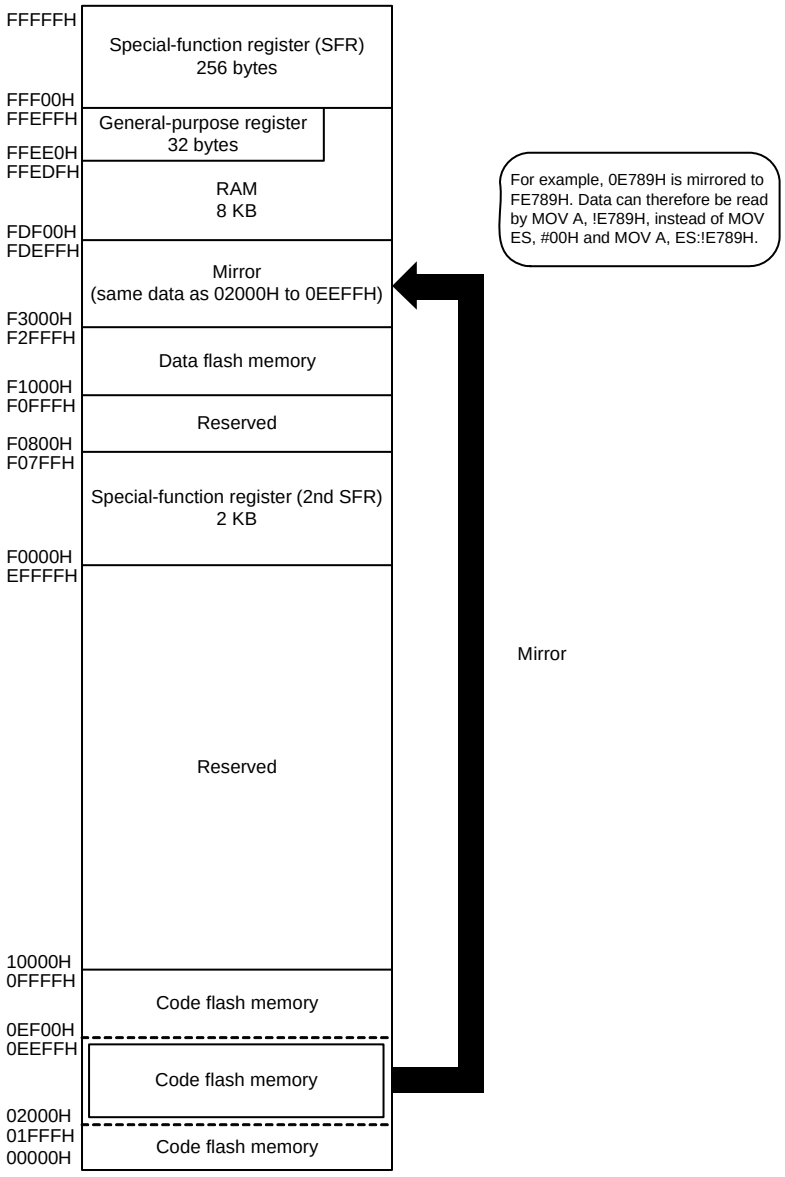
A 10-byte area of 000C4H to 000CDH and 010C4H to 010CDH can be used as an on-chip debug security ID setting area. Set the on-chip debug security ID of 10 bytes at 000C4H to 000CDH when the boot swap is not used and at 000C4H to 000CDH and at 010C4H to 010CDH when the boot swap is used. For details, see **CHAPTER 31 ON-CHIP DEBUG FUNCTION**.

3.1.2 Mirror area

The RL78/L1C mirrors the code flash area of 00000H to 0FFFFH, to F0000H to FFFFFH. The products with 96 KB or more flash memory mirror the code flash area of 00000H to 0FFFFH or 10000H to 1FFFFH, 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 R5F110xE, R5F111xE (x = M, N, P) (Flash memory: 64 KB, RAM: 8 KB)



The PMC register is described below.

• Processor mode control register (PMC)

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

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

Reset signal generation sets this register to 00H.

Figure 3 - 6 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	10000H to 1FFFFH is mirrored to F0000H to FFFFFH

Caution 1. In products with 64-KB flash memory, be sure to set bit 0 (MAA) to 0 (initial value).

Caution 2. Set the PMC register only once during the initial settings prior to operating the DTC (data transfer controller).

Rewriting the PMC register other than during the initial settings is prohibited.

Caution 3. 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/L1C products incorporate the following RAMs.

Table 3 - 6 Internal RAM Capacity

Part Number	Internal RAM
R5F110xE, R5F111xE (x = M, N, P)	8192 × 8 bits (FDF00H to FFEFFH)
R5F110xF, R5F111xF (x = M, N, P)	10240 × 8 bits (FD700H to FFEFFH)
R5F110xG, R5F111xG (x = M, N, P)	12288 × 8 bits (FCF00H to FFEFFH)
R5F110xH, R5F111xH (x = M, N, P)	16384 × 8 bits (FBF00H to FFEFFH)
R5F110xJ, R5F111xJ (x = M, N, P)	16384 × 8 bits (FBF00H to FFEFFH)

The internal RAM can be used as a data area and a program area where instructions are written and executed (it is prohibited to use the general-purpose register area for executing 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.

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

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

Caution 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.
R5F110xH, R5F111xH (x = M, N, P): FBF00H to FC309H
R5F110xJ, R5F111xJ (x = M, N, P): FBF00H to FC309H

Caution 4. The internal RAM area in the following products cannot be used as the stack memory when using the on-chip debugging trace function.
R5F110xH, R5F111xH (x = M, N, P): FC300H to FC6FFH
R5F110xJ, R5F111xJ (x = M, N, P): FC300H to FC6FFH

3.1.4 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FFF00H to FFFFFH (see **Tables 3 - 7** to **3 - 10** in **3.2.4 Special function registers (SFRs)**).

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

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

On-chip peripheral hardware special function registers (2nd SFRs) are allocated in the area F0000H to F07FFH (see **Tables 3 - 11** to **3 - 25** in **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**).

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

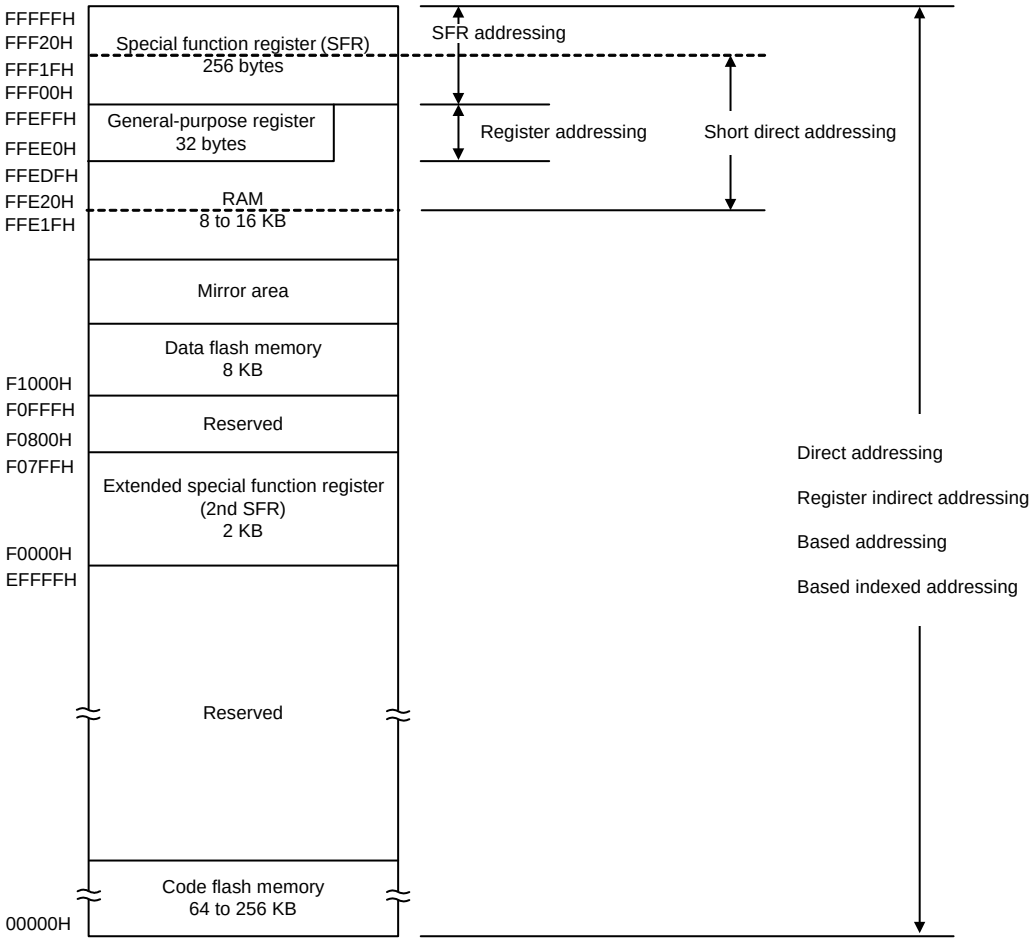
Caution Do not access addresses to which extended SFRs are not assigned.

3.1.6 Data memory addressing

Addressing refers to the method of specifying the address of the instruction to be executed next or the address of the register or memory relevant to the execution of instructions.
Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the RL78/L1C, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of the special function registers (SFR) and general-purpose registers are available for use. Figure 3 - 7 shows the correspondence between data memory and addressing.

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

Figure 3 - 7 Correspondence Between Data Memory and Addressing



3.2 Processor Registers

The RL78/L1C products incorporate the following processor registers.

3.2.1 Control registers

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

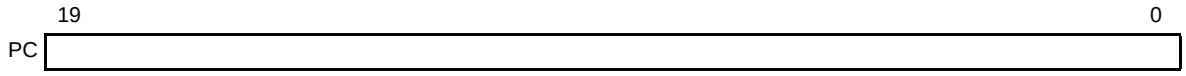
(1) Program counter (PC)

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

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

Reset signal generation sets the reset vector table values at addresses 00000H and 00001H to the 16 lower-order bits of the program counter. The four higher-order bits of the program counter are cleared to 0000.

Figure 3 - 8 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 - 9 Format of Program Status Word

	7							0
PSW	IE	Z	RBS1	AC	RBS0	ISP1	ISP0	CY

(a) Interrupt enable flag (IE)

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

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

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

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

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

(c) Register bank select flags (RBS0, RBS1)
These are 2-bit flags to select one of the four register banks.
In these flags, the 2-bit information that indicates the register bank selected by SEL RBn instruction execution is stored.

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

(e) In-service priority flags (ISP1, ISP0)
This flag manages the priority of acknowledgeable maskable vectored interrupts. Vectored interrupt requests specified lower than the value of ISP0 and ISP1 flags by the priority specification flag registers (PRn0L, PRn0H, PRn1L, PRn1H, PRn2L, PRn2H, PRn3L) (see **21.3.3**) cannot be acknowledged. Actual request acknowledgment is controlled by the interrupt enable flag (IE).

Remark n = 0, 1

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

(3) Stack pointer (SP)

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

Figure 3 - 10 Format of Stack Pointer

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

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

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

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

Caution 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 DTC transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

Caution 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.

R5F110xH, R5F111xH (x = M, N, P): FBF00H to FC309H

R5F110xJ, R5F111xJ (x = M, N, P): FBF00H to FC309H

Caution 5. Use of the RAM areas of the following products is prohibited when using the on-chip debugging trace function and rewriting the data flash memory, because these areas are used as stack memory.

R5F110xH, R5F111xH (x = M, N, P): FC300H to FC6FFH

R5F110xJ, R5F111xJ (x = M, N, P): FC300H to FC6FFH

3.2.2 General-purpose registers

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

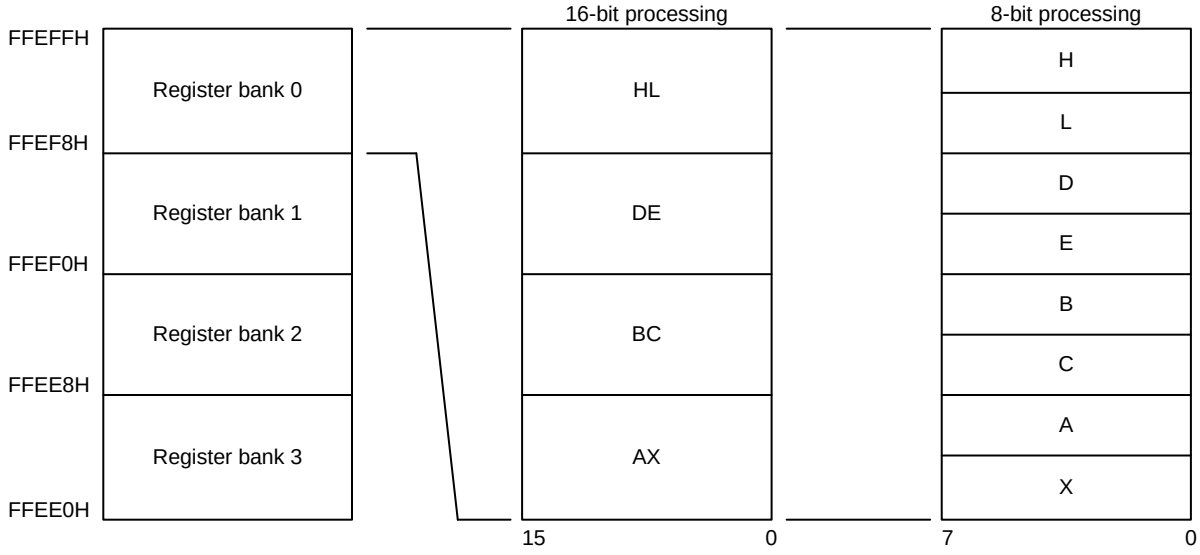
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

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

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

Figure 3 - 11 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.

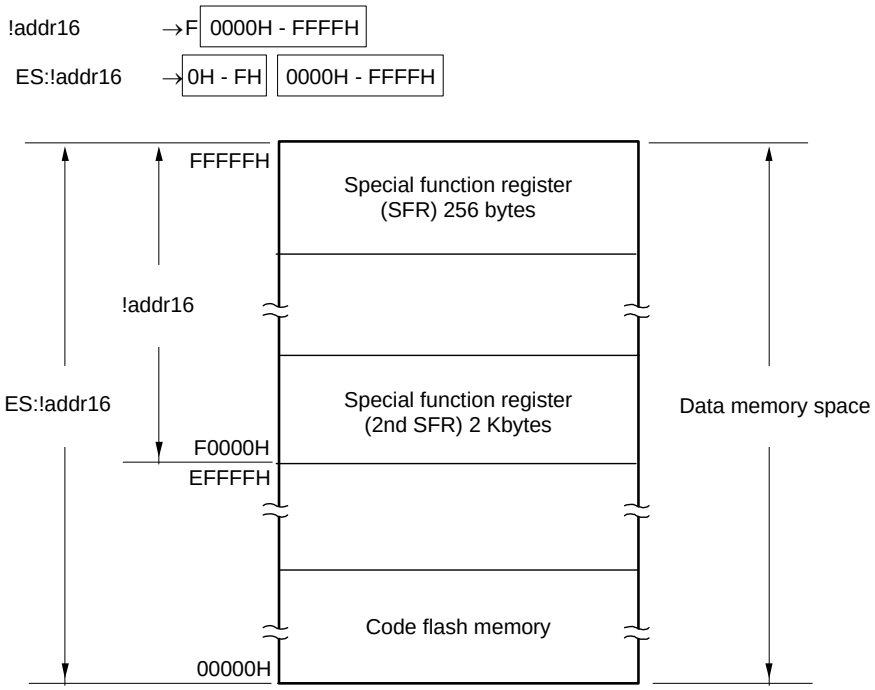
Figure 3 - 12 Configuration of ES and CS Registers

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

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

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

Figure 3 - 13 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 reserved 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 reserved by the assembler for the 16-bit manipulation instruction operand (sfrp).
  When specifying an address, describe an even address.

Tables 3 - 7 to 3 - 10 give lists of the SFRs. The meanings of items in the table are as follows.

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

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

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

Table 3 - 7 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
FFF05H	Port register 5	P5		R/W	√	√	—	00H
FFF06H	Port register 6	P6		R/W	√	√	—	00H
FFF07H	Port register 7	P7		R/W	√	√	—	00H
FFF08H	Port register 8	P8		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
FFF0FH	Port register 15	P15		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		—			—	—		
FFF14H	Serial data register 12	TXD3/ SIO30	SDR12	R/W	—	√	√	0000H
FFF15H		—			—	—		
FFF16H	Serial data register 13	RXD3	SDR13	R/W	—	√	√	0000H
FFF17H		—			—	—		
FFF18H	Timer data register 00	TDR00		R/W	—	—	√	0000H
FFF19H								
FFF1AH	Timer data register 01	TDR01L	TDR01	R/W	—	√	√	00H
FFF1BH		TDR01H			—	√		00H
FFF1EH	12-bit A/D conversion result register	ADCR		R	—	—	√	0000H
FFF1FH	8-bit A/D conversion result register	ADCRH		R	—	√	—	00H
FFF20H	Port mode register 0	PM0		R/W	√	√	—	FFH
FFF21H	Port mode register 1	PM1		R/W	√	√	—	FFH
FFF22H	Port mode register 2	PM2		R/W	√	√	—	FFH
FFF23H	Port mode register 3	PM3		R/W	√	√	—	FFH
FFF24H	Port mode register 4	PM4		R/W	√	√	—	FFH
FFF25H	Port mode register 5	PM5		R/W	√	√	—	FFH
FFF26H	Port mode register 6	PM6		R/W	√	√	—	FFH
FFF27H	Port mode register 7	PM7		R/W	√	√	—	FFH
FFF2CH	Port mode register 12	PM12		R/W	√	√	—	FFH
FFF2EH	Port mode register 14	PM14		R/W	√	√	—	FFH
FFF2FH	Port mode register 15	PM15		R/W	√	√	—	FFH
FFF30H	A/D converter mode register 0	ADM0		R/W	√	√	—	00H

Table 3 - 8 SFR List (2/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF31H	Analog input channel specification register	ADS		R/W	√	√	—	00H
FFF32H	A/D converter mode register 1	ADM1		R/W	√	√	—	00H
FFF34H	Key return control register	KRCTL		R/W	√	√	—	00H
FFF35H	Key return flag register	KRF		R/W	—	√	—	00H
FFF37H	Key return mode register	KRM0		R/W	√	√	—	00H
FFF38H	External interrupt rising edge enable register 0	EGP0		R/W	√	√	—	00H
FFF39H	External interrupt falling edge enable register 0	EGN0		R/W	√	√	—	00H
FFF40H	LCD mode register 0	LCDM0		R/W	—	√	—	00H
FFF41H	LCD mode register 1	LCDM1		R/W	√	√	—	00H
FFF42H	LCD clock control register 0	LCDC0		R/W	—	√	—	00H
FFF43H	LCD boost level control register	VLCD		R/W	—	√	—	04H
FFF44H	Serial data register 02	TXD1/ SIO10	SDR02	R/W	—	√	√	0000H
FFF45H		—			—	—		
FFF46H	Serial data register 03	RXD1	SDR03	R/W	—	√	√	0000H
FFF47H		—			—	—		
FFF48H	Serial data register 10	TXD2/ SIO20	SDR10	R/W	—	√	√	0000H
FFF49H		—			—	—		
FFF4AH	Serial data register 11	RXD2	SDR11	R/W	—	√	√	0000H
FFF4BH		—			—	—		
FFF50H	IICA shift register 0	IICA0		R/W	—	√	—	00H
FFF51H	IICA status register 0	IICS0		R	√	√	—	00H
FFF52H	IICA flag register 0	IICF0		R/W	√	√	—	00H
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								

Table 3 - 9 SFR List (3/4)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
FFF74H	D/A conversion value setting register 0	DACS0	R/W	—	√	—	00H
FFF75H	D/A conversion value setting register 1	DACS1	R/W	—	√	—	00H
FFF76H	D/A converter mode register	DAM	R/W	√	√	—	00H
FFF90H	12-bit interval timer control register	ITMC	R/W	—	—	√	0FFFH
FFF91H							
FFF92H	Second count register	SEC	R/W	—	√	—	Undefined
FFF93H	Minute count register	MIN	R/W	—	√	—	Undefined
FFF94H	Hour count register	HOUR	R/W	—	√	—	Undefined
FFF95H	Week count register	WEEK	R/W	—	√	—	Undefined
FFF96H	Day count register	DAY	R/W	—	√	—	Undefined
FFF97H	Month count register	MONTH	R/W	—	√	—	Undefined
FFF98H	Year count register	YEAR	R/W	—	√	—	Undefined
FFF9AH	Alarm minute register	ALARMWM	R/W	—	√	—	Undefined
FFF9BH	Alarm hour register	ALARMWH	R/W	—	√	—	Undefined
FFF9CH	Alarm week register	ALARMWW	R/W	—	√	—	Undefined
FFF9DH	Real-time clock control register 0	RTCC0	R/W	√	√	—	00H Note 1
FFF9EH	Real-time clock control register 1	RTCC1	R/W	√	√	—	00H Note 1
FFFA0H	Clock operation mode control register	CMC	R/W	—	√	—	00H Note 1
FFFA1H	Clock operation status control register	CSC	R/W	√	√	—	C0H Note 1
FFFA2H	Oscillation stabilization time counter status register	OSTC	R	√	√	—	00H
FFFA3H	Oscillation stabilization time select register	OSTS	R/W	—	√	—	07H
FFFA4H	System clock control register	CKC	R/W	√	√	—	00H
FFFA5H	Clock output select register 0	CKS0	R/W	√	√	—	00H
FFFA6H	Clock output select register 1	CKS1	R/W	√	√	—	00H
FFFA8H	Reset control flag register	RESF	R	—	√	—	Undefined Note 2
FFFA9H	Voltage detection register	LVIM	R/W	√	√	—	00H Note 2
FFFAAH	Voltage detection level register	LVIS	R/W	√	√	—	00H/01H/81H Note 2
FFFABH	Watchdog timer enable register	WDTE	R/W	—	√	—	9AH/1AH Note 3
FFFACH	CRC input register	CRCIN	R/W	—	√	—	00H

Note 1. This register is reset only by a power-on reset.

Note 2. The register states change depending on reset sources as shown below.

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

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

Table 3 - 10 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
FFFD2H	Interrupt request flag register 3L	IF3L	IF3	R/W	√	√	√	00H
FFFD4H	Interrupt mask flag register 2	MK2L	MK2	R/W	√	√	√	FFH
FFFD5H		MK2H		R/W	√	√		FFH
FFFD6H	Interrupt mask flag register 3L	MK3L	MK3	R/W	√	√	√	FFH
FFFD8H	Priority specification flag register 02	PR02L	PR02	R/W	√	√	√	FFH
FFFD9H		PR02H		R/W	√	√		FFH
FFFDAH	Priority specification flag register 03L	PR03L	PR03	R/W	√	√	√	FFH
FFFDCH	Priority specification flag register 12	PR12L	PR12	R/W	√	√	√	FFH
FFFDDH		PR12H		R/W	√	√		FFH
FFFDEH	Priority specification flag register 13L	PR13L	PR13	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	Multiply and accumulation register (L)	MACRL		R/W	—	—	√	0000H
FFFF1H								
FFFF2H	Multiply and accumulation register (H)	MACRH		R/W	—	—	√	0000H
FFFF3H								
FFFFEH	Processor mode control register	PMC		R/W	√	√	—	00H

Remark For extended SFRs (2nd SFRs), see **Tables 3 - 11** to **3 - 25 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 reserved 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 reserved by the assembler for the 16-bit manipulation instruction operand (!addr16). When specifying an address, describe an even address.

Tables 3 - 11 to 3 - 25 give lists of the extended SFRs. The meanings of items in the table are as follows.

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

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

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

Table 3 - 11 Extended SFR (2nd SFR) List (1/15)

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
F0032H	Pull-up resistor option register 2	PU2	R/W	√	√	—	00H
F0033H	Pull-up resistor option register 3	PU3	R/W	√	√	—	00H
F0034H	Pull-up resistor option register 4	PU4	R/W	√	√	—	01H
F0035H	Pull-up resistor option register 5	PU5	R/W	√	√	—	00H
F0037H	Pull-up resistor option register 7	PU7	R/W	√	√	—	00H
F0038H	Pull-up resistor option register 8	PU8	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
F0042H	Port input mode register 2	PIM2	R/W	√	√	—	00H
F0043H	Port input mode register 3	PIM3	R/W	√	√	—	00H
F0044H	Port input mode register 4	PIM4	R/W	√	√	—	00H
F0050H	Port output mode register 0	POM0	R/W	√	√	—	00H
F0051H	Port output mode register 1	POM1	R/W	√	√	—	00H
F0052H	Port output mode register 2	POM2	R/W	√	√	—	00H
F0053H	Port output mode register 3	POM3	R/W	√	√	—	00H
F0054H	Port output mode register 4	POM4	R/W	√	√	—	00H
F0062H	Port mode control register 2	PMC2	R/W	√	√	—	FFH
F0064H	Port mode control register 4	PMC4	R/W	√	√	—	00H
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
F0077H	Peripheral I/O redirection register	PIOR	R/W	—	√	—	00H
F0078H	Invalid memory access detection control register	IAWCTL	R/W	—	√	—	00H
F0079H	Timer output select register	TOS	R/W	√	√	—	00H
F007AH	Peripheral enable register 1	PER1	R/W	√	√	—	00H
F007BH	Port mode select register	PMS	R/W	√	√	—	00H

Table 3 - 12 Extended SFR (2nd SFR) List (2/15)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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	—	√	—	The value set by FRQSEL2 to FRQSEL0 of the option byte (000C2H) Note 2
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
F00F9H	Power-on-reset status register	PORSR		R/W	—	√	—	00H Note 3
F00FDH	Peripheral enable register 2	PER2		R/W	√	√	—	00H
F00FEH	BCD correction result register	BCDADJ		R	—	√	—	Undefined
F0100H	Serial status register 00	SSR00L	SSR00	R	—	√	√	0000H
F0101H		—			—	—		
F0102H	Serial status register 01	SSR01L	SSR01	R	—	√	√	0000H
F0103H		—			—	—		
F0104H	Serial status register 02	SSR02L	SSR02	R	—	√	√	0000H
F0105H		—			—	—		
F0106H	Serial status register 03	SSR03L	SSR03	R	—	√	√	0000H
F0107H		—			—	—		
F0108H	Serial flag clear trigger register 00	SIR00L	SIR00	R/W	—	√	√	0000H
F0109H		—			—	—		
F010AH	Serial flag clear trigger register 01	SIR01L	SIR01	R/W	—	√	√	0000H
F010BH		—			—	—		
F010CH	Serial flag clear trigger register 02	SIR02L	SIR02	R/W	—	√	√	0000H
F010DH		—			—	—		
F010EH	Serial flag clear trigger register 03	SIR03L	SIR03	R/W	—	√	√	0000H
F010FH		—			—	—		
F0110H	Serial mode register 00	SMR00		R/W	—	—	√	0020H
F0111H								
F0112H	Serial mode register 01	SMR01		R/W	—	—	√	0020H
F0113H								
F0114H	Serial mode register 02	SMR02		R/W	—	—	√	0020H
F0115H								
F0116H	Serial mode register 03	SMR03		R/W	—	—	√	0020H
F0117H								
F0118H	Serial communication operation setting register 00	SCR00		R/W	—	—	√	0087H
F0119H								
F011AH	Serial communication operation setting register 01	SCR01		R/W	—	—	√	0087H
F011BH								
F011CH	Serial communication operation setting register 02	SCR02		R/W	—	—	√	0087H
F011DH								

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

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

Note 3. This register is reset only by a power-on reset.

Table 3 - 13 Extended SFR (2nd SFR) List (3/15)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F011EH	Serial communication operation setting register 03	SCR03		R/W	—	—	√	0087H
F011FH								
F0120H	Serial channel enable status register 0	SE0L	SE0	R	√	√	√	0000H
F0121H		—			—	—		
F0122H	Serial channel start register 0	SS0L	SS0	R/W	√	√	√	0000H
F0123H		—			—	—		
F0124H	Serial channel stop register 0	ST0L	ST0	R/W	√	√	√	0000H
F0125H		—			—	—		
F0126H	Serial clock select register 0	SPS0L	SPS0	R/W	—	√	√	0000H
F0127H		—			—	—		
F0128H	Serial output register 0	SO0		R/W	—	—	√	0F0FH
F0129H								
F012AH	Serial output enable register 0	SOE0L	SOE0	R/W	√	√	√	0000H
F012BH		—			—	—		
F0134H	Serial output level register 0	SOL0L	SOL0	R/W	—	√	√	0000H
F0135H		—			—	—		
F0138H	Serial standby control register 0	SSC0L	SSC0	R/W	—	√	√	0000H
F0139H		—			—	—		
F0140H	Serial status register 10	SSR10L	SSR10	R	—	√	√	0000H
F0141H		—			—	—		
F0142H	Serial status register 11	SSR11L	SSR11	R	—	√	√	0000H
F0143H		—			—	—		
F0144H	Serial status register 12	SSR12L	SSR12	R	—	√	√	0000H
F0145H		—			—	—		
F0146H	Serial status register 13	SSR13L	SSR13	R	—	√	√	0000H
F0147H		—			—	—		
F0148H	Serial flag clear trigger register 10	SIR10L	SIR10	R/W	—	√	√	0000H
F0149H		—			—	—		
F014AH	Serial flag clear trigger register 11	SIR11L	SIR11	R/W	—	√	√	0000H
F014BH		—			—	—		
F014CH	Serial flag clear trigger register 12	SIR12L	SIR12	R/W	—	√	√	0000H
F014DH		—			—	—		
F014EH	Serial flag clear trigger register 13	SIR13L	SIR13	R/W	—	√	√	0000H
F014FH		—			—	—		
F0150H	Serial mode register 10	SMR10		R/W	—	—	√	0020H
F0151H								
F0152H	Serial mode register 11	SMR11		R/W	—	—	√	0020H
F0153H								
F0154H	Serial mode register 12	SMR12		R/W	—	—	√	0020H
F0155H								

Table 3 - 14 Extended SFR (2nd SFR) List (4/15)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0156H	Serial mode register 13	SMR13		R/W	—	—	√	0020H
F0157H								
F0158H	Serial communication operation setting register 10	SCR10		R/W	—	—	√	0087H
F0159H								
F015AH	Serial communication operation setting register 11	SCR11		R/W	—	—	√	0087H
F015BH								
F015CH	Serial communication operation setting register 12	SCR12		R/W	—	—	√	0087H
F015DH								
F015EH	Serial communication operation setting register 13	SCR13		R/W	—	—	√	0087H
F015FH								
F0160H	Serial channel enable status register 1	SE1L	SE1	R	√	√	√	0000H
F0161H		—			—	—		
F0162H	Serial channel start register 1	SS1L	SS1	R/W	√	√	√	0000H
F0163H		—			—	—		
F0164H	Serial channel stop register 1	ST1L	ST1	R/W	√	√	√	0000H
F0165H		—			—	—		
F0166H	Serial clock select register 1	SPS1L	SPS1	R/W	—	√	√	0000H
F0167H		—			—	—		
F0168H	Serial output register 1	SO1		R/W	—	—	√	0F0FH
F0169H								
F016AH	Serial output enable register 1	SOE1L	SOE1	R/W	√	√	√	0000H
F016BH		—			—	—		
F0174H	Serial output level register 1	SOL1L	SOL1	R/W	—	√	√	0000H
F0175H		—			—	—		
F0178H	Serial standby control register 1	SSC1L	SSC1	R/W	—	√	√	0000H
F0179H		—			—	—		
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								

Table 3 - 15 Extended SFR (2nd SFR) List (5/15)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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								
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		—			—	—		

Table 3 - 16 Extended SFR (2nd SFR) List (6/15)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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		—			—	—		
F01C0H	Event link setting register 00	ELSELR00		R/W	—	√	—	00H
F01C1H	Event link setting register 01	ELSELR01		R/W	—	√	—	00H
F01C2H	Event link setting register 02	ELSELR02		R/W	—	√	—	00H
F01C3H	Event link setting register 03	ELSELR03		R/W	—	√	—	00H
F01C4H	Event link setting register 04	ELSELR04		R/W	—	√	—	00H
F01C5H	Event link setting register 05	ELSELR05		R/W	—	√	—	00H
F01C6H	Event link setting register 06	ELSELR06		R/W	—	√	—	00H
F01C7H	Event link setting register 07	ELSELR07		R/W	—	√	—	00H
F01C8H	Event link setting register 08	ELSELR08		R/W	—	√	—	00H
F01C9H	Event link setting register 09	ELSELR09		R/W	—	√	—	00H
F01CAH	Event link setting register 10	ELSELR10		R/W	—	√	—	00H
F01CBH	Event link setting register 11	ELSELR11		R/W	—	√	—	00H
F01CCH	Event link setting register 12	ELSELR12		R/W	—	√	—	00H
F01CDH	Event link setting register 13	ELSELR13		R/W	—	√	—	00H
F01CEH	Event link setting register 14	ELSELR14		R/W	—	√	—	00H
F01CFH	Event link setting register 15	ELSELR15		R/W	—	√	—	00H
F01D0H	Event link setting register 16	ELSELR16		R/W	—	√	—	00H
F01D1H	Event link setting register 17	ELSELR17		R/W	—	√	—	00H
F01D2H	Event link setting register 18	ELSELR18		R/W	—	√	—	00H
F01D3H	Event link setting register 19	ELSELR19		R/W	—	√	—	00H
F01D4H	Event link setting register 20	ELSELR20		R/W	—	√	—	00H
F01D5H	Event link setting register 21	ELSELR21		R/W	—	√	—	00H
F01D6H	Event link setting register 22	ELSELR22		R/W	—	√	—	00H
F01D7H	Event link setting register 23	ELSELR23		R/W	—	√	—	00H
F01D8H	Event link setting register 24	ELSELR24		R/W	—	√	—	00H
F01D9H	Event link setting register 25	ELSELR25		R/W	—	√	—	00H

Table 3 - 17 Extended SFR (2nd SFR) List (7/15)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F01DAH	Event link setting register 26	ELSELR26	R/W	—	√	—	00H
F01DBH	Event link setting register 27	ELSELR27	R/W	—	√	—	00H
F01DCH	Event link setting register 28	ELSELR28	R/W	—	√	—	00H
F01DDH	Event link setting register 29	ELSELR29	R/W	—	√	—	00H
F01DEH	Event link setting register 30	ELSELR30	R/W	—	√	—	00H
F0230H	IICA control register 00	IICCTL00	R/W	√	√	—	00H
F0231H	IICA control register 01	IICCTL01	R/W	√	√	—	00H
F0232H	IICA low-level width setting register 0	IICWL0	R/W	—	√	—	FFH
F0233H	IICA high-level width setting register 0	IICWH0	R/W	—	√	—	FFH
F0234H	Slave address register 0	SVA0	R/W	—	√	—	00H
F0240H	16-bit timer KB2 compare register 10	TKBCR10	R/W	—	—	√	0000H
F0242H	16-bit timer KB2 compare register 11	TKBCR11	R/W	—	—	√	0000H
F0244H	16-bit timer KB2 compare register 12	TKBCR12	R/W	—	—	√	0000H
F0246H	16-bit timer KB2 compare register 13	TKBCR13	R/W	—	—	√	0000H
F0248H	16-bit timer KB2 trigger compare register 1	TKBTGCR1	R/W	—	—	√	0000H
F024AH	16-bit timer KB2 smooth start initial duty register 10	TKBSIR10	R/W	—	—	√	0000H
F024CH	16-bit timer KB2 smooth start initial duty register 11	TKBSIR11	R/W	—	—	√	0000H
F024EH	16-bit timer KB2 dithering count register 10	TKBDNR10	R/W	—	√	—	00H
F024FH	16-bit timer KB2 smooth start step width register 10	TKBSSR10	R/W	—	√	—	00H
F0250H	16-bit timer KB2 dithering count register 11	TKBDNR11	R/W	—	√	—	00H
F0251H	16-bit timer KB2 smooth start step width register 11	TKBSSR11	R/W	—	√	—	00H
F0252H	16-bit timer KB2 trigger register 1	TKBTRG1	W	√	√	—	00H
F0253H	16-bit timer KB2 flag register 1	TKBFLG1	R	√	√	—	00H
F0254H	16-bit timer KB2 compare 1L & dithering count register 10	TKBCRLD10	R/W	—	—	√	0000H
F0256H	16-bit timer KB2 compare 1L & dithering count register 11	TKBCRLD10	R/W	—	—	√	0000H
F0260H	16-bit timer counter register 1	TKBCNT1	R	—	—	√	0000H
F0262H	16-bit timer KB2 operation control register 10	TKBCTL10	R/W	—	—	√	0000H
F0264H	16-bit timer KB2 maximum frequency limit setting register 1	TKBMFR1	R/W	—	—	√	0000H

Table 3 - 18 Extended SFR (2nd SFR) List (8/15)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0266H	16-bit timer KB2 output control register 10	TKBIOC10	R/W	√	√	—	00H
F0267H	16-bit timer KB2 flag clear trigger register 1	TKBCLR1	W	√	√	—	00H
F0268H	16-bit timer KB2 output control register 11	TKBIOC11	R/W	√	√	—	00H
F0269H	16-bit timer KB2 operation control register 11	TKBCTL11	R/W	√	√	—	00H
F026AH	16-bit timer KB2 count clock division ratio select register 1	TKBPSCS1	R/W	—	√	—	00H
F0270H	Forced output stop function control register 10	TKBPACTL10	R/W	—	—	√	0000H
F0272H	Forced output stop function control register 11	TKBPACTL11	R/W	—	—	√	0000H
F0274H	Forced output stop function 1 start register 1	TKBPAHFS1	W	√	√	—	00H
F0275H	Forced output stop function stop trigger register 1	TKBPAHFT1	W	√	√	—	00H
F0276H	Forced output stop function flag register 1	TKBPAFLG1	R	√	√	—	00H
F0277H	Forced output stop function control register 12	TKBPACTL12	R/W	√	√	—	00H
F0280H	16-bit timer KB2 compare register 20	TKBCR20	R/W	—	—	√	0000H
F0282H	16-bit timer KB2 compare register 21	TKBCR21	R/W	—	—	√	0000H
F0284H	16-bit timer KB2 compare register 22	TKBCR22	R/W	—	—	√	0000H
F0286H	16-bit timer KB2 compare register 23	TKBCR23	R/W	—	—	√	0000H
F0288H	16-bit timer KB2 trigger compare register 2	TKBTGCR2	R/W	—	—	√	0000H
F028AH	16-bit timer KB2 smooth start initial duty register 20	TKBSIR20	R/W	—	—	√	0000H
F028CH	16-bit timer KB2 smooth start initial duty register 21	TKBSIR21	R/W	—	—	√	0000H
F028EH	16-bit timer KB2 dithering count register 20	TKBDNR20	R/W	—	√	—	00H
F028FH	16-bit timer KB2 smooth start step width register 20	TKBSSR20	R/W	—	√	—	00H
F0290H	16-bit timer KB2 dithering count register 21	TKBDNR21	R/W	—	√	—	00H
F0291H	16-bit timer KB2 smooth start step width register 21	TKBSSR21	R/W	—	√	—	00H
F0292H	16-bit timer KB2 trigger register 2	TKBTRG2	W	√	√	—	00H
F0293H	16-bit timer KB2 flag register 2	TKBFLG2	R	√	√	—	00H
F0294H	16-bit timer KB2 compare 1L & dithering count register 20	TKBCRLD20	R/W	—	—	√	0000H
F0296H	16-bit timer KB2 compare 1L & dithering count register 21	TKBCRLD21	R/W	—	—	√	0000H

Table 3 - 19 Extended SFR (2nd SFR) List (9/15)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F02A0H	16-bit timer counter register 2	TKBCNT2	R	—	—	√	0000H
F02A2H	16-bit timer KB2 operation control register 20	TKBCTL20	R/W	—	—	√	0000H
F02A4H	16-bit timer KB2 maximum frequency limit setting register 2	TKBMFR2	R/W	—	—	√	0000H
F02A6H	16-bit timer KB2 output control register 20	TKBIOC20	R/W	√	√	—	00H
F02A7H	16-bit timer KB2 flag clear trigger register 2	TKBCLR2	W	√	√	—	00H
F02A8H	16-bit timer KB2 output control register 21	TKBIOC21	R/W	√	√	—	00H
F02A9H	16-bit timer KB2 operation control register 21	TKBCTL21	R/W	√	√	—	00H
F02AAH	16-bit timer KB2 count clock division ratio select register 2	TKBPSCS2	R/W	—	√	—	00H
F02B0H	Forced output stop function control register 20	TKBPACTL20	R/W	—	—	√	0000H
F02B2H	Forced output stop function control register 21	TKBPACTL21	R/W	—	—	√	0000H
F02B4H	Forced output stop function 1 start trigger register 2	TKBPAHFS2	W	√	√	—	00H
F02B5H	Forced output stop function stop trigger register 2	TKBPAHFT2	W	√	√	—	00H
F02B6H	Forced output stop function flag register 2	TKBPAFLG2	R	√	√	—	00H
F02B7H	Forced output stop function control register 22	TKBPACTL22	R/W	√	√	—	00H
F02E0H	DTC base address register	DTCBAR	R/W	—	√	—	FDH
F02E5H	PLL control register Note 1	DSCCTL	R/W	√	√	—	00H
F02E6H	Main clock control register	MCKC	R/W	√	√	—	00H
F02E8H	DTC activation enable register 0	DTCEN0	R/W	√	√	—	00H
F02E9H	DTC activation enable register 1	DTCEN1	R/W	√	√	—	00H
F02EAH	DTC activation enable register 2	DTCEN2	R/W	√	√	—	00H
F02EBH	DTC activation enable register 3	DTCEN3	R/W	√	√	—	00H
F02ECH	DTC activation enable register 4	DTCEN4	R/W	√	√	—	00H
F02F0H	Flash memory CRC control register	CRC0CTL	R/W	√	√	—	00H
F02F2H	Flash memory CRC operation result register	PGCRCL	R/W	—	—	√	0000H
F02FAH	CRC data register	CRCD	R/W	—	—	√	0000H
F0300H	LCD port function register 0	PFSEG0	R/W	√	√	—	F0H
F0301H	LCD port function register 1	PFSEG1	R/W	√	√	—	FFH
F0302H	LCD port function register 2	PFSEG2	R/W	√	√	—	FFH
F0303H	LCD port function register 3	PFSEG3	R/W	√	√	—	FFH
F0304H	LCD port function register 4	PFSEG4	R/W	√	√	—	FFH
F0305H	LCD port function register 5	PFSEG5	R/W	√	√	—	FFH
F0306H	LCD port function register 6	PFSEG6	R/W	√	√	—	FFH
F0308H	LCD input switch control register	ISCLCD	R/W	√	√	—	00H
F0310H	Watch error correction register	SUBCUD	R/W	—	—	√	0020H Note 2
F0311H							

Note 1. Products with USB only.

Note 2. This register is reset only by a power-on reset.

Table 3 - 20 Extended SFR (2nd SFR) List (10/15)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0340H	Comparator mode setting register	COMPMDR	R/W	√	√	—	00H
F0341H	Comparator filter control register	COMPFIR	R/W	√	√	—	00H
F0342H	Comparator output control register	COMPOCR	R/W	√	√	—	00H
F0400H	LCD display data memory 0	SEG0	R/W	—	√	—	00H
F0401H	LCD display data memory 1	SEG1	R/W	—	√	—	00H
F0402H	LCD display data memory 2	SEG2	R/W	—	√	—	00H
F0403H	LCD display data memory 3	SEG3	R/W	—	√	—	00H
F0404H	LCD display data memory 4	SEG4	R/W	—	√	—	00H
F0405H	LCD display data memory 5	SEG5	R/W	—	√	—	00H
F0406H	LCD display data memory 6	SEG6	R/W	—	√	—	00H
F0407H	LCD display data memory 7	SEG7	R/W	—	√	—	00H
F0408H	LCD display data memory 8	SEG8	R/W	—	√	—	00H
F0409H	LCD display data memory 9	SEG9	R/W	—	√	—	00H
F040AH	LCD display data memory 10	SEG10	R/W	—	√	—	00H
F040BH	LCD display data memory 11	SEG11	R/W	—	√	—	00H
F040CH	LCD display data memory 12	SEG12	R/W	—	√	—	00H
F040DH	LCD display data memory 13	SEG13	R/W	—	√	—	00H
F040EH	LCD display data memory 14	SEG14	R/W	—	√	—	00H
F040FH	LCD display data memory 15	SEG15	R/W	—	√	—	00H
F0410H	LCD display data memory 16	SEG16	R/W	—	√	—	00H
F0411H	LCD display data memory 17	SEG17	R/W	—	√	—	00H
F0412H	LCD display data memory 18	SEG18	R/W	—	√	—	00H
F0413H	LCD display data memory 19	SEG19	R/W	—	√	—	00H
F0414H	LCD display data memory 20	SEG20	R/W	—	√	—	00H
F0415H	LCD display data memory 21	SEG21	R/W	—	√	—	00H
F0416H	LCD display data memory 22	SEG22	R/W	—	√	—	00H
F0417H	LCD display data memory 23	SEG23	R/W	—	√	—	00H
F0418H	LCD display data memory 24	SEG24	R/W	—	√	—	00H
F0419H	LCD display data memory 25	SEG25	R/W	—	√	—	00H
F041AH	LCD display data memory 26	SEG26	R/W	—	√	—	00H
F041BH	LCD display data memory 27	SEG27	R/W	—	√	—	00H
F041CH	LCD display data memory 28	SEG28	R/W	—	√	—	00H
F041DH	LCD display data memory 29	SEG29	R/W	—	√	—	00H
F041EH	LCD display data memory 30	SEG30	R/W	—	√	—	00H
F041FH	LCD display data memory 31	SEG31	R/W	—	√	—	00H

Table 3 - 21 Extended SFR (2nd SFR) List (11/15)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0420H	LCD display data memory 32	SEG32	R/W	—	√	—	00H
F0421H	LCD display data memory 33	SEG33	R/W	—	√	—	00H
F0422H	LCD display data memory 34	SEG34	R/W	—	√	—	00H
F0423H	LCD display data memory 35	SEG35	R/W	—	√	—	00H
F0424H	LCD display data memory 36	SEG36	R/W	—	√	—	00H
F0425H	LCD display data memory 37	SEG37	R/W	—	√	—	00H
F0426H	LCD display data memory 38	SEG38	R/W	—	√	—	00H
F0427H	LCD display data memory 39	SEG39	R/W	—	√	—	00H
F0428H	LCD display data memory 40	SEG40	R/W	—	√	—	00H
F0429H	LCD display data memory 41	SEG41	R/W	—	√	—	00H
F042AH	LCD display data memory 42	SEG42	R/W	—	√	—	00H
F042BH	LCD display data memory 43	SEG43	R/W	—	√	—	00H
F042CH	LCD display data memory 44	SEG44	R/W	—	√	—	00H
F042DH	LCD display data memory 45	SEG45	R/W	—	√	—	00H
F042EH	LCD display data memory 46	SEG46	R/W	—	√	—	00H
F042FH	LCD display data memory 47	SEG47	R/W	—	√	—	00H
F0430H	LCD display data memory 48	SEG48	R/W	—	√	—	00H
F0431H	LCD display data memory 49	SEG49	R/W	—	√	—	00H
F0432H	LCD display data memory 50	SEG50	R/W	—	√	—	00H
F0433H	LCD display data memory 51	SEG51	R/W	—	√	—	00H
F0434H	LCD display data memory 52	SEG52	R/W	—	√	—	00H
F0435H	LCD display data memory 53	SEG53	R/W	—	√	—	00H
F0436H	LCD display data memory 54	SEG54	R/W	—	√	—	00H
F0437H	LCD display data memory 55	SEG55	R/W	—	√	—	00H
F0500H	16-bit timer KB2 compare register 00	TKBCR00	R/W	—	—	√	0000H
F0502H	16-bit timer KB2 compare register 01	TKBCR01	R/W	—	—	√	0000H
F0504H	16-bit timer KB2 compare register 02	TKBCR02	R/W	—	—	√	0000H
F0506H	16-bit timer KB2 compare register 03	TKBCR03	R/W	—	—	√	0000H
F0508H	16-bit timer KB2 trigger compare register 0	TKBTGCR0	R/W	—	—	√	0000H
F050AH	16-bit timer KB2 smooth start initial duty register 00	TKBSIR00	R/W	—	—	√	0000H
F050CH	16-bit timer KB2 smooth start initial duty register 01	TKBSIR01	R/W	—	—	√	0000H
F050EH	16-bit timer KB2 dithering count register 00	TKBDNR00	R/W	—	√	—	00H
F050FH	16-bit timer KB2 smooth start step width register 00	TKBSSR00	R/W	—	√	—	00H

Table 3 - 22 Extended SFR (2nd SFR) List (12/15)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0510H	16-bit timer KB2 dithering count register 01	TKBDNR01	R/W	—	√	—	00H
F0511H	16-bit timer KB2 smooth start step width register 01	TKBSSR01	R/W	—	√	—	00H
F0512H	16-bit timer KB2 trigger register 0	TKBTRG0	W	√	√	—	00H
F0513H	16-bit timer KB2 flag register 0	TKBFLG0	R	√	√	—	00H
F0514H	16-bit timer KB2 compare 1L & dithering count register 00	TKBCRLD00	R/W	—	—	√	0000H
F0516H	16-bit timer KB2 compare 1L & dithering count register 01	TKBCRLD01	R/W	—	—	√	0000H
F0520H	16-bit timer counter register 0	TKBCNT0	R	—	—	√	0000H
F0522H	16-bit timer KB2 operation control register 00	TKBCTL00	R/W	—	—	√	0000H
F0524H	16-bit timer KB2 maximum frequency limit setting register 0	TKBMFR0	R/W	—	—	√	0000H
F0526H	16-bit timer KB2 output control register 00	TKBIOC00	R/W	√	√	—	00H
F0527H	16-bit timer KB2 flag clear trigger register 0	TKBCLR0	W	√	√	—	00H
F0528H	16-bit timer KB2 output control register 01	TKBIOC01	R/W	√	√	—	00H
F0529H	16-bit timer KB2 operation control register 01	TKBCTL01	R/W	√	√	—	00H
F052AH	16-bit timer KB2 count clock division ratio select register 0	TKBPSCS0	R/W	—	√	—	00H
F0530H	Forced output stop function control register 00	TKBPACTL00	R/W	—	—	√	0000H
F0532H	Forced output stop function control register 01	TKBPACTL01	R/W	—	—	√	0000H
F0534H	Forced output stop function 1 start trigger register 0	TKBPAHFS0	W	√	√	—	00H
F0535H	Forced output stop function stop trigger register 0	TKBPAHFT0	W	√	√	—	00H
F0536H	Forced output stop function flag register 0	TKBPAFLG0	R	√	√	—	00H
F0537H	Forced output stop function control register 02	TKBPACTL02	R/W	√	√	—	00H
F0580H	DTC transfer D0FIFO port register Note	D0FIFOD00	R/W	—	√	√	0000H
F0581H							
F05C0H	DTC transfer D1FIFO port register Note	D1FIFOD00	R/W	—	√	√	0000H
F05C1H							
F0600H	System configuration control register Note	SYSCFG	R/W	—	—	√	0000H
F0601H							
F0604H	System configuration status register 0 Note	SYSSTS0	R	—	—	√	00000000 000000XXB
F0605H							
F0608H	Device state control register 0 Note	DVSTCTR0	R/W	—	—	√	0000H
F0609H							

Note Products with USB only.

Table 3 - 23 Extended SFR (2nd SFR) List (13/15)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0614H	CFIFO port register Note	CFIFOML	CFIFOM	R/W	—	√	√	0000H
F0615H		—			—	—		
F0618H	D0FIFO port register Note	D0FIFOML	D0FIFOM	R/W	—	√	√	0000H
F0619H		—			—	—		
F061CH	D1FIFO port register Note	D1FIFOML	D1FIFOM	R/W	—	√	√	0000H
F061DH		—			—	—		
F0620H	CFIFO port select register Note	CFIFOSEL		R/W	—	—	√	0000H
F0621H								
F0622H	CFIFO port control register Note	CFIFOCTR		R/W	—	—	√	0000H
F0623H								
F0628H	D0FIFO port selection register Note	D0FIFOSEL		R/W	—	—	√	0000H
F0629H								
F062AH	D0FIFO port control register Note	D0FIFOCTR		R/W	—	—	√	0000H
F062BH								
F062CH	D1FIFO port selection register Note	D1FIFOSEL		R/W	—	—	√	0000H
F062DH								
F062EH	D1FIFO port control register Note	D1FIFOCTR		R/W	—	—	√	0000H
F062FH								
F0630H	Interrupt enable register 0 Note	INTENB0		R/W	—	—	√	0000H
F0631H								
F0632H	Interrupt enable register 1 Note	INTENB1		R/W	—	—	√	0000H
F0633H								
F0636H	BRDY interrupt enable register Note	BRDYENB		R/W	—	—	√	0000H
F0637H								
F0638H	NRDY interrupt enable register Note	NRDYENB		R/W	—	—	√	0000H
F0639H								
F063AH	BEMP interrupt enable register Note	BEMPENB		R/W	—	—	√	0000H
F063BH								
F063CH	SOF output configuration register Note	SOFCFG		R/W	—	—	√	0000H
F063DH								
F0640H	Interrupt status register 0 Note	INTSTS0		R/W	—	—	√	00000000 X0000000B
F0641H								
F0642H	Interrupt status register 1 Note	INTSTS1		R/W	—	—	√	0000H
F0643H								
F0646H	BRDY interrupt status register Note	BRDYSTS		R/W	—	—	√	0000H
F0647H								
F0648H	NRDY interrupt status register Note	NRDYSTS		R/W	—	—	√	0000H
F0649H								

Note Products with USB only.

Table 3 - 24 Extended SFR (2nd SFR) List (14/15)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F064AH	BEMP interrupt status register Note	BEMPSTS	R/W	—	—	√	0000H
F064BH							
F064CH	Frame number register Note	FRMNUM	R	—	—	√	0000H
F064DH							
F0650H	USB address register Note	USBADDR	R	—	—	√	0000H
F0651H							
F0654H	USB request type register Note	USBREQ	R	—	—	√	0000H
F0655H							
F0656H	USB request value register Note	USBVAL	R	—	—	√	0000H
F0657H							
F0658H	USB request index register Note	USBINDX	R	—	—	√	0000H
F0659H							
F065AH	USB request length register Note	USBLENG	R	—	—	√	0000H
F065BH							
F065CH	DCP configuration register Note	DCPCFG	R/W	—	—	√	0000H
F065DH							
F065EH	DCP maximum packet size register Note	DCPMAXP	R/W	—	—	√	0040H
F065FH							
F0660H	DCP control register Note	DCPCTR	R/W	—	—	√	0040H
F0661H							
F0664H	Pipe window selection register Note	PIPESEL	R/W	—	—	√	0000H
F0665H							
F0668H	Pipe configuration register Note	PIPECFG	R/W	—	—	√	0000H
F0669H							
F066CH	Pipe maximum packet size register Note	PIPEMAXP	R/W	—	—	√	0000H/0040H
F066DH							
F0676H	Pipe 4 control register Note	PIPE4CTR	R/W	—	—	√	0000H
F0677H							
F0678H	Pipe 5 control register Note	PIPE5CTR	R/W	—	—	√	0000H
F0679H							
F067AH	Pipe 6 control register Note	PIPE6CTR	R/W	—	—	√	0000H
F067BH							
F067CH	Pipe 7 control register Note	PIPE7CTR	R/W	—	—	√	0000H
F067DH							
F069CH	Pipe 4 transaction counter enable register Note	PIPE4TRE	R/W	—	—	√	0000H
F069DH							
F069EH	Pipe 4 transaction counter register Note	PIPE4TRN	R/W	—	—	√	0000H
F069FH							

Note Products with USB only.

Table 3 - 25 Extended SFR (2nd SFR) List (15/15)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F06A0H	Pipe 5 transaction counter enable register Note	PIPE5TRE	R/W	—	—	√	0000H
F06A1H							
F06A2H	Pipe 5 transaction counter register Note	PIPE5TRN	R/W	—	—	√	0000H
F06A3H							
F06A8H	DTC0 to FIFO pin configuration register Note	DTC0PCFG	R/W	—	—	√	0000H
F06A9H							
F06ACH	DTC1 to FIFO pin configuration register Note	DTC1PCFG	R/W	—	—	√	0000H
F06ADH							
F06B0H	BC control register 0 Note	USBBCCTRL0	R/W	—	—	√	0000H
F06B1H							
F06C4H	USB clock selection register Note	UCKSEL	R/W	—	—	√	0000H
F06C5H							
F06B8H	BC option control register 0 Note	USBBCOPT0	R/W	—	—	√	0000H
F06B9H							
F06CCH	USB module control register Note	USBMC	R/W	—	—	√	0002H
F06CDH							

Note Products with USB only.

Remark For SFRs in the SFR area, see **Tables 3 - 7** to **3 - 10 SFR List**.

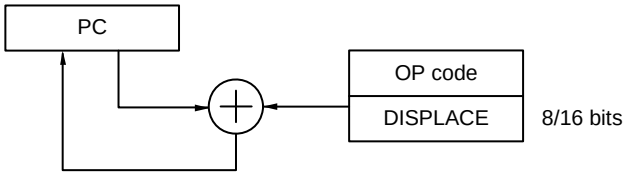
3.3 Instruction Address Addressing

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 - 14 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 - 15 Example of CALL !!addr20/BR !!addr20

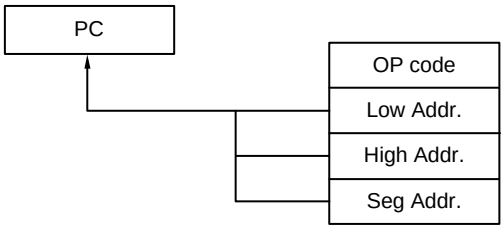
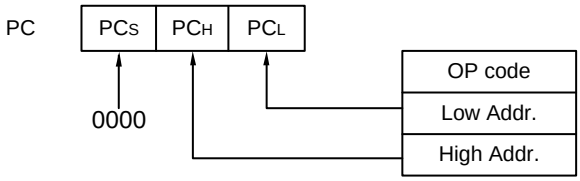


Figure 3 - 16 Example of CALL !addr16/BR !addr16



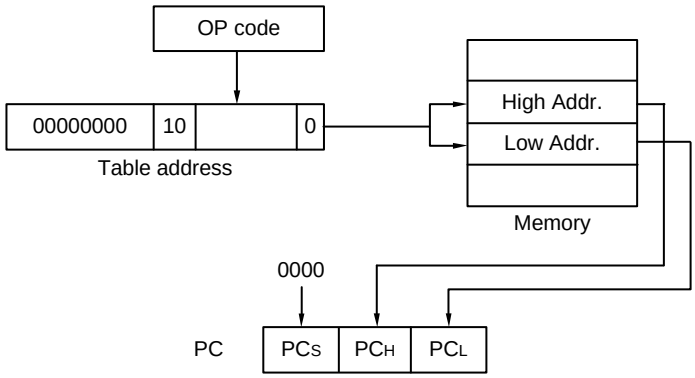
3.3.3 Table indirect addressing

[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 - 17 Outline of Table Indirect Addressing

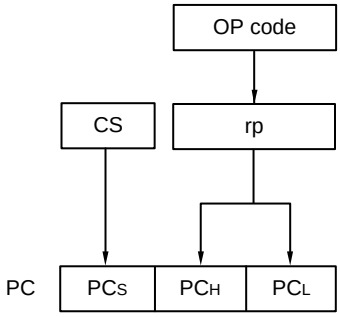


3.3.4 Register direct addressing

[Function]

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

Figure 3 - 18 Outline of Register Direct Addressing



3.4 Addressing for Processing Data Addresses

3.4.1 Implied addressing

[Function]

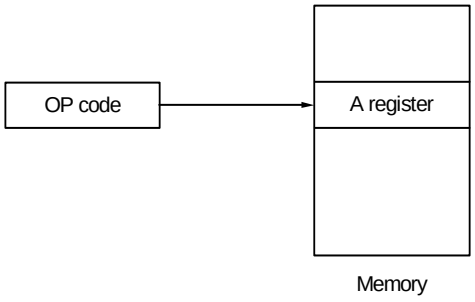
Instructions for accessing registers (such as accumulators) that have special functions are directly specified with the instruction word, without using any register specification field in the instruction word.

[Operand format]

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

Implied addressing can be applied only to MULU X.

Figure 3 - 19 Outline of Implied Addressing



3.4.2 Register addressing

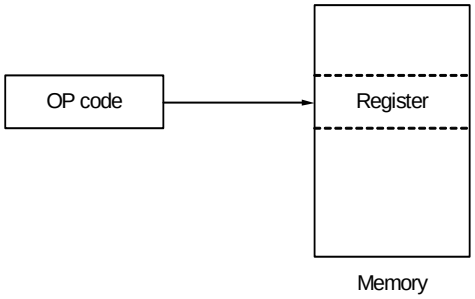
[Function]

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 - 20 Outline of Register Addressing



3.4.3 Direct addressing

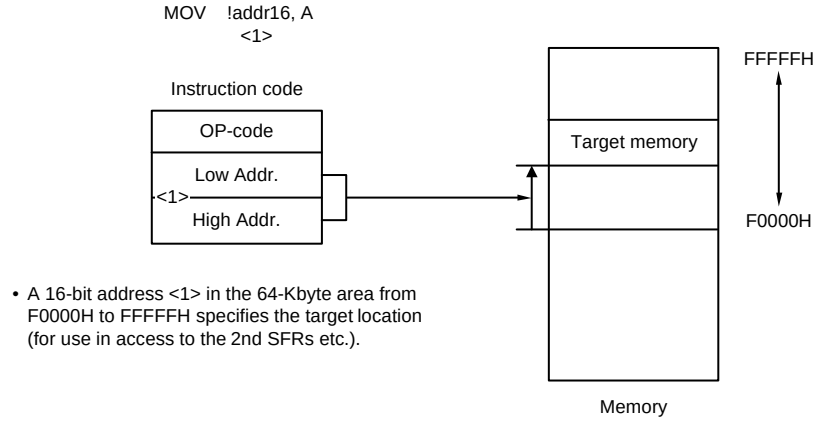
[Function]

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

[Operand format]

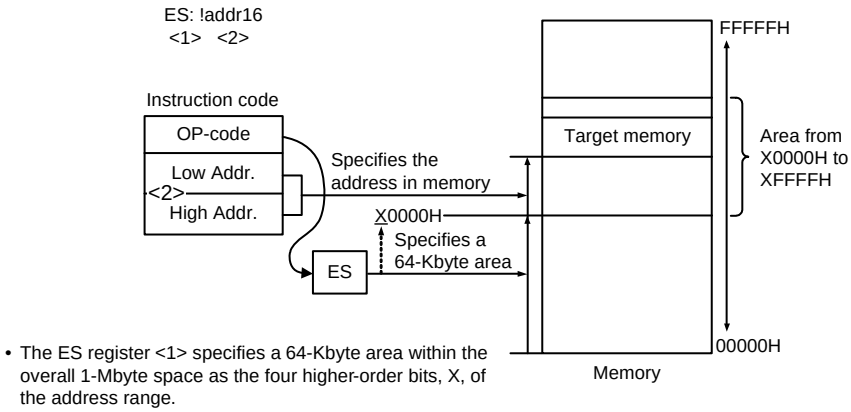
Identifier	Description
!addr16	Label or 16-bit immediate data (only the space from F0000H to FFFFFH is specifiable)
ES:!addr16	Label or 16-bit immediate data (higher 4-bit addresses are specified by the ES register)

Figure 3 - 21 Example of !addr16



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

Figure 3 - 22 Example of ES:!addr16



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

3.4.4 Short direct addressing

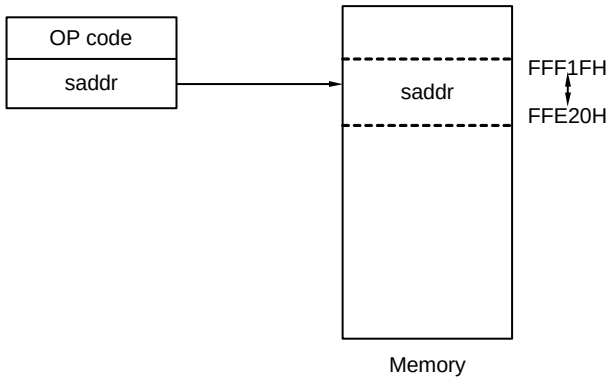
[Function]

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

[Operand format]

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

Figure 3 - 23 Outline of Short Direct Addressing



Remark SADDR and SADDRP are used to describe the values of addresses FE20H to FF1FH with 16-bit immediate data (higher 4 bits of actual address are omitted), and the values of addresses FFE20H to FFF1FH with 20-bit immediate data.
Regardless of whether SADDR or SADDRP is used, addresses within the space from FFE20H to FFF1FH are specified for the memory.

3.4.5 SFR addressing

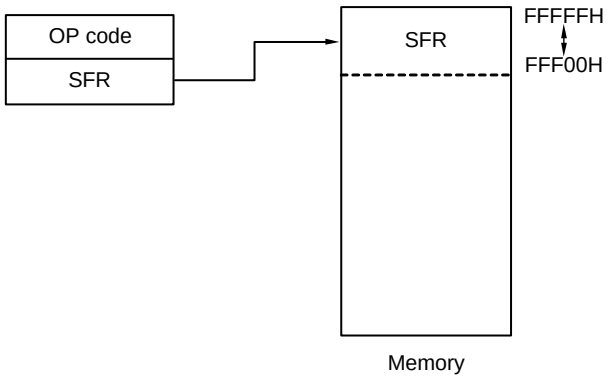
[Function]

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

[Operand format]

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

Figure 3 - 24 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 - 25 Example of [DE], [HL]

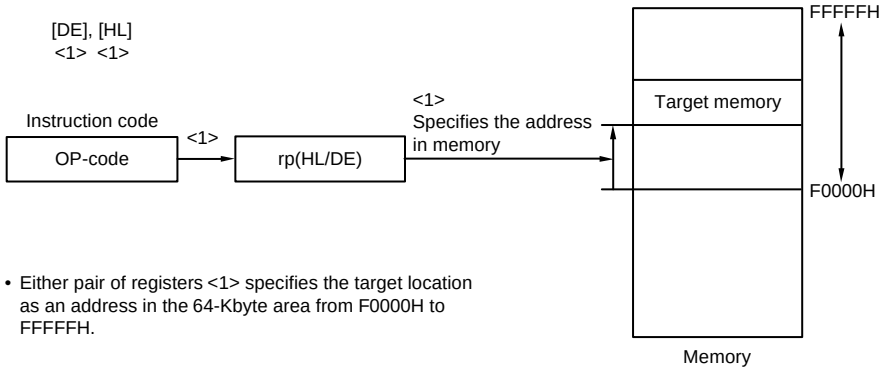
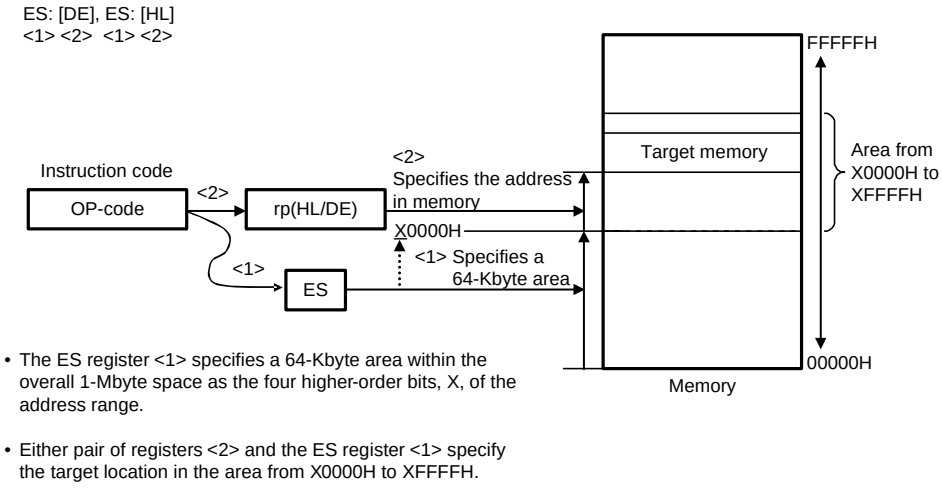


Figure 3 - 26 Example of ES:[DE], ES:[HL]



3.4.7 Based addressing

[Function]

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

[Operand format]

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

Figure 3 - 27 Example of [SP+byte]

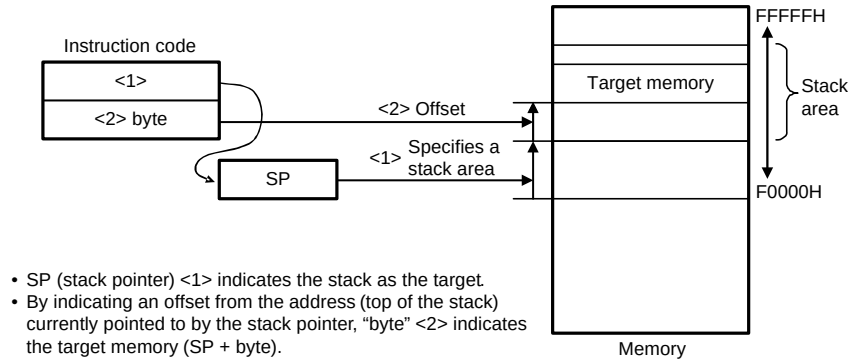
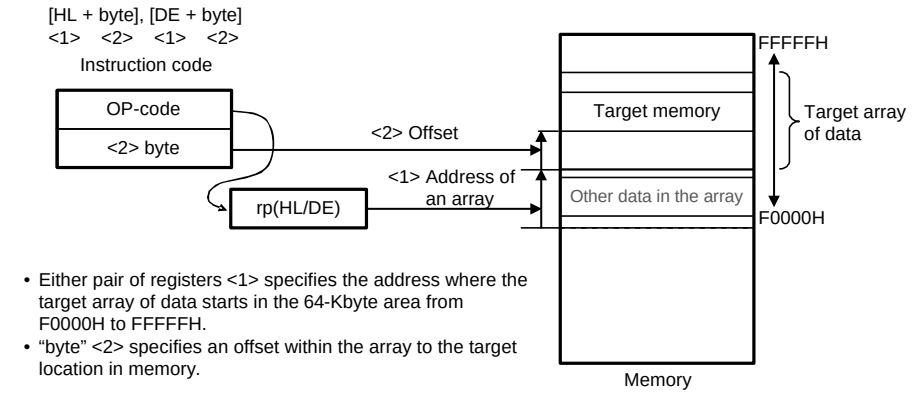
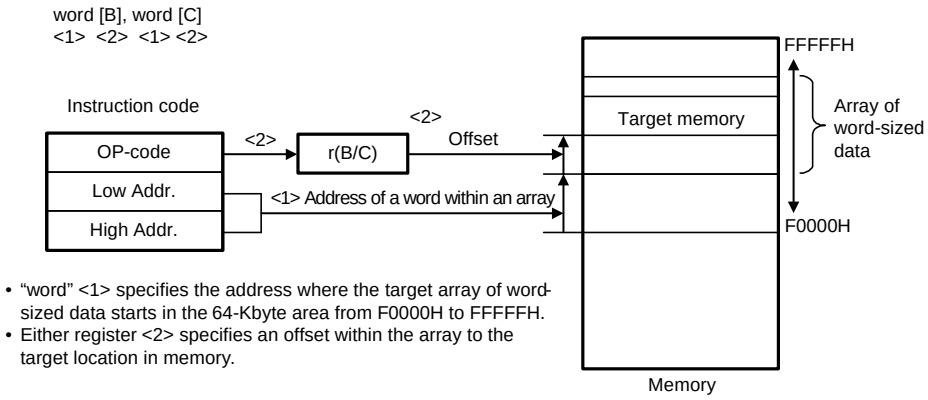


Figure 3 - 28 Example of [HL + byte], [DE + byte]]



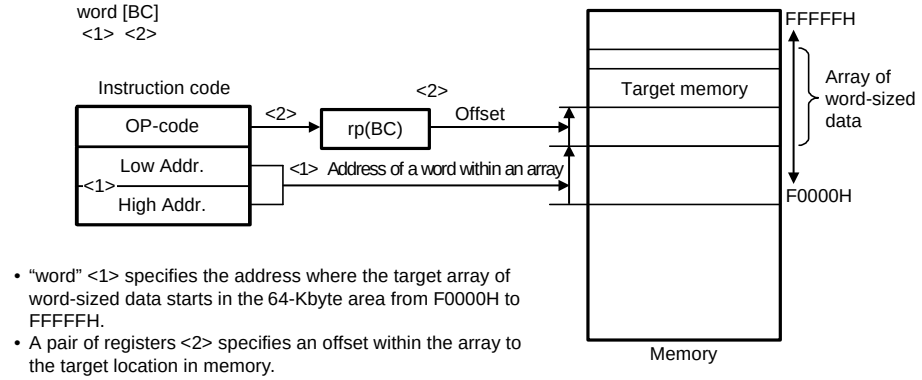
- Either pair of registers <1> specifies the address where the target array of data starts in the 64-Kbyte area from F0000H to FFFFFH.
- "byte" <2> specifies an offset within the array to the target location in memory.

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



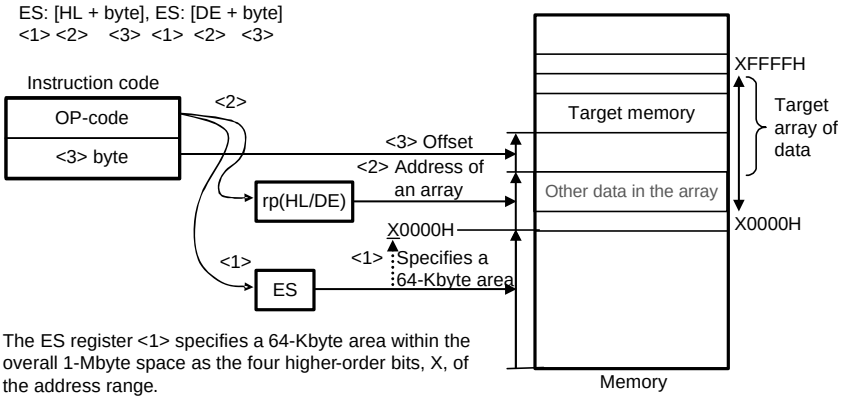
- "word" <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- Either register <2> specifies an offset within the array to the target location in memory.

Figure 3 - 30 Example of word[BC]



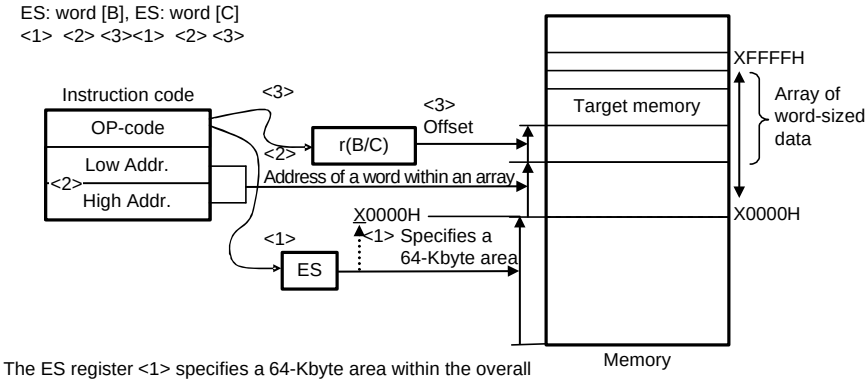
- "word" <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- A pair of registers <2> specifies an offset within the array to the target location in memory.

Figure 3 - 31 Example of ES:[HL + byte], ES:[DE + byte]]



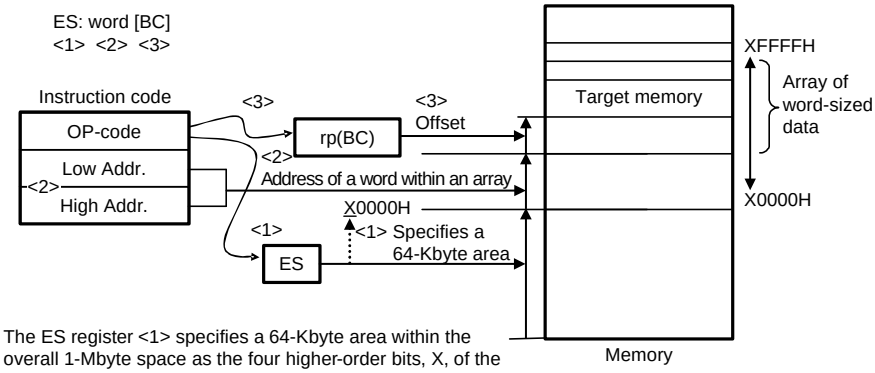
- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- Either pair of registers <2> specifies the address where the target array of data starts in the 64-Kbyte area specified in the ES register <1>.
- “byte” <3> specifies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

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



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

Figure 3 - 33 Example of ES:word[BC]



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

3.4.8 Based indexed addressing

[Function]

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

[Operand format]

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

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

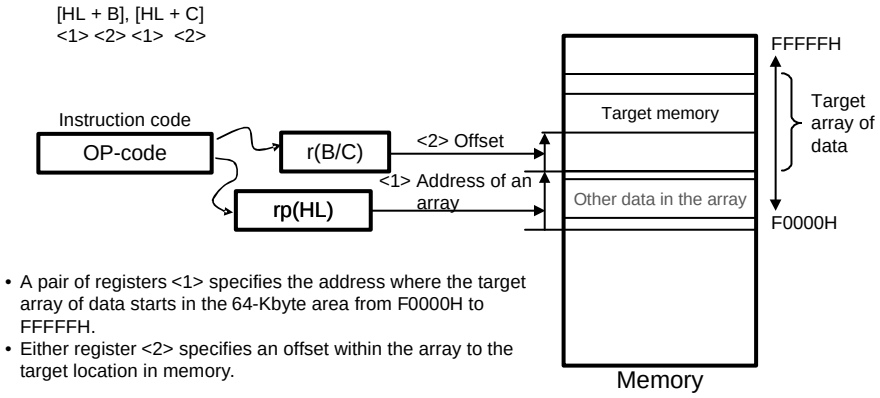
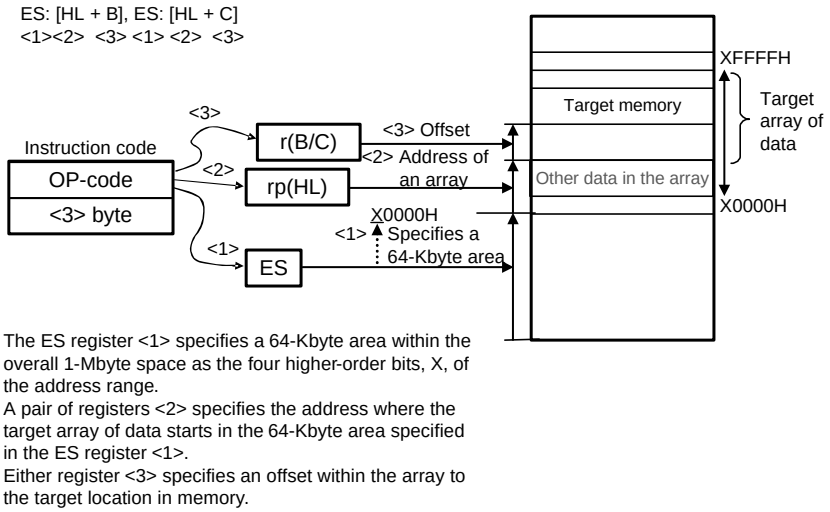


Figure 3 - 35 Example of ES:[HL+B], ES:[HL+C]



3.4.9 Stack addressing

[Function]

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

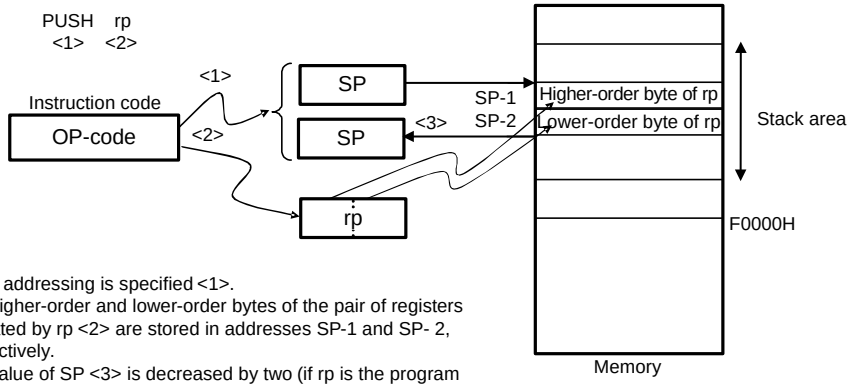
Only the internal RAM area can be set as the stack area.

[Description format]

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

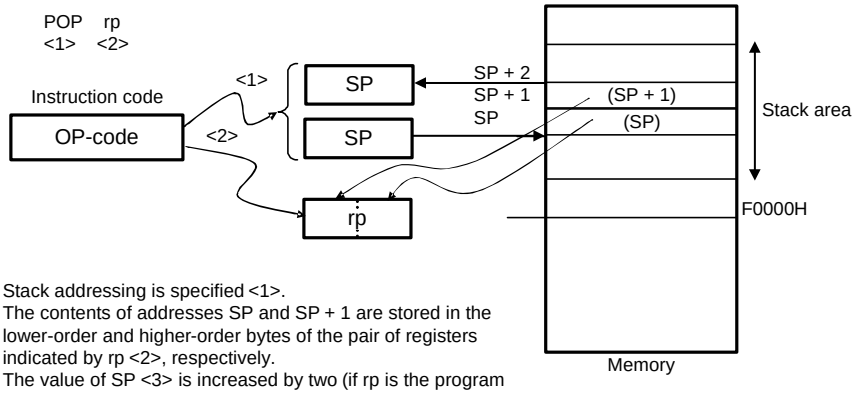
The data to be saved/restored by each stack operation is shown in Figures 3 - 36 to 3 - 41.

Figure 3 - 36 Example of PUSH rp



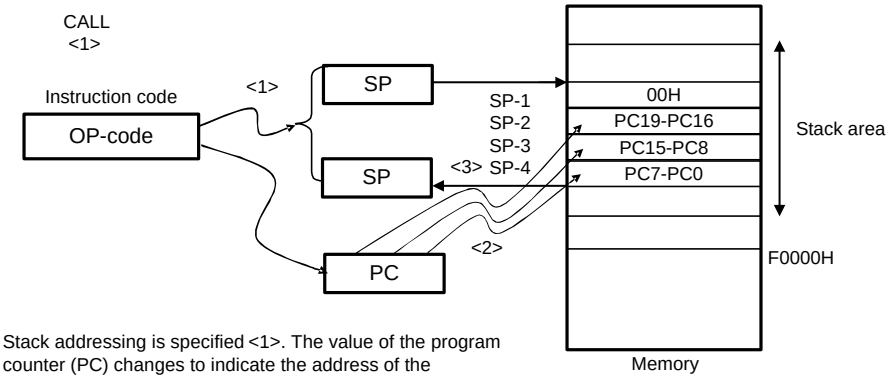
- Stack addressing is specified <1>.
- The higher-order and lower-order bytes of the pair of registers indicated by rp <2> are stored in addresses SP-1 and SP- 2, respectively.
- The value of SP <3> is decreased by two (if rp is the program status word (PSW), the value of the PSW is stored in SP-1 and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

Figure 3 - 37 Example of POP



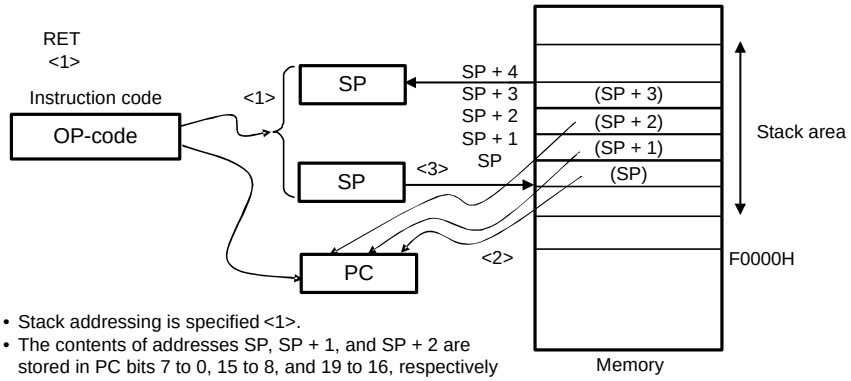
- Stack addressing is specified <1>.
- The contents of addresses SP and SP + 1 are stored in the lower-order and higher-order bytes of the pair of registers indicated by rp <2>, respectively.
- The value of SP <3> is increased by two (if rp is the program status word (PSW), the content of address SP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

Figure 3 - 38 Example of CALL, CALLT



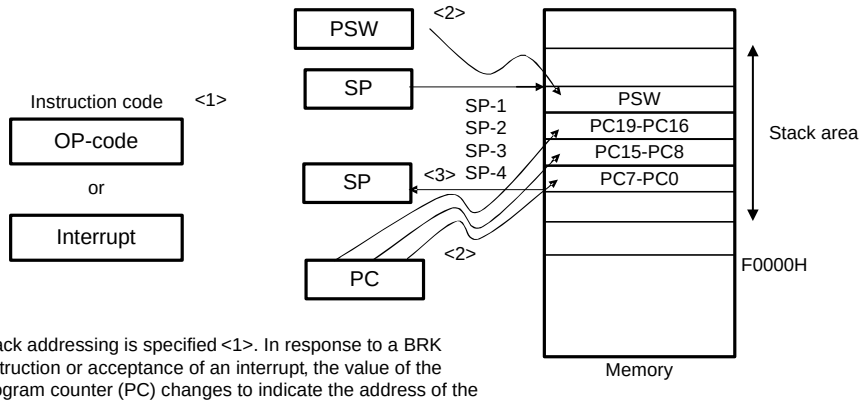
- Stack addressing is specified <1>. The value of the program counter (PC) changes to indicate the address of the instruction following the CALL instruction.
- 00H, the values of PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP-1, SP-2, SP-3, and SP-4, respectively <2>.
- The value of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

Figure 3 - 39 Example of RET



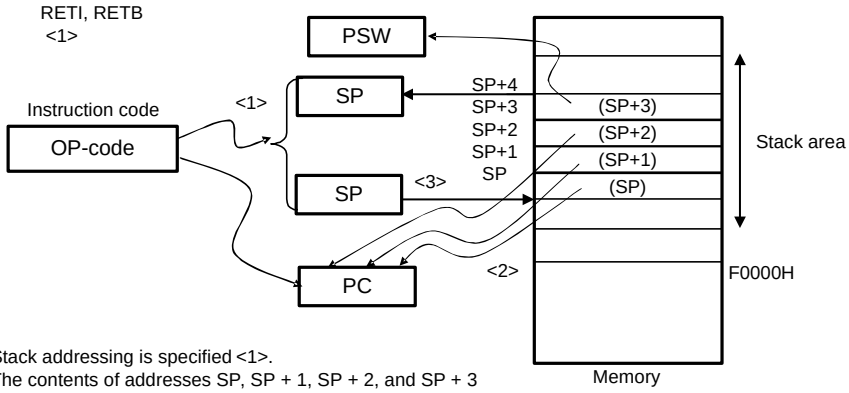
- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, and SP + 2 are stored in PC bits 7 to 0, 15 to 8, and 19 to 16, respectively <2>.
- The value of SP <3> is increased by four.

Figure 3 - 40 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 - 41 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/L1C 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 PM8, PM12, PM14, PM15) Port registers (P0 to P8, P12 to P15) Pull-up resistor option registers (PU0 to PU5, PU7, PU8, PU12, PU14) Port input mode registers (PIM0 to PIM4) Port output mode registers (POM0 to POM4) Port mode control registers (PMC2, PMC4, PMC14) A/D port configuration register (ADPC) Peripheral I/O redirection registers (PIOR) LCD port function register (PFSEG0 to PFSEG6) LCD input switch control register (ISCLCD)
Port	• 80/85-pin products (with USB) Total: 59 (CMOS I/O: 51 (N-ch open drain I/O [V_{DD} tolerance]: 12), CMOS input: 5, CMOS output: 1, N-ch open-drain I/O [6 V tolerance]: 2) • 80/85-pin products (without USB) Total: 63 (CMOS I/O: 55 (N-ch open drain I/O [V_{DD} tolerance]: 12), CMOS input: 5, CMOS output: 1, N-ch open-drain I/O [6 V tolerance]: 2) • 100-pin products (with USB) Total: 77 (CMOS I/O: 69 (N-ch open drain I/O [V_{DD} tolerance]: 15), CMOS input: 5, CMOS output: 1, N-ch open-drain I/O [6 V tolerance]: 2) • 100-pin products (without USB) Total: 81 (CMOS I/O: 73 (N-ch open drain I/O [V_{DD} tolerance]: 15), CMOS input: 5, CMOS output: 1, N-ch open-drain I/O [6 V tolerance]: 2)

4.2.1 Port 0

Port 0 is an I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to P07 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0 (PU0).

Input to the P00 and P01 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 0 (PIM0).

Output from the P00 to P02 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 0 (POM0).

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

Reset signal generation sets port 0 to the digital input invalid mode Note.

Note “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

4.2.2 Port 1

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

Input to the P10 and 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, external interrupt request input, comparator output, and segment output of LCD controller/driver.

Reset signal generation sets port 1 to the digital input invalid mode Note.

Note “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

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). To use the P20 to P27 pins as input ports, use of an on-chip pull-up resistor can be specified in 1-bit units by setting pull-up resistor option register 2 (PU2).

Input to the P24 and P25 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 2 (PIM2).

Output from the P24 to P26 pin can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 2 (POM2).

To use the P20 and P21 pins as digital I/O ports, set them to digital I/O using port mode control register 2 (PMC2) (Can be specified in 1-bit units).

This port can also be used for A/D converter analog input, timer I/O, serial interface clock I/O, and data I/O, clock/buzzer output, programming UART output, and input, and segment output of LCD controller/driver.

Reset signal generation sets port 2 to the digital input invalid mode Note.

Note “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

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 to P37 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 3 (PU3).

Input to the P33 and P34 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 3 (PIM3).

Output from the P33 to P35 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 3 (POM3).

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

Reset signal generation sets port 3 to the digital input invalid mode Note.

Note “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

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). To use the P40 to P46 pins as input ports, use of an on-chip pull-up resistor can be specified by setting pull-up resistor option register 4 (PU4).

Input to the P43 and P44 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 4 (PIM4).

Output from the P42 to P44 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 4 (POM4).

To use the P41 to P46 pins as digital I/O ports, set them to digital I/O using port mode control register 4 (PMC4) (Can be specified in 1-bit units).

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

Reset signal generation sets port 4 to input mode.

4.2.6 Port 5

Port 5 is an I/O port with an output latch. Port 5 can be set to the input mode or output mode in 1-bit units using port mode register 5 (PM5). When the P50 to P57 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 5 (PU5).

This port can also be used for external interrupt request input, segment output of LCD controller/driver.

Reset signal generation sets port 5 to the digital input invalid mode Note.

Note “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

4.2.7 Port 6

Port 6 is an I/O port with an output latch. Port 6 can be set to the input mode or output mode in 1-bit units using port mode register 6 (PM6).

The output of the P60 and P61 pins is N-ch open-drain output (6 V tolerance).

This port can also be used for serial interface data I/O and clock I/O.

Reset signal generation sets port 6 to input mode.

4.2.8 Port 7

Port 7 is an I/O port with an output latch. Port 7 can be set to the input mode or output mode in 1-bit units using port mode register 7 (PM7). When the P70 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 key interrupt input, timer I/O, and segment output of LCD controller/driver.

Reset signal generation sets port 7 to the digital input invalid mode Note.

Note “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

4.2.9 Port 8

Port 8 is an I/O port with an output latch. Port 8 can be set to the input mode or output mode in 1-bit units using port mode register 8 (PM8). When the P80 to P83 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 8 (PU8).

Reset signal generation sets port 8 to input mode.

4.2.10 Port 12

P125 to P127 are I/O ports with output latches. Port 12 can be set to the input mode or output mode in 1-bit units using port mode register 12 (PM12). When the P125 to P127 pins are used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

P121 to P124 are 4-bit input ports.

This port can also be used for connecting resonator for main system clock, connecting resonator for subsystem clock, external clock input for main system clock, external clock input for subsystem clock, connecting a capacitor for LCD controller/driver, and power supply voltage pin for driving the LCD.

Reset signal generation sets P121 to P124 to input mode, and sets P125 to P127 to digital input invalid Note.

Note “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

4.2.11 Port 13

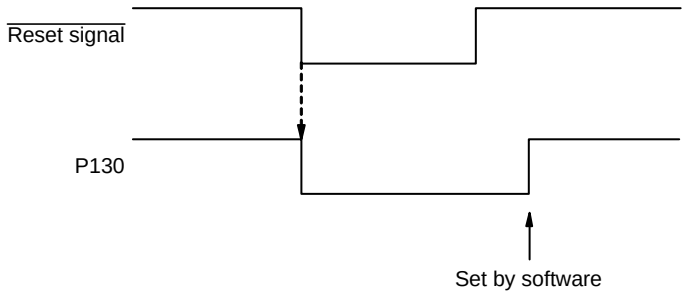
P130 is a 1-bit output-only port with an output latch.

P137 is a 1-bit input-only port.

P130 is fixed an output port, and P137 is fixed an input ports.

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

Remark When reset is effected, P130 outputs a low level. If P130 is set to output a high level before reset is effected, the output signal of P130 can be dummy-output as the CPU reset signal.



4.2.12 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 P140 to P143 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 14 (PU14).

The P140 to P143 pins can be set to digital I/O or analog input using port mode control register 14 (PMC14).

This port can also be used for A/D converter analog input, segment output of LCD controller/driver.

Reset signal generation sets port 14 to the digital input invalid mode Note.

Note “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

4.2.13 Port 15

Port 15 is an I/O port with an output latch. Port 15 can be set to the input mode or output mode in 1-bit units using port mode register 15 (PM15).

This port can also be used for A/D converter analog input and reference voltage (positive side and negative side) input.

To use P150/ANI0 to P156/ANI6 as digital I/O pins, set them to digital I/O using the A/D port configuration register (ADPC). Use these pins starting from the upper bit.

To use P150/ANI0 to P156/ANI6 as digital output pins, set them in the digital I/O mode by using the ADPC register and in the output mode by using the PM15 register.

To use P150/ANI0 to P156/ANI6 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 PM15 register. Use these pins starting from the lower bit.

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 (PIOR)
- LCD port function register (PFSEG0 to PFSEG6)
- LCD input switch control register (ISCLCD)

Caution Which registers and bits are included depends on the product. For registers and bits mounted on each product, see Tables 4 - 2 to 4 - 5. Be sure to set bits that are not mounted to their initial values.

Table 4 - 2 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and the bits mounted on each product (1/4)

Port		Bit Name						Products with USB		Products without USB	
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register	100-pin	80/85-pin	100-pin	80/85-pin
Port 0	0	PM00	P00	PU00	PIM00	POM00	—	√	√	√	√
	1	PM01	P01	PU01	PIM01	POM01	—	√	√	√	√
	2	PM02	P02	PU02	—	POM02	—	√	√	√	√
	3	PM03	P03	PU03	—	—	—	√	√	√	√
	4	PM04	P04	PU04	—	—	—	√	√	√	√
	5	PM05	P05	PU05	—	—	—	√	√	√	√
	6	PM06	P06	PU06	—	—	—	√	√	√	√
	7	PM07	P07	PU07	—	—	—	√	√	√	√
Port 1	0	PM10	P10	PU10	PIM10	POM10	—	√	√	√	√
	1	PM11	P11	PU11	PIM11	POM11	—	√	√	√	√
	2	PM12	P12	PU12	—	POM12	—	√	√	√	√
	3	PM13	P13	PU13	—	—	—	√	—	√	—
	4	PM14	P14	PU14	—	—	—	√	—	√	—
	5	PM15	P15	PU15	—	—	—	√	—	√	—
	6	PM16	P16	PU16	—	—	—	√	—	√	—
	7	PM17	P17	PU17	—	—	—	√	—	√	—
Port 2	0	PM20	P20	PU20	—	—	PMC20	√	√	√	√
	1	PM21	P21	PU21	—	—	PMC21	√	√	√	√
	2	PM22	P22	PU22	—	—	—	√	√	√	√
	3	PM23	P23	PU23	—	—	—	√	√	√	√
	4	PM24	P24	PU24	PIM24	POM24	—	√	√	√	√
	5	PM25	P25	PU25	PIM25	POM25	—	√	√	√	√
	6	PM26	P26	PU26	—	POM26	—	√	√	√	√
	7	PM27	P27	PU27	—	—	—	√	√	√	√

Table 4 - 3 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and the bits mounted on each product (2/4)

Port		Bit Name						Products with USB		Products without USB	
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register	100-pin	80/85-pin	100-pin	80/85-pin
Port 3	0	PM30	P30	PU30	—	—	—	√	√	√	√
	1	PM31	P31	PU31	—	—	—	√	√	√	√
	2	PM32	P32	PU32	—	—	—	√	√	√	√
	3	PM33	P33	PU33	PIM33	POM33	—	√	√	√	√
	4	PM34	P34	PU34	PIM34	POM34	—	√	√	√	√
	5	PM35	P35	PU35	—	POM35	—	√	√	√	√
	6	PM36	P36	PU36	—	—	—	√	—	√	—
	7	PM37	P37	PU37	—	—	—	√	—	√	—
Port 4	0	PM40	P40	PU40	—	—	—	√	√	√	√
	1	PM41	P41	PU41	—	—	PMC41	√	—	√	—
	2	PM42	P42	PU42	—	POM42	PMC42	√	—	√	—
	3	PM43	P43	PU43	PIM43 Note	POM43 Note	PMC43	√	√	√	√
	4	PM44	P44	PU44	PIM44 Note	POM44 Note	PMC44	√	√	√	√
	5	PM45	P45	PU45	—	—	PMC45	√	√	√	√
	6	PM46	P46	PU46	—	—	PMC46	√	√	√	√
	7	—	—	—	—	—	—	—	—	—	—
Port 5	0	PM50	P50	PU50	—	—	—	√	√	√	√
	1	PM51	P51	PU51	—	—	—	√	√	√	√
	2	PM52	P52	PU52	—	—	—	√	√	√	√
	3	PM53	P53	PU53	—	—	—	√	—	√	—
	4	PM54	P54	PU54	—	—	—	√	—	√	—
	5	PM55	P55	PU55	—	—	—	√	—	√	—
	6	PM56	P56	PU56	—	—	—	√	—	√	—
	7	PM57	P57	PU57	—	—	—	√	—	√	—
Port 6	0	PM60	P60	—	—	—	—	√	√	√	√
	1	PM61	P61	—	—	—	—	√	√	√	√
	2	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—

Note 100-pin products only.

Table 4 - 4 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and the bits mounted on each product (3/4)

Port		Bit Name						Products with USB		Products without USB	
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register	100-pin	80/85-pin	100-pin	80/85-pin
Port 7	0	PM70	P70	PU70	—	—	—	√	√	√	√
	1	PM71	P71	PU71	—	—	—	√	√	√	√
	2	PM72	P72	PU72	—	—	—	√	√	√	√
	3	PM73	P73	PU73	—	—	—	√	√	√	√
	4	PM74	P74	PU74	—	—	—	√	√	√	√
	5	PM75	P75	PU75	—	—	—	√	√	√	√
	6	PM76	P76	PU76	—	—	—	√	√	√	√
	7	PM77	P77	PU77	—	—	—	√	√	√	√
Port 8	0	PM80	P80	PU80	—	—	—	—	—	√	—
	1	PM81	P81	PU81	—	—	—	—	—	√	—
	2	PM82	P82	PU82	—	—	—	—	—	√	√
	3	PM83	P83	PU83	—	—	—	—	—	√	√
	4	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—
Port 12	0	—	—	—	—	—	—	—	—	—	—
	1	—	P121	—	—	—	—	√	√	√	√
	2	—	P122	—	—	—	—	√	√	√	√
	3	—	P123	—	—	—	—	√	√	√	√
	4	—	P124	—	—	—	—	√	√	√	√
	5	PM125	P125	PU125	—	—	—	√	√	√	√
	6	PM126	P126	PU126	—	—	—	√	√	√	√
	7	PM127	P127	PU127	—	—	—	√	√	√	√
Port 13	0	—	P130	—	—	—	—	√	√	√	√
	1	—	—	—	—	—	—	—	—	—	—
	2	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—
	7	—	P137	—	—	—	—	√	√	√	√

Table 4 - 5 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and the bits mounted on each product (4/4)

Port		Bit Name						Products with USB		Products without USB	
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register	100-pin	80/85-pin	100-pin	80/85-pin
Port 14	0	PM140	P140	PU140	—	—	PMC140	√	√	√	√
	1	PM141	P141	PU141	—	—	PMC141	√	√	√	√
	2	PM142	P142	PU142	—	—	PMC142	√	√	√	√
	3	PM143	P143	PU143	—	—	PMC143	√	√	√	√
	4	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—
Port 15	0	PM150	P150	—	—	—	—	√	√	√	√
	1	PM151	P151	—	—	—	—	√	√	√	√
	2	PM152	P152	—	—	—	—	√	√	√	√
	3	PM153	P153	—	—	—	—	√	—	√	—
	4	PM154	P154	—	—	—	—	√	—	√	—
	5	PM155	P155	—	—	—	—	√	—	√	√
	6	PM156	P156	—	—	—	—	√	—	√	√
	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	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FFF23H	FFH	R/W
PM4	1	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FFF24H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FFF25H	FFH	R/W
PM6	1	1	1	1	1	1	PM61	PM60	FFF26H	FFH	R/W
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
PM8	1	1	1	1	PM83	PM82	PM81	PM80	FFF28H	FFH	R/W
PM12	PM127	PM126	PM125	1	1	1	1	1	FFF2CH	FFH	R/W
PM14	1	1	1	1	PM143	PM142	PM141	PM140	FFF2EH	FFH	R/W
PM15	1	PM156	PM155	PM154	PM153	PM152	PM151	PM150	FFF2FH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 8, 12, 14, 15; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

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

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 P20, P21, P140 to P143, and P150 to P156 are set up as analog inputs of the A/D converter or P41 to P44 are set up as analog inputs of the comparator, or 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	P07	P06	P05	P04	P03	P02	P01	P00	FFF00H	00H (output latch)	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	FFF01H	00H (output latch)	R/W
P2	P27	P26	P25	P24	P23	P22	P21	P20	FFF02H	00H (output latch)	R/W
P3	P37	P36	P35	P34	P33	P32	P31	P30	FFF03H	00H (output latch)	R/W
P4	0	P46	P45	P44	P43	P42	P41	P40	FFF04H	00H (output latch)	R/W
P5	P57	P56	P55	P54	P53	P52	P51	P50	FFF05H	00H (output latch)	R/W
P6	0	0	0	0	0	0	P61	P60	FFF06H	00H (output latch)	R/W
P7	P77	P76	P75	P74	P73	P72	P71	P70	FFF07H	00H (output latch)	R/W
P8	0	0	0	0	P83	P82	P81	P80	FFF08H	00H (output latch)	R/W
P12	P127	P126	P125	P124	P123	P122	P121	0	FFF0CH	Undefined	R/W Note
P13	P137	0	0	0	0	0	0	P130	FFF0DH	Undefined	R/W Note
P14	0	0	0	0	P143	P142	P141	P140	FFF0EH	00H (output latch)	R/W
P15	0	P156	P155	P154	P153	P152	P151	P150	FFF0FH	00H (output latch)	R/W

Pmn	m = 0 to 8, 12 to 15; n = 0 to 7	
	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

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

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 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 a different potential device to the TTL buffer, pull up to the power supply of the different potential device via an external 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	PU07	PU06	PU05	PU04	PU03	PU02	PU01	PU00	F0030H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU2	PU27	PU26	PU25	PU24	PU23	PU22	PU21	PU20	F0032H	00H	R/W
PU3	PU37	PU36	PU35	PU34	PU33	PU32	PU31	PU30	F0033H	00H	R/W
PU4	0	PU46	PU45	PU44	PU43	PU42	PU41	PU40	F0034H	01H	R/W
PU5	PU57	PU56	PU55	PU54	PU53	PU52	PU51	PU50	F0035H	00H	R/W
PU7	PU77	PU76	PU75	PU74	PU73	PU72	PU71	PU70	F0037H	00H	R/W
PU8	0	0	0	0	PU83	PU82	PU81	PU80	F0038H	00H	R/W
PU12	PU127	PU126	PU125	0	0	0	0	0	F003CH	00H	R/W
PU14	0	0	0	0	PU143	PU142	PU141	PU140	F003EH	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0 to 5, 7, 8, 12, 14; 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.

Port input mode 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	0	0	PIM01	PIM00	F0040H	00H	R/W
PIM1	0	0	0	0	0	0	PIM11	PIM10	F0041H	00H	R/W
PIM2	0	0	PIM25	PIM24	0	0	0	0	F0042H	00H	R/W
PIM3	0	0	0	PIM34	PIM33	0	0	0	F0043H	00H	R/W
PIM4	0	0	0	PIM44	PIM43	0	0	0	F0044H	00H	R/W

PIMmn	Pmn pin input buffer selection (m = 0 to 4; n = 0, 1, 3, 4)
0	Normal input buffer
1	TTL input buffer

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

4.3.5 Port output mode registers (POMxx)

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

N-ch open-drain output (V_{DD} tolerance) mode can be selected during serial communication with an external device of the different potential, and for the SDA00, SDA10, SDA20, and SDA30 pins during simplified I^2C 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.

Port output mode 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	POM01	POM00	F0050H	00H	R/W
POM1	0	0	0	0	0	POM12	POM11	POM10	F0051H	00H	R/W
POM2	0	POM26	POM25	POM24	0	0	0	0	F0052H	00H	R/W
POM3	0	0	POM35	POM34	POM33	0	0	0	F0053H	00H	R/W
POM4	0	0	0	POM44	POM43	POM42	0	0	F0054H	00H	R/W

POMmn	Pmn pin output mode selection (m = 0 to 4; n = 0 to 5)
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.

Port mode control registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to FFH (Only PMC4 is set to 00H).

Figure 4 - 6 Format of Port mode control register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMC2	1	1	1	1	1	1	PMC21	PMC20	F0062H	FFH	R/W
PMC4	0	PMC46	PMC45	PMC44	PMC43	PMC42	PMC41	0	F0064H	00H	R/W
PMC14	1	1	1	1	PMC143	PMC142	PMC141	PMC140	F006EH	FFH	R/W

PMCmn	Pmn pin digital I/O/analog input selection (m = 2, 4, 14; n = 0 to 6)
0	Digital I/O (alternate function other than analog input)
1	Analog input

Caution 1. Select input mode using port mode registers 2 and 14 (PM2, PM14) for the ports which are set by the PMCxx register as analog input.

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

Caution 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/P150 to ANI6/P156 pins to analog input of A/D converter or digital I/O of port.

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

Reset signal generation clears 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	0	ADPC2	ADPC1	ADPC0

ADPC2	ADPC1	ADPC0	Analog input (A)/digital I/O (D) switching						
			ANI6/P156	ANI5/P155	ANI4/P154	ANI3/P153	ANI2/P152	ANI1/P151	ANI0/P150
0	0	0	A	A	A	A	A	A	A
0	0	1	D	D	D	D	D	D	D
0	1	0	D	D	D	D	D	D	A
0	1	1	D	D	D	D	D	A	A
1	0	0	D	D	D	D	A	A	A
1	0	1	D	D	D	A	A	A	A
1	1	0	D	D	A	A	A	A	A
1	1	1	D	A	A	A	A	A	A

Caution 1. Set the channel used for A/D conversion to the input mode by using port mode register 15 (PM15).

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

Caution 3. When using AVREFP and AVREFM, specify ANI0 and ANI1 as the analog input channels and specify input mode by using the port mode register.

4.3.8 Peripheral I/O redirection register (PIOR)

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 PIOR 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 (PIOR)

Address: F0077H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIOR	0	0	0	0	PIOR3	PIOR2	PIOR1	PIOR0

Bit	Alternate Function	100-pin		80/85-pin	
		Setting value		Setting value	
		0	1	0	1
PIOR3	INTP7	P10	P43	P10	P43
	INTP5	P06	P27	P06	P27
PIOR2	PCLBUZ0	P10	P02	P10	P02
PIOR1	TxD1	P02	P42	Cannot be used as the alternate function. Set to 0 (initial value).	
	RxD1	P01	P43		
	SCL10	P00	P44		
	SDA10	P01	P43		
	SI10	P01	P43		
	SO10	P02	P42		
	SCK10	P00	P44		
PIOR0	TO00	P03	P40	P03	P40
	TI00	P03	P40	P03	P40
	TO01	P32	P60	P32	P60
	TI01	P32	P60	P32	P60
	TI02	P05	P61	P05	P61
	TO02	P05	P61	P05	P61
	TI03	P30	P127	P30	P127
	TO03	P30	P127	P30	P127
	TI04	P22	P126	P22	P126
	TO04	P22	P126	P22	P126
	TI05	P42	P27	P27	P27
	TO05	P42	P27	P27	P27
	TI06	P07	P125	P07	P125
	TO06	P07	P125	P07	P125
	TI07	P23	P41	P23	P23
	TO07	P23	P41	P23	P23
	REMOOUT	P30	P127	P30	P127

Remark Cannot be used as the alternate function.

4.3.9 LCD port function registers 0 to 6 (PFSEG0 to PFSEG6)

These registers specify whether to use pins P00 to P07, P10 to P17, P20 to P27, P30 to P37, P50 to P57, P70 to P77, P140 to P143 as port pins (other than segment output pins) or segment output pins.
These registers are set by using a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets these registers to FFH (PFSEG0 is F0H).

Remark The correspondence between the segment output pins (SEGxx) and the PFSEG register (PFSEGxx bits) and the existence of SEGxx pins in each product are shown in Table 4 - 6 Segment Output Pins in Each Product and Correspondence with PFSEG Register (PFSEG Bits).

Figure 4 - 9 Format of LCD Port Function Registers 0 to 6

Address: F0300H After reset: F0H R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG0	PFSEG07	PFSEG06	PFSEG05	PFSEG04	0	0	0	0

Address: F0301H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG1	PFSEG15	PFSEG14	PFSEG13	PFSEG12	PFSEG11	PFSEG10	PFSEG09	PFSEG08

Address: F0302H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG2	PFSEG23	PFSEG22	PFSEG21	PFSEG20	PFSEG19	PFSEG18	PFSEG17	PFSEG16

Address: F0303H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG3	PFSEG31	PFSEG30	PFSEG29	PFSEG28	PFSEG27	PFSEG26	PFSEG25	PFSEG24

Address: F0304H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG4	PFSEG39	PFSEG38	PFSEG37	PFSEG36	PFSEG35	PFSEG34	PFSEG33	PFSEG32

Address: F0305H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG5	PFSEG47	PFSEG46	PFSEG45	PFSEG44	PFSEG43	PFSEG42	PFSEG41	PFSEG40

Address: F0306H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG6	PFSEG55	PFSEG54	PFSEG53	PFSEG52	PFSEG51	PFSEG50	PFSEG49	PFSEG48

PFSEGxx (xx = 04 to 55)	Port (other than segment output)/segment outputs specification of Pmn pins (mn = 00 to 07, 10 to 17, 20 to 27, 30 to 37, 50 to 57, 70 to 77, 140 to 143)
0	Used as port (other than segment output)
1	Used as segment output

Caution To use the Pmn pins as segment output pins (PFSEGxx = 1), be sure to set the PUmn bit of the PUm register, POMmn bit of the POMm register, and PIMmn bit of the PIMm register to "0".

Table 4 - 6 Segment Output Pins in Each Product and Correspondence with PFSEG Register (PFSEG Bits)

Bit name of PFSEG register	Corresponding SEGxx pins	Alternate port	100-pin	80/85-pin
PFSEG04	SEG4	P50	√	√
PFSEG05	SEG5	P51	√	√
PFSEG06	SEG6	P52	√	√
PFSEG07	SEG7	P53	√	—
PFSEG08	SEG8	P54	√	—
PFSEG09	SEG9	P55	√	—
PFSEG10	SEG10	P56	√	—
PFSEG11	SEG11	P57	√	—
PFSEG12	SEG12	P70	√	√
PFSEG13	SEG13	P71	√	√
PFSEG14	SEG14	P72	√	√
PFSEG15	SEG15	P73	√	√
PFSEG16	SEG16	P74	√	√
PFSEG17	SEG17	P75	√	√
PFSEG18	SEG18	P76	√	√
PFSEG19	SEG19	P77	√	√
PFSEG20	SEG20	P30	√	√
PFSEG21	SEG21	P31	√	√
PFSEG22	SEG22	P32	√	√
PFSEG23	SEG23	P33	√	√
PFSEG24	SEG24	P34	√	√
PFSEG25	SEG25	P35	√	√
PFSEG26	SEG26	P36	√	—
PFSEG27	SEG27	P37	√	—
PFSEG28	SEG28	P140	√	√
PFSEG29	SEG29	P141	√	√
PFSEG30	SEG30	P142	√	√
PFSEG31	SEG31	P143	√	√
PFSEG32	SEG32	P20	√	√
PFSEG33	SEG33	P21	√	√
PFSEG34	SEG34	P22	√	√
PFSEG35	SEG35	P23	√	√
PFSEG36	SEG36	P24	√	√
PFSEG37	SEG37	P25	√	√
PFSEG38	SEG38	P26	√	√
PFSEG39	SEG39	P27	√	√
PFSEG40	SEG40	P10	√	√
PFSEG41	SEG41	P11	√	√
PFSEG42	SEG42	P12	√	√
PFSEG43	SEG43	P13	√	—
PFSEG44	SEG44	P14	√	—
PFSEG45	SEG45	P15	√	—
PFSEG46	SEG46	P16	√	—
PFSEG47	SEG47	P17	√	—
PFSEG48	SEG48	P00	√	√
PFSEG49	SEG49	P01	√	√
PFSEG50	SEG50	P02	√	√
PFSEG51	SEG51	P03	√	√
PFSEG52	SEG52	P04	√	√
PFSEG53	SEG53	P05	√	√
PFSEG54	SEG54	P06	√	√
PFSEG55	SEG55	P07	√	√

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

4.3.10 LCD input switch control register (ISCLCD)

The CAPL/P126, CAPH/P127, and V_{L3}/P125 pins are internally connected with a Schmitt trigger buffer.

Input to the Schmitt trigger buffer must be disabled until the CAPL/P126, CAPH/P127, and V_{L3}/P125 pins are set to operate as LCD function pins in order to prevent through-current from entering.

This register is set by using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets ISCLCD to 00H.

Figure 4 - 10 Format of LCD input switch control register (ISCLCD)

Address: F0308H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISCLCD	0	0	0	0	0	0	ISCVL3	ISCCAP

ISCVL3	V_{L3}/P125 pin Schmitt trigger buffer control
0	Makes digital input ineffective
1	Makes digital input effective

ISCCAP	CAPL/P126, CAPH/P127 pins Schmitt trigger buffer control
0	Makes digital input ineffective
1	Makes digital input effective

Caution If ISCVL3 = 0 and ISCCAP = 0, set the corresponding port registers as follows:
PU127 bit of PU12 register = 0, P127 bit of P12 register = 0
PU126 bit of PU12 register = 0, P126 bit of P12 register = 0
PU125 bit of PU12 register = 0, P125 bit of P12 register = 0

4.4 Port Function Operations

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

4.4.1 Writing to I/O port

(1) Output mode

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

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

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

(2) Input mode

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

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

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

4.4.2 Reading from I/O port

(1) Output mode

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

(2) Input mode

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

4.4.3 Operations on I/O port

(1) Output mode

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

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

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

(2) Input mode

The pin level is read and an operation is performed on its contents. The result of the operation is written to the output latch, but since the output buffer is off, the pin status does not change. 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 (1.8 V, 2.5 V) by using I/O buffers

It is possible to connect an external device operating on a different potential (1.8 V or 2.5 V) by switching I/O buffers with the port input mode register (PIMxx) and port output mode register (POMxx). When receiving input from an external device with a different potential (1.8 V or 2.5 V), set port input mode registers 0 to 4 (PIM0 to PIM4) on a bit-by-bit basis to enable normal input (CMOS)/TTL switching.

When outputting data to an external device with a different potential (1.8 V or 2.5 V), set port output mode registers 0 to 4 (POM0 to POM4) on a bit-by-bit basis to enable normal output (CMOS)/N-ch open drain (V_{DD} tolerance) switching.

Connection of a serial interface is described as follows.

(1) Setting procedure when using input ports of UART0 to UART3, CSI00, CSI01, CSI10, CSI20, and CSI30 functions for the TTL input buffer

In case of UART0:	P25
In case of UART1:	P01 (P43)
In case of UART2:	P11
In case of UART3:	P34
In case of CSI00:	P24, P25
In case of CSI10:	P00, P01 (P44, P43)
In case of CSI20:	P10, P11
In case of CSI30:	P33, P34

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

<1> Using an external resistor, pull up externally the pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).

<2> Set the corresponding bit of the PIM0 to PIM4 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 UART/Simplified SPI (CSI[Note]) mode.

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

(2) Setting procedure when using output ports of UART0 to UART3, CSI00, CSI01, CSI10, CSI20, and CSI30 functions in N-ch open-drain output mode

In case of UART0:	P26
In case of UART1:	P02 (P42)
In case of UART2:	P12
In case of UART3:	P35
In case of CSI00:	P24, P26
In case of CSI10:	P00, P02 (P44, P42)
In case of CSI20:	P10, P12
In case of CSI30:	P33, P35

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

<1> Using an external resistor, pull up externally the pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).
<2> After reset release, the port mode is the input mode (Hi-Z).
<3> Set the output latch of the corresponding port to 1.
<4> Set the corresponding bit of the POM0 to POM4 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 UART/Simplified SPI (CSI) mode.
<6> Set the corresponding bit of the PM0 to PM4 registers to output mode.
At this time, the output data is high level, so the pin is in the Hi-Z state.

(3) Setting procedure when using I/O ports of simplified IIC00, IIC10, IIC20, and IIC30 functions with a different potential (1.8 V, 2.5 V)

In case of simplified IIC00: P24, P25
In case of simplified IIC10: P00, P01 (P44, P43)
In case of simplified IIC20: P10, P11
In case of simplified IIC30: P33, P34

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

<1> Using an external resistor, pull up externally the pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).
<2> After reset release, the port mode is the input mode (Hi-Z).
<3> Set the output latch of the corresponding port to 1.
<4> Set the corresponding bit of the POM0 to POM4 registers to 1 to set N-ch open drain output (V_{DD} tolerance) mode.
<5> Set the corresponding bit of the POM0 to POM4 registers to 1 to switch to the TTL input buffer. For V_{IH} and V_{IL}, refer to the DC characteristics when the TTL input buffer is selected.
<6> Enable the operation of the serial array unit and set the mode to simplified I^2C mode.
<7> Set the corresponding bit of the PM0 to PM4 registers to output mode (data I/O is possible in output mode).
At this time, the output data is high level, so the pin is in the Hi-Z state.

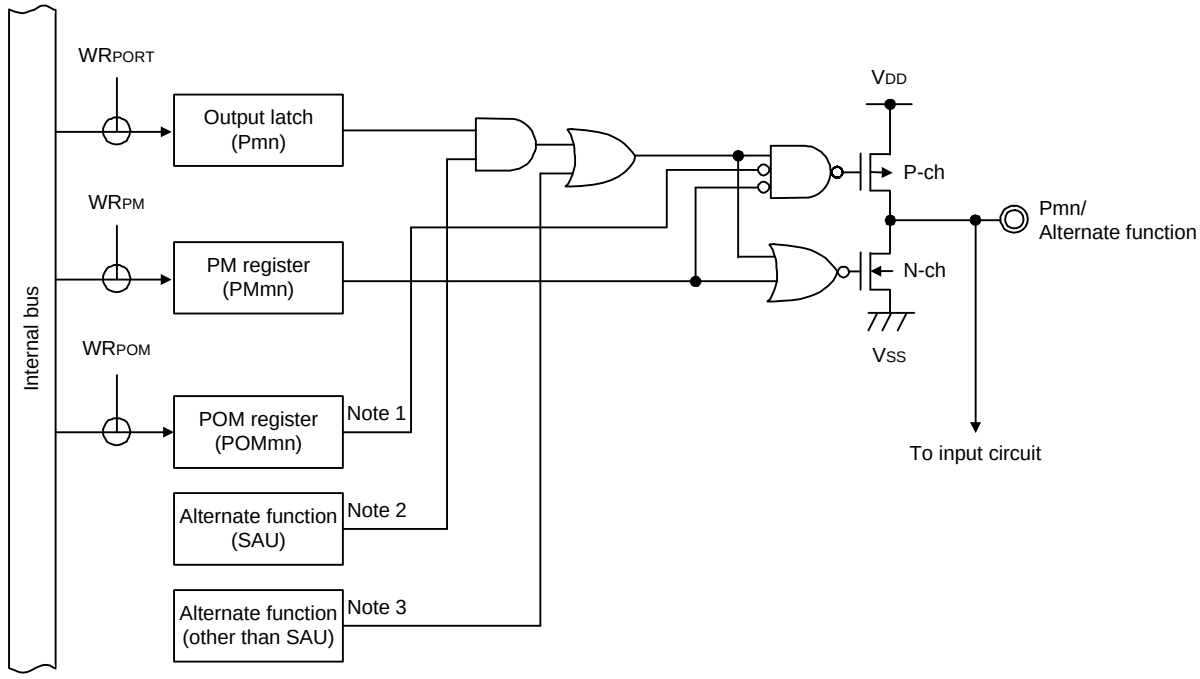
4.5 Register Settings When Using Alternate Function

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 - 11 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, RTC2, clock/buzzer output, 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 - 7.

Figure 4 - 11 Basic Configuration of Output Circuit for Pins



Note 1. When there is no POM register, this signal should be considered to be low level (0).
Note 2. When there is no alternate function, this signal should be considered to be high level (1).
Note 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 - 7 Concept of Basic Settings

Output Function of Used Pin	Output Settings of Unused Alternate Function		
	Output Function for Port	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 (PIOR). 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.

(5) PCLBUZn = 0 (setting when clock/buzzer output is not used)
When the clock/buzzer output is not used, set the PCLOEn bit in clock output select register n (CKSn) to 0 (output disabled). This is the same setting as the initial state.

(6) RTC1HZ = 0 (setting when real-time clock 2 output is not used)
When the real-time clock 2 output is not used, set the RCLOE1 bit in real-time clock control register 0 (RTCC0) to 0 (output disabled). This is the same setting as the initial state.

(7) VCOUTn = 0 (setting when comparator output is not used)
When the comparator output is not used, set the CnOE bit in comparator output control register (COMPOCR) to 0 (output disabled). This is the same setting as the initial state.

(8) TKBOnp = 0 (setting when timer KB2 output is not used)
When the timer KB2 output is not used, set the TKBCEn bit in 16-bit timer KB2 operation control register n1 (TKBCTLn1) to 0 (timer operation stopped). This is the same setting as the initial state.

(9) ANOn = 0 (setting when D/A converter output is not used)
When the D/A converter output is not used, set the DACEn bit in D/A converter mode register (DAM) to 0 (stops D/A conversion operation). 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 Tables 4 - 8 to 4 - 17. The registers used to control the port functions should be set as shown in Tables 4 - 8 to 4 - 17. See the following remark for legends used in Tables 4 - 8 to 4 - 17.

Remark : Not supported
×: don't care
PIORx: Peripheral I/O redirection register
PFSEGXX: LCD port function register
POMxx: Port output mode register
PMCxx: Port mode control register
PMxx: Port mode register
Pxx: Port output latch
Functions in parentheses can be assigned via settings in the peripheral I/O redirection register (PIOR).
For details about ports that also serve as segment output pins (SEGxx), see **4.5.4 Operation of ports that alternately function as SEGxx pins**.
For details about ports that also serve as V_{L3}, CAPL, and CAPH pins, see **4.5.5 Operation of ports that alternately function as V_{L3}, CAPL, and CAPH pins**.

Table 4 - 8 Setting Examples of Registers and Output Latches When Using Pin Function (1/10)

Pin Name	Used Function		PIORXX	PFSEGXX	POMXX	PMCXX	PMXX	PXX	Alternate Function Output		With USB		Without USB	
	Function Name	I/O							SAU Output Function	Other than SAU	80/85-pin	100-pin	80/85-pin	100-pin
P00	P00	Input	—	PFSEG48 = 0	×	—	1	×	×	—	√	√	√	√
		Output	—	PFSEG48 = 0	0	—	0	0/1	SCK10/ SCL10 = 1	—	√	√	√	√
		N-ch OD output	—	PFSEG48 = 0	1	—	0	0/1			√	√	√	√
	SCK10	Input	PIOR1 = 0 Note	PFSEG48 = 0	×	—	1	×	×	—	√	√	√	√
		Output	PIOR1 = 0 Note	PFSEG48 = 0	0/1	—	0	1	×	—	√	√	√	√
	SCL10	Output	PIOR1 = 0 Note	PFSEG48 = 0	0/1	—	0	1	×	—	√	√	√	√
	SEG48	Output	—	PFSEG48 = 1	0	—	0	0	×	—	√	√	√	√
P01	P01	Input	—	PFSEG49 = 0	×	—	1	×	×	—	√	√	√	√
		Output	—	PFSEG49 = 0	0	—	0	0/1	SDA10 = 1	—	√	√	√	√
		N-ch OD output	—	PFSEG49 = 0	1	—	0	0/1			√	√	√	√
	SI10	Input	PIOR1 = 0 Note	PFSEG49 = 0	×	—	1	×	×	—	√	√	√	√
	RxD1	Input	PIOR1 = 0 Note	PFSEG49 = 0	×	—	1	×	×	—	√	√	√	√
	SDA10	I/O	PIOR1 = 0 Note	PFSEG49 = 0	1	—	0	1	×	—	√	√	√	√
	SEG49	Output	—	PFSEG49 = 1	0	—	0	0	×	—	√	√	√	√
P02	P02	Input	—	PFSEG50 = 0	×	—	1	×	×	×	√	√	√	√
		Output	—	PFSEG50 = 0	0	—	0	0/1	TxD1/ SO10 = 1	(PCLBUZ0) = 0	√	√	√	√
		N-ch OD output	—	PFSEG50 = 0	1	—	0	0/1			√	√	√	√
	SO10	Output	PIOR1 = 0 Note	PFSEG50 = 0	0/1	—	0	1	×	(PCLBUZ0) = 0	√	√	√	√
	TxD1	Output	PIOR1 = 0 Note	PFSEG50 = 0	0/1	—	0	1	×	(PCLBUZ0) = 0	√	√	√	√
	(PCLBUZ0)	Output	PIOR2 = 1	PFSEG50 = 0	0	—	0	0	TxD1/ SO10 = 1	×	√	√	√	√
	SEG50	Output	—	PFSEG50 = 1	0	—	0	0	×	×	√	√	√	√
P03	P03	Input	—	PFSEG51= 0	—	—	1	×	—	×	√	√	√	√
		Output	—	PFSEG51= 0	—	—	0	0/1	—	TO00 = 0	√	√	√	√
	TI00	Input	PIOR0 = 0	PFSEG51 = 0	—	—	1	×	—	×	√	√	√	√
	TO00	Output	PIOR0 = 0	PFSEG51 = 0	—	—	0	0	—	×	√	√	√	√
	INTP1	Input	—	PFSEG51 = 0	—	—	1	×	—	×	√	√	√	√
	SEG51	Output	—	PFSEG51 = 1	—	—	0	0	—	×	√	√	√	√
P04	P04	Input	—	PFSEG52= 0	—	—	1	×	—	—	√	√	√	√
		Output	—	PFSEG52= 0	—	—	0	0/1	—	—	√	√	√	√
	INTP2	Input	—	PFSEG52 = 0	—	—	1	×	—	—	√	√	√	√
	SEG52	Output	—	PFSEG52 = 1	—	—	0	0	—	—	√	√	√	√
P05	P05	Input	—	PFSEG53= 0	—	—	1	×	—	×	√	√	√	√
		Output	—	PFSEG53= 0	—	—	0	0/1	—	TO02 = 0	√	√	√	√
	TI02	Input	PIOR0 = 0	PFSEG53 = 0	—	—	1	×	—	×	√	√	√	√
	TO02	Output	PIOR0 = 0	PFSEG53 = 0	—	—	0	0	—	×	√	√	√	√
	SEG53	Output	—	PFSEG53 = 1	—	—	0	0	—	×	√	√	√	√
P06	P06	Input	—	PFSEG54= 0	—	—	1	×	—	—	√	√	√	√
		Output	—	PFSEG54= 0	—	—	0	0/1	—	—	√	√	√	√
	INTP5	Input	PIOR3=0	PFSEG54 = 0	—	—	1	×	—	—	√	√	√	√
	SEG54	Output	—	PFSEG54 = 1	—	—	0	0	—	—	√	√	√	√
P07	P07	Input	—	PFSEG55= 0	—	—	1	×	—	×	√	√	√	√
		Output	—	PFSEG55= 0	—	—	0	0/1	—	TO06 = 0	√	√	√	√
	TI06	Input	PIOR0 = 0	PFSEG55 = 0	—	—	1	×	—	×	√	√	√	√
	TO06	Output	PIOR0 = 0	PFSEG55 = 0	—	—	0	0	—	×	√	√	√	√
	SEG55	Output	—	PFSEG55 = 1	—	—	0	0	—	×	√	√	√	√

Note 100-pin products only.

Table 4 - 9 Setting Examples of Registers and Output Latches When Using Pin Function (2/10)

Pin Name	Used Function		PIORXX	PFSEGXX	POMXX	PMCXX	PMXX	PXX	Alternate Function Output		With USB		Without USB	
	Function Name	I/O							SAU Output Function	Other than SAU	80/85-pin	100-pin	80/85-pin	100-pin
P10	P10	Input	—	PFSEG40 = 0	×	—	1	×	×	×	√	√	√	√
		Output	—	PFSEG40 = 0	0	—	0	0/1	SCK20/ SCL20 = 1	PCLBUZ0 = 0	√	√	√	√
		N-ch OD output	—	PFSEG40 = 0	1	—	0	0/1			√	√	√	√
	INTP7	Input	PIOR3=0	PFSEG40 = 0	×	—	1	×	×	×	√	√	√	√
	PCLBUZ0	Output	PIOR2=0	PFSEG40 = 0	0	—	0	0	SCK20/ SCL20 = 1	×	√	√	√	√
	SCK20	Input	—	PFSEG40 = 0	×	—	1	×	×	×	√	√	√	√
		Output	—	PFSEG40 = 0	0/1	—	0	1	×	PCLBUZ0 = 0	√	√	√	√
	SCL20	Output	—	PFSEG40 = 0	0/1	—	0	1	×	PCLBUZ0 = 0	√	√	√	√
	SEG40	Output	—	PFSEG40 = 1	0	—	0	0	×	×	√	√	√	√
P11	P11	Input	—	PFSEG41= 0	×	—	1	×	×	×	√	√	√	√
		Output	—	PFSEG41 = 0	0	—	0	0/1	SDA20 = 1	VCOUT0 = 0	√	√	√	√
		N-ch OD output	—	PFSEG41 = 0	1	—	0	0/1			√	√	√	√
	RxD2	Input	—	PFSEG41 = 0	×	—	1	×	×	×	√	√	√	√
	SI20	Input	—	PFSEG41 = 0	×	—	1	×	×	×	√	√	√	√
	SDA20	I/O	—	PFSEG41 = 0	1	—	0	1	×	VCOUT0 = 0	√	√	√	√
	SEG41	Output	—	PFSEG41 = 1	0	—	0	0	×	×	√	√	√	√
	VCOUT0	Output	—	PFSEG41 = 0	0	—	0	0	SDA20 = 1	×	√	√	√	√
P12	P12	Input	—	PFSEG42 = 0	×	—	1	×	×	×	√	√	√	√
		Output	—	PFSEG42 = 0	0	—	0	0/1	TxD2/ SO20 = 1	VCOUT1 = 0	√	√	√	√
		N-ch OD output	—	PFSEG42 = 0	1	—	0	0/1			√	√	√	√
	TxD2	Output	—	PFSEG42 = 0	0/1	—	0	1	×	VCOUT1 = 0	√	√	√	√
	SO20	Output	—	PFSEG42 = 0	0/1	—	0	1	×	VCOUT1 = 0	√	√	√	√
	SEG42	Output	—	PFSEG42 = 1	0	—	0	0	×	×	√	√	√	√
	VCOUT1	Output	—	PFSEG42 = 0	0	—	0	0	TxD2/ SO20 = 1	×	√	√	√	√
P13	P13	Input	—	PFSEG43 = 0	—	—	1	×	—	—	×	√	×	√
		Output	—	PFSEG43 = 0	—	—	0	0/1	—	—	×	√	×	√
	SEG43	Output	—	PFSEG43 = 1	—	—	0	0	—	—	×	√	×	√
P14	P14	Input	—	PFSEG44 = 0	—	—	1	×	—	—	×	√	×	√
		Output	—	PFSEG44 = 0	—	—	0	0/1	—	—	×	√	×	√
	SEG44	Output	—	PFSEG44 = 1	—	—	0	0	—	—	×	√	×	√
P15	P15	Input	—	PFSEG45 = 0	—	—	1	×	—	—	×	√	×	√
		Output	—	PFSEG45 = 0	—	—	0	0/1	—	—	×	√	×	√
	SEG45	Output	—	PFSEG45 = 1	—	—	0	0	—	—	×	√	×	√
P16	P16	Input	—	PFSEG46 = 0	—	—	1	×	—	—	×	√	×	√
		Output	—	PFSEG46 = 0	—	—	0	0/1	—	—	×	√	×	√
	SEG46	Output	—	PFSEG46 = 1	—	—	0	0	—	—	×	√	×	√
P17	P17	Input	—	PFSEG47 = 0	—	—	1	×	—	—	×	√	×	√
		Output	—	PFSEG47 = 0	—	—	0	0/1	—	—	×	√	×	√
	SEG47	Output	—	PFSEG47 = 1	—	—	0	0	—	—	×	√	×	√
P20	P20	Input	—	PFSEG32 = 0	—	0	1	×	—	—	√	√	√	√
		Output	—	PFSEG32 = 0	—	0	0	0/1	—	—	√	√	√	√
	ANI20	Input	—	PFSEG32 = 0	—	1	1	×	—	—	√	√	√	√
	SEG32	Output	—	PFSEG32 = 1	—	0	0	0	—	—	√	√	√	√
P21	P21	Input	—	PFSEG33 = 0	—	0	1	×	—	—	√	√	√	√
		Output	—	PFSEG33 = 0	—	0	0	0/1	—	—	√	√	√	√
	ANI21	Input	—	PFSEG33 = 0	—	1	1	×	—	—	√	√	√	√
	SEG33	Output	—	PFSEG33 = 1	—	0	0	0	—	—	√	√	√	√

Table 4 - 10 Setting Examples of Registers and Output Latches When Using Pin Function (3/10)

Pin Name	Used Function		PIORXX	PFSEGXX	POMXX	PMCXX	PMXX	PXX	Alternate Function Output		With USB		Without USB	
	Function Name	I/O							SAU Output Function	Other than SAU	80/85-pin	100-pin	80/85-pin	100-pin
P22	P22	Input	—	PFSEG34= 0	—	—	1	×	—	×	√	√	√	√
		Output	—	PFSEG34= 0	—	—	0	0/1	—	TO04 = 0	√	√	√	√
	TI04	Input	PIOR0 = 0	PFSEG34 = 0	—	—	1	×	—	×	√	√	√	√
	TO04	Output	PIOR0 = 0	PFSEG34 = 0	—	—	0	0	—	×	√	√	√	√
	SEG34	Output	—	PFSEG34 = 1	—	—	0	0	—	×	√	√	√	√
P23	P23	Input	—	PFSEG35= 0	—	—	1	×	—	×	√	√	√	√
		Output	—	PFSEG35= 0	—	—	0	0/1	—	TO07 = 0	√	√	√	√
	TI07	Input	PIOR0 = 0	PFSEG35 = 0	—	—	1	×	—	×	√	√	√	√
	TO07	Output	PIOR0 = 0	PFSEG35 = 0	—	—	0	0	—	×	√	√	√	√
	SEG35	Output	—	PFSEG35 = 1	—	—	0	0	—	×	√	√	√	√
P24	P24	Input	—	PFSEG36 = 0	×	—	1	×	×	×	√	√	√	√
		Output	—	PFSEG36 = 0	0	—	0	0/1	SCK00/ SCL00 = 1	—	√	√	√	√
		N-ch OD output	—	PFSEG36 = 0	1	—	0	0/1			√	√	√	√
	SCK00	Input	—	PFSEG36 = 0	×	—	1	×	×	—	√	√	√	√
		Output	—	PFSEG36 = 0	0/1	—	0	1	×	—	√	√	√	√
	SCL00	Output	—	PFSEG36 = 0	0/1	—	0	1	×	—	√	√	√	√
	SEG36	Output	—	PFSEG36 = 1	0	—	0	0	×	—	√	√	√	√
P25	P25	Input	—	PFSEG37 = 0	×	—	1	×	×	—	√	√	√	√
		Output	—	PFSEG37 = 0	0	—	0	0/1	SDA00 = 1	—	√	√	√	√
		N-ch OD output	—	PFSEG37 = 0	1	—	0	0/1			√	√	√	√
	SI00	Input	—	PFSEG37 = 0	×	—	1	×	×	—	√	√	√	√
	RxD0	Input	—	PFSEG37 = 0	×	—	1	×	×	—	√	√	√	√
	TOOLRxD	Input	—	PFSEG37 = 0	×	—	1	×	×	—	√	√	√	√
	SDA00	I/O	—	PFSEG37 = 0	1	—	0	1	×	—	√	√	√	√
	SEG37	Output	—	PFSEG37 = 1	0	—	0	0	×	—	√	√	√	√
P26	P26	Input	—	PFSEG38 = 0	×	—	1	×	×	×	√	√	√	√
		Output	—	PFSEG38 = 0	0	—	0	0/1	TxD0/ SO00 = 1	TOOLTxD = 0	√	√	√	√
		N-ch OD output	—	PFSEG38 = 0	1	—	0	0/1			√	√	√	√
	SO00	Output	—	PFSEG38 = 0	0/1	—	0	1	×	TOOLTxD = 0	√	√	√	√
	TxD0	Output	—	PFSEG38 = 0	0/1	—	0	1	×	TOOLTxD = 0	√	√	√	√
	TOOLTxD	Output	—	PFSEG38 = 0	0/1	—	0	1	TxD0/ SO00 = 1	×	√	√	√	√
	SEG38	Output	—	PFSEG38 = 1	0	—	0	0	×	×	√	√	√	√
P27	P27	Input	—	PFSEG39 = 0	—	—	1	×	—	×	√	√	√	√
		Output	—	PFSEG39 = 0	—	—	0	0/1	—	(TO05) = 0 PCLBUZ1 = 0	√	√	√	√
	(TI05)	Input	PIOR0 = 1	PFSEG39 = 0	—	—	1	×	—	×	√	√	√	√
	(TO05)	Output	PIOR0 = 1	PFSEG39 = 0	—	—	0	0	—	PCLBUZ1 = 0	√	√	√	√
	(INTP5)	Input	PIOR3 = 1	PFSEG39 = 0	—	—	1	×	—	×	√	√	√	√
	PCLBUZ1	Output	—	PFSEG39 = 0	—	—	0	0	—	(TO05) = 0	√	√	√	√
	SEG39	Output	—	PFSEG39 = 1	—	—	0	0	—	×	√	√	√	√
P30	P30	Input	—	PFSEG20 = 0	—	—	1	×	—	×	√	√	√	√
		Output	—	PFSEG20 = 0	—	—	0	0/1	—	TO03 = 0	√	√	√	√
	TI03	Input	PIOR0 = 0	PFSEG20 = 0	—	—	1	×	—	×	√	√	√	√
	TO03	Output	PIOR0 = 0	PFSEG20 = 0	—	—	0	0	—	×	√	√	√	√
	REMOOUT	Output	PIOR0 = 0	PFSEG20 = 0	—	—	0	0	—	×	√	√	√	√
	SEG20	Output	—	PFSEG20 = 1	—	—	0	0	—	×	√	√	√	√

Table 4 - 11 Setting Examples of Registers and Output Latches When Using Pin Function (4/10)

Pin Name	Used Function		PIORXX	PFSEGXX	POMXX	PMCXX	PMXX	PXX	Alternate Function Output		With USB		Without USB	
	Function Name	I/O							SAU Output Function	Other than SAU	80/85-pin	100-pin	80/85-pin	100-pin
P31	P31	Input	—	PFSEG21 = 0	—	—	1	×	—	×	√	√	√	√
		Output	—	PFSEG21 = 0	—	—	0	0/1	—	RTC1HZ = 0	√	√	√	√
	INTP3	Input	—	PFSEG21 = 0	—	—	1	×	—	×	√	√	√	√
	RTC1HZ	Output	—	PFSEG21 = 0	—	—	0	0	—	×	√	√	√	√
	SEG21	Output	—	PFSEG21 = 1	—	—	0	0	—	×	√	√	√	√
P32	P32	Input	—	PFSEG22 = 0	—	—	1	×	—	×	√	√	√	√
		Output	—	PFSEG22 = 0	—	—	0	0/1	—	TO01 = 0	√	√	√	√
	TI01	Input	PIOR0 = 0	PFSEG22 = 0	—	—	1	×	—	×	√	√	√	√
	TO01	Output	PIOR0 = 0	PFSEG22 = 0	—	—	0	0	—	×	√	√	√	√
	SEG22	Output	—	PFSEG22 = 1	—	—	0	0	—	×	√	√	√	√
P33	P33	Input	—	PFSEG23 = 0	×	—	1	×	×	—	√	√	√	√
		Output	—	PFSEG23 = 0	0	—	0	0/1	SCK30/ SCL30 = 1	—	√	√	√	√
		N-ch OD output	—	PFSEG23 = 0	1	—	0	0/1			√	√	√	√
	INTP4	Input	—	PFSEG23 = 0	×	—	1	×	×	—	√	√	√	√
	SCK30	Input	—	PFSEG23 = 0	×	—	1	×	×	—	√	√	√	√
		Output	—	PFSEG23 = 0	0/1	—	0	1	×	—	√	√	√	√
	SCL30	Output	—	PFSEG23 = 0	0/1	—	0	1	×	—	√	√	√	√
	SEG23	Output	—	PFSEG23 = 1	×	—	0	0	×	—	√	√	√	√
P34	P34	Input	—	PFSEG24 = 0	×	—	1	×	×	—	√	√	√	√
		Output	—	PFSEG24 = 0	0	—	0	0/1	SDA30 = 1	—	√	√	√	√
		N-ch OD output	—	PFSEG24 = 0	1	—	0	0/1			√	√	√	√
	SI30	Input	—	PFSEG24 = 0	×	—	1	×	×	—	√	√	√	√
	RxD3	Input	—	PFSEG24 = 0	×	—	1	×	×	—	√	√	√	√
	SDA30	I/O	—	PFSEG24 = 0	1	—	0	1	×	—	√	√	√	√
	SEG24	Output	—	PFSEG24 = 1	0	—	0	0	×	—	√	√	√	√
P35	P35	Input	—	PFSEG25 = 0	×	—	1	×	×	—	√	√	√	√
		Output	—	PFSEG25 = 0	0	—	0	0/1	TxD3/ SO30 = 1	—	√	√	√	√
		N-ch OD output	—	PFSEG25 = 0	1	—	0	0/1			√	√	√	√
	SO30	Output	—	PFSEG25 = 0	0/1	—	0	1	×	—	√	√	√	√
	TxD3	Output	—	PFSEG25 = 0	0/1	—	0	1	×	—	√	√	√	√
	SEG25	Output	—	PFSEG25 = 1	0	—	0	0	×	—	√	√	√	√
P36	P36	Input	—	PFSEG26 = 0	×	—	1	×	—	—	×	√	×	√
		Output	—	PFSEG26 = 0	0	—	0	0/1	—	—	×	√	×	√
	SEG26	Output	—	PFSEG26 = 1	—	—	0	0	—	—	×	√	×	√
P37	P37	Input	—	PFSEG27 = 0	×	—	1	×	—	—	×	√	×	√
		Output	—	PFSEG27 = 0	0	—	0	0/1	—	—	×	√	×	√
	SEG27	Output	—	PFSEG27 = 1	—	—	0	0	—	—	×	√	×	√
P40	P40	Input	—	—	×	—	1	×	—	×	√	√	√	√
		Output	—	—	0	—	0	0/1	—	(TO00) = 0	√	√	√	√
	TOOL0	I/O	—	—	—	—	×	×	—	(TO00) = 0	√	√	√	√
	(TI00)	Input	PIOR0 = 1	—	—	—	1	×	—	×	√	√	√	√
	(TO00)	Output	PIOR0 = 1	—	—	—	0	0	—	×	√	√	√	√
P41	P41	Input	—	—	×	0	1	×	—	×	×	√	×	√
		Output	—	—	0	0	0	0/1	—	(TO07) = 0	×	√	×	√
	(TI07)	Input	PIOR0 = 1 Note	—	—	0	1	×	—	×	×	√	×	√
	(TO07)	Output	PIOR0 = 1 Note	—	—	0	0	0	—	×	×	√	×	√
	IVREF1	Input	—	—	—	1	1	×	—	×	×	√	×	√

Note 100-pin products only.

Table 4 - 12 Setting Examples of Registers and Output Latches When Using Pin Function (5/10)

Pin Name	Used Function		PIORXX	PFSEGXX	POMXX	PMCXX	PMXX	PXX	Alternate Function Output		With USB		Without USB	
	Function Name	I/O							SAU Output Function	Other than SAU	80/85-pin	100-pin	80/85-pin	100-pin
P42	P42	Input	—	—	×	0	1	×	×	×	×	√	×	√
		Output	—	—	0	0	0	0/1	(TxD1)/ (SO10) = 1	TO05 = 0	×	√	×	√
		N-ch OD output	—	—	1	0	0	0/1			×	√	×	√
	TI05	Input	PIOR0 = 0 Note	—	×	0	1	×	×	×	×	√	×	√
	TO05	Output	PIOR0 = 0 Note	—	0	0	0	0	(TxD1)/ (SO10) = 1	×	×	√	×	√
	(SO10)	Output	PIOR1 = 1 Note	—	0/1	0	0	1	×	TO05 = 0	×	√	×	√
	(TxD1)	Output	PIOR1 = 1 Note	—	0/1	0	0	1	×	TO05 = 0	×	√	×	√
	IVCMP1	Input	—	—	×	1	1	×	×	×	×	√	×	√
P43	P43	Input	—	—	×	0	1	×	×	—	√	√	√	√
		Output	—	—	0	0	0	0/1	(SDA10) = 1	—	√	√	√	√
		N-ch OD output	—	—	1	0	0	0/1			×	√	×	√
	(INTP7)	Input	PIOR3 = 1	—	×	0	1	×	×	—	√	√	√	√
	(SI10)	Input	PIOR1 = 1 Note	—	×	0	1	×	×	—	×	√	×	√
	(RxD1)	Input	PIOR1 = 1 Note	—	×	0	1	×	×	—	×	√	×	√
	(SDA10)	I/O	PIOR1 = 1 Note	—	1	0	0	1	×	—	×	√	×	√
	IVCMP0	Input	—	—	×	1	1	×	×	—	√	√	√	√
P44	P44	Input	—	—	×	0	1	×	×	—	√	√	√	√
		Output	—	—	0	0	0	0/1	(SCK10)/ (SCL10) = 1	—	√	√	√	√
		N-ch OD output	—	—	1	0	0	0/1			×	√	×	√
	(SCK10)	Input	PIOR1 = 1 Note	—	×	0	1	×	×	—	×	√	×	√
		Output	PIOR1 = 1 Note	—	0/1	0	0	1	×	—	×	√	×	√
	(SCL10)	Output	PIOR1 = 1 Note	—	0/1	0	0	1	×	—	×	√	×	√
	IVREF0	Input	—	—	×	1	1	×	×	—	√	√	√	√
P45	P45	Input	—	—	—	0	1	×	—	—	√	√	√	√
		Output	—	—	—	0	0	0/1	—	—	√	√	√	√
	ANO0	Analog output	—	—	—	1	1	×	—	—	√	√	√	√
P46	P46	Input	—	—	—	0	1	×	—	—	√	√	√	√
		Output	—	—	—	0	0	0/1	—	—	√	√	√	√
	ANO1	Analog output	—	—	—	1	1	×	—	—	√	√	√	√
P50	P50	Input	—	PFSEG04 = 0	—	—	1	×	—	—	√	√	√	√
		Output	—	PFSEG04 = 0	—	—	0	0/1	—	—	√	√	√	√
	SEG4	Output	—	PFSEG04 = 1	—	—	0	0	—	—	√	√	√	√
	INTP6	Input	—	PFSEG04 = 0	—	—	1	×	—	—	√	√	√	√
P51	P51	Input	—	PFSEG05 = 0	—	—	1	×	—	—	√	√	√	√
		Output	—	PFSEG05 = 0	—	—	0	0/1	—	—	√	√	√	√
	SEG5	Output	—	PFSEG05 = 1	—	—	0	0	—	—	√	√	√	√
P52	P52	Input	—	PFSEG06 = 0	—	—	1	×	—	—	√	√	√	√
		Output	—	PFSEG06 = 0	—	—	0	0/1	—	—	√	√	√	√
	SEG6	Output	—	PFSEG06 = 1	—	—	0	0	—	—	√	√	√	√
P53	P53	Input	—	PFSEG07 = 0	—	—	1	×	—	—	×	√	×	√
		Output	—	PFSEG07 = 0	—	—	0	0/1	—	—	×	√	×	√
	SEG7	Output	—	PFSEG07 = 1	—	—	0	0	—	—	×	√	×	√
P54	P54	Input	—	PFSEG08 = 0	—	—	1	×	—	—	×	√	×	√
		Output	—	PFSEG08 = 0	—	—	0	0/1	—	—	×	√	×	√
	SEG8	Output	—	PFSEG08 = 1	—	—	0	0	—	—	×	√	×	√

Note 100-pin products only.

Table 4 - 13 Setting Examples of Registers and Output Latches When Using Pin Function (6/10)

Pin Name	Used Function		PIORXX	PFSEGXX	POMXX	PMCXX	PMXX	PXX	Alternate Function Output		With USB		Without USB	
	Function Name	I/O							SAU Output Function	Other than SAU	80/85-pin	100-pin	80/85-pin	100-pin
P55	P55	Input	—	PFSEG09 = 0	—	—	1	×	—	—	×	√	×	√
		Output	—	PFSEG09 = 0	—	—	0	0/1	—	—	×	√	×	√
	SEG9	Output	—	PFSEG09 = 1	—	—	0	0	—	—	×	√	×	√
P56	P56	Input	—	PFSEG10 = 0	—	—	1	×	—	—	×	√	×	√
		Output	—	PFSEG10 = 0	—	—	0	0/1	—	—	×	√	×	√
	SEG10	Output	—	PFSEG10 = 1	—	—	0	0	—	—	×	√	×	√
P57	P57	Input	—	PFSEG11 = 0	—	—	1	×	—	—	×	√	×	√
		Output	—	PFSEG11 = 0	—	—	0	0/1	—	—	×	√	×	√
	SEG11	Output	—	PFSEG11 = 1	—	—	0	0	—	—	×	√	×	√
P60	P60	Input	—	—	—	—	1	×	—	×	√	√	√	√
		N-ch OD output (6 V tolerance)	—	—	—	—	0	0/1	—	SCLA0 = 0 (TO01) = 0	√	√	√	√
	SCLA0	I/O	—	—	—	—	0	0	—	(TO01) = 0	√	√	√	√
	(TI01)	Input	PIOR0 = 1	—	—	—	1	×	—	×	√	√	√	√
	(TO01)	Output	PIOR0 = 1	—	—	—	0	0	—	SCLA0 = 0	√	√	√	√
P61	P61	Input	—	—	—	—	1	×	—	×	√	√	√	√
		N-ch OD output (6 V tolerance)	—	—	—	—	0	0/1	—	SDAA0 = 0 (TO02) = 0	√	√	√	√
	SDAA0	I/O	×	—	—	—	0	0	—	(TO02) = 0	√	√	√	√
	(TI02)	Input	PIOR0 = 1	—	—	—	1	×	—	×	√	√	√	√
	(TO02)	Output	PIOR0 = 1	—	—	—	0	0	—	SDAA0 = 0	√	√	√	√
P70	P70	Input	—	PFSEG12 = 0	—	—	1	×	—	—	√	√	√	√
		Output	—	PFSEG12 = 0	—	—	0	0/1	—	—	√	√	√	√
	KR7	Input	×	PFSEG12 = 0	—	—	1	×	—	—	√	√	√	√
	SEG12	Output	×	PFSEG12 = 1	—	—	0	0	—	—	√	√	√	√
P71	P71	Input	—	PFSEG13 = 0	—	—	1	×	—	—	√	√	√	√
		Output	—	PFSEG13 = 0	—	—	0	0/1	—	—	√	√	√	√
	KR6	Input	×	PFSEG13 = 0	—	—	1	×	—	—	√	√	√	√
	SEG13	Output	×	PFSEG13 = 1	—	—	0	0	—	—	√	√	√	√
P72	P72	Input	—	PFSEG14 = 0	—	—	1	×	—	×	√	√	√	√
		Output	—	PFSEG14 = 0	—	—	0	0/1	—	TKBO20 = 0	√	√	√	√
	KR5	Input	×	PFSEG14 = 0	—	—	1	×	—	×	√	√	√	√
	TKBO20	Output	×	PFSEG14 = 0	—	—	0	0	—	×	√	√	√	√
	SEG14	Output	×	PFSEG14 = 1	—	—	0	0	—	×	√	√	√	√
P73	P73	Input	—	PFSEG15 = 0	—	—	1	×	—	×	√	√	√	√
		Output	—	PFSEG15 = 0	—	—	0	0/1	—	TKBO21 = 0	√	√	√	√
	KR4	Input	×	PFSEG15 = 0	—	—	1	×	—	×	√	√	√	√
	TKBO21	Output	×	PFSEG15 = 0	—	—	0	0	—	×	√	√	√	√
	SEG15	Output	×	PFSEG15 = 1	—	—	0	0	—	×	√	√	√	√
P74	P74	Input	—	PFSEG16 = 0	—	—	1	×	—	×	√	√	√	√
		Output	—	PFSEG16 = 0	—	—	0	0/1	—	TKBO10 = 0	√	√	√	√
	KR3	Input	×	PFSEG16 = 0	—	—	1	×	—	×	√	√	√	√
	TKBO10	Output	×	PFSEG16 = 0	—	—	0	0	—	×	√	√	√	√
	SEG16	Output	×	PFSEG16 = 1	—	—	0	0	—	×	√	√	√	√
P75	P75	Input	—	PFSEG17 = 0	—	—	1	×	—	×	√	√	√	√
		Output	—	PFSEG17 = 0	—	—	0	0/1	—	TKBO11 = 0	√	√	√	√
	KR2	Input	×	PFSEG17 = 0	—	—	1	×	—	×	√	√	√	√
	TKBO11	Output	×	PFSEG17 = 0	—	—	0	0	—	×	√	√	√	√
	SEG17	Output	×	PFSEG17 = 1	—	—	0	0	—	×	√	√	√	√

Table 4 - 14 Setting Examples of Registers and Output Latches When Using Pin Function (7/10)

Pin Name	Used Function		PIORXX	PFSEGXX	POMXX	PMCXX	PMXX	PXX	Alternate Function Output		With USB		Without USB	
	Function Name	I/O							SAU Output Function	Other than SAU	80/85-pin	100-pin	80/85-pin	100-pin
P76	P76	Input	—	PFSEG18 = 0	—	—	1	×	—	×	√	√	√	√
		Output	—	PFSEG18 = 0	—	—	0	0/1	—	TKBO00 = 0	√	√	√	√
	KR1	Input	×	PFSEG18 = 0	—	—	1	×	—	×	√	√	√	√
	TKBO00	Output	×	PFSEG18 = 0	—	—	0	0	—	×	√	√	√	√
	SEG18	Output	×	PFSEG18 = 1	—	—	0	0	—	×	√	√	√	√
P77	P77	Input	—	PFSEG19 = 0	—	—	1	×	—	×	√	√	√	√
		Output	—	PFSEG19 = 0	—	—	0	0/1	—	TKBO01 = 0	√	√	√	√
	KR0	Input	×	PFSEG19 = 0	—	—	1	×	—	×	√	√	√	√
	TKBO01	Output	×	PFSEG19 = 0	—	—	0	0	—	×	√	√	√	√
	SEG19	Output	×	PFSEG19 = 1	—	—	0	0	—	×	√	√	√	√
P80	P80	Input	—	—	—	—	1	×	—	—	×	×	×	√
		Output	—	—	—	—	0	0/1	—	—	×	×	×	√
P81	P81	Input	—	—	—	—	1	×	—	—	×	×	×	√
		Output	—	—	—	—	0	0/1	—	—	×	×	×	√
P82	P82	Input	—	—	—	—	1	×	—	—	×	×	√	√
		Output	—	—	—	—	0	0/1	—	—	×	×	√	√
P83	P83	Input	—	—	—	—	1	×	—	—	×	×	√	√
		Output	—	—	—	—	0	0/1	—	—	×	×	√	√

Table 4 - 15 Setting Examples of Registers and Output Latches When Using Pin Function (8/10)

Pin Name	Used Function		CMC (EXCLK,OSCSEL, EXCLKS, OSCSELS)	Pxx	With USB		Without USB	
	Function Name	I/O			80/85-pin	100-pin	80/85-pin	100-pin
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	P123	Input	xx 00/xx 10	×	√	√	√	√
	XT2	—	xx 01	—	√	√	√	√
	EXCLKS	Input	xx 11	—	√	√	√	√

Table 4 - 16 Setting Examples of Registers and Output Latches When Using Pin Function (9/10)

Pin Name	Used Function		PIORXX	PFSEGXX	POMXX	PMCXX	PMXX	PXX	ISCLCD	Alternate Function Output		With USB		Without USB	
	Function Name	I/O								SAU Output Function	Other than SAU	80/85-pin	100-pin	80/85-pin	100-pin
P125	P125	Input	—	—	—	—	1	×	ISCVL3 = 1	—	×	√	√	√	√
		Output	—	—	—	—	0	0/1	ISCVL3 = 1	—	(TO06) = 0	√	√	√	√
	V_{L3}	I/O	×	—	—	—	1	0	ISCVL3 = 0	—	×	√	√	√	√
	(TI06)	Input	PIOR0 = 1	—	—	—	1	×	ISCVL3 = 1	—	×	√	√	√	√
	(TO06)	Output	PIOR0 = 1	—	—	—	0	0	ISCVL3 = 1	—	×	√	√	√	√
P126	P126	Input	—	—	—	—	1	×	ISCCAP = 1	—	×	√	√	√	√
		Output	—	—	—	—	0	0/1	ISCCAP = 1	—	(TO04) = 0	√	√	√	√
	CAPL	Output	×	—	—	—	1	0	ISCCAP = 0	—	×	√	√	√	√
	(TI04)	Input	PIOR0 = 1	—	—	—	1	×	ISCCAP = 1	—	×	√	√	√	√
	(TO04)	Output	PIOR0 = 1	—	—	—	0	0	ISCCAP = 1	—	×	√	√	√	√
P127	P127	Input	—	—	—	—	1	×	ISCCAP = 1	—	×	√	√	√	√
		Output	—	—	—	—	0	0/1	ISCCAP = 1	—	(TO03) = 0	√	√	√	√
	CAPH	Output	×	—	—	—	1	0	ISCCAP = 0	—	×	√	√	√	√
	(TI03)	Input	PIOR0 = 1	—	—	—	1	×	ISCCAP = 1	—	×	√	√	√	√
	(TO03)	Output	PIOR0 = 1	—	—	—	0	0	ISCCAP = 1	—	×	√	√	√	√
	(REMOOUT)	Output	PIOR0 = 1	—	—	—	0	0	ISCCAP = 1	—	×	√	√	√	√
P130	P130	Output	—	—	—	—	—	0/1	—	—	—	√	√	√	√
P137	P137	Input	—	—	—	—	—	×	—	—	—	√	√	√	√
	INTP0	Input	×	—	—	—	—	×	—	—	—	√	√	√	√
P140	P140	Input	—	PFSEG28 = 0	—	0	1	×	—	—	—	√	√	√	√
		Output	—	PFSEG28 = 0	—	0	0	0/1	—	—	—	√	√	√	√
	ANI16	Input	×	PFSEG28 = 0	—	1	1	×	—	—	—	√	√	√	√
	SEG28	Output	×	PFSEG28 = 1	—	0	0	0	—	—	—	√	√	√	√
P141	P141	Input	—	PFSEG29 = 0	—	0	1	×	—	—	—	√	√	√	√
		Output	—	PFSEG29 = 0	—	0	0	0/1	—	—	—	√	√	√	√
	ANI17	Input	×	PFSEG29 = 0	—	1	1	×	—	—	—	√	√	√	√
	SEG29	Output	×	PFSEG29 = 1	—	0	0	0	—	—	—	√	√	√	√
P142	P142	Input	—	PFSEG30 = 0	—	0	1	×	—	—	—	√	√	√	√
		Output	—	PFSEG30 = 0	—	0	0	0/1	—	—	—	√	√	√	√
	ANI18	Input	×	PFSEG30 = 0	—	1	1	×	—	—	—	√	√	√	√
	SEG30	Output	×	PFSEG30 = 1	—	0	0	0	—	—	—	√	√	√	√
P143	P143	Input	—	PFSEG31 = 0	—	0	1	×	—	—	—	√	√	√	√
		Output	—	PFSEG31 = 0	—	0	0	0/1	—	—	—	√	√	√	√
	ANI19	Input	×	PFSEG31 = 0	—	1	1	×	—	—	—	√	√	√	√
	SEG31	Output	×	PFSEG31 = 1	—	0	0	0	—	—	—	√	√	√	√

Table 4 - 17 Setting Examples of Registers and Output Latches When Using Pin Function (10/10)

Pin Name	Used Function		ADPC	ADM2	PMxx	Pxx	With USB		Without USB	
	Function Name	I/O					80/85-pin	100-pin	80/85-pin	100-pin
P150	P150	Input	ADPC = 01H	×	1	×	√	√	√	√
		Output	ADPC = 01H	×	0	0/1				
	ANI0	Analog input	ADPC = 00H/02H to 07H	00x0xx0x, 10x0xx0x	1	×	√	√	√	√
	AVREFP	Reference voltage	ADPC = 00H/02H to 07H	01x0xx0x	1	×	√	√	√	√
P151	P151	Input	ADPC = 01H/02H	×	1	×	√	√	√	√
		Output	ADPC = 01H/02H	×	0	0/1				
	ANI1	Analog input	ADPC = 00H/03H to 07H	xx00xx0x	1	×	√	√	√	√
	AVREFM	Reference voltage	ADPC = 00H/03H to 07H	xx10xx0x	1	×	√	√	√	√
P152	P152	Input	ADPC = 01H to 03H	×	1	×	√	√	√	√
		Output	ADPC = 01H to 03H	×	0	0/1				
	ANI2	Analog input	ADPC = 00H/04H to 07H	×	1	×	√	√	√	√
P153	P153	Input	ADPC = 01H to 04H	×	1	×	×	√	×	√
		Output	ADPC = 01H to 04H	×	0	0/1				
	ANI3	Analog input	ADPC = 00H/05H to 07H	×	1	×	×	√	×	√
P154	P154	Input	ADPC = 01H to 05H	×	1	×	×	√	×	√
		Output	ADPC = 01H to 05H	×	0	0/1				
	ANI4	Analog input	ADPC = 00H/06H to 07H	×	1	×	×	√	×	√
P155	P155	Input	ADPC = 01H to 06H	×	1	×	×	√	√	√
		Output	ADPC = 01H to 06H	×	0	0/1				
	ANI5	Analog input	ADPC = 00H/07H	×	1	×	×	√	√	√
P156	P156	Input	ADPC = 01H to 07H	×	1	×	×	√	√	√
		Output	ADPC = 01H to 07H	×	0	0/1				
	ANI6	Analog input	ADPC = 00H	×	1	×	×	√	√	√

4.5.4 Operation of ports that alternately function as SEGxx pins

The functions of ports that also serve as segment output pins (SEGxx) can be selected by using the port mode control register (PMCxx), port mode register (PMxx), and LCD port function registers 0 to 6 (PFSEG0 to PFSEG6).

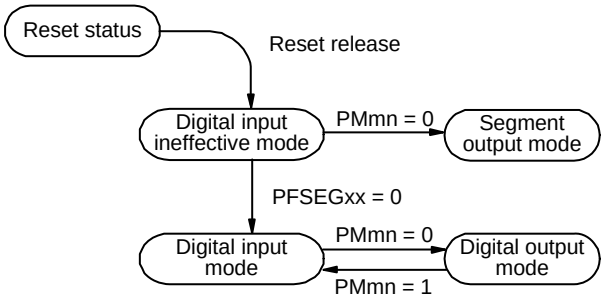
- P00 to P07, P10 to P17, P22 to P27, P30 to P37, P50 to P57, P70 to P77
  (ports that do not serve as analog input pins (ANIxx))

Table 4 - 18 Settings of SEGxx/Port Pin Function

PFSEGxx Bit of PFSEG0 to PFSEG6 Registers	PMxx Bit of PMxx Register	Pin Function	Initial Status
1	1	Digital input ineffective mode	√
0	0	Digital output mode	—
0	1	Digital input mode	—
1	0	Segment output mode	—

The following shows the SEGxx/port pin function status transitions.

Figure 4 - 12 SEGxx/Port Pin Function Status Transitions



Caution Be sure to set the segment output mode before segment output starts (while SCOC bit of LCD mode register 1 (LCDM1) is 0).

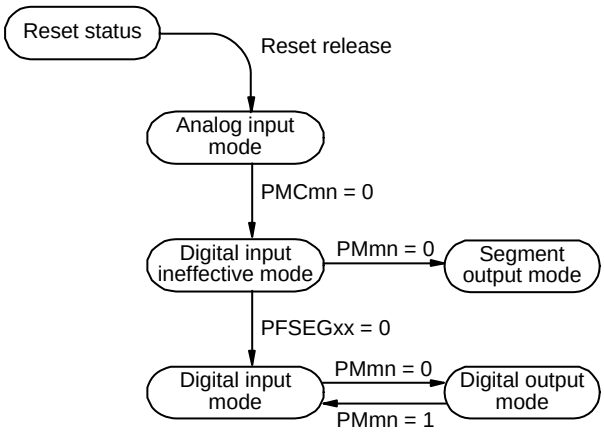
• P20, P21, P140 to P143 (ports that serve as analog input pins (ANIxx))

Table 4 - 19 Settings of ANIxx/SEGxx/Port Pin Function

PMCxx Bit of PMCxx Register	PFSEGxx Bit of PFSEG2 and PFSEG3 Registers	PMxx Bit of PMxx Register	Pin Function	Initial Status
1	1	1	Analog input mode	√
0	0	0	Digital output mode	—
0	0	1	Digital input mode	—
0	1	0	Segment output mode	—
0	1	1	Digital input ineffective mode	—
Other than above			Setting prohibited	

The following shows the ANIxx/SEGxx/port pin function status transitions.

Figure 4 - 13 ANIxx/SEGxx/Port Pin Function Status Transitions



Caution Be sure to set the segment output mode before segment output starts (while SCOC bit of LCD mode register 1 (LCDM1) is 0).

4.5.5 Operation of ports that alternately function as V_{L3}, CAPL, and CAPH pins

The functions of the V_{L3}/P125, CAPL/P126, and CAPH/P127 pins can be selected by using the LCD input switch control register (ISCLCD), LCD mode register 0 (LCDM0), and port mode register 12 (PM12).

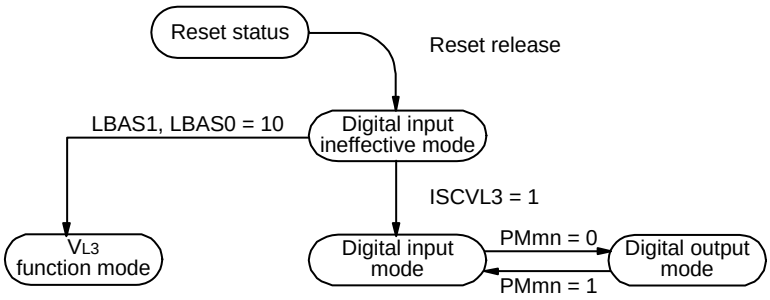
• V_{L3}/P125

Table 4 - 20 Settings of V_{L3}/P125 Pin Function

Bias Setting (LBAS1 and LBAS0 Bits of LCDM0 Register)	ISCVL3 Bit of ISCLCD Register	PM125 Bit of PM12 Register	Pin Function	Initial Status
Other than 1/4 bias method (LBAS1, LBAS0 = 00 or 01)	0	1	Digital input ineffective mode	√
	1	0	Digital output mode	—
	1	1	Digital input mode	—
1/4 bias method (LBAS1, LBAS0 = 10)	0	1	V_{L3} function mode	—
Other than above			Setting prohibited	

The following shows the V_{L3}/P125 pin function status transitions.

Figure 4 - 14 V_{L3}/P125 Pin Function Status Transitions



Caution Be sure to set the V_{L3} function mode before segment output starts (while SCOC bit of LCD mode register 1 (LCDM1) is 0).

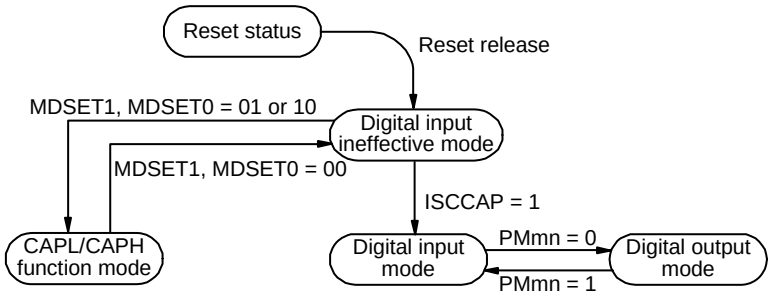
- **CAPL/P126, CAPH/P127**

Table 4 - 21 Settings of CAPL/P126 and CAPH/P127 Pin Functions

LCD Drive Voltage Generator (MDSET1 and MDSET0 Bits of LCDM0 Register)	ISCCAP Bit of ISCLCD Register	PM126 and PM127 Bits of PM12 Register	Pin Function	Initial Status
External resistance division (MDSET1, MDSET0 = 00)	0	1	Digital input ineffective mode	√
	1	0	Digital output mode	—
	1	1	Digital input mode	—
Internal voltage boosting or capacitor split (MDSET1, MDSET0 = 01 or 10)	0	1	CAPL/CAPH function mode	—
Other than above			Setting prohibited	

The following shows the CAPL/P126 and CAPH/P127 pin function status transitions.

Figure 4 - 15 CAPL/P126 and CAPH/P127 Pin Function Status Transitions



Caution Be sure to set the CAPL/CAPH function mode before segment output starts (while SCOC bit of LCD mode register 1 (LCDM1) is 0).

4.6 Cautions When Using Port Function

4.6.1 Cautions on 1-Bit Manipulation Instruction for Port Register n (Pn)

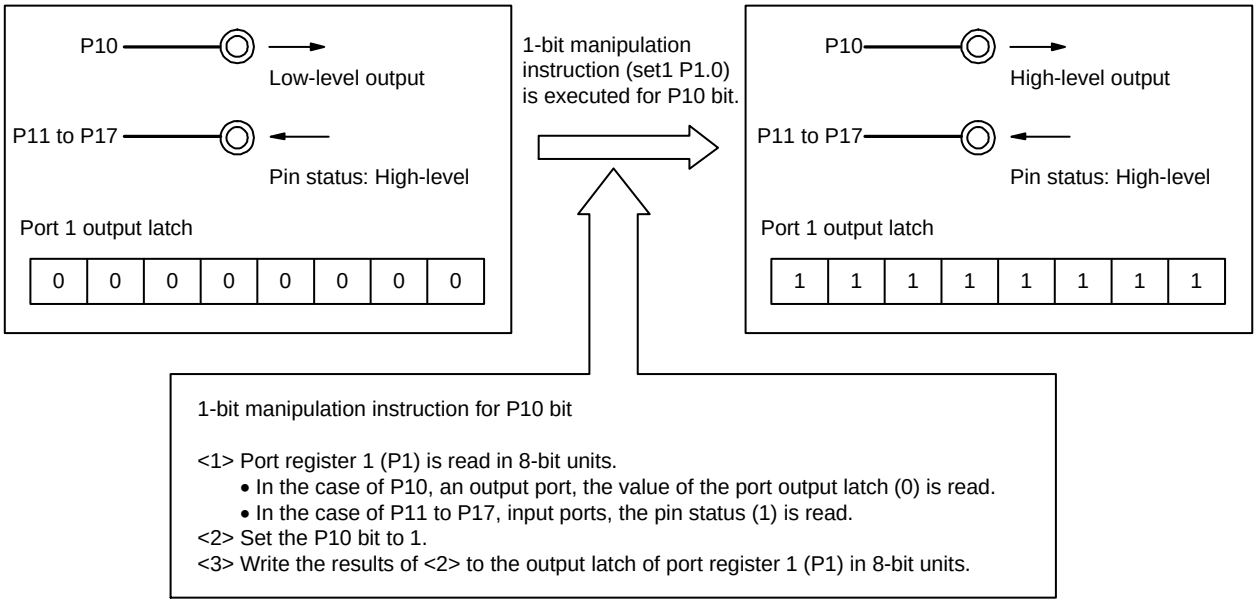
When a 1-bit manipulation instruction is executed on a port that provides both input and output functions, the output latch value of an input port that is not subject to manipulation may be written in addition to the targeted bit.
Therefore, it is recommended to rewrite the output latch when switching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

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

Explanation: The targets of writing to and reading from the Pn register of a port whose PMnm bit is 1 are the output latch and pin status, respectively.
A 1-bit manipulation instruction is executed in the following order in the RL78/L1C.

<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.
In step <1>, the output latch value (0) of P10, which is an output port, is read, while the pin statuses of P11 to P17, which are input ports, are read. If the pin statuses of P11 to P17 are high level at this time, the read value is FEH.
The value is changed to FFH by the manipulation in <2>.
FFH is written to the output latch by the manipulation in <3>.

Figure 4 - 16 Bit Manipulation Instruction (P10)



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 (PIOR). For details about the alternate output function, 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).

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

CHAPTER 5 CLOCK GENERATOR

5.1 Functions of Clock Generator

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

(1) Main system clock

<1> X1 oscillator

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

<2> High-speed on-chip oscillator

The frequency at which to oscillate can be selected from among f_{HOCO} = 48, 24, 16, 12, 8, 6, 4, 3, 2, or 1 MHz (TYP.) by using the option byte (000C2H).
When 48 MHz is selected as f_{HOCO}, f_{IH} is set to 24 MHz. When 24 MHz or less is selected as f_{HOCO}, f_{IH} is not divided and set to the same frequency as f_{HOCO}. After a reset release, the CPU always starts operating with this high-speed on-chip oscillator clock Note 1. Oscillation can be stopped by executing the STOP instruction or setting of the HIOSTOP bit (bit 0 of the CSC register).
The frequency specified by using the option byte can be changed by using the high-speed on-chip oscillator frequency select register (HOCODIV). For details about the frequency, see **Figure 5 - 13 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.
When using the high-speed on-chip oscillator clock f_{HOCO} Note 2 as the USB/function controller clock, use the high-speed on-chip oscillator frequency select register (HOCODIV) to set the oscillation frequency to 48 MHz.

Power Supply Voltage	Oscillation Frequency (MHz)										
	1	2	3	4	6	8	12	16	24	32	48
2.7 V ≤ V_{DD} ≤ 3.6 V	√	√	√	√	√	√	√	√	√	—	√
2.4 V ≤ V_{DD} ≤ 3.6 V	√	√	√	√	√	√	√	√	—	—	—
1.8 V ≤ V_{DD} ≤ 3.6 V	√	√	√	√	√	√	—	—	—	—	—
1.6 V ≤ V_{DD} ≤ 3.6 V	√	√	√	√	—	—	—	—	—	—	—

Note 1. When 48 MHz is selected (FRQSEL4 = 1 in the option byte (000C2H)), the 48-MHz clock (f_{HOCO}) is supplied to 16-bit timers KB20, KB21, and KB22 or the USB function controller. The f_{HOCO} divided by 2, 4, or 8 is supplied to the other functions including the CPU.

Note 2. When selecting 48 MHz for the high-speed on-chip oscillator clock f_{HOCO} (FRQSEL4 = 1 in the option byte (000C2H)) to use it for the USB/function controller clock or 16-bit timers KB20, KB21, and KB22, be sure to set f_{CLK} to f_{IH}.

<3> High-speed system clock multiplication function using PLL (phase locked loop) (products with USB)
This clock function is mainly used for clock supply to the USB function controller. Set the DSCCTL register so that the PLL oscillation frequency (f_{PLL}) is 48 MHz. When the CKSELR bit is set to 1, a clock generated by dividing the f_{PLL} frequency by 2, 4, or 8 by setting the RDIV0 and RDIV1 bits is selected as the main system clock source (f_{IH}) Note.
The PLL can be operated or stopped by setting the DSCON bit (bit 0 in the DSCCTL register).
For details on PLL settings and the connection with the USB clock, see **Figure 5 - 16** and **Table 5 - 3**.

Note When using PLL (f_{PLL}) for the USB/function controller clock or 16-bit timers KB20, KB21, and KB22, be sure to set f_{CLK} to f_{IH}.

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)). However, note that the usable frequency range of the main system clock differs depending on the setting of the power supply voltage (V_{DD}). The operating voltage of the flash memory must be set by using the CMODE0 and CMODE1 bits of the option byte (000C2H) (see **CHAPTER 29 OPTION BYTE**).

(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 XT1 pin and XT2 pin.
Oscillation can be stopped by setting the XTSTOP bit (bit 6 of the clock operation status control register (CSC)).

An external subsystem clock (f_{EXT} = 32.768 kHz) can also be supplied from the EXCLKS/XT2/P124 pin. An external subsystem clock input can be disabled by the setting of the XTSTOP bit.

(3) Low-speed on-chip oscillator
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 2
• 12-bit interval timer
• LCD controller/driver

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 2 count clock when the fixed-cycle interrupt function is used.

Remark f_X: X1 clock oscillation frequency
f_{HOCO}: High-speed on-chip oscillator clock frequency (48 MHz max.)
f_{IH}: Main system clock source frequency when the high-speed on-chip oscillator clock divided 1, 2, 4, or 8, or the PLL clock divided by 2, 4, or 8 is selected (24 MHz max.)
f_{EX}: External main system clock frequency
f_{XT}: XT1 clock oscillation frequency
f_{EXT}: External subsystem clock frequency
f_{IL}: Low-speed on-chip oscillator frequency

5.2 Configuration of Clock Generator

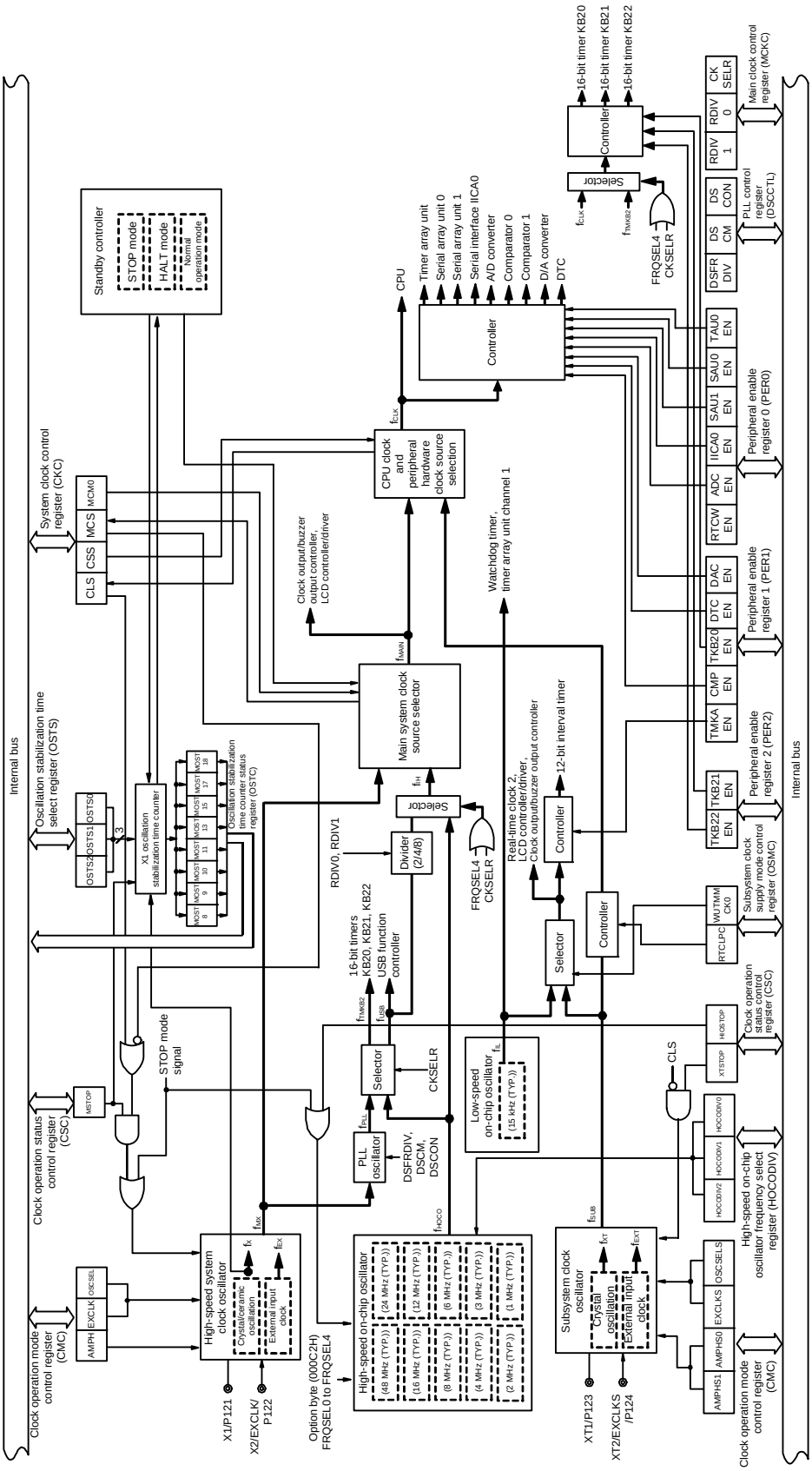
The clock generator includes the following hardware.

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) Peripheral enable registers 0, 1, 2 (PER0, PER1, 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) PLL control register (DSCCTL) Note Main clock control register (MCKC) Note USB clock selection register (UCKSEL) Note
Oscillators	X1 oscillator XT1 oscillator High-speed on-chip oscillator Low-speed on-chip oscillator PLL oscillator Note

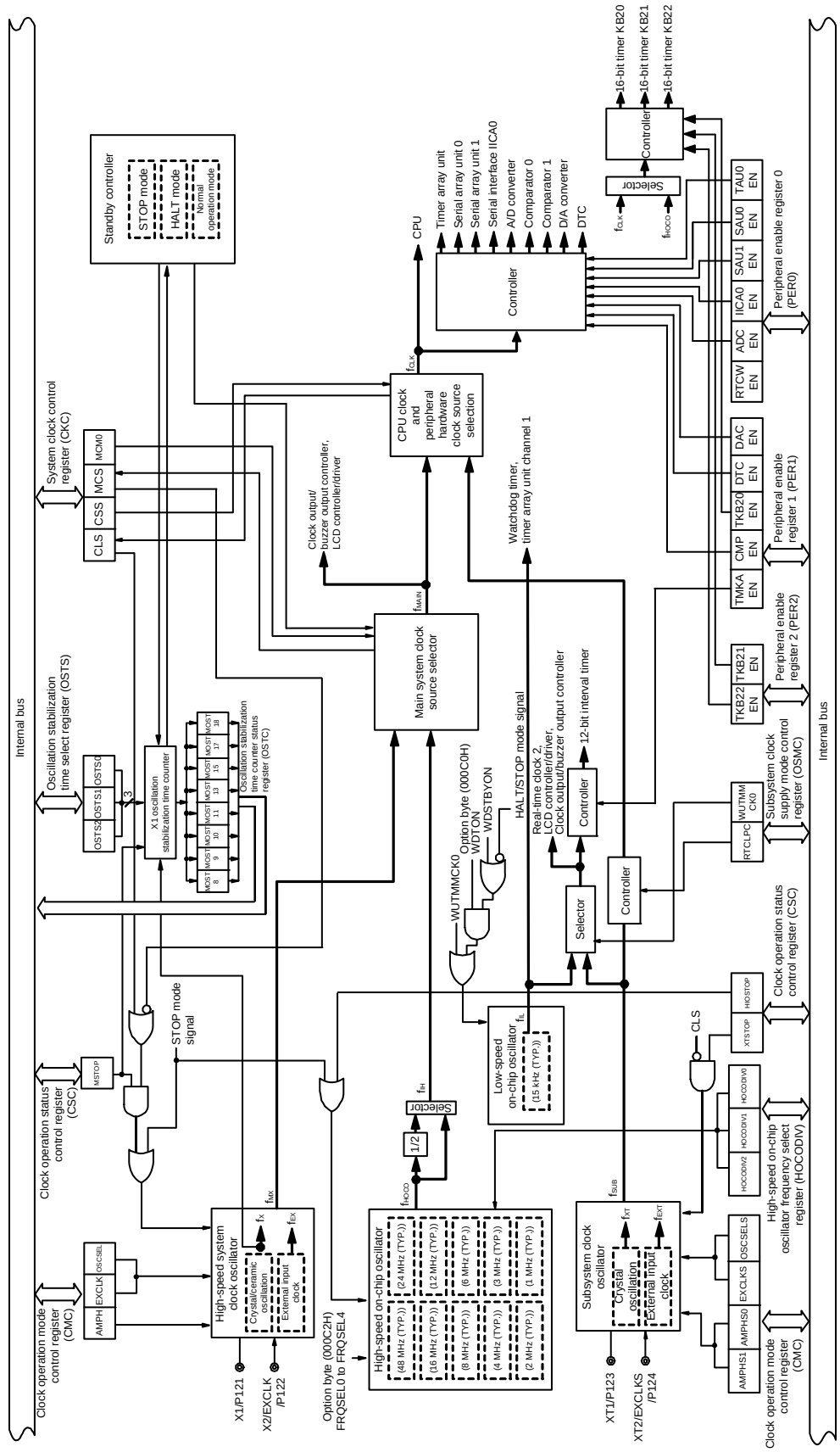
Note Products with USB

Figure 5 - 1 Block Diagram of Clock Generator (Products with USB)



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

Figure 5 - 2 Block Diagram of Clock Generator (Products without USB)



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

Remark f_X: X1 clock oscillation frequency
f_{HOCO}: High-speed on-chip oscillator clock frequency (48 MHz max.)
f_{IH}: Main system clock source frequency when the high-speed on-chip oscillator clock divided 1, 2, 4, or 8, or the PLL clock divided by 2, 4, or 8 is selected (24 MHz max.)
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_{EXT}: 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 clock frequency
f_{USB}: USB clock frequency

5.3 Registers Controlling Clock Generator

The following registers are used to control the clock generator.

- Clock operation mode control register (CMC)
- System clock control register (CKC)
- Clock operation status control register (CSC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- Peripheral enable registers 0, 1, 2 (PER0, PER1, 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)
- PLL control register (DSCCTL) Note
- Main clock control register (MCKC) Note
- USB clock selection register (UCKSEL) Note

Note Products with USB

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.

Caution The EXCLKS, OSCSELS, AMPHS1, and AMPHS0 bits are reset only by a power-on reset; they retain the previous values when a reset caused by another factor occurs.

Figure 5 - 3 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 Note	OSCSELS Note	0	AMPHS1 Note	AMPHS0 Note	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 Note	OSCSELS Note	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/ceramic resonator connection	
1	0	Input port mode	Input port	
1	1	External clock input mode	Input port	External clock input

AMPHS1 Note	AMPHS0 Note	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 ≤ fx ≤ 10 MHz
1	10 MHz < fx ≤ 20 MHz

Note The EXCLKS, OSCSELS, AMPHS1, and AMPHS0 bits are reset only by a power-on reset; they retain the values when a reset caused by another factor occurs.

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

Caution 1. The CMC register can be written only once after a reset ends, 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. A malfunction caused by mistakenly writing a value other than 00H is unrecoverable.

Caution 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).

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

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

Caution 5. Count the f_{XT} oscillation stabilization time by using software.

Caution 6. Although the maximum system clock frequency is 24 MHz, the maximum frequency of the X1 oscillator is 20 MHz.

Caution 7. If a reset other than a power-on reset occurs after the CMC register is written and then the reset ends, be sure to set the CMC register to the value specified before the reset occurred, to prevent a malfunction if a program loop occurs.

Caution 8. 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 parasitic capacitance and 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 - 4 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 Note 2	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 Note 2	Subsystem clock (f_{SUB})

MCS	Status of Main system clock (f_{MAIN})
0	Main system clock source frequency when the high-speed on-chip oscillator clock divided 1, 2, 4, or 8, or the PLL clock divided by 2, 4, or 8 is selected (24 MHz max.) (f_{IH}) Note 3
1	High-speed system clock (f_{MX})

MCM0 Note 2	Main system clock (f_{MAIN}) operation control
0	Selects the main system clock source frequency when the high-speed on-chip oscillator clock divided 1, 2, 4, or 8, or the PLL clock divided by 2, 4, or 8 is selected (24 MHz max.) (f_{IH}) Note 3 as the main system clock (f_{MAIN})
1	Selects the high-speed system clock (f_{MX}) as the main system clock (f_{MAIN})

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

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

Note 3. The PLL clock (f_{PLL}) can be selected in products with USB.

Remark f_{HOCO}: High-speed on-chip oscillator clock frequency (48 MHz max.)
f_{IH}: Main system clock source frequency when the high-speed on-chip oscillator clock divided 1, 2, 4, or 8, or the PLL clock divided by 2, 4, or 8 is selected (24 MHz max.)
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency
f_{SUB}: Subsystem clock frequency
f_{PLL}: PLL clock frequency

(**Cautions** are listed on the next page.)

Caution 1. Be sure to set bits 0 to 3 to 0.

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

Caution 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 34 or CHAPTER 35 ELECTRICAL SPECIFICATIONS.

Caution 4. When selecting f_{HOCO} as the count source clock for 16-bit timer KB20, KB21, KB22, select f_{IH} as f_{CLK} before setting bit 4 (TKB20EN) of peripheral enable register 1 (PER1), and bits 0 and 1 (TKB21EN, TKB22EN) of peripheral enable register 2 (PER2). When changing f_{CLK} to a clock other than f_{IH}, first clear bit 4 (TKB20EN) of peripheral enable register 1 (PER1), and bits 0 and 1 (TKB21EN, TKB22EN) of peripheral enable register 2 (PER2).

Caution 5. When using the high-speed on-chip oscillator clock f_{HOCO} set to 48 MHz (FRQSEL4 = 1 in the option byte (000C2H)) or using the PLL clock (48 MHz) for the USB/function controller or 16-bit timers KB20, KB21, and KB22, be sure to set CSS to 0.

Caution 6. When using the high-speed on-chip oscillator clock f_{HOCO} set to 48 MHz (FRQSEL4 = 1 in the option byte (000C2H)) or using the PLL clock (48 MHz) for the USB/function controller or 16-bit timers KB20, KB21, and KB22, be sure to set MCM0 to 0.

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.

Caution The XTSTOP bit is reset only by a power-on reset; it retains the value when a reset caused by another factor occurs.

Figure 5 - 5 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 Note	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 Note	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

Note The XTSTOP bit is reset only by a power-on reset; it retains the value when a reset caused by another factor occurs.

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

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

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

Caution 4. When starting XT1 oscillation by setting the XSTOP bit, wait for oscillation of the subsystem clock to stabilize by setting a wait time using software.

Caution 5. Do not stop the clock selected for the CPU peripheral hardware clock (f_{CLK}) with the OSC register.

Caution 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 the Clock

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

Remark The oscillation stabilization time counter starts counting in the following cases.
- When oscillation of the X1 clock starts (EXCLK, OSCSEL = 0, 1 → MSTOP = 0)
- When the STOP mode is released

Figure 5 - 6 Format of Oscillation stabilization time counter status register (OSTC)

Address: FFFA2H After reset: 00H R

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

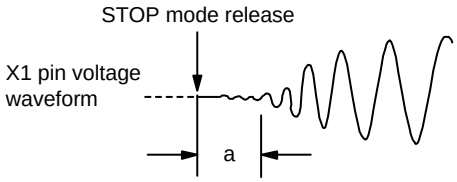
MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status		
									fx = 10 MHz	fx = 20 MHz
0	0	0	0	0	0	0	0	2^8/fx max.	25.6 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	2^8/fx min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	2^9/fx min.	51.2 μs min.	25.6 μs min.
1	1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

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

Caution 2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by the oscillation stabilization time select register (OSTS).
In the following cases, set the oscillation stabilization time of the OSTS register to a value greater than the count value to be monitored by using 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.)

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



Remark fx: X1 clock oscillation frequency

5.3.5 Oscillation stabilization time select register (OSTS)

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

When the X1 clock is 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. The oscillation stabilization time can be checked up to the time set using the OSTC register.

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

Reset signal generation sets the OSTS register to 07H.

Figure 5 - 7 Format of Oscillation stabilization time select register (OSTS)

Address: FFFA3H After reset: 07H R/W

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

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection		
				fx = 10 MHz	fx = 20 MHz
0	0	0	2^8/fx	25.6 μs	12.8 μs
0	0	1	2^9/fx	51.2 μs	25.6 μs
0	1	0	2^{10}/fx	102 μs	51 μs
0	1	1	2^{11}/fx	204 μs	102 μs
1	0	0	2^{13}/fx	819 μs	409 μs
1	0	1	2^{15}/fx	3.27 ms	1.63 ms
1	1	0	2^{17}/fx	13.1 ms	6.55 ms
1	1	1	2^{18}/fx	26.2 ms	13.1 ms

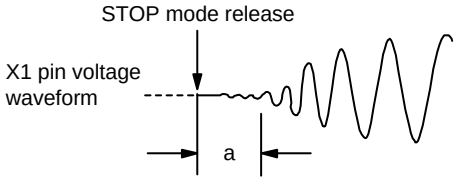
Caution 1. Change the setting of the OSTS register before setting the MSTOP bit of the clock operation status control register (CSC) to 0.

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

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

- **If the X1 clock starts oscillation while the high-speed on-chip oscillator clock 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.)**

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



Remark fx: X1 clock oscillation frequency

5.3.6 Peripheral enable registers 0, 1, 2 (PER0, PER1, 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 these registers, set (1) the bit corresponding to each function before specifying the initial settings of the peripheral functions.

- Real-time clock 2
- A/D converter
- Serial interface IICA0
- Serial array unit 1
- Serial array unit 0
- Timer array unit
- 12-bit interval timer
- Comparators 0 and 1
- DTC
- D/A converter
- 16-bit timer KB20, KB21, KB22

The PER0, PER1, and 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	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

RTCWEN	Control of real-time clock 2 (RTC2) input clock supply
0	Stops input clock supply. (Stops f_{CLK} clock supply) • SFRs used by the real-time clock 2 (RTC2) cannot be written. • The real-time clock 2 (RTC2) is operable.
1	Enables input clock supply. • SFRs used by the real-time clock 2 (RTC2) can be read and written. • The real-time clock 2 (RTC2) is operable.

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

Caution Be sure to clea bits 6 and 1 to 0.

Figure 5 - 9 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	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

IICA0EN	Control of serial interface IICA0 input clock supply
0	Stops input clock supply. • SFRs used by the serial interface IICA0 cannot be written. • The serial interface IICA0 is in the reset status.
1	Enables input clock supply. • SFRs used by the serial interface IICA0 can be read and written.

SAU1EN	Control of serial array unit 1 input clock supply
0	Stops input clock supply. • SFRs used by serial array unit 1 cannot be written. • Serial array unit 1 is in the reset status.
1	Enables input clock supply. • SFRs used by serial array unit 1 can be read and written.

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

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

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

Figure 5 - 10 Format of Peripheral enable register 1 (PER1)

Address: F007AH After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER1	TMKAEN	0	CMPEN	TKB20EN Note	DTCEN	0	0	DACEN

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

CMPEN	Control of comparator 0/1 input clock supply
0	Stops input clock supply. • SFRs used by comparators 0 and 1 cannot be written. • Comparators 0 and 1 are in the reset status.
1	Enables input clock supply. • SFRs used by comparators 0 and 1 can be read and written.

TKB20EN Note	Control of 16-bit timer KB20 input clock supply
0	Stops input clock supply. • SFRs used by 16-bit timer KB20 cannot be written. • 16-bit timer KB20 is in the reset status.
1	Enables input clock supply. • SFRs used by 16-bit timer KB20 can be read and written.

DTCEN	Control of DTC input clock supply
0	Stops input clock supply. • DTC cannot run.
1	Enables input clock supply. • DTC can run.

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

Note When FRQSEL4 in the user option byte (000C2H) is 1, select f_{IH} as f_{CLK} before setting bit 4 (TKB20EN) of peripheral enable register 1 (PER1).
When changing f_{CLK} to a clock other than f_{IH}, first clear bit 4 (TKB20EN) of peripheral enable register 1 (PER1).
In products with USB, when the CKSELR bit in the MCKC register is set to 1, select f_{PLL} as f_{CLK} before setting bit 4 (TKB20EN) of peripheral enable register 1 (PER1). When changing f_{CLK} to a clock other than f_{PLL}, clear bit 4 (TKB20EN) of peripheral enable register 1 (PER1) first.

Caution Be sure to clear bits 6, 2, and 1 to 0.

Figure 5 - 11 Format of Peripheral enable register 2 (PER2)

Address: F00FDH After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
PER2	0	0	0	0	0	0	TKB22EN	TKB21EN

TKB22EN Note	Control of 16-bit timer KB22 input clock supply
0	Stops input clock supply. • SFRs used by 16-bit timer KB22 cannot be written. • 16-bit timer KB22 is in the reset status.
1	Enables input clock supply. • SFRs used by 16-bit timer KB22 can be read and written.

TKB21EN Note	Control of 16-bit timer KB21 input clock supply
0	Stops input clock supply. • SFRs used by 16-bit timer KB21 cannot be written. • 16-bit timer KB21 is in the reset status.
1	Enables input clock supply. • SFRs used by 16-bit timer KB21 can be read and written.

Note When FRQSEL4 in the user option byte (000C2H) is 1, select f_{IH} as f_{CLK} before setting bits 0 (TKB21EN), 1 (TKB22EN) of peripheral enable register 2 (PER2).
When changing f_{CLK} to a clock other than f_{IH}, first clear bits 0 (TKB21EN), 1 (TKB22EN) of peripheral enable register 2 (PER2). In products with USB, when the CKSELR bit in the MCKC register is set to 1, select f_{PLL} as f_{CLK} before setting bit 0 (TKB21EN) and bit 1 (TKB22EN) of peripheral enable register 2 (PER2). When changing f_{CLK} to a clock other than f_{PLL}, clear bit 0 (TKB21EN) and bit 1 (TKB22EN) of peripheral enable register 2 (PER2) first.

Caution Be sure to clear bits 7 to 2 to “0”.

5.3.7 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 other than the real-time clock 2, 12-bit interval timer, clock output/buzzer output controller, and LCD controller/driver is stopped in STOP mode or in HALT mode while the subsystem clock is selected as the CPU clock.

The OSMC register can be used to select the count clock of the real-time clock 2 and 12-bit interval timer, and the operating clock of the clock output/buzzer output and LCD driver/controller.

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

Reset signal generation clears this register to 00H.

Figure 5 - 12 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 Note	0	0	0	0

RTCLPC	Setting in STOP mode or in HALT mode while subsystem clock is selected as CPU clock
0	Enables supplying the subsystem clock to peripheral functions (See **Tables 23 - 1** and **23 - 2** for peripheral functions whose operations are enabled.)
1	Stops supplying the subsystem clock to peripheral functions other than the real-time clock 2, 12-bit interval timer, clock output/buzzer output controller, and LCD controller/driver.

WUTMMCK0 Note	Selection of operation clock for real-time clock 2, 12-bit interval timer, and LCD controller/driver	Selection of clock output from PCLBUZn pin of clock output/buzzer output controller
0	Subsystem clock (f_{SUB})	Selecting the subsystem clock (f_{SUB}) is enabled.
1	Low-speed on-chip oscillator clock (f_{IL})	Selecting the subsystem clock (f_{SUB}) is disabled.

Note Be sure to select the subsystem clock (WUTMMCK0 bit = 0) while the subsystem clock is oscillating.

Caution 1. The subsystem clock and low-speed on-chip oscillator clock can only be switched by using the WUTMMCK0 bit if the real-time clock 2, 12-bit interval timer, and LCD controller/driver are all stopped. These are stopped as follows:
Real-time clock 2: Set the RTCE bit to 0.
12-bit interval timer: Set the RINTE bit to 0.
LCD controller/driver: Set the SCOC and VLCON bits to 0.

Caution 2. Do not select f_{SUB} as the clock output or buzzer output clock when the WUTMMCK0 bit is 1.

Remark RTCE: Bit 7 of real-time clock control register 0 (RTCC0)
RINTE: Bit 15 of the 12-bit interval timer control register (ITMC)
SCOC: Bit 6 of LCD mode register 1 (LCDM1)
VLCON: Bit 5 of LCD mode register 1 (LCDM1)

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

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

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

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

Figure 5 - 13 Format of High-speed on-chip oscillator frequency select register (HOCODIV)

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

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

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

Note 1. See the MCKC register for division ratio settings.

Note 2. When using the high-speed on-chip oscillator clock to operate the USB/function controller, be sure to set f_{HOCO} = 48 MHz.

Caution 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			
0	0	LV (low-voltage main) mode	1 to 4 MHz	1.6 to 3.6 V
1	0	LS (low-speed main) mode	1 to 8 MHz	1.8 to 3.6 V
1	1	HS (high-speed main) mode	1 to 16 MHz	2.4 to 3.6 V
			1 to 24 MHz	2.7 to 3.6 V

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

Caution 3. After 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**

Caution 4. To change the frequency of the high-speed on-chip oscillator when X1 oscillation, external oscillation input or sub clock is set for the system clock, stop the high-speed on-chip oscillator by setting bit 0 (HIOSTOP) of the CSC register to 1 and then change the frequency.

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

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

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

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

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

Figure 5 - 14 Format of High-speed on-chip oscillator trimming register (HIOTRM)

Address: F00A0H After reset: Note R/W

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

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

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

Remark 1. The HIOTRM register holds a six-bit value used to adjust the high-speed on-chip oscillator with an increment of 1 corresponding to an increase of frequency by about 0.05%.

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

5.3.10 PLL control register (DSCCTL)

This register is used to control the operations of the PLL oscillator Note.

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

Reset signal generation clears this register to 00H.

Note Products with USB.

Figure 5 - 15 Format of PLL control register (DSCCTL)

Address: F02E5H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
DSCCTL	0	0	0	0	0	DSFRDIV	DSCM	DSCON

DSFRDIV	PLL reference clock divider control
0	No division
1	Divided by 2

Remark PLL reference clock is the high-speed system clock (fMX).

DSCM	PLL multiplication selection
0	12 times (6 times)
1	16 times (8 times)

Remark The frequency is divided by 2 in the last stage of the PLL oscillator, therefore the multiplication ratio becomes the value in parentheses.

DSCON	PLL oscillation and output control
0	Stop
1	Oscillation, output

Caution 1. Be sure to clear bits 3 to 7 to 0.

Caution 2. Be sure to set the DSCON bit to 0before changing DSFRDIVand DSCM.

Caution 3. Do not set the DSCONbit to0while the PLL clock is selected as the system clock.

The combination which user can select as the USB clock when the PLL is used is shown below.

Figure 5 - 16 Relationship between the PLL and the USB Clock

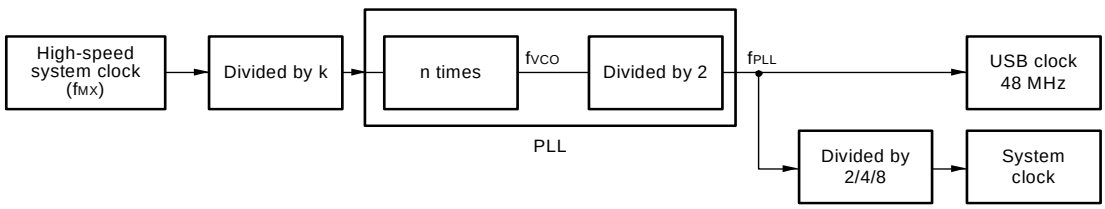


Table 5 - 3 USB Clock Frequency Setting Example

High-speed system clock (fMX)	Divided by k DSFRDIV	n times DSCM		Frequency after n times (fVCO)	USB clock (fPLL)
16 MHz	Divided by 2	0	12 times	96 MHz	48 MHz
12 MHz	Divided by 2	1	16 times	96 MHz	48 MHz

Table 5 - 3 USB Clock Frequency Setting Example

High-speed system clock (f_{MX})	Divided by k DSFRDIV	n times DSCM		Frequency after n times (f_{VCO})	USB clock (f_{PLL})
8 MHz	No division	0	12 times	96 MHz	48 MHz
6 MHz	No division	1	16 times	96 MHz	48 MHz

5.3.11 Main clock control register (MCKC)

This register is used to control the operations of the main clock Note.

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

Reset signal generation clears this register to 00H.

Note Products with USB.

Figure 5 - 17 Format of Main clock control register (MCKC)

Address: F02E6H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MCKC	0	0	0	0	0	RDIV1	RDIV0	CKSELR

RDIV1	RDIV0	High-speed on-chip oscillator clock/PLL clock division ratio (divided by 2/4/8) selection
0	0	Divided by 2
0	1	Divided by 4
1	0	Divided by 8
1	1	Setting prohibited

CKSELR	Selection of USB clock/timers KB20, KB21, and KB22/system clock source
0	High-speed on-chip oscillator clock (f_{HOCO}) only when 48 MHz is selected by the option byte
1	PLL clock (f_{PLL})

Caution 1. When the clock is switched between the PLL clock (f_{PLL}) and the high-speed on-chip oscillator clock (f_{HOCO}), both clocks must be stopped.

Caution 2. When selecting the high-speed on-chip oscillator clock as the USB clock (f_{USB}), set the UCKSELC bit in the USB clock select register (UCKSEL) to 1.

Remark A clock selected by this bit when the MCM0 bit is 0 becomes the main clock.

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

5.3.12 USB clock selection register (UCKSEL)

The UCKSEL register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 5 - 18 Format of USB clock selection register (UCKSEL)

Address: F06C4H, F06C5H After reset: 0000H R/W

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

UCKSELC	USB clock selection
0	High-speed on-chip oscillator clock (f_{HOCO}) is not selected as USB clock
1	High-speed on-chip oscillator clock (f_{HOCO}) is selected as USB clock

Caution 1. When selecting the high-speed on-chip oscillator clock (f_{HOCO}) as the USB clock, set UCKSELC = 1 and the CKSELR bit in the MCKC register to 0 at the same time.

Caution 2. The USB clock select register can be rewritten only when the USB is disconnected.

Caution 3. The high-speed on-chip oscillator clock can be selected only when T_A = -20 to +85°C.

Caution 4. If the high-speed on-chip oscillator clock is selected as the USB clock, when the USB is suspended, perform USB suspended processing while the high-speed on-chip oscillator clock is selected (UCKSELC = 1).

Caution 5. If the high-speed on-chip oscillator clock is selected as the USB clock, when the USB is disconnected, perform USB stopped processing (including setting DPRPU = 0) before setting UCKSELC = 0.

5.4 System Clock Oscillator

5.4.1 X1 oscillator

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

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

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

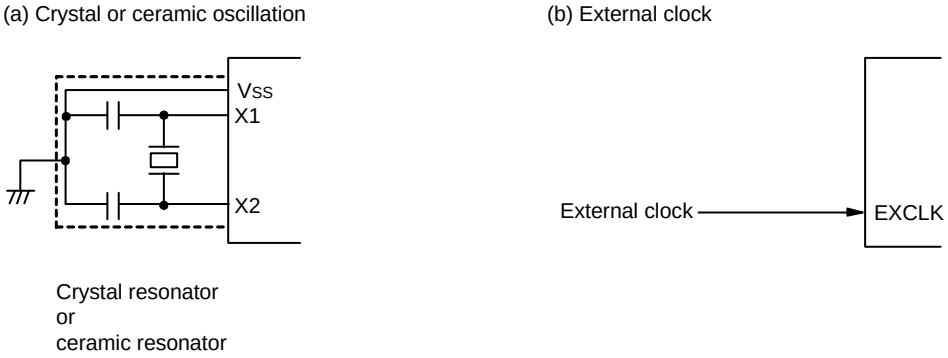
- Crystal or ceramic oscillation: EXCLK, OSCSEL = 0, 1
- External clock input: EXCLK, OSCSEL = 1, 1

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

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

Figure 5 - 19 shows an example of the external circuit of the X1 oscillator.

Figure 5 - 19 Example of External Circuit of X1 Oscillator



(**Cautions** are listed on the next page.)

5.4.2 XT1 oscillator

The XT1 oscillator oscillates with a crystal resonator (standard: 32.768 kHz) 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

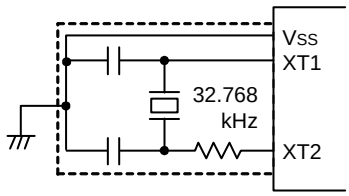
When the XT1 oscillator is not used, set the input port mode (EXCLKS, OSCSELS = 0, 0).

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

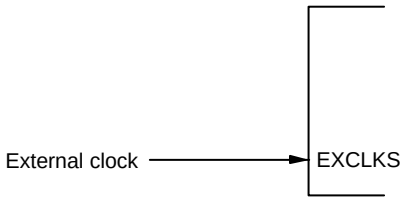
Figure 5 - 20 shows an example of the external circuit of the XT1 oscillator.

Figure 5 - 20 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 - 19 and 5 - 20 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.**

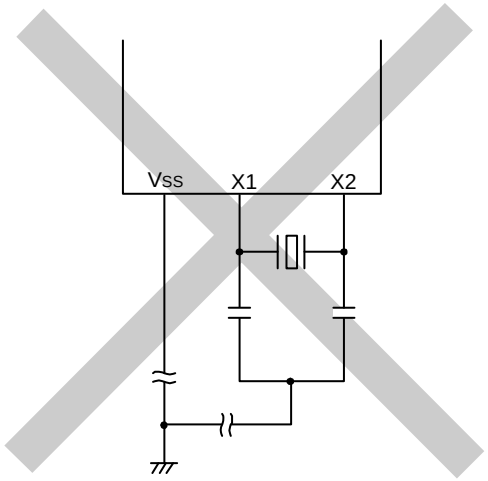
The XT1 oscillator is a circuit with low amplification in order to achieve low-power consumption. Note the following points when designing the circuit.

- **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.**

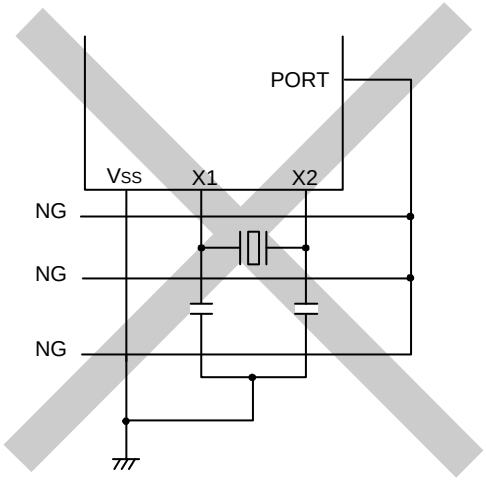
Figure 5 - 21 shows examples of incorrect resonator connection.

Figure 5 - 21 Examples of Incorrect Resonator Connection (1/2)

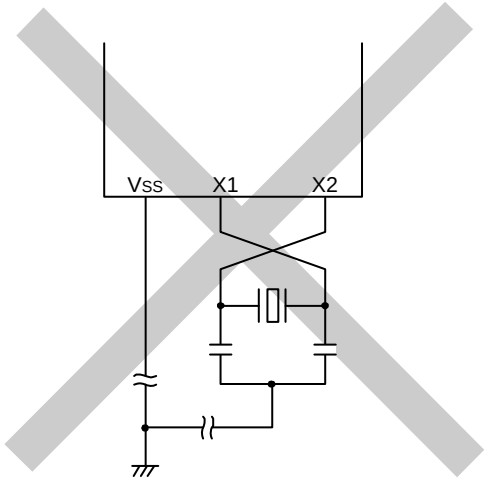
(a) Too long wiring



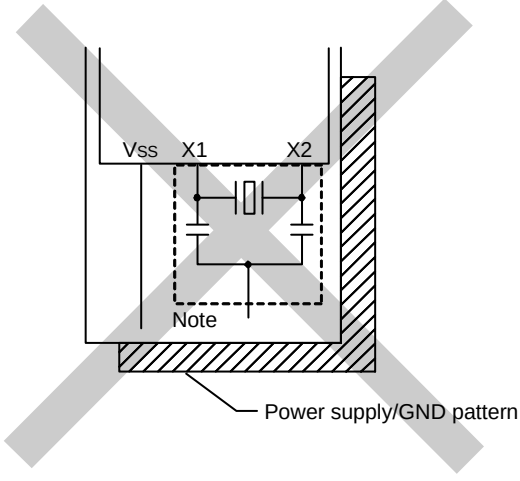
(b) Crossed signal line



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



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

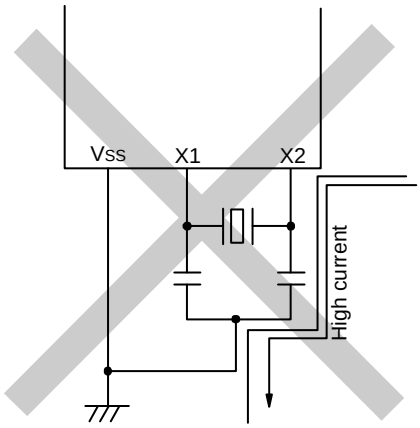


Note Do not place a power supply/GND pattern under the wiring section (section indicated by a broken line in the figure) of the X1 and X2 pins and the resonators in a multi-layer board or double-sided board.
Do not configure a layout that will cause capacitance elements and affect the oscillation characteristics.

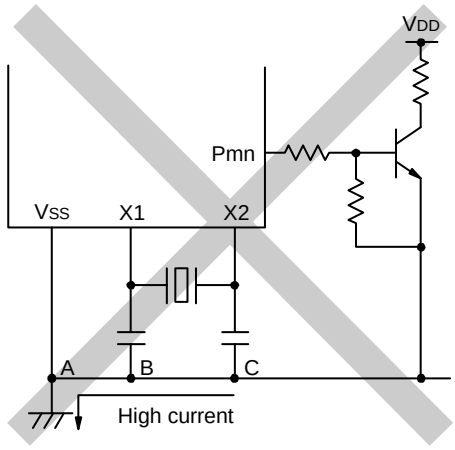
Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

Figure 5 - 22 Examples of Incorrect Resonator Connection (2/2)

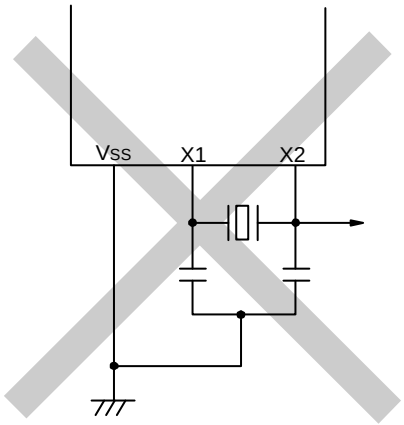
(e) Wiring near high alternating current



(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/L1C. The frequency can be selected from among 48, 24, 16, 12, 8, 6, 4, 3, 2, or 1 MHz by using the option byte (000C2H). When 48 MHz is selected, the two frequency division of the selected clock is supplied to CPU clock. Oscillation can be controlled by bit 0 (HIOSTOP) of the clock operation status control register (CSC).

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

5.4.4 Low-speed on-chip oscillator

The low-speed on-chip oscillator is incorporated in the RL78/L1C.

The low-speed on-chip oscillator clock is used only as the clock for the watchdog timer, real-time clock 2, 12-bit interval timer, and the LCD controller/driver. The low-speed on-chip oscillator clock cannot be used as the CPU clock.

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

The low-speed on-chip oscillator is stopped when the watchdog timer is stopped and WUTMMCK0 is set to 0.

5.4.5 PLL (Phase Locked Loop)

The PLL circuit is incorporated in the RL78/L1C Note.

The PLL can be used to multiply the high-speed system clock.

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

Note Products with USB

Caution 1. When switching from PLL mode to the internal high-speed oscillation clock and the high-speed system clock, stop the function (USB function controller) that provides the PLL output clock (f_{PLL}).

Caution 2. Do not set the DSCON bit to 1 to start the PLL operating while the subsystem clock is the operating clock for the CPU.

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**).

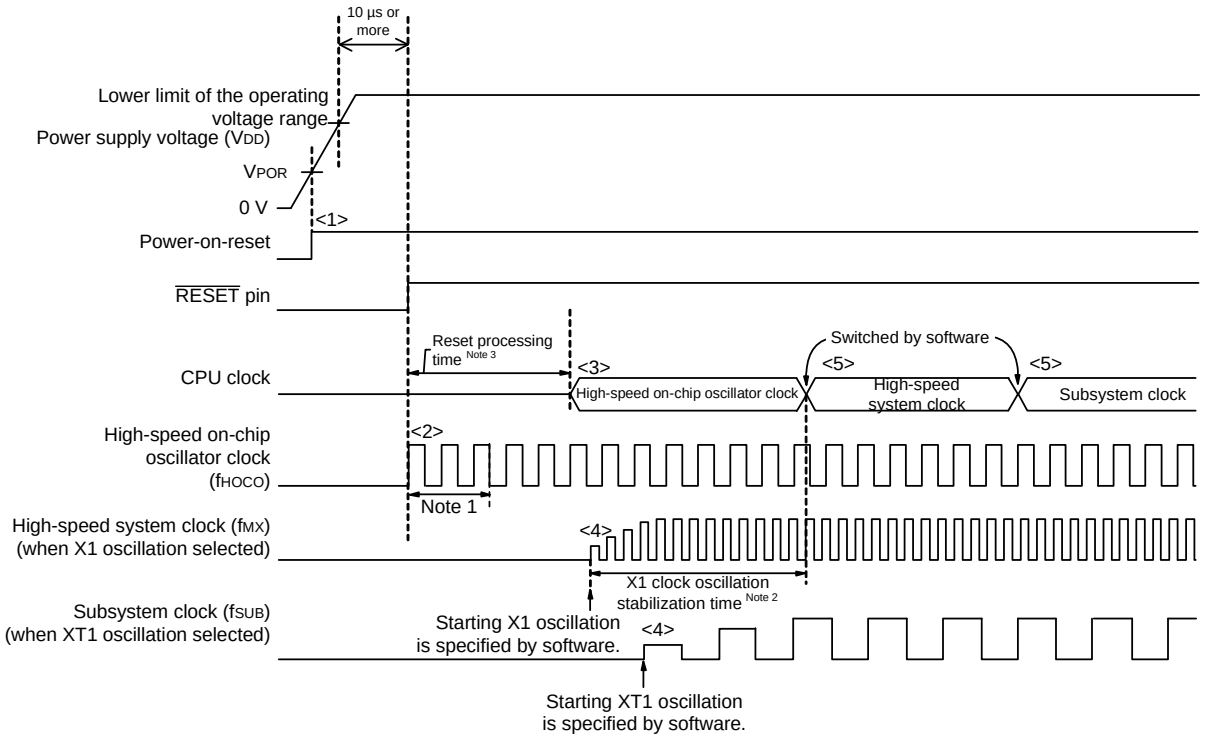
- Main system clock f_{MAIN}
  - High-speed system clock f_{MX}
    X1 clock f_X
    External main system clock f_{EX}
  - High-speed on-chip oscillator clock f_{HOCO}
  - PLL clock f_{PLL} Note

Note Products with USB

- Subsystem clock f_{SUB}
  - XT1 clock f_{XT}
  - External subsystem clock f_{EXT}
- Low-speed on-chip oscillator clock f_{IL}
- CPU/peripheral hardware clock f_{CLK}

The CPU starts operation when the high-speed on-chip oscillator starts outputting after a reset release in the RL78/L1C.
When the power supply voltage is turned on, the clock generator operation is shown in Figure 5 - 23.

Figure 5 - 23 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 **34.4** or **35.4 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**).

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

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

Note 3. For details on the reset processing time, refer to **CHAPTER 25 POWER-ON-RESET CIRCUIT**.

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

5.6 Controlling Clock

5.6.1 Example of setting high-speed on-chip oscillator

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. The frequency of the high-speed on-chip oscillator can be selected from 48, 24, 16, 12, 8, 6, 4, 3, 2, and 1 MHz by using FRQSEL0 to FRQSEL4 of the option byte (000C2H). In addition, oscillation can be changed by the high-speed on-chip oscillator frequency select register (HOCODIV).

[Option byte setting]

Address: 000C2H

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

CMODE1	CMODE0	Setting of flash operation mode	
0	0	LV (low voltage main) mode	V_{DD} = 1.6 V to 3.6 V @ 1 MHz to 4 MHz
1	0	LS (low speed main) mode	V_{DD} = 1.8 V to 3.6 V @ 1 MHz to 8 MHz
1	1	HS (high speed main) mode Note 1	V_{DD} = 2.4 V to 3.6 V @ 1 MHz to 16 MHz V_{DD} = 2.7 V to 3.6 V @ 1 MHz to 24 MHz
Other than above		Setting prohibited	

FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator	
					f_{HOCO}	f_{IH}
1	0	0	0	0	48 MHz Note 2	24/12/6 MHz Note 3
0	0	0	0	0	24 MHz	24 MHz
0	1	0	0	1	16 MHz	16 MHz
0	0	0	0	1	12 MHz	12 MHz
0	1	0	1	0	8 MHz	8 MHz
0	0	0	1	0	6 MHz	6 MHz
0	1	0	1	1	4 MHz	4 MHz
0	0	0	1	1	3 MHz	3 MHz
0	1	1	0	0	2 MHz	2 MHz
0	1	1	0	1	1 MHz	1 MHz
Other than above					Setting prohibited	

Note 1. When you use PLL, please choose HS (high speed main) mode.

Note 2. When you use PLL, set it in FRQSEL4 = 0, and, please do not choose 48 MHz.

Note 3. See the MCKC register for division ratio settings.

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

Address: F00A8H

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

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

Note See the MCKC register for division ratio settings.

5.6.2 Example of setting X1 oscillation clock

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

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

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

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

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

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

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

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

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

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

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

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

Caution 1. The EXCLKS, OSCSELS, AMPHS1, AMPHS0, XTSTOP, and RTCWEN bits are reset only by a power on reset; they retain the previous values when a reset caused by another factor occurs.

Caution 2. Keep the operating voltage within the range that allows operation of the flash memory as set in an option byte (000C2H) before and after changes to the main system clock (f_{MAIN}) by using the system clock control register (CKC).

Option Byte (000C2H) Value		Flash Operation Mode	Operating Frequency Range	Operating Voltage Range
CMODE1	CMODE0			
0	0	LV (low-voltage main) mode	1 MHz to 4 MHz	1.6 V to 5.5 V
1	0	LS (low-speed main) mode	1 MHz to 8 MHz	1.8 V to 5.5 V
1	1	HS (high-speed main) mode	1 MHz to 16 MHz	2.4 V to 5.5 V
			1 MHz to 32 MHz	2.7 V to 5.5 V

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 2, 12-bit interval timer, and LCD controller/driver on the subsystem clock (for ultra-low current consumption) in the STOP mode or HALT mode during CPU operation on the subsystem clock.

	7	6	5	4	3	2	1	0
OSMC	RTCLPC 0/1	0	0	WUTMMCK0 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 0	OSCSEL 0	EXCLKS 0	OSCSELS 1	0	AMPHS1 0/1	AMPHS0 0/1	AMPH 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 1	XTSTOP 0	0	0	0	0	0	HIOSTOP 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 0	CSS 1	MCS 0	MCM0 0	0	0	0	0

Caution The EXCLKS, OSCSELS, AMPHS1, AMPHS0, XTSTOP, and RTCWEN bits are reset only by a power on reset; they retain the previous values when a reset caused by another factor occurs.

5.6.4 Example of setting PLL circuit

After setting the high-speed system clock (see **5.6.2 Example of setting X1 oscillation clock**), use the PLL control register (DSCCTL) to control the PLL circuit Note.

Note Products with USB

[Register settings] Set the register in the order of <1> to <8> below.

<1> Set the HIOSTOP bit in the CSC register to make the high-speed on-chip oscillator run.

	7	6	5	4	3	2	1	0
CSC	0/1	0/1	0	0	0	0	0	HIOSTOP 0Note 1

<2> Set the DSFRDIV bit and DSCM bit in the DSCCTL register to set the PLL multiplication and division.

	7	6	5	4	3	2	1	0
DSCCTL	0	0	0	0	0	DSFRDIV 0/1	DSCM 0/1	DSCON 0

<3> Set the RDIV1, RDIV0 bits of the MCKC register to set the division of the system clock.

	7	6	5	4	3	2	1	0
MCKC	0	0	0	0	0	RDIV1 0/1	RDIV0 0/1	CKSELR 0Note 1

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

<4> Set (1) the DSCON bit of the DSCCTL register to operate the PLL circuit Note2.

	7	6	5	4	3	2	1	0
DSCCTL	0	0	0	0	0	DSFRDIV 0/1	DSCM 0/1	DSCON 1

<5> Set (1) the CKSELR bit of the MCKC register to select PLL output for the system clock.

	7	6	5	4	3	2	1	0
MCKC	0	0	0	0	0	RDIV1 0/1	RDIV0 0/1	CKSELR 1

<6> Use software to set up a wait of 65 μs.Note3

<7> Set the HIOSTOP bit in the CSC register to stop the high-speed on-chip oscillator.[Note2].

	7	6	5	4	3	2	1	0
CSC	0/1	0/1	0	0	0	0/1	0/1	HIOSTOP 1[Note1]

<8> When the PLL clock frequency divided by 2, 4, or 8 is selected as the main system clock (f_{MAIN}), set the MCM0 bit in the CKC register to select the source for deriving the main system clock as a signal with a frequency (f_{IH}) of up to 24 MHz..

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

Note 1. No setting is required to change to the PLL while the CKSELR bit is 1. When setting the CKSELR bit to 1, ensure that the high-speed on-chip oscillator is running.

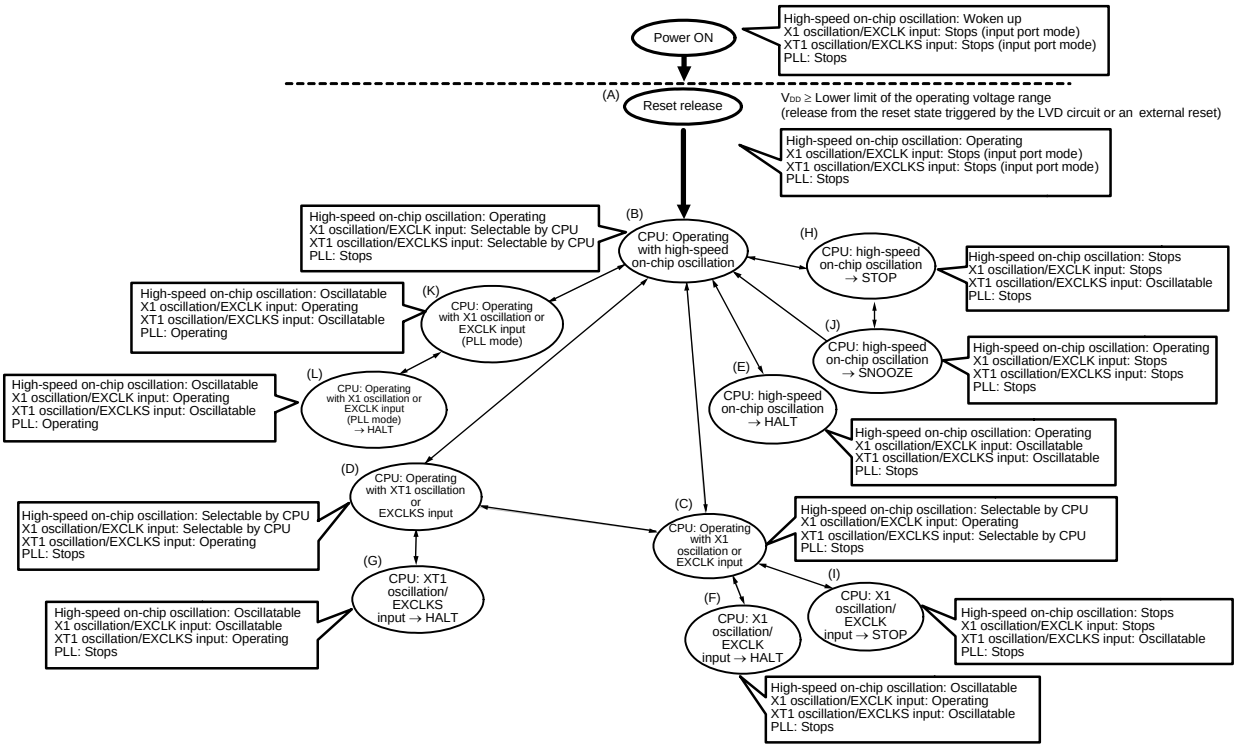
Note 2. After oscillation by the X1 oscillator clock has become stable, allow at least 1 μs to elapse before starting the PLL. When restarting the PLL after it has been stopped, wait for at least 4 μs before using it in operations.

Note 3. Wait for 40 μs for oscillation by the oscillator clock to become stabled if the HIOSTOP bit is not set to 0.

5.6.5 CPU clock status transition diagram

Figures 5 - 24 and 5 - 25 show the CPU Clock Status Transition Diagram (Products with USB) of this product.

Figure 5 - 24 CPU Clock Status Transition Diagram (Products with USB)



Tables 5 - 4 to 5 - 10 show transition of the CPU clock and examples of setting the SFR registers.

Table 5 - 4 CPU Clock Transition and SFR Register Setting Examples (1/7)

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

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

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

(Setting sequence of SFR registers) →

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

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

Note 2. Set the oscillation stabilization time as follows.
- Desired the oscillation stabilization time counter status register (OSTC) oscillation stabilization time ≤ Oscillation stabilization time set by the oscillation stabilization time select register (OSTS)

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 34 or CHAPTER 35 ELECTRICAL SPECIFICATIONS).

(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.

Remark 1. ×: don't care

Remark 2. (A) to (L) in Tables 5 - 4 to 5 - 10 correspond to (A) to (L) in Figure 5 - 24.

Table 5 - 5 CPU Clock Transition and SFR Register Setting Examples (2/7)

(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	Need not be checked	1

Unnecessary if these registers are already set

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

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

Note 2. Set the oscillation stabilization time as follows.

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

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 34 or CHAPTER 35 ELECTRICAL SPECIFICATIONS).

(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, 0	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.

Remark 1. ×: don't care

Remark 2. (A) to (L) in Tables 5 - 4 to 5 - 10 correspond to (A) to (L) in Figure 5 - 24.

Table 5 - 6 CPU Clock Transition and SFR Register Setting Examples (3/7)

(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	Note	0

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

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

Remark 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 sub system 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	Note	0

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

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

Remark 1. (A) to (L) in Tables 5 - 4 to 5 - 10 correspond to (A) to (L) in Figure 5 - 24.
Remark 2. The oscillation accuracy stabilization time changes according to the temperature conditions and the STOP mode period.

Table 5 - 7 CPU Clock Transition and SFR Register Setting Examples (4/7)

(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 ≤ fx ≤ 10 MHz)	Note	0	Must be checked	0
(D) → (C) (X1 clock: 10 MHz < fx ≤ 20 MHz)	Note	0	Must be checked	0
(D) → (C) (external main clock)	Note	0	Need 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 34 or CHAPTER 35 ELECTRICAL SPECIFICATIONS).**

Remark (A) to (L) in Tables 5 - 4 to 5 - 10 correspond to (A) to (L) in Figure 5 - 24.

Table 5 - 8 CPU Clock Transition and SFR Register Setting Examples (5/7)

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

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	DSCCTL Register Note 1			OSTS Register	CSC Register	OSTC Register	DSCCTL Register	
	DEXCLK	OSCSEL	AMPH		MSTOP		DSFRDIV	DSCM
(B) → (K) (divided by 2)	0/1	1	0/1	Note 2	0	Must be checked	0/1	0/1
(B) → (K) (divided by 4)	0/1	1	0/1	Note 2	0	Must be checked	0/1	0/1
(B) → (K) (divided by 8)	0/1	1	0/1	Note 2	0	Must be checked	0/1	0/1

→

MCKC Register		Waiting for Oscillation Stabilization	DSCCTL Register	Waiting for Oscillation Stabilization	MCKC Register
RDIV1	RDIV0		DSCON		CKSELR
0	0	1 μs	1	40 μs	1
0	1		1		1
1	0		1		1

Note 1. Writing to the clock operating mode control register (CMC) can only proceed once and must be by an 8-bit memory manipulation instruction after release from the reset state.

Note 2. Set the oscillation stabilization time in the oscillation stabilization time select register (OSTS) as follows.

- Desired oscillation stabilization time setting of the oscillation stabilization time counter status register (OSTC) ≤ Oscillation stabilization time set in the OSTS register

Caution Completion of clock switching after the CKSELR bit has been set to 1 requires up to 2 clock cycles when the FRQSEL4 bit is 1, and up to 10 clock cycles when the FRQSEL4 bit is 0. Until the clock switching is completed, do not stop the high-speed on-chip oscillator.

Table 5 - 9 CPU Clock Transition and SFR Register Setting Examples (6/7)

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

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register HIOSTOP	Waiting for Oscillation Stabilization	MCKC Register CKSELR	Waiting for clock change	DSCCTL Register DSCON
(K) → (B) FRQSEL4=0	0	18 to 65 μs	0	256 clock	0
(K) → (B) FRQSEL4=1		18 ro 135 μs		16 clock	

Table 5 - 10 CPU Clock Transition and SFR Register Setting Examples (7/7)

(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 system clock (PLL mode) (K)

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

Remark (A) to (L) in Tables 5 - 4 to 5 - 10 correspond to (A) to (L) in Figure 5 - 24.

(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) →

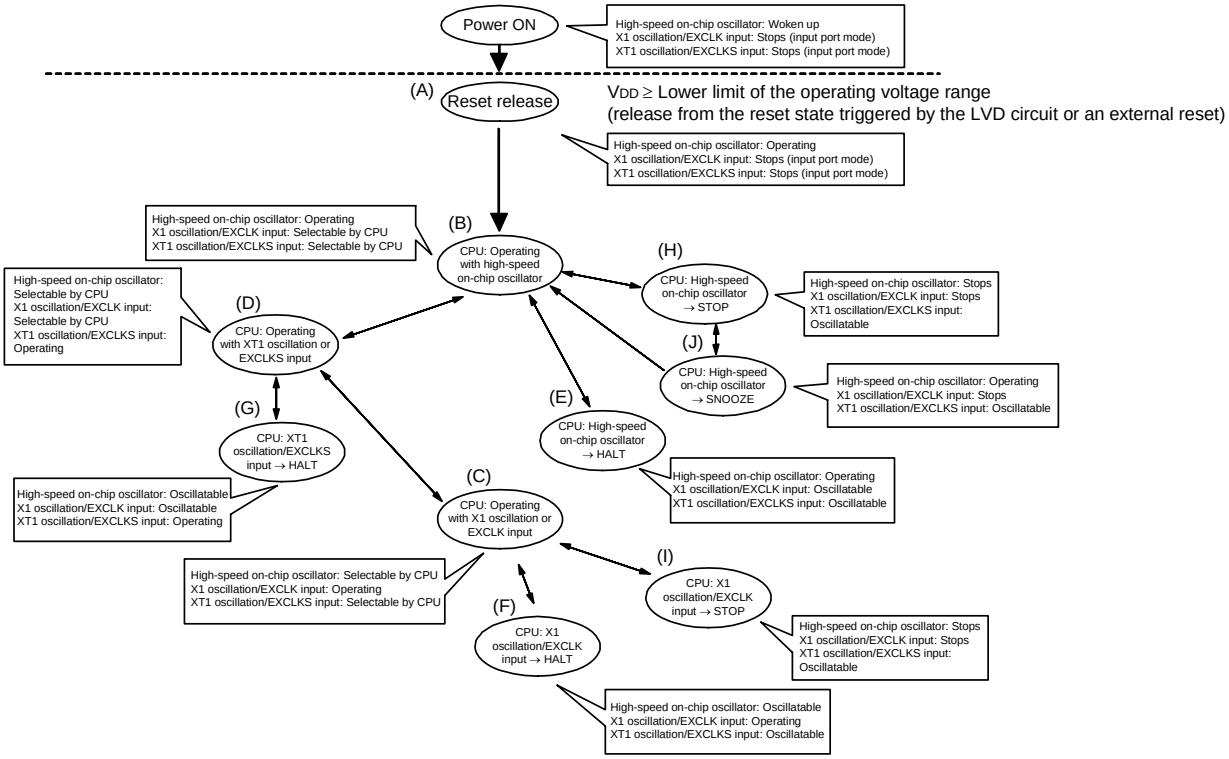
Status Transition		Setting		
(B) → (H)		Stopping peripheral functions that are disabled in STOP mode	—	Executing STOP instruction
(C) → (I)	In X1 oscillation		Sets the OSTS register	
	External main system clock		—	

(14) CPU changing from STOP mode (H) to SNOOZE mode (J)
For details about the setting for switching from the STOP mode to the SNOOZE mode, see **12.8 SNOOZE Mode Function**, **15.5.7 SNOOZE mode function** and **15.6.3 SNOOZE mode function**.

(15) • Changing to STOP mode (I) from the high-speed system clock (PLL mode) as the operating clock for the CPU (K) Switch to high-speed system clock operation from PLL mode, stop the PLL (DSCON = 0), and then execute the STOP instruction.

Remark (A) to (L) in Tables 5 - 4 to 5 - 10 correspond to (A) to (L) in Figure 5 - 24.

Figure 5 - 25 CPU Clock Status Transition Diagram (Products without USB)



Tables 5 - 11 to 5 - 15 show transition of the CPU clock and examples of setting the SFR registers.

Table 5 - 11 CPU Clock Transition and SFR Register Setting Examples (1/5)

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

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

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

(Setting sequence of SFR registers) →

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

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

Note 2. Set the oscillation stabilization time as follows.
- Desired the oscillation stabilization time counter status register (OSTC) oscillation stabilization time ≤ Oscillation stabilization time set by the oscillation stabilization time select register (OSTS)

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 34 or CHAPTER 35 ELECTRICAL SPECIFICATIONS).

(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.

Remark 1. ×: don't care

Remark 2. (A) to (J) in Tables 5 - 11 to 5 - 15 correspond to (A) to (J) in Figure 5 - 25.

Table 5 - 12 CPU Clock Transition and SFR Register Setting Examples (2/5)

(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	Need not be checked	1

Unnecessary if these registers are already set

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

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

Note 2. Set the oscillation stabilization time as follows.

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

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 34 or CHAPTER 35 ELECTRICAL SPECIFICATIONS).

(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, 0	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.

Remark 1. ×: don't care

Remark 2. (A) to (J) in Tables 5 - 11 to 5 - 15 correspond to (A) to (J) in Figure 5 - 25.

Table 5 - 13 CPU Clock Transition and SFR Register Setting Examples (3/5)

(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	Note	0

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

Note When FRQSEL4 = 0: 18 μs to 65 μs
When FRQSEL4 = 1: 18 μs to 75 μs

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	Note	0

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

Note When FRQSEL4 = 0: 18 μs to 65 μs
When FRQSEL4 = 1: 18 μs to 75 μs

Remark 1. (A) to (J) in Tables 5 - 11 to 5 - 15 correspond to (A) to (J) in Figure 5 - 25.
Remark 2. The oscillation accuracy stabilization time changes according to the temperature conditions and the STOP mode period.

Table 5 - 14 CPU Clock Transition and SFR Register Setting Examples (4/5)

(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 ≤ fx ≤ 10 MHz)	Note	0	Must be checked	0
(D) → (C) (X1 clock: 10 MHz < fx ≤ 20 MHz)	Note	0	Must be checked	0
(D) → (C) (external main clock)	Note	0	Need 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 34 or CHAPTER 35 ELECTRICAL SPECIFICATIONS).

(10) • 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)

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

Remark (A) to (J) in Tables 5 - 11 to 5 - 15 correspond to (A) to (J) in Figure 5 - 25.

Table 5 - 15 CPU Clock Transition and SFR Register Setting Examples (5/5)

(11) • 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		—	

(12) CPU changing from STOP mode (H) to SNOOZE mode (J)
For details about the setting for switching from the STOP mode to the SNOOZE mode, see **12.8 SNOOZE Mode Function**, **15.5.7 SNOOZE mode function**, and **15.6.3 SNOOZE mode function**.

Remark (A) to (J) in Tables 5 - 11 to 5 - 15 correspond to (A) to (J) in Figure 5 - 25.

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 - 16 Changing CPU Clock (1/3)

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	
	PLL clock	Stabilization of X1 oscillation • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time Or enabling input of external clock from the EXCLKS pin • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0 Oscillation of PLL • DSCON = 1	
X1 clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • The oscillation accuracy stabilization time has elapsed	X1 oscillation can be stopped (MSTOP = 1) after checking that the CPU clock is changed.
	External main system clock	Transition not possible	—
	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.
	PLL clock	Oscillation of PLL • DSCON = 1 Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • The oscillation accuracy stabilization time has elapsed	—

Table 5 - 17 Changing CPU Clock (2/3)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
External main system clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • The oscillation accuracy stabilization time has elapsed	External main system clock input can be disabled (MSTOP = 1).
	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).
	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).
	PLL clock	Oscillation of PLL • DSCON = 1 Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • The oscillation accuracy stabilization time has elapsed	—
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)
	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	—
	PLL clock	Transition not possible	—

Table 5 - 18 Changing CPU Clock (3/3)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
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).
	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	—
	PLL clock	Transition not possible	—
PLL clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator • HIOSTOP = 0	Operating current can be reduced by stopping PLL (DSCON = 0).
	X1 clock	Stabilization of X1 oscillation • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time	
	External main system clock	Enabling input of external clock from the EXCLK pin • OSCSEL = 1, EXCLK = 1, MSTOP = 0	
	XT1 clock	Transition not possible	—
	External subsystem clock	Transition not possible	—

5.6.7 Time required for switchover of CPU clock and main 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 - 19** to **5 - 21**).

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 - 19 Maximum Time Required for Main System Clock Switchover

Clock A	Switching directions	Clock B	Remark
f_{IH}	⟷	f_{MX}	See **Table 5 - 20**
f_{MAIN}	⟷	f_{SUB}	See **Table 5 - 21**

Table 5 - 20 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 ($f_{MAIN} = f_{IH}$)	$f_{MX} \geq f_{IH}$	$2f_{MX}/f_{IH}$ clock	
	$f_{MX} < f_{IH}$	2 clock	

Table 5 - 21 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	

Remark 1. The number of clocks listed in Tables 5 - 20 and 5 - 21 is the number of CPU clocks before switchover.

Remark 2. Calculate the number of clocks in Tables 5 - 20 and 5 - 21 by rounding up the number after the decimal position.

Example When switching the main system clock from the high-speed system clock to the high-speed on-chip oscillator clock (when f_{IH} = 8 MHz, f_{MX} = 10 MHz)

$2f_{MX}/f_{IH}$ cycles = 2 (10/8) = 2.5 → 3 clock cycles

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 - 22 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

5.7 Resonator and Oscillator Constants

For the resonators for which the operation has been verified and their oscillation constants (for reference), see the page for the corresponding product at the Renesas Web site (http://www.renesas.com).

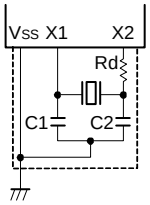
The resonators for which the operation is verified and their oscillator constants are shown below.

Caution 1. The constants for these oscillator circuits are reference values based on specific environments set up for evaluation by the manufacturers. For actual applications, request evaluation by the manufacturer of the oscillator circuit mounted on a board. Furthermore, if you are switching from a different product to this microcontroller, and whenever you change the board, again request evaluation by the manufacturer of the oscillator circuit mounted on the new board.

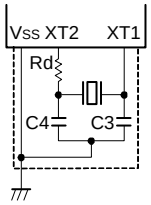
Caution 2. The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the RL78 microcontroller so that the internal operation conditions are within the specifications of the DC and AC characteristics.

Figure 5 - 26 Example of External Circuit

(a) X1 oscillation



(b) XT1 oscillation



(1) X1 oscillation

As of Aug 2016

Manufacturer	Resonator	Part Number Note 2	SMD/ Lead	Frequency (MHz)	Flash Operation Mode Note 1	Circuit Constants (Reference)			Voltage Range (V)	
						C1 (pF)	C2 (pF)	Rd (kΩ)	MIN.	MAX.
Murata Manufacturing Co., Ltd. Note 3	Crystal resonator	CSTCC2M00G56-R0	SMD	2.0	LV	(47)	(47)	0	1.6	3.6
		CSTCR4M00G3.6-R0	SMD	4.0		(39)	(39)	0		
		CSTLS4M00G53-B0	Lead			(15)	(15)	0		
		CSTCC2M00G56-R0	SMD	2.0	LS	(47)	(47)	0	1.8	3.6
		CSTCR4M00G3.6-R0	SMD	4.0		(39)	(39)	0		
		CSTLS4M00G53-B0	Lead			(15)	(15)	0		
		CSTCR4M19G55-R0	SMD	4.194		(39)	(39)	0		
		CSTLS4M19G53-B0	Lead			(15)	(15)	0		
		CSTCR4M91G53-R0	SMD	4.915		(15)	(15)	0		
		CSTLS4M91G53-B0	Lead			(15)	(15)	0		
		CSTCR5M00G53-R0	SMD	5.0		(15)	(15)	0		
		CSTLS5M00G53-B0	Lead			(15)	(15)	0		
		CSTCR6M00G53-R0	SMD	6.0		(15)	(15)	0		
		CSTLS6M00G53-B0	Lead			(15)	(15)	0		
		CSTCE8M00G52-R0	SMD	8.0		(10)	(10)	0		
		CSTLS8M00G53-B0	Lead			(15)	(15)	0		
		CSTCE8M38G52-R0	SMD	8.388	HS	(10)	(10)	0	2.4	3.6
		CSTLS8M38G53-B0	Lead			(15)	(15)	0		
		CSTCE10M0G52-R0	SMD	10.0		(10)	(10)	0		
		CSTLS10M0G53-B0	Lead			(15)	(15)	0		
		CSTCE12M0G52-R0	SMD	12.0		(10)	(10)	0		
		CSTCE16M0V53-R0	SMD	16.0		(15)	(15)	0		
		CSTLS16M0X51-B0	Lead			(5)	(5)	0		
		CSTCE20M0V51-R0	SMD	20.0		(5)	(5)	0	2.7	3.6
		CSTLS20M0X51-B0	Lead			(5)	(5)	0		

Note 1. Set the flash operation mode by using the CMODE1 and CMODE0 bits of the option byte (000C2H).

Note 2. Products compatible with 105°C have different part numbers. Contact Murata Manufacturing Company, Ltd. (http://www.murata.co.jp) for more information.

Note 3. When using this resonator, for details about the matching, contact Murata Manufacturing Co., Ltd. (http://www.murata.com).

Remark Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.

HS (High-speed main) mode: 2.7 V ≤ V_{DD} ≤ 3.6 V@1 MHz to 24 MHz
2.4 V ≤ V_{DD} ≤ 3.6 V@1 MHz to 16 MHz
LS (Low-speed main) mode: 1.8 V ≤ V_{DD} ≤ 3.6 V@1 MHz to 8 MHz
LV (Low-voltage main) mode: 1.6 V ≤ V_{DD} ≤3.6 V@1 MHz to 4 MHz

(2) XT1 oscillation (crystal resonator)

As of Aug 2016

Manufacturer	Part Number	SMD/ Lead	Frequency (kHz)	Load Capacitance CL (pF)	X1 oscillation Mode Note 1	Circuit Constants (Reference)			Voltage Range (V)	
						C3 (pF)	C4 (pF)	Rd (kΩ)	MIN.	MAX.
Seiko Instruments Inc. Note 2	SSP-T7-FL	SMD	32.768	6.0	Normal oscillation	10	9	0	1.6	3.6
				4.4	Low power consumption oscillation	7	5	0		
				3.7	Ultra-low power consumption oscillation	6	3	0		
	VT-200-FL	Lead		6.0	Normal oscillation	10	9	0		
				4.4	Low power consumption oscillation	7	5	0		
				3.7	Ultra-low power consumption oscillation	6	3	0		
Nihon Dempa Kogyo Co., Ltd. Note 3	NX3215SA	SMD	32.768	6.0	Normal oscillation	Note 3			1.6	3.6
					Low power consumption oscillation					
					Ultra-low power consumption oscillation					
RIVER ELETEC CORPORATION	TFX-02-32.768 kHz-J20986 Note 4	SMD	32.768	9	Normal oscillation	12	10	0	1.6	3.6
					Low power consumption oscillation					
	TFX-03-32.768 kHz-J13375 Note 4	SMD	32.768	9	Normal oscillation	12	10	0		

Note 1. Set the XT1 oscillation mode by using the AMPHS0 and AMPHS1 bits of the clock operation mode control register (CMC).

Note 2. When using these resonators, contact Seiko Instruments Inc., Ltd (http://www.sii-crystal.com) for more information on matching.

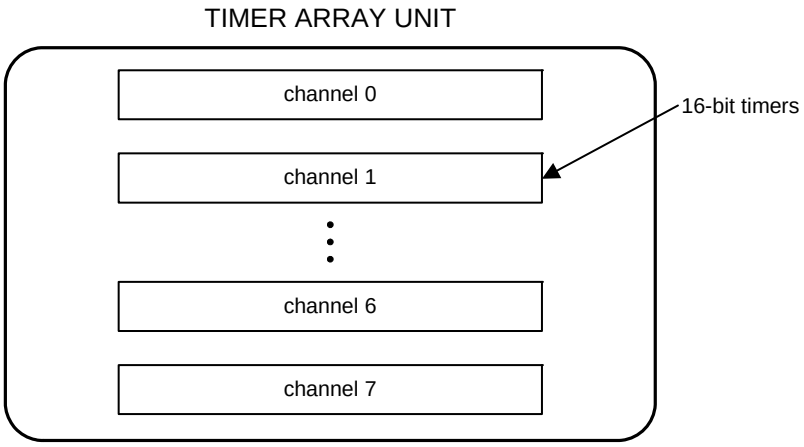
Note 3. When using these resonators, contact Nihon Dempa Kogyo Co., Ltd (http://www.ndk.com/en) for more information on matching.

Note 4. When using this resonator, for details about the matching, contact RIVER ELETEC CORPORATION (http://www.river-ele.co.jp/english/index.html).

CHAPTER 6 TIMER ARRAY UNIT

Caution Most of the following descriptions in this chapter use the 100-pin products as an example.

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 (→ refer to **6.7.2**) • Square wave output (→ refer to **6.7.2**) • External event counter (→ refer to **6.8.2**) • Input pulse interval measurement (→ refer to **6.8.3**) • Measurement of high-/low-level width of input signal (→ refer to **6.8.4**) • Delay counter (→ refer to **6.8.5**)	• One-shot pulse output (→ refer to **6.9.1**) • PWM output (→ refer to **6.9.2**) • Multiple PWM output (→ refer to **6.9.3**) • Remote control output function (→ refer to **6.9.4**)

It is possible to use the 16-bit timer of channels 1 and 3 as two 8-bit timers (higher and lower). The functions that can use channels 1 and 3 as 8-bit timers are as follows:

- Interval timer (upper or lower 8-bit timer)/square wave output (lower 8-bit timer only)
- External event counter (lower 8-bit timer only)
- Delay counter (lower 8-bit timer only)

Channel 7 can be used to realize LIN-bus communication operating in combination with UART0 of the serial array unit.

6.1 Functions of Timer Array Unit

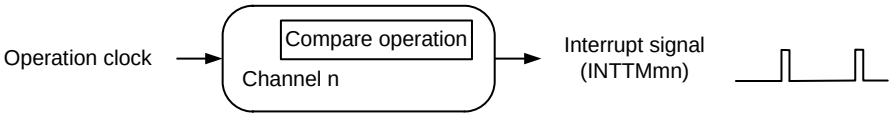
Timer array unit has the following functions.

6.1.1 Independent channel operation function

By operating a channel independently, it can be used for the following purposes without being affected by the operation mode of other channels.

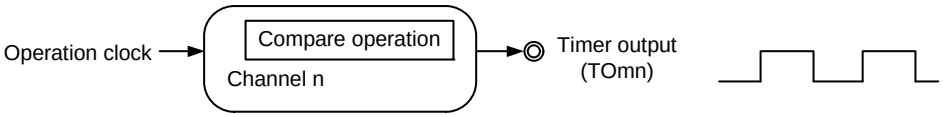
(1) Interval timer

Each timer of a unit can be used as a reference timer that generates an interrupt (INTTMmn) at fixed intervals.



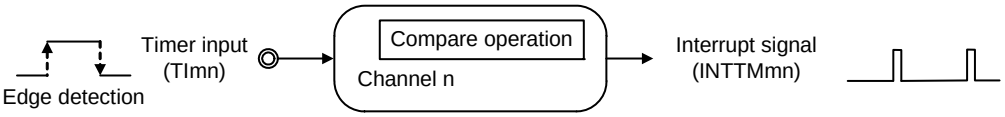
(2) Square wave output

A toggle operation is performed each time INTTMmn interrupt is generated and a square wave with a duty factor of 50% is output from a timer output pin (TOmn).



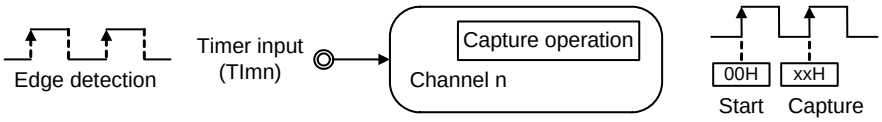
(3) External event counter

Each timer of a unit can be used as an event counter that generates an interrupt when the number of the valid edges of a signal input to the timer input pin (TImn) has reached a specific value.



(4) Input pulse interval measurement

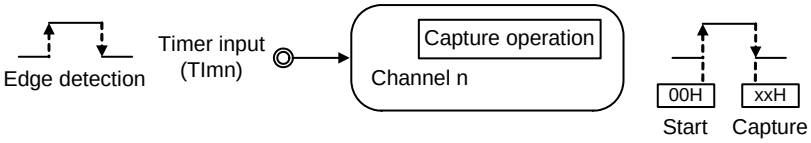
Counting is started by the valid edge of a pulse signal input to a timer input pin (TImn). The count value of the timer is captured at the valid edge of the next pulse. In this way, the interval of the input pulse can be measured.



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

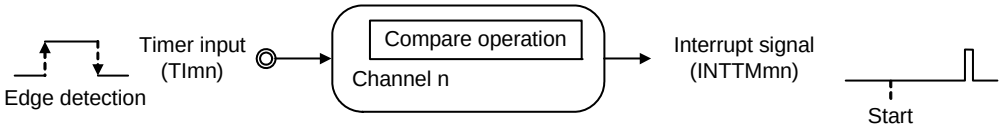
(5) Measurement of high-/low-level width of input signal

Counting is started by a single edge of the signal input to the timer input pin (TImn), and the count value is captured at the other edge. In this way, the high-level or low-level width of the input signal can be measured.



(6) Delay counter

Counting is started at the valid edge of the signal input to the timer input pin (TImn), and an interrupt is generated after any delay period.



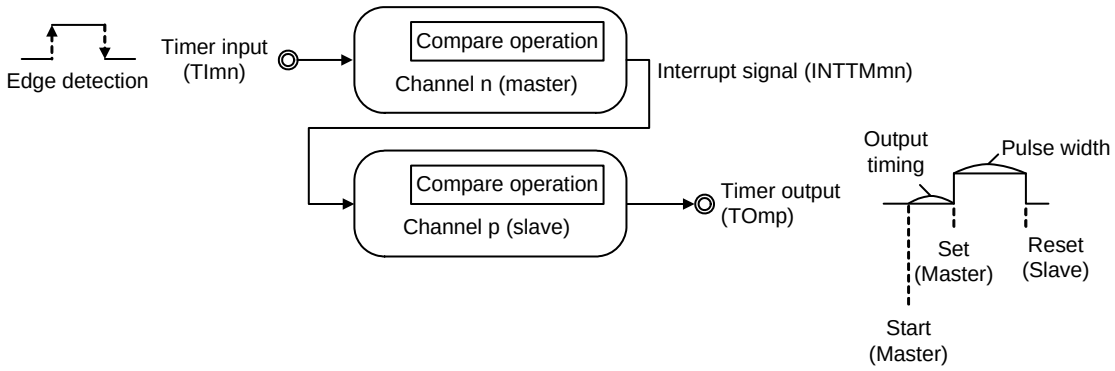
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.1.2 Simultaneous channel operation function

By using the combination of a master channel (a reference timer mainly controlling the cycle) and slave channels (timers operating according to the master channel), channels can be used for the following purposes.

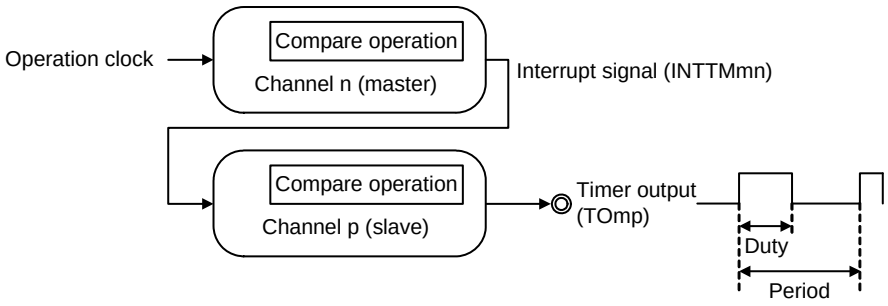
(1) One-shot pulse output

Two channels are used as a set to generate a one-shot pulse with a specified output timing and a specified pulse width.



(2) PWM (Pulse Width Modulation) output

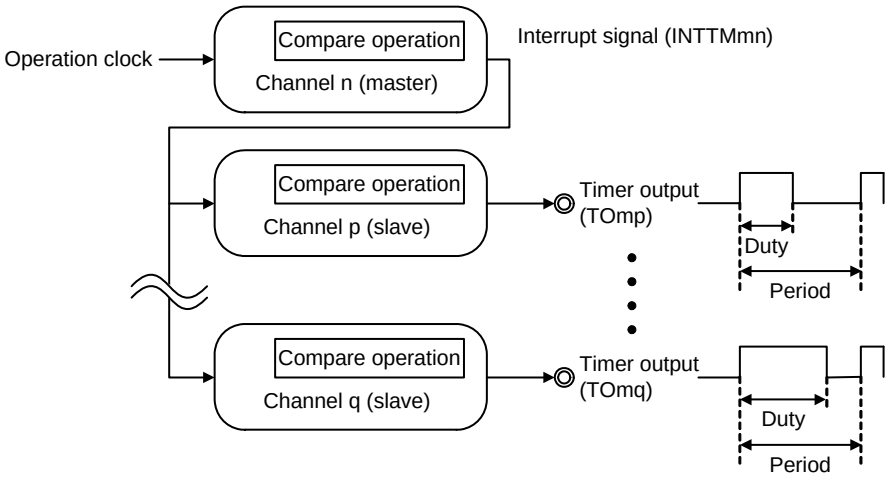
Two channels are used as a set to generate a pulse with a specified period and a specified duty factor.



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

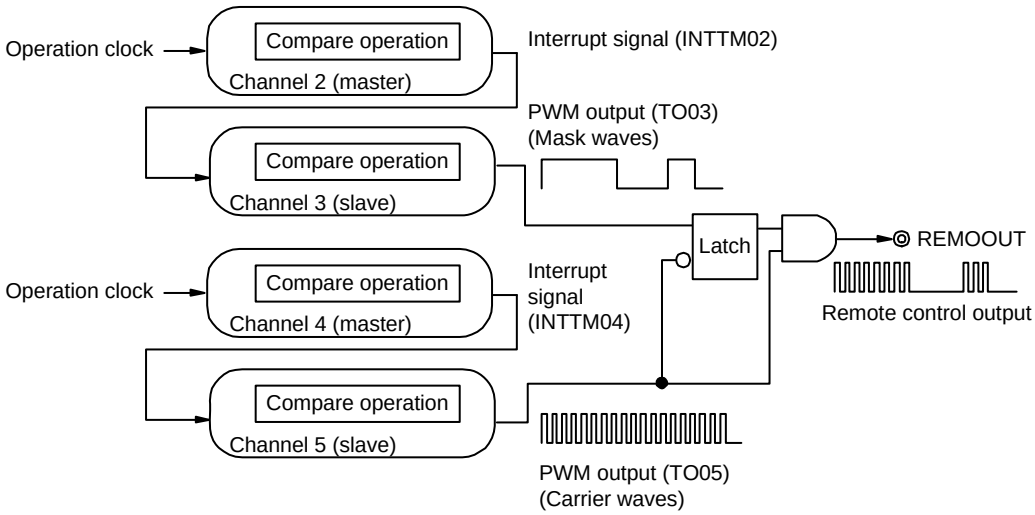
(3) Multiple PWM (Pulse Width Modulation) output

By extending the PWM function and using one master channel and two or more slave channels, up to seven types of PWM signals that have a specific period and a specified duty factor can be generated.



(4) Remote control output function

The pairings of channels 2 and 3 and channels 4 and 5 are used to output the PWM signal. The PWM signal output from channel 3 is used as a mask waves, the PWM signal output from channel 5 is used as a carrier waves, and the logical products of these signals are output as remote control output.



Caution For details about the rules of simultaneous channel operation function, see 6.4.1 Basic rules of simultaneous channel operation function.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7),
p, q: Slave channel number (n < p < q ≤ 7)

6.1.3 8-bit timer operation function (channels 1 and 3 only)

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels. This function can only be used for channels 1 and 3.

Caution There are several rules for using 8-bit timer operation function.
For details, see 6.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only).

6.1.4 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 sync 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 sync 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 **6.3.14 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 count register mn (TCRmn)
Register	Timer data register mn (TDRmn)
Timer input	TI00 to TI07, RxD0 pin (for LIN-bus)
Timer output	TO00 to TO07, output controller
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Timer clock select register m (TPSm) • Timer channel enable status register m (TEm) • Timer channel start register m (TSm) • Timer channel stop register m (TTm) • Timer input select register 0 (TIS0) • Timer output select register (TOS) • Timer output enable register m (TOEm) • Timer output register m (TOm) • Timer output level register m (TOLm) • Timer output mode register m (TOMm)
	<Registers of each channel> • Timer mode register mn (TMRmn) • Timer status register mn (TSRmn) • Input switch control register (ISC) • Noise filter enable register 1 (NFEN1) • Port mode control register (PMCxx) Note • Port mode register (PMxx) Note • Port register (Pxx) Note

Note The Port mode registers (PMxx) and port registers (Pxx) to be set differ depending on the product. For details, see **4.5.3 Register setting examples for used port and alternate functions**.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

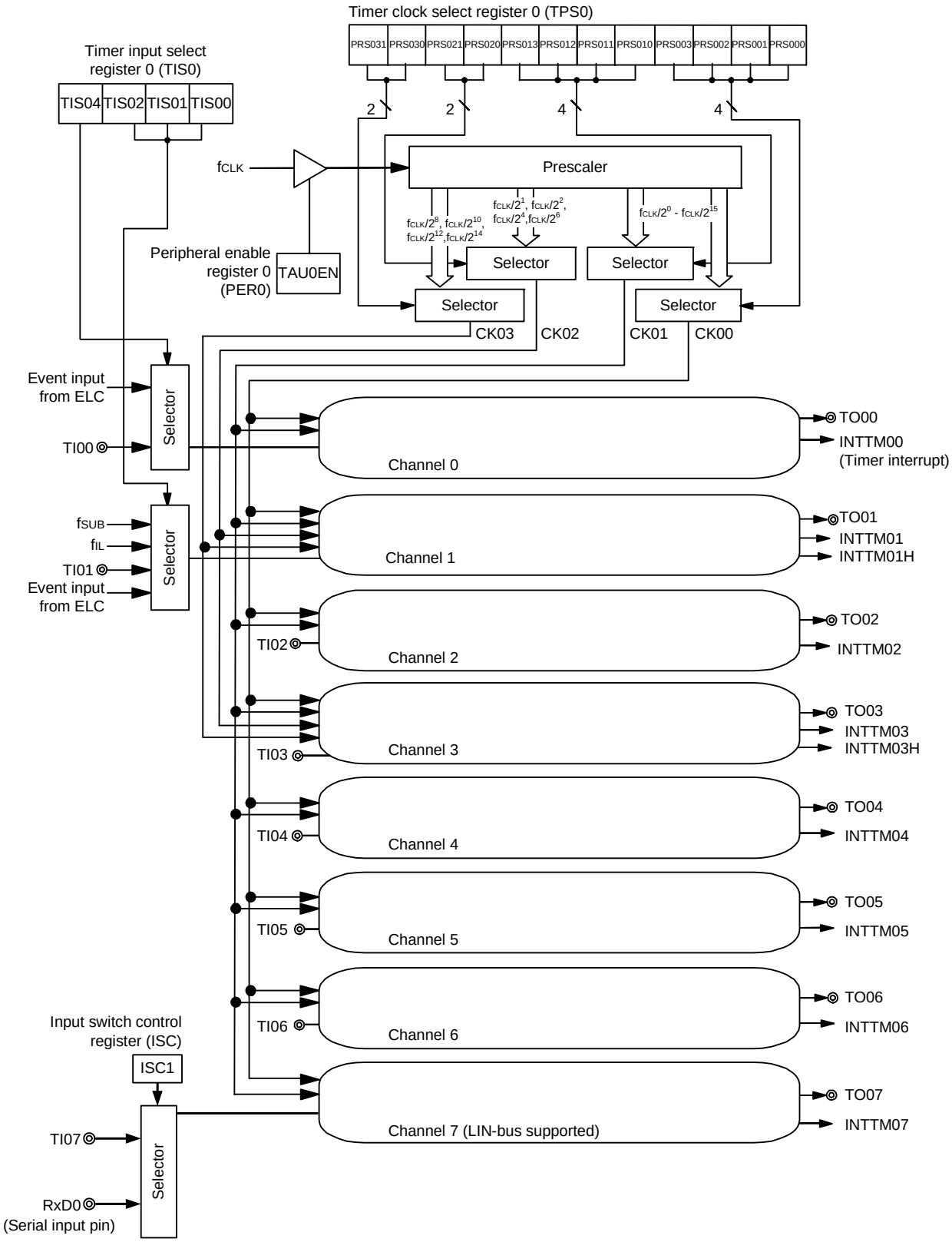
The port pins alternatively used as timer I/O pins in each timer array unit channel depend on the product.

Table 6 - 2 Timer I/O Pins provided in Each Product

Timer array unit channels	100-pin	80/85-pin
Channel 0	TI00/TO00	
Channel 1	TI01/TO01	
Channel 2	TI02/TO02	
Channel 3	TI03/TO03/REMOOUT	
Channel 4	TI04/TO04	
Channel 5	TI05/TO05	TI05/TO05
Channel 6	TI06/TO06	
Channel 7	TI07/TO07	TI07/TO07

Figure 6 - 1 shows the block diagram of the timer array unit.

Figure 6 - 1 Entire Configuration of Timer Array Unit 0



Remark fSUB: Subsystem clock frequency

fIL: Low-speed on-chip oscillator clock frequency

Figure 6 - 2 Internal Block Diagram of Channel 0 of Timer Array Unit

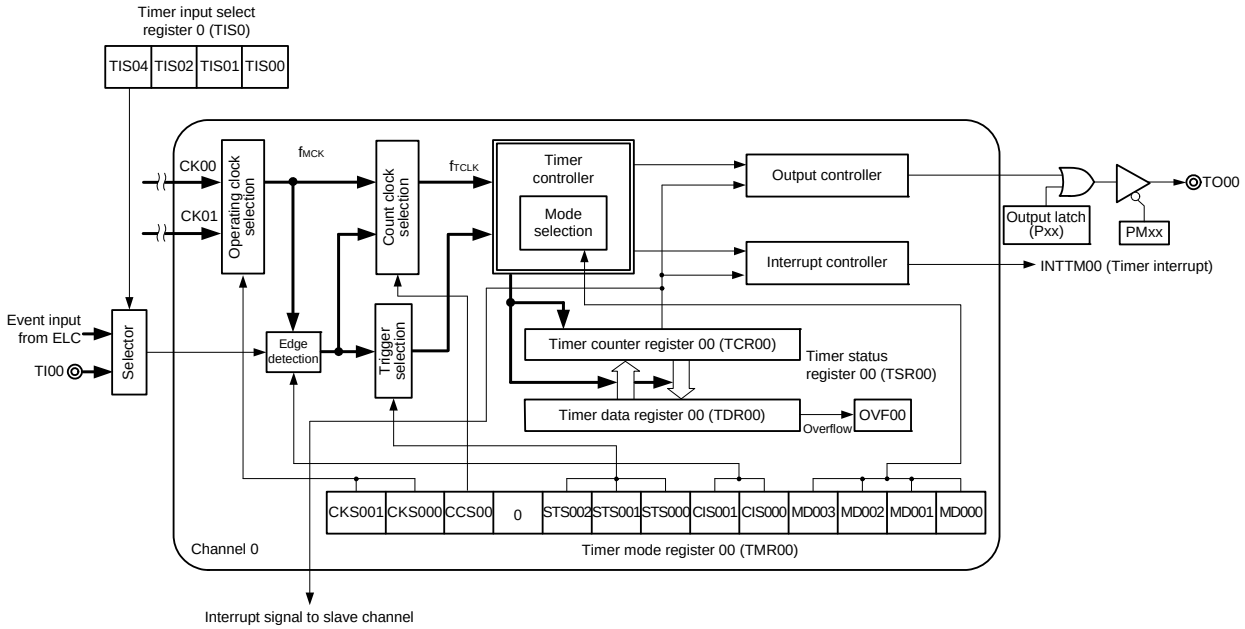


Figure 6 - 3 Internal Block Diagram of Channel 1 of Timer Array Unit

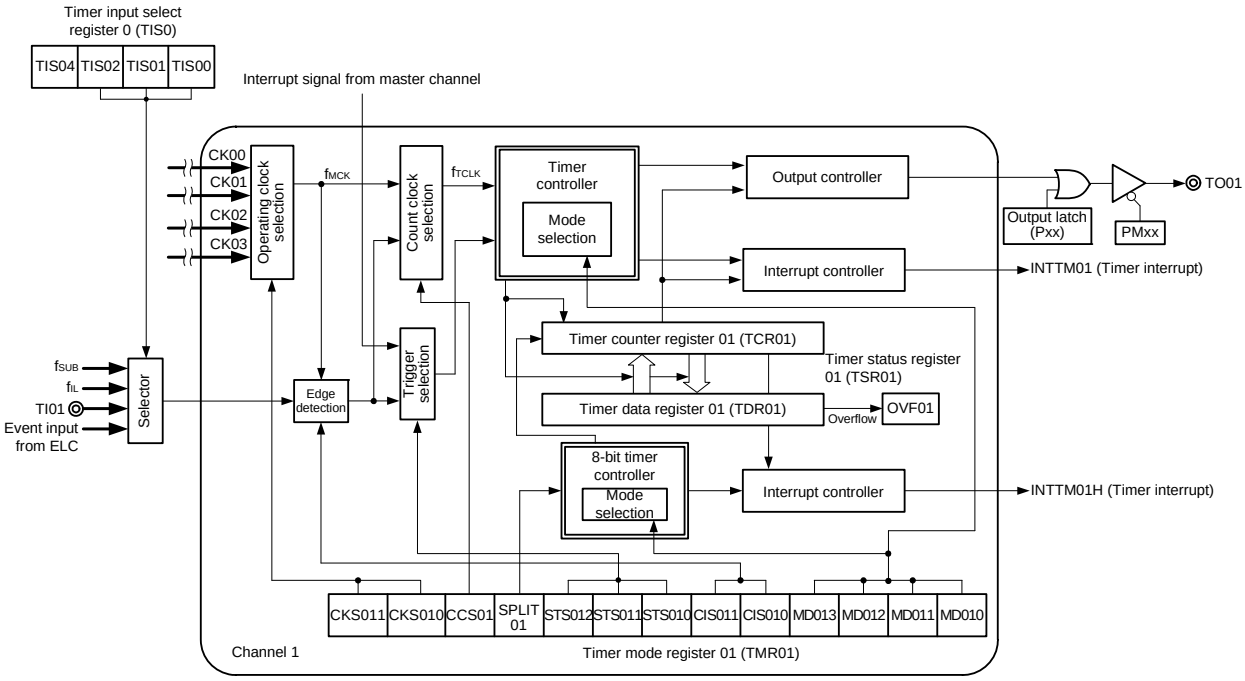
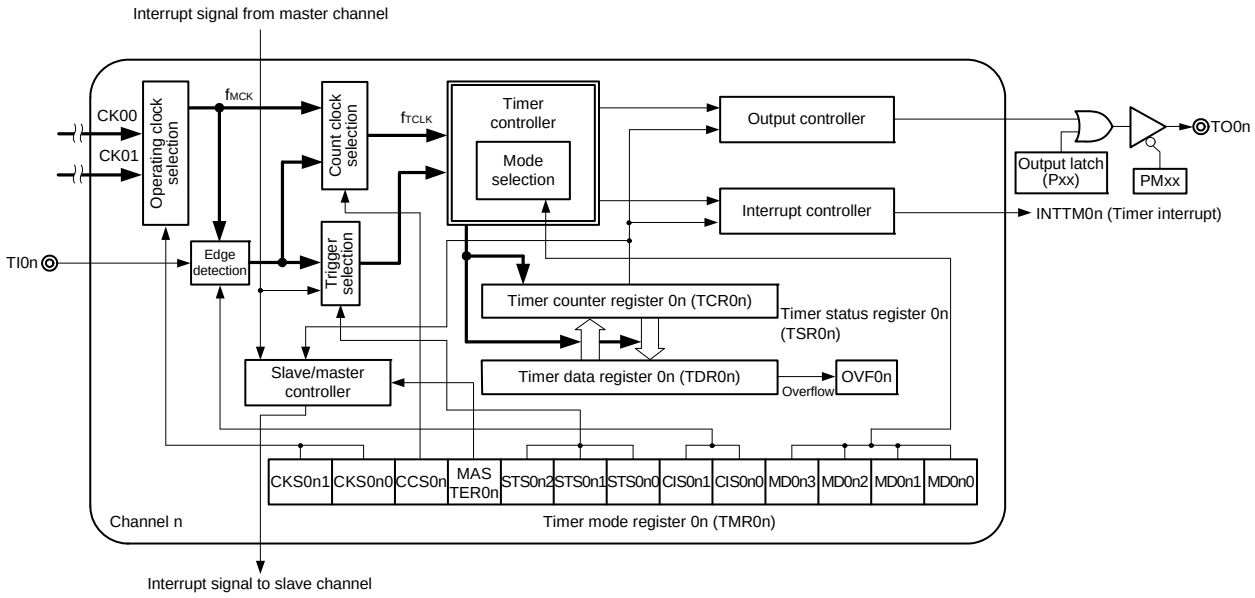


Figure 6 - 4 Internal Block Diagram of Channel n of Timer Array Unit



Remark n = 2, 4, 6

Figure 6 - 5 Internal Block Diagram of Channel 3 of Timer Array Unit

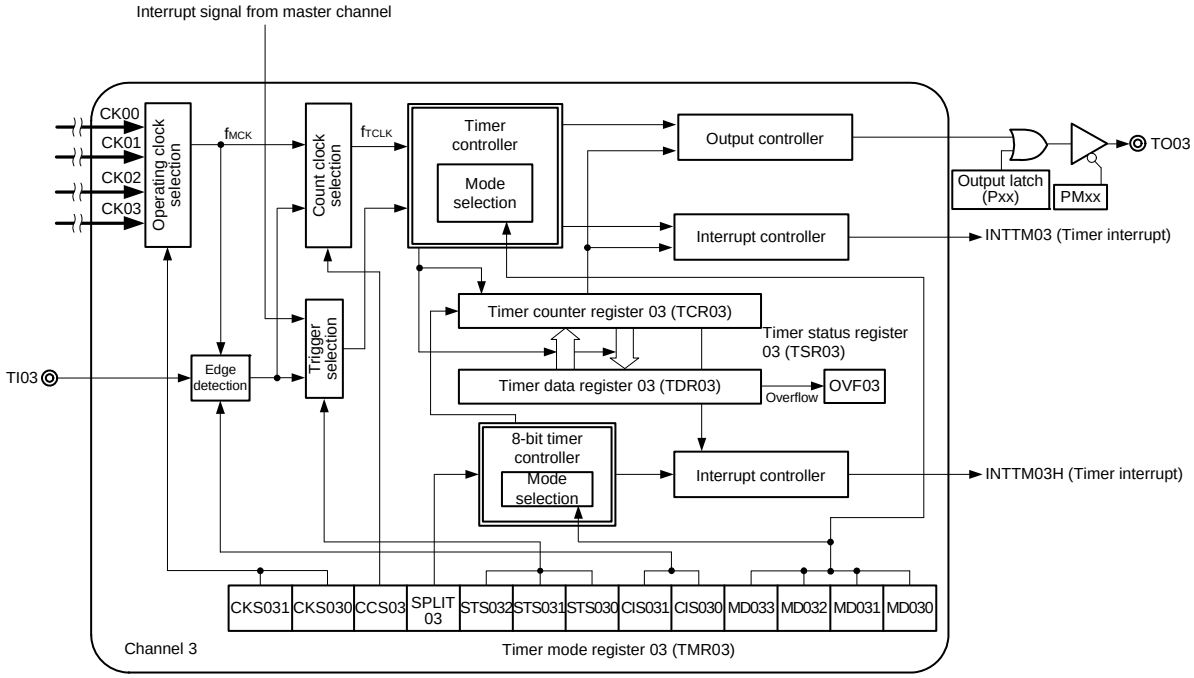


Figure 6 - 6 Internal Block Diagram of Channel 5 of Timer Array Unit

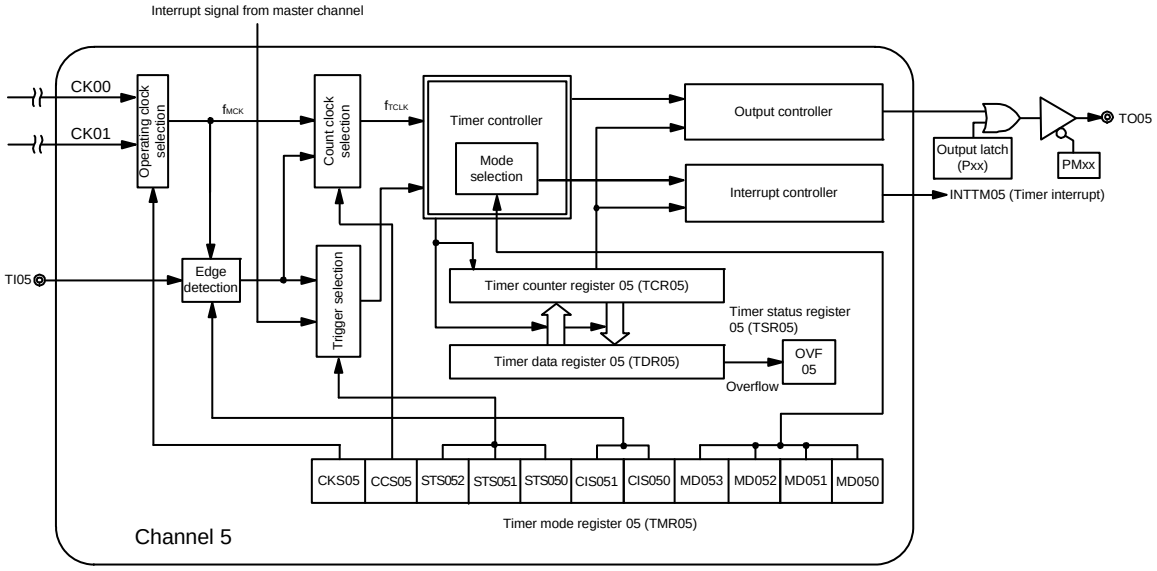
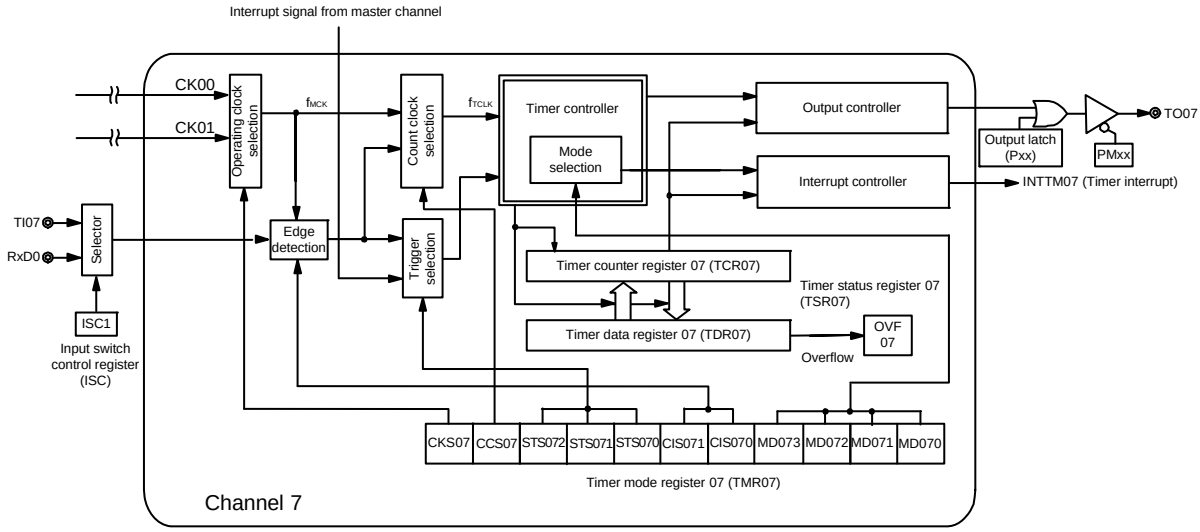


Figure 6 - 7 Internal Block Diagram of Channel 7 of Timer Array Unit



6.2.1 Timer count register mn (TCRmn)

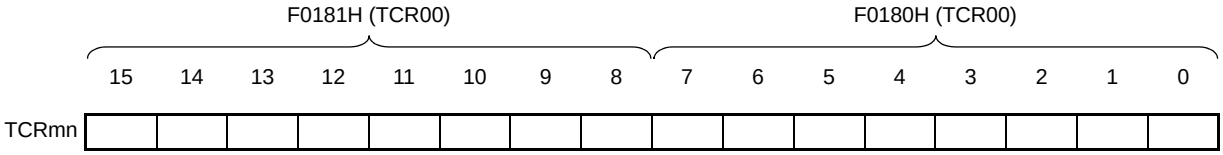
The TCRmn register is a 16-bit read-only register and is used to count clocks.

The value of this counter is incremented or decremented in synchronization with the rising edge of a count clock.

Whether the counter is incremented or decremented depends on the operation mode that is selected by the MDmn3 to MDmn0 bits of timer mode register mn (TMRmn) (refer to **6.3.3 Timer mode register mn (TMRmn)**).

Figure 6 - 8 Format of Timer count register mn (TCRmn)

Address: F0180H, F0181H (TCR00) to F018EH, F018FH (TCR07) After reset: FFFFH R



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

The count value can be read by reading timer count register mn (TCRmn).

The count value is set to FFFFH in the following cases.

- When the reset signal is generated
- When the TAUmEN bit of peripheral enable register 0 (PER0) is cleared
- When counting of the slave channel has been completed in the PWM output mode
- When counting of the slave channel has been completed in the delay count mode
- When counting of the master/slave channel has been completed in the one-shot pulse output mode
- When counting of the slave channel has been completed in the multiple PWM output mode
- The count value is cleared to 0000H in the following cases.
- When the start trigger is input in the capture mode
- When capturing has been completed in the capture mode

Caution The count value is not captured to timer data register mn (TDRmn) even when the TCRmn register is read.

The TCRmn register read value differs as follows according to operation mode changes and the operating status.

Table 6 - 3 Timer Count Register mn (TCRmn) Read Value in Various Operation Modes

Operation Mode	Count Mode	Timer count register mn (TCRmn) Read Value Note			
		Value if the operation mode was changed after releasing reset	Value if the operation was restarted after count operation paused (TTmn = 1)	Value if the operation mode was changed after count operation paused (TTmn = 1)	Value when waiting for a start trigger after one count
Interval timer mode	Count down	FFFFH	Value if stop	Undefined	—
Capture mode	Count up	0000H	Value if stop	Undefined	—
Event counter mode	Count down	FFFFH	Value if stop	Undefined	—
One-count mode	Count down	FFFFH	Value if stop	Undefined	FFFFH
Capture & one-count mode	Count up	0000H	Value if stop	Undefined	Capture value of TDRmn register + 1

Note This indicates the value read from the TCRmn register when channel n has stopped operating as a timer (TEmn = 0) and has been enabled to operate as a counter (TSmn = 1). The read value is held in the TCRmn register until the count operation starts.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

6.2.2 Timer data register mn (TDRmn)

This is a 16-bit register from which a capture function and a compare function can be selected.

The capture or compare function can be switched by selecting an operation mode by using the MDmn3 to MDmn0 bits of timer mode register mn (TMRmn).

The value of the TDRmn register can be changed at any time.

This register can be read or written in 16-bit units.

In addition, for the TDRm1 and TDRm3 registers, while in the 8-bit timer mode (when the SPLIT bits of timer mode registers 01 and 03 (TMRm1, TMRm3) are 1), it is possible to rewrite the data in 8-bit units, with TDRm1H and TDRm3H used as the higher 8 bits, and TDRm1L and TDRm3L used as the lower 8 bits. However, reading is only possible in 16-bit units.

Reset signal generation clears this register to 0000H.

Figure 6 - 9 Format of Timer data register mn (TDRmn) (n = 0, 2, 4 to 7)

Address: FFF18H, FFF19H (TDR00), FFF64H, FFF65H (TDR02) After reset: 0000H R/W
FFF68H, FFF69H (TDR04) to FFF6EH, FFF6FH (TDR07)

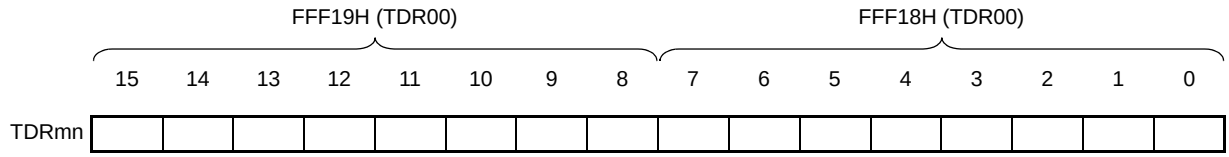
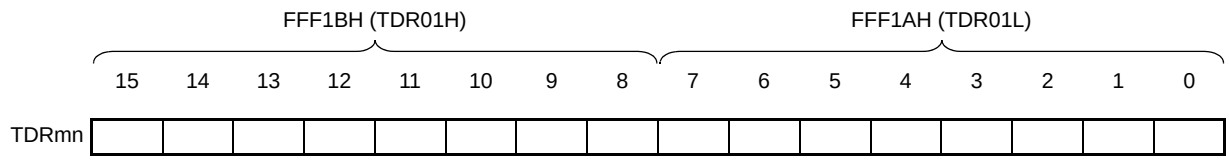


Figure 6 - 10 Format of Timer data register mn (TDRmn) (n = 1, 3)

Address: FFF1AH, FFF1BH (TDR01), FFF66H, FFF67H (TDR03) After reset: 00H R/W



(i) When timer data register mn (TDRmn) is used as compare register

Counting down is started from the value set to the TDRmn register. When the count value reaches 0000H, an interrupt signal (INTTMmn) is generated. The TDRmn register holds its value until it is rewritten.

Caution The TDRmn register does not perform a capture operation even if a capture trigger is input, when it is set to the compare function.

(ii) When timer data register mn (TDRmn) is used as capture register

The count value of timer count register mn (TCRmn) is captured to the TDRmn register when the capture trigger is input.

A valid edge of the TImn pin can be selected as the capture trigger. This selection is made by timer mode register mn (TMRmn).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 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 m (TPSm)
- Timer mode register mn (TMRmn)
- Timer status register mn (TSRmn)
- Timer channel enable status register m (TEm)
- Timer channel start register m (TSm)
- Timer channel stop register m (TTm)
- Timer input select register 0 (TIS0)
- Timer output select register (TOS)
- Timer output enable register m (TOEm)
- Timer output register m (TOm)
- Timer output level register m (TOLm)
- Timer output mode register m (TOMm)
- Input switch control register (ISC)
- Noise filter enable register 1 (NFEN1)
- Port mode control register (PMCxx)
- Port mode register (PMxx)
- Port register (Pxx)

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

Remark m: Unit number (m = 0, 1), 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 0 is used, be sure to set bit 0 (TAU0EN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Figure 6 - 11 Format of Peripheral enable register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	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.

Caution 1. When setting the timer array unit, be sure to set the following registers first while the TAUmEN bit is set to 1. If TAUmEN = 0, 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 (TIS0), timer output select register (TOS), input switch control register (ISC), noise filter enable register 1 (NFEN1), port mode control register 4 (PMC4), port mode registers 0, 2 to 4 (PM0, PM2 to PM14), and port registers 0, 2 to 4 (P0, P2 to P4)).

- **Timer clock select register m (TPSm)**
- **Timer mode register mn (TMRmn)**
- **Timer status register mn (TSRmn)**
- **Timer channel enable status register m (TEm)**
- **Timer channel start register m (TSm)**
- **Timer channel stop register m (TTm)**
- **Timer output enable register m (TOEm)**
- **Timer output register m (TOm)**
- **Timer output level register m (TOLm)**
- **Timer output mode register m (TOMm)**

Caution 2. Be sure to clear bits 1 and 6 to 0.

6.3.2 Timer clock select register m (TPSm)

The TPSm register is a 16-bit register that is used to select two types or four types of operation clocks (CKm0, CKm1, CKm2, CKm3) that are commonly supplied to each channel. CKm0 is selected by using bits 3 to 0 of the TPSm register, and CKm1 is selected by using bits 7 to 4 of the TPSm register. In addition, only for channels 1 and 3, CKm2 and CKm3 can be also selected. CKm2 is selected by using bits 9 and 8 of the TPSm register, and CKm3 is selected by using bits 13 and 12 of the TPSm register.
Rewriting of the TPSm register during timer operation is possible only in the following cases.

If the PRSm00 to PRSm03 bits can be rewritten (n = 0 to 7):
All channels for which CKm0 is selected as the operation clock (CKSmn1, CKSmn0 = 0, 0) are stopped (TEmn = 0).
If the PRSm10 to PRSm13 bits can be rewritten (n = 0 to 7):
All channels for which CKm2 is selected as the operation clock (CKSmn1, CKSmn0 = 0, 1) are stopped (TEmn = 0).
If the PRSm20 and PRSm21 bits can be rewritten (n = 1, 3):
All channels for which CKm1 is selected as the operation clock (CKSmn1, CKSmn0 = 1, 0) are stopped (TEmn = 0).
If the PRSm30 and PRSm31 bits can be rewritten (n = 1, 3):
All channels for which CKm3 is selected as the operation clock (CKSmn1, CKSmn0 = 1, 1) are stopped (TEmn = 0).

The TPSm register can be set by a 16-bit memory manipulation instruction.
Reset signal generation clears this register to 0000H.

Figure 6 - 12 Format of Timer clock select register m (TPSm) (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
TPSm	0	0	PRSm 31	PRSm 30	0	0	PRSm 21	PRSm 20	PRSm 13	PRSm 12	PRSm 11	PRSm 10	PRSm 03	PRSm 02	PRSm 01	PRSm 00

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Selection of operation clock (CKmk) Note (k = 0, 1)					
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz	24 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	12 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	6 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	3 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz	1.5 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz	750 kHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz	375 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	187.5 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	93.8 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	46.9 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	23.4 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	11.7 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	5.86 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz	2.93 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.46 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61.0 Hz	153 Hz	305 Hz	610 Hz	732 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit (TTm = 00FFH).

Caution 1. Be sure to clear bits 15, 14, 11, and 10 to "0".

Caution 2. If f_{CLK} (undivided) is selected as the operation clock (CKmk) and TDRnm is set to 0000H (n = 0, m = 0 to 7), interrupt requests output from timer array units cannot be used.

Remark 1. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 2. The above $f_{CLK}/2^r$ is not a signal which is simply divided f_{CLK} by 2r, but a signal which becomes high level for one period of f_{CLK} from its rising edge (r = 1 to 15). For details, see **6.5.1 Count clock (f_{TCLK})**.

Figure 6 - 13 Format of Timer clock select register m (TPSm) (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
TPSm	0	0	PRSm31	PRSm30	0	0	PRSm21	PRSm20	PRSm13	PRSm12	PRSm11	PRSm10	PRSm03	PRSm02	PRSm01	PRSm00

PRS m21	PRS m20	Selection of operation clock (CKm2) Note					
			f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz
0	0	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	12 MHz
0	1	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	6 MHz
1	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz	1.5 MHz
1	1	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156.2 kHz	313 kHz	375 kHz

PRS m31	PRS m30	Selection of operation clock (CKm3) Note					
			f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz
0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	93.8 kHz
0	1	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	23.4 kHz
1	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	5.86 kHz
1	1	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.46 kHz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop the timer array unit (TTm = 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 TImn pin is selected.

Caution Be sure to clear bits 15, 14, 11, and 10 to "0".

By using channels 1 and 3 in the 8-bit timer mode and specifying CKm2 or CKm3 as the operation clock, the interval times shown in Table 6 - 4 can be achieved by using the interval timer function.

Table 6 - 4 Interval Times Available for Operation Clock CKSm2 or CKSm3

Clock		Interval time Note (f_{CLK} = 20 MHz)			
		16 μs	160 μs	1.6 ms	16 ms
CKm2	$f_{CLK}/2$	√	—	—	—
	$f_{CLK}/2^2$	√	—	—	—
	$f_{CLK}/2^4$	√	√	—	—
	$f_{CLK}/2^6$	√	√	—	—
CKm3	$f_{CLK}/2^8$	—	√	√	—
	$f_{CLK}/2^{10}$	—	√	√	—
	$f_{CLK}/2^{12}$	—	—	√	√
	$f_{CLK}/2^{14}$	—	—	√	√

Note The margin is within 5%.

Remark 1. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 2. For details of a signal of $f_{CLK}/2^i$ selected with the TPSm register, see **6.5.1 Count clock (f_{TCLK})**.

6.3.3 Timer mode register mn (TMRmn)

The TMRmn register sets an operation mode of channel n. This register is used to select the operation clock (f_{MCK}), select the count clock, select the master/slave, select the 16 or 8-bit timer (only for channels 1 and 3), specify the start trigger and capture trigger, select the valid edge of the timer input, and specify the operation mode (interval, capture, event counter, one-count, or capture and one-count).

Rewriting the TMRmn register is prohibited when the register is in operation (when TEmn = 1). However, bits 7 and 6 (CISmn1, CISmn0) can be rewritten even while the register is operating with some functions (when TEmn = 1) (for details, see **6.7 Timer Input (TImn) Control** and **6.9 Simultaneous Channel Operation Function of Timer Array Unit**).

The TMRmn register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Caution The bits mounted depend on the channels in bit 11 of TMRmn register.
TMRm2, TMRm4, TMRm6: MASTERmn bit (n = 2, 4, 6)
TMRm1, TMRm3: SPLITmn bit (n = 1, 3)
TMRm0, TMRm5, TMRm7: Fixed to 0

Figure 6 - 14 Format of Timer mode register mn (TMRmn) (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
TMRmn (n = 2, 4, 6)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKSmn1	CKSmn0	0	CCSmn	SPLITmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0, 5, 7)	CKSmn1	CKSmn0	0	CCSmn	0 Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

CKSmn1	CKSmn0	Selection of operation clock (f_{MCK}) of channel n
0	0	Operation clock CKm0 set by timer clock select register m (TPSm)
0	1	Operation clock CKm2 set by timer clock select register m (TPSm)
1	0	Operation clock CKm1 set by timer clock select register m (TPSm)
1	1	Operation clock CKm3 set by timer clock select register m (TPSm)
Operation clock (f_{MCK}) is used by the edge detector. A count clock (f_{TCLK}) and a sampling clock are generated depending on the setting of the CCSmn bit. The operation clocks CKm2 and CKm3 can only be selected for channels 1 and 3.		

CCSmn	Selection of count clock (f_{TCLK}) of channel n
0	Operation clock (f_{MCK}) specified by the CKSmn0 and CKSmn1 bits
1	Valid edge of input signal input from the TImn pin In channel 1, Valid edge of input signal selected by TIS0
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.

Caution 1. Be sure to clear bits 13, 5, and 4 to "0".

Caution 2. The timer array unit must be stopped (TTm = 00FFH) if the clock selected for f_{CLK} is changed (by changing the value of the system clock control register (CKC)), even if the operating clock specified by using the CKSmn0 and CKSmn1 bits (f_{MCK}) or the valid edge of the signal input from the TImn pin is selected as the count clock (f_{TCLK}).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

Figure 6 - 15 Format of Timer mode register mn (TMRmn) (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
TMRmn (n = 2, 4, 6)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKSmn1	CKSmn0	0	CCSmn	SPLITmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0, 5, 7)	CKSmn1	CKSmn0	0	CCSmn	0 Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

(Bit 11 of TMRmn (n = 2, 4, 6))

MASTERmn	Selection between using channel n independently or simultaneously with another channel (as a slave or master)
0	Operates in independent channel operation function or as slave channel in simultaneous channel operation function.
1	Operates as master channel in simultaneous channel operation function.
Only channel 2, 4, 6 can be set as a master channel (MASTERmn = 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 MASTERmn bit to 0 for a channel that is used with the independent channel operation function.	

(Bit 11 of TMRmn (n = 1, 3))

SPLITmn	Selection of 8 or 16-bit timer operation for channels 1 and 3
0	Operates as 16-bit timer. (Operates in independent channel operation function or as slave channel in simultaneous channel operation function.)
1	Operates as 8-bit timer.

STSmn2	STSmn1	STSmn0	Setting of start trigger or capture trigger of channel n
0	0	0	Only software trigger start is valid (other trigger sources are unselected).
0	0	1	Valid edge of the TImn pin input is used as both the start trigger and capture trigger.
0	1	0	Both the edges of the TImn pin input are used as a start trigger and a capture trigger.
1	0	0	Interrupt signal of the master channel is used (when the channel is used as a slave channel with the simultaneous channel operation function).
Other than above			Setting prohibited

Note Bit 11 is fixed at 0 of read only, write is ignored.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

Figure 6 - 16 Format of Timer mode register mn (TMRmn) (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
TMRmn (n = 2, 4, 6)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKSmn1	CKSmn0	0	CCSmn	SPLITmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0, 5, 7)	CKSmn1	CKSmn0	0	CCSmn	0 Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

(When the input source is other than an event input signal from the ELC by setting the TIS0 register)

CISmn1	CISmn0	Selection of TImn pin input valid edge (n = 0, 1)
0	0	Falling edge
0	1	Rising edge
1	0	Both edges (when low-level width is measured) Start trigger: Falling edge, Capture trigger: Rising edge
1	1	Both edges (when high-level width is measured) Start trigger: Rising edge, Capture trigger: Falling edge
If both the edges are specified when the value of the STSmn2 to STSmn0 bits is other than 010B, set the CISmn1 to CISmn0 bits to 10B.		

When the input source is an event input signal from the ELC by setting the TIS0 register.

CISmn1	CISmn0	Selection of TImn pin input valid edge (n = 0, 1)
0	0	Set to 00 (event input signal from the ELC).
Other than above		Setting prohibited

Note Bit 11 is fixed at 0 of read only, write is ignored.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

Figure 6 - 17 Format of Timer mode register mn (TMRmn) (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
TMRmn (n = 2, 4, 6)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKSmn1	CKSmn0	0	CCSmn	SPLITmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0, 5, 7)	CKSmn1	CKSmn0	0	CCSmn	0 Note 1	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

MD mn3	MD mn2	MD mn1	Operation mode of channel n	Corresponding function	Count operation of TCR
0	0	0	Interval timer mode	Interval timer / Square wave output / 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 MDmn0 bit (see the table below).					

Operation mode (Value set by the MDmn3 to MDmn1 bits (see the table above))	MDmn0	Setting of starting counting and interrupt
• Interval timer mode (0, 0, 0) • Capture mode (0, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
	1	Timer interrupt is generated when counting is started (timer output also changes).
• Event counter mode (0, 1, 1)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• One-count mode Note 2 (1, 0, 0)	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated, either.
	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, either.

Note 1. Bit 11 is fixed at 0 of read only, write is ignored.

Note 2. In one-count mode, interrupt output (INTTMmn) when starting a count operation and TOmn output are not controlled.

Note 3. If the start trigger (TSmn = 1) is issued during operation, the counter is initialized, an interrupt is generated, and recounting is started (does not occur the interrupt request).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

6.3.4 Timer status register mn (TSRmn)

The TSRmn register indicates the overflow status of the counter of channel n.

The TSRmn register is valid only in the capture mode (MDmn3 to MDmn1 = 010B) and capture & one-count mode (MDmn3 to MDmn1 = 110B). See **Table 6 - 5** for the operation of the OVF bit in each operation mode and set/clear conditions.

The TSRmn register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TSRmn register can be set with an 8-bit memory manipulation instruction with TSRmnL.

Reset signal generation clears this register to 0000H.

Figure 6 - 18 Format of Timer status register mn (TSRmn)

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
TSRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OVF

OVF	Counter overflow status of channel n
0	Overflow does not occur.
1	Overflow occurs.
When OVF = 1, this flag is cleared (OVF = 0) when the next value is captured without overflow.	

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 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 m (TEm)

The TEm register is used to enable or stop the timer operation of each channel.

Each bit of the TEm register corresponds to each bit of the timer channel start register m (TSm) and the timer channel stop register m (TTm). When a bit of the TSm register is set to 1, the corresponding bit of this register is set to 1. When a bit of the TTm register is set to 1, the corresponding bit of this register is cleared to 0.

The TEm register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TEm register can be set with a 1-bit or 8-bit memory manipulation instruction with TEmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 19 Format of Timer channel enable status register m (TEm)

Address: F01B0H, F01B1H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEm	0	0	0	0	TEHm3	0	TEHm1	0	TEm7	TEm6	TEm5	TEm4	TEm3	TEm2	TEm1	TEm0

TEHm3	Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 3 is in the 8-bit timer mode
0	Operation is stopped.
1	Operation is enabled.

TEHm1	Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 1 is in the 8-bit timer mode
0	Operation is stopped.
1	Operation is enabled.

TEmn	Indication of operation enable/stop status of channel n
0	Operation is stopped.
1	Operation is enabled.
This bit displays whether operation of the lower 8-bit timer for TEm1 and TEm3 is enabled or stopped when channel 1 or 3 is in the 8-bit timer mode.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

6.3.6 Timer channel start register m (TSm)

The TSm register is a trigger register that is used to initialize timer count register mn (TCRmn) and start the counting operation of each channel.

When a bit of this register is set to 1, the corresponding bit of timer channel enable status register m (TEm) is set to 1. The TSmn, TSHm1, TSHm3 bits are immediately cleared when operation is enabled (TEmn, TEHm1, TEHm3 = 1), because they are trigger bits.

The TSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TSm register can be set with a 1-bit or 8-bit memory manipulation instruction with TSmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 20 Format of Timer channel start register m (TSm)

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
TSm	0	0	0	0	TSHm3	0	TSHm1	0	TSm7	TSm6	TSm5	TSm4	TSm3	TSm2	TSm1	TSm0

TSHm3	Trigger to enable operation (start operation) of the higher 8-bit timer when channel 3 is in the 8-bit timer mode
0	No trigger operation
1	The TEHm3 bit is set to 1 and the count operation becomes enabled. The TCRm3 register count operation start in the interval timer mode in the count operation enabled state (see **Table 6 - 6** in **6.5.2 Start timing of counter**).

TSHm1	Trigger to enable operation (start operation) of the higher 8-bit timer when channel 1 is in the 8-bit timer mode
0	No trigger operation
1	The TEHm1 bit is set to 1 and the count operation becomes enabled. The TCRm1 register count operation start in the interval timer mode in the count operation enabled state (see **Table 6 - 6** in **6.5.2 Start timing of counter**).

TSmn	Operation enable (start) trigger of channel n
0	No trigger operation
1	The TEmn bit is set to 1 and the count operation becomes enabled. The TCRmn register count operation start in the count operation enabled state varies depending on each operation mode (see **Table 6 - 6** in **6.5.2 Start timing of counter**). This bit is the trigger to enable operation (start operation) of the lower 8-bit timer for TSm1 and TSm3 when channel 1 or 3 is in the 8-bit timer mode.

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

Caution 1. Be sure to clear bits 15 to 12, 10, and 8 to “0”.

Caution 2. When switching from a function that does not use TImn pin input to one that does, the following wait period is required from when timer mode register mn (TMRmn) is set until the TSmn (TSHm1, TSHm3) bit is set to 1.
When the TImn pin noise filter is enabled (TNFENmn = 1): Four cycles of the operation clock (f_{MCK})
When the TImn pin noise filter is disabled (TNFENmn = 0): Two cycles of the operation clock (f_{MCK})

Remark 1. When the TSm register is read, 0 is always read.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.3.7 Timer channel stop register m (TTm)

The TTm register is a trigger register that is used to stop the counting operation of each channel.

When a bit of this register is set to 1, the corresponding bit of timer channel enable status register m (TEm) is cleared to 0. The TTmn, TTHm1, TTHm3 bits are immediately cleared when operation is stopped (TEmn, TEHm1, TEHm3 = 0), because they are trigger bits.

The TTm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TTm register can be set with a 1-bit or 8-bit memory manipulation instruction with TTmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 21 Format of Timer channel stop register m (TTm)

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
TTm	0	0	0	0	TTHm3	0	TTHm1	0	TTm7	TTm6	TTm5	TTm4	TTm3	TTm2	TTm1	TTm0

TTHm3	Trigger to stop operation of the higher 8-bit timer when channel 3 is in the 8-bit timer mode
0	No trigger operation
1	TEHm3 bit is cleared to 0 and the count operation is stopped.

TTHm1	Trigger to stop operation of the higher 8-bit timer when channel 1 is in the 8-bit timer mode
0	No trigger operation
1	TEHm1 bit is cleared to 0 and the count operation is stopped.

TTmn	Operation stop trigger of channel n
0	No trigger operation
1	TEmn bit is cleared to 0 and the count operation is stopped. This bit is the trigger to stop operation of the lower 8-bit timer for TTm1 and TTm3 when channel 1 or 3 is in the 8-bit timer mode.

Caution Be sure to clear bits 15 to 12, 10, and 8 of the TTm register to “0”.

Remark 1. When the TTm register is read, 0 is always read.

Remark 2. m: Unit number (m = 0, 1), 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 0 and 1 timer input.

The TIS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6 - 22 Format of Timer input select register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	TIS04	0	TIS02	TIS01	TIS00

TIS04	Selection of timer input used with channel 0
0	Input signal of timer input pin (TI00)
1	Event input signal from ELC

TIS02	TIS01	TIS00	Selection of timer input used with channel 1
0	0	0	Input signal of timer input pin (TI01)
0	1	0	
0	1	1	
0	0	1	Event input signal from ELC
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
1	0	1	Subsystem clock (f_{SUB})
Other than above			Setting prohibited

Caution 1. At least $1/f_{MCK}$ + 10 ns is necessary as the high-level and low-level widths of the timer input to be selected. Thus, the TIS02 bit cannot be set to 1 when f_{SUB} is selected as f_{CLK} (CSS in CKC register = 1).

Caution 2. When selecting an event input signal from the ELC using timer input select register 0 (TIS0), select f_{CLK} using timer clock select register 0 (TPS0).

6.3.9 Timer output select register (TOS)

The TOS register is used to enable the remote control output function.

Remote control output are generated by using the PWM output signal generated by channels 2 and 3 (mask waveform) to mask the PWM output signal generated by channels 4 and 5 (carrier waveform).

Rewriting the TOS register is only possible before counting starts (TE02, TE03, TE04, TE05 = 0).

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

Reset signal generation clears this register to 00H.

Figure 6 - 23 Format of Timer output select register (TOS)

Address: F0079H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
TOS	0	0	0	0	0	0	0	TOS0

TOS0	Remote control output setting
0	Disable (channels 2, 3, 4, and 5 is used for timer output)
1	Enable (remote control output to the REMOOUT pin)

Caution Channels 2, 3, 4, and 5 cannot be used for any other function when remote control output is enabled (TOS0 = 1).

6.3.10 Timer output enable register m (TOEm)

The TOEm register is used to enable or disable timer output of each channel.

Channel n for which timer output has been enabled becomes unable to rewrite the value of the TOmn bit of timer output register m (TOm) described later by software, and the value reflecting the setting of the timer output function through the count operation is output from the timer output pin (TOmn).

The TOEm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOEm register can be set with a 1-bit or 8-bit memory manipulation instruction with TOEmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 24 Format of Timer output enable register m (TOEm)

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
TOEm	0	0	0	0	0	0	0	0	TOEm7	TOEm6	TOEm5	TOEm4	TOEm3	TOEm2	TOEm1	TOEm0

TOEmn	Timer output enable/disable of channel n
0	Timer output is disabled. Timer operation is not applied to the TOmn bit and the output is fixed. Writing to the TOmn bit is enabled and the level set in the TOmn bit is output from the TOmn pin.
1	Timer output is enabled. Timer operation is applied to the TOmn bit and an output waveform is generated. Writing to the TOmn bit is ignored.

Caution Be sure to clear bits 15 to 8 to “0”.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.3.11 Timer output register m (TOm)

The TOm register is a buffer register of timer output of each channel.

The value of each bit in this register is output from the timer output pin (TOmn) of each channel.

The TOmn bit of this register can be rewritten by software only when timer output is disabled (TOEmn = 0). When timer output is enabled (TOEmn = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.

To use the P03/TI00/TO00, P32/TI01/TO01, P05/TI02/TO02, P30/TI03/TO03, P22/TI04/TO04, P42/TI05/TO05, P07/TI06/TO06, P23/TI07/TO07 pin as a port function pin, set the corresponding TOmn bit to “0”.

The TOm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOm register can be set with an 8-bit memory manipulation instruction with TOmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 25 Format of Timer output register m (TOm)

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
TOm	0	0	0	0	0	0	0	0	TOm7	TOm6	TOm5	TOm4	TOm3	TOm2	TOm1	TOm0

TOmn	Timer output of channel n
0	Timer output value is “0”.
1	Timer output value is “1”.

Caution Be sure to clear bits 15 to 8 to “0”.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.3.12 Timer output level register m (TOLm)

The TOLm register is a register that controls the timer output level of each channel.

The setting of the inverted output of channel n by this register is reflected at the timing of set or reset of the timer output signal while the timer output is enabled (TOEmn = 1) in the Slave channel output mode (TOMmn = 1). In the master channel output mode (TOMmn = 0), this register setting is invalid.

The TOLm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOLm register can be set with an 8-bit memory manipulation instruction with TOLmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 26 Format of Timer output level register m (TOLm)

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
TOLm	0	0	0	0	0	0	0	0	TOLm7	TOLm6	TOLm5	TOLm4	TOLm3	TOLm2	TOLm1	0

TOLmn	Control of timer output level of channel n
0	Positive logic output (active-high)
1	Negative logic output (active-low)

Caution Be sure to clear bits 15 to 8, and 0 to “0”.

Remark 1. If the value of this register is rewritten during timer operation, the timer output logic is inverted when the timer output signal changes next, instead of immediately after the register value is rewritten.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.3.13 Timer output mode register m (TOMm)

The TOMm register is used to control the timer output mode of each channel.

When a channel is used for the independent channel operation function, set the corresponding bit of the channel to be used to 0.

When a channel is used for the simultaneous channel operation function (PWM output, one-shot pulse output, or multiple PWM output), set the corresponding bit of the master channel to 0 and the corresponding bit of the slave channel to 1.

The setting of each channel n by this register is reflected at the timing when the timer output signal is set or reset while the timer output is enabled (TOEmn = 1).

The TOMm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOMm register can be set with an 8-bit memory manipulation instruction with TOMmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 27 Format of Timer output mode register m (TOMm)

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

TOM mn	Control of timer output mode of channel n
0	Master channel output mode (to produce toggle output by timer interrupt request signal (INTTMmn)
1	Slave channel output mode (output is set by the timer interrupt request signal (INTTMmn) of the master channel, and reset by the timer interrupt request signal (INTTM0p) of the slave channel)

Caution **Be sure to clear bits 15 to 8, and 0 to “0”.**

Remark m: Unit number (m = 0)
n: Channel number
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
p: Slave channel number
n < p ≤ 7
(For details of the relation between the master channel and slave channel, refer to **6.4.1 Basic rules of simultaneous channel operation function**.)

6.3.14 Input switch control register (ISC)

The ISC1 and ISC0 bits of the ISC register are used to implement LIN-bus communication operation by using channel 7 in association with the serial array unit. When the ISC1 bit is set to 1, the input signal of the serial data input pin (RxD0) is selected as a timer input signal.

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

Figure 6 - 28 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
0	Uses the input signal of the TI07 pin as a timer input (normal operation).
1	Input signal of the RXD0 pin is used as timer input (detects the wakeup signal and measures the low width of the break field and the pulse width of the sync field).

ISC0	Switching external interrupt (INTP0) input
0	Uses the input signal of the INTP0 pin as an external interrupt (normal operation).
1	Uses the input signal of the RXD0 pin as an external interrupt (wakeup signal detection).

Caution Be sure to clear bits 7 to 2 to “0”.

Remark When the LIN-bus communication function is used, select the input signal of the RxD0 pin by setting ISC1 to 1.

6.3.15 Noise filter enable register 1 (NFEN1)

The NFEN1 register is used to set whether the noise filter can be used for the timer input signal to each channel.
Enable the noise filter by setting the corresponding bits to 1 on the pins in need of noise removal.
When the noise filter is enabled, after synchronization with the operating clock (f_{MCK}) for the target channel, whether the signal keeps the same value for two clock cycles is detected. When the noise filter is disabled, the input signal is only synchronized with the operating clock (f_{MCK}) for the target channel Note.
The NFEN1 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Note For details, see **6.5.1 (2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1)**, **6.5.2 Start timing of counter**, and **6.7 Timer Input (TImn) Control**.

Figure 6 - 29 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	TNFEN04	TNFEN03	TNFEN02	TNFEN01	TNFEN00

TNFEN07	Enable/disable using noise filter of TI07 pin or 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

TNFEN04	Enable/disable using noise filter of TI04 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

TNFEN02	Enable/disable using noise filter of TI02 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN01	Enable/disable using noise filter of TI01 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN00	Enable/disable using noise filter of TI00 pin input signal
0	Noise filter OFF
1	Noise filter ON

Note The applicable pin can be switched by setting the ISC1 bit of the ISC register.
ISC1 = 0: Whether or not to use the noise filter of the TI07 pin can be selected.
ISC1 = 1: Whether or not to use the noise filter of the RxD0 pin can be selected.

6.3.16 Registers that control port functions of timer input/output pins

Using the timer array unit requires setting of the registers that control the port functions for the port pins with which the timer array unit pin functions for the target channel are multiplexed (port mode register (PMxx), port register (Pxx), and port mode control register (PMCxx)). For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.2 Port registers (Pxx)**, and **4.3.6 Port mode control registers (PMCxx)**.
The port mode register (PMxx), port register (Pxx), and port mode control register (PMCxx) to be set depend on the product. For details, see **4.5.3 Register setting examples for used port and alternate functions**.

Using a port pin which is multiplexed with a timer output pin function (e.g. P03/TI00/TO00, P05/TI02/TO02) for timer output requires setting the corresponding bits in the port mode register (PMxx) and port register (Pxx) to 0.

Example When P05/TO02/TI02 is to be used for timer output
Set the PM05 bit of port mode register 0 to 0.
Set the P05 bit of port register 0 to 0.

Using a port pin which is multiplexed with a timer input pin function (e.g. P03/TI00/TO00, P05/TI02/TO02) for timer input requires setting the corresponding bit in the port mode register (PMxx) to 1. At this time, the value of the corresponding bit in the port register (Pxx) may be 0 or 1.

Example When P05/TO02/TI02 is to be used for timer input
Set the PM05 bit of port mode register 0 to 1.
Set the P05 bit of port register 0 to 0 or 1.

Remark 1. The P42/TI05/TO05/IVCMP1 pin is multiplexed with an analog input pin function. When using the timer I/O function, be sure to set the corresponding bit of the PMC4 register which switches between digital I/O and analog input to “0”.
Remark 2. When using a port pin which is multiplexed with a segment output pin function for timer I/O, be sure to clear the corresponding bit of LCD port function registers 0 to 6 (PFSEG0 to PFSEG6) to “0”.
Remark 3. When using the P125/(TI06)/(TO06)/V_{L3} pin for timer I/O, be sure to set the ISCVL3 bit of the LCD Input switch control register (ISCLCD) to “1”.
Remark 4. When using the P126/(TI04)/(TO04)/CAPL and P127/(TI03)/(TO03)/CAPH pins for timer I/O, be sure to set the ISCCAP bit of the LCD Input switch control register (ISCLCD) 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 (channel 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) The slave channel must be lower than the master channel.
Example If channel 2 is set as a master channel, channel 3 or those that follow (channels 3, 4, 5, etc.) can be set as a slave channel.

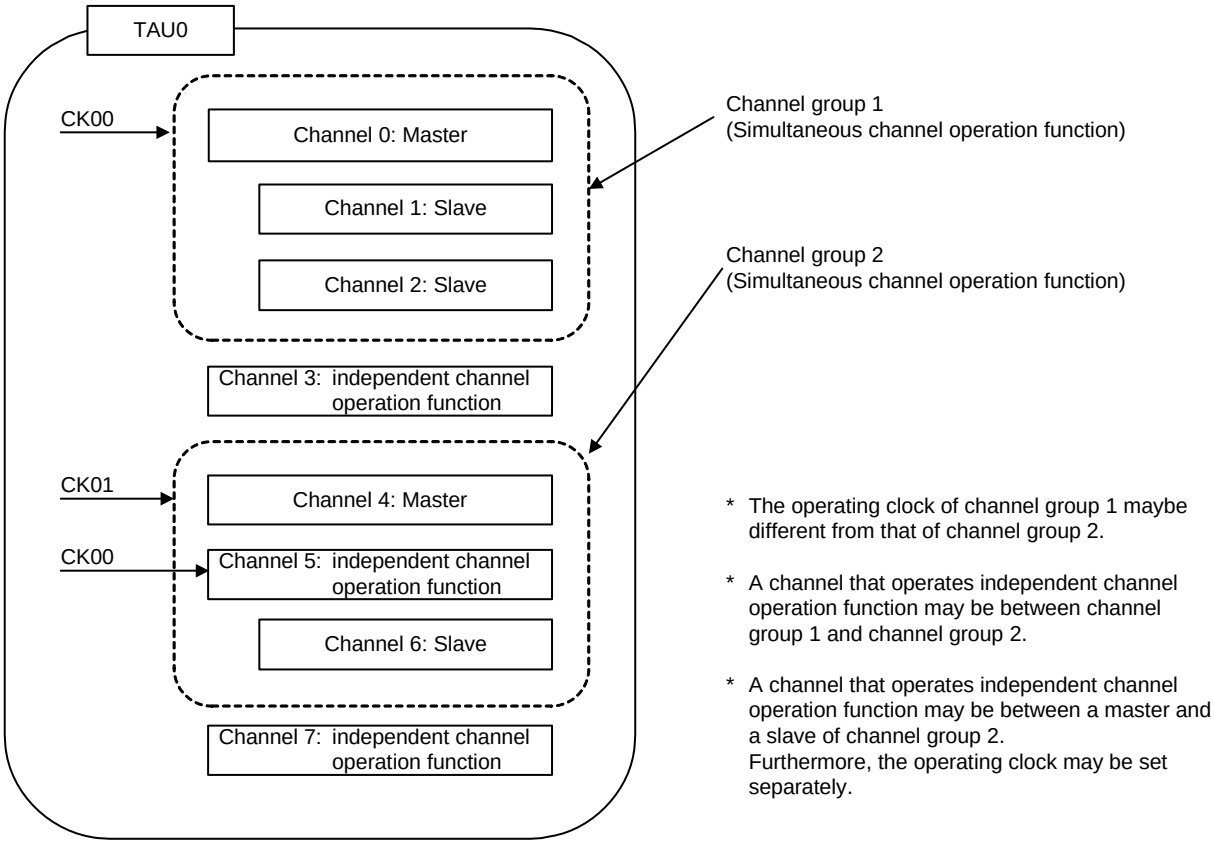
(4) Two or more slave channels can be set for one master channel.
(5) When two or more master channels are to be used, slave channels with a master channel between them may not be set.
Example If channels 0 and 4 are set as master channels, channels 1 to 3 can be set as the slave channels of master channel 0. Channels 5 to 7 cannot be set as the slave channels of master channel 0.

(6) The operating clock for a slave channel in combination with a master channel must be the same as that of the master channel. The CKSmn0, CKSmn1 bits (bit 15, 14 of timer mode register mn (TMRmn)) of the slave channel that operates in combination with the master channel must be the same value as that of the master channel.
(7) A master channel can transmit INTTMmn (interrupt), start software trigger, and count clock to the lower channels.
(8) A slave channel can use INTTMmn (interrupt), a start software trigger, or the count clock of the master channel as a source clock, but cannot transmit its own INTTMmn (interrupt), start software trigger, or count clock to channels with lower channel numbers.
(9) A master channel cannot use INTTMmn (interrupt), a start software trigger, or the count clock from the other higher master channel as a source clock.
(10) To simultaneously start channels that operate in combination, the channel start trigger bit (TSmn) of the channels in combination must be set at the same time.
(11) During the counting operation, a TSmn bit of a master channel or TSmn bits of all channels which are operating simultaneously can be set. It cannot be applied to TSmn bits of slave channels alone.
(12) To stop the channels in combination simultaneously, the channel stop trigger bit (TTmn) of the channels in combination must be set at the same time.
(13) CKm2/CKm3 cannot be selected while channels are operating simultaneously, because the operating clocks of master channels and slave channels have to be synchronized.
(14) Timer mode register m0 (TMRm0) has no master bit (it is fixed as “0”). However, as channel 0 is the highest channel, it can be used as a master channel during simultaneous operation.

The rules of the simultaneous channel operation function are applied in a channel group (a master channel and slave channels forming one simultaneous channel operation function).
If two or more channel groups that do not operate in combination are specified, the basic rules of the simultaneous channel operation function in **6.4.1 Basic rules of simultaneous channel operation function** do not apply to the channel groups.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

Example



6.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only)

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels.
This function can only be used for channels 1 and 3, and there are several rules for using it.
The basic rules for this function are as follows:

(1) The 8-bit timer operation function applies only to channels 1 and 3.
(2) When using 8-bit timers, set the SPLIT bit of timer mode register mn (TMRmn) to 1.
(3) The higher 8 bits can be operated as the interval timer function.
(4) At the start of operation, the higher 8 bits output INTTMm1H/INTTMm3H (an interrupt) (which is the same operation performed when MDmn0 is set to 1).
(5) The operation clock of the higher 8 bits is selected according to the CKSmn1 and CKSmn0 bits of the lower-bit TMRmn register.
(6) For the higher 8 bits, the TSHm1/TSHm3 bit is manipulated to start channel operation and the TTHm1/TTHm3 bit is manipulated to stop channel operation. The channel status can be checked using the TEHm1/TEHm3 bit.
(7) The lower 8 bits operate according to the TMRmn register settings. The following three functions support operation of the lower 8 bits:
  • Interval timer function
  • External event counter function
  • Delay count function

(8) For the lower 8 bits, the TSm1/TSm3 bit is manipulated to start channel operation and the TTm1/TTm3 bit is manipulated to stop channel operation. The channel status can be checked using the TEm1/TEm3 bit.
(9) During 16-bit operation, manipulating the TSHm1, TSHm3, TTHm1, and TTHm3 bits is invalid. The TSm1, TSm3, TTm1, and TTm3 bits are manipulated to operate channels 1 and 3. The TEHm3 and TEHm1 bits are not changed.
(10) For the 8-bit timer function, the simultaneous operation functions (one-shot pulse, PWM, and multiple PWM) cannot be used.

Remark m: Unit number (m = 0), n: Channel number (n = 1, 3)

6.5 Operation 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 CCSmn bit of timer mode register mn (TMRmn).

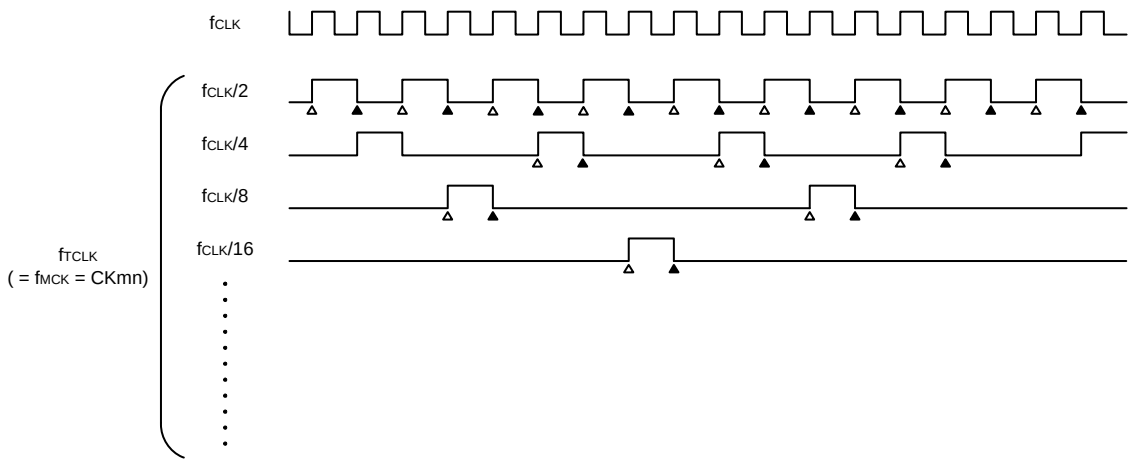
- Operation clock (f_{MCK}) specified by the CKSmn0 and CKSmn1 bits
- Valid edge of input signal input from the TImn pin

Because the timer array unit is designed to operate in synchronization with f_{CLK}, the timings of the count clock (f_{TCLK}) are shown below.

(1) When operation clock (f_{MCK}) specified by the CKSmn0 and CKSmn1 bits is selected (CCSmn = 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 TPSmn 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 mn (TCRmn) delayed by one period of f_{CLK} from rising edge of the count clock, because of synchronization with f_{CLK}. But, this is described as “counting at rising edge of the count clock”, as a matter of convenience.

Figure 6 - 30 Timing of f_{CLK} and count clock (f_{TCLK}) (When CCSmn = 0)



Remark 1. △ : Rising edge of the count clock
▲ : Synchronization, increment/decrement of counter

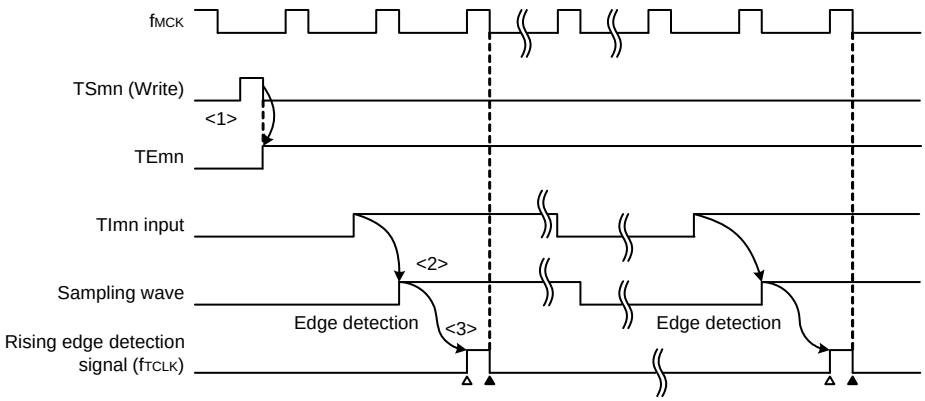
Remark 2. f_{CLK}: CPU/peripheral hardware clock

(2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1)

The count clock (f_{TCLK}) becomes the signal that detects valid edge of input signal via the TImn pin and synchronizes next rising f_{MCK}. The count clock (f_{TCLK}) is delayed for 1 to 2 periods of f_{MCK} from the input signal via the TImn pin (when a noise filter is used, the delay becomes 3 to 4 clocks).

Counting of timer count register mn (TCRmn) delayed by one period of f_{CLK} from rising edge of the count clock, because of synchronization with f_{CLK}. But, this is described as “counting at valid edge of input signal via the TImn pin”, as a matter of convenience.

Figure 6 - 31 Timing of f_{CLK} and count clock (f_{TCLK}) (When CCSmn = 1, noise filter unused)



<1> Setting TSmn bit to 1 enables the timer to be started and to become wait state for valid edge of input signal via the TImn pin.
<2> The rise of input signal via the TImn pin is sampled by f_{MCK}.
<3> The edge is detected by the rising of the sampled signal and the detection signal (count clock) is output.

Remark 1. △ : Rising edge of the count clock
▲ : Synchronization, increment/decrement of counter

Remark 2. f_{CLK}: CPU/peripheral hardware clock
f_{MCK}: Operation clock of channel n

Remark 3. The waveform of the input signal via TImn pin of the input pulse interval measurement, the measurement of high/low width of input signal, and the delay counter, and the one-shot pulse output are the same as that shown in Figure 6 - 31.

6.5.2 Start timing of counter

Timer count register mn (TCRmn) becomes enabled to operation by setting of TSmn bit of timer channel start register m (TSm).

Operations from count operation enabled state to timer count Register mn (TCRmn) count start is shown in Table 6 - 6.

Table 6 - 6 Operations from Count Operation Enabled State to Timer count Register mn (TCRmn) Count Start

Timer operation mode	Operation when TSmn = 1 is set
• Interval timer mode	No operation is carried out from start trigger detection (TSmn = 1) until count clock generation. The first count clock loads the value of the TDRmn register to the TCRmn register and the subsequent count clock performs count down operation (see **6.5.3 (1) Operation of interval timer mode**).
• Event counter mode	Writing 1 to the TSmn bit loads the value of the TDRmn register to the TCRmn register. If detect edge of TImn input. The subsequent count clock performs count down operation. (see **6.5.3 (2) Operation of event counter mode**).
• Capture mode	No operation is carried out from start trigger detection (TSmn = 1) until count clock generation. The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see **6.5.3 (3) Operation of capture mode (input pulse interval measurement)**).
• One-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads the value of the TDRmn register to the TCRmn register and the subsequent count clock performs count down operation (see **6.5.3 (4) Operation of one-count mode**).
• Capture & one-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see **6.5.3 (5) Start timing in capture & one-count mode (when high-level width is measured)**).

6.5.3 Operation of counter

Here, the counter operation in each mode is explained.

(1) Operation of interval timer mode

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit. Timer count register mn (TCRmn) holds the initial value until count clock generation.

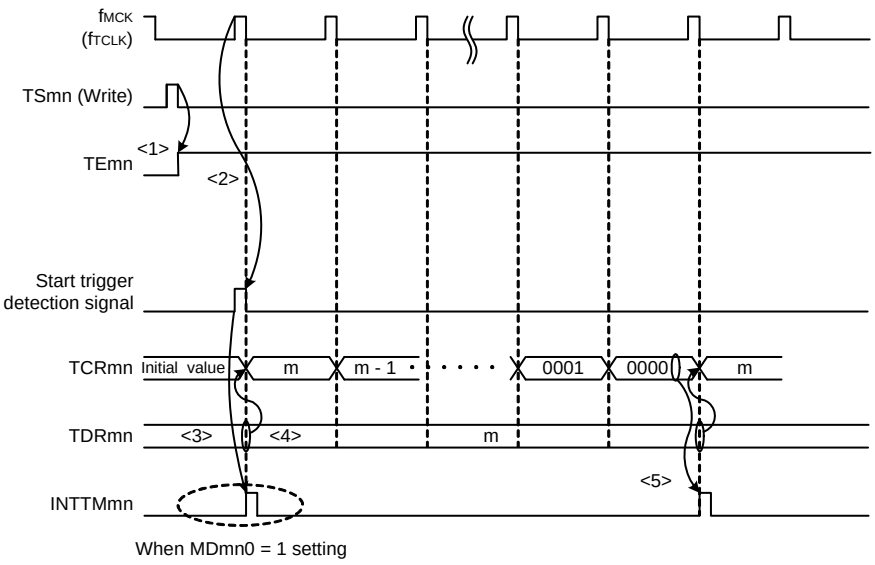
<2> A start trigger is generated at the first count clock after operation is enabled.

<3> When the MDmn0 bit is set to 1, INTTMmn is generated by the start trigger.

<4> By the first count clock after the operation enable, the value of timer data register mn (TDRmn) is loaded to the TCRmn register and counting starts in the interval timer mode.

<5> When the TCRmn register counts down and its count value is 0000H, INTTMmn is generated and the value of timer data register mn (TDRmn) is loaded to the TCRmn register and counting keeps on.

Figure 6 - 32 Operation Timing (In Interval Timer Mode)



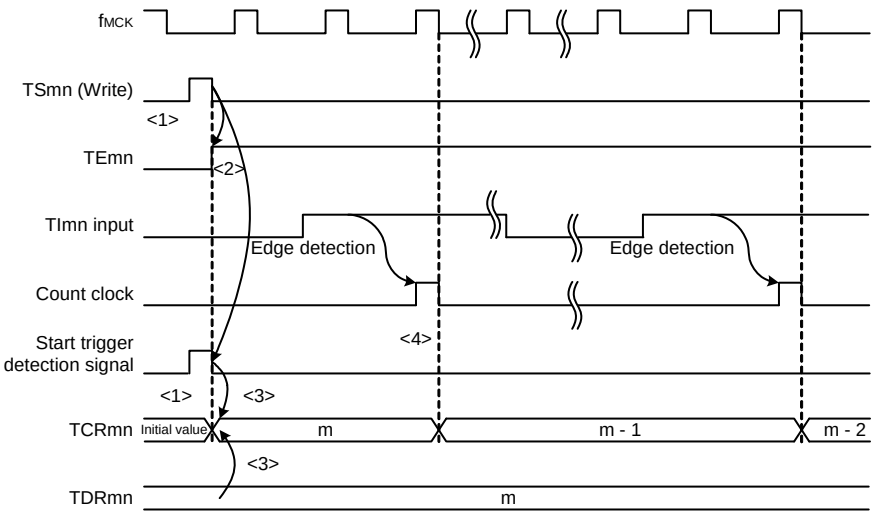
Caution **In the first cycle operation of count clock after writing the TSmn bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.**

Remark fMCK, the start trigger detection signal, and INTTMmn become active between one clock in synchronization with fCLK.

(2) Operation of event counter mode

<1> Timer count register mn (TCRmn) holds its initial value while operation is stopped (TEmn = 0).
<2> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.
<3> As soon as 1 has been written to the TSmn bit and 1 has been set to the TEmn bit, the value of timer data register mn (TDRmn) is loaded to the TCRmn register to start counting.
<4> After that, the TCRmn register value is counted down according to the count clock of the valid edge of the TImn input.

Figure 6 - 33 Operation Timing (In Event Counter Mode)

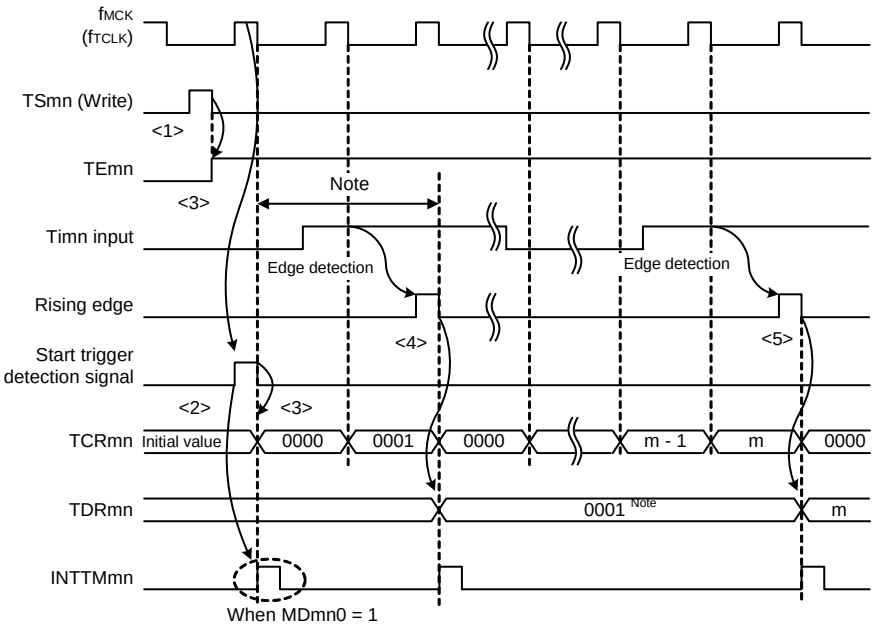


Remark The above figure shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (fMCK).

(3) Operation of capture mode (input pulse interval measurement)

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.
<2> Timer count register mn (TCRmn) holds the initial value until count clock generation.
<3> A start trigger is generated at the first count clock after operation is enabled. And the value of 0000H is loaded to the TCRmn register and counting starts in the capture mode. (When the MDmn0 bit is set to 1, INTTMmn is generated by the start trigger.)
<4> On detection of the valid edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated. However, this capture value is no meaning. The TCRmn register keeps on counting from 0000H.
<5> On next detection of the valid edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated.

Figure 6 - 34 Operation Timing (In Capture Mode: Input Pulse Interval Measurement)



Note If a clock has been input to TImn (the trigger exists) when capturing starts, counting starts when a trigger is detected, even if no edge is detected. Therefore, the first captured value (<4>) does not determine a pulse interval (in the above figure, 0001 just indicates two clock cycles but does not determine the pulse interval) and so the user can ignore it.

Caution **In the first cycle operation of count clock after writing the TSmn bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.**

Remark 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 TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (fMCK).

(4) Operation of one-count mode

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.

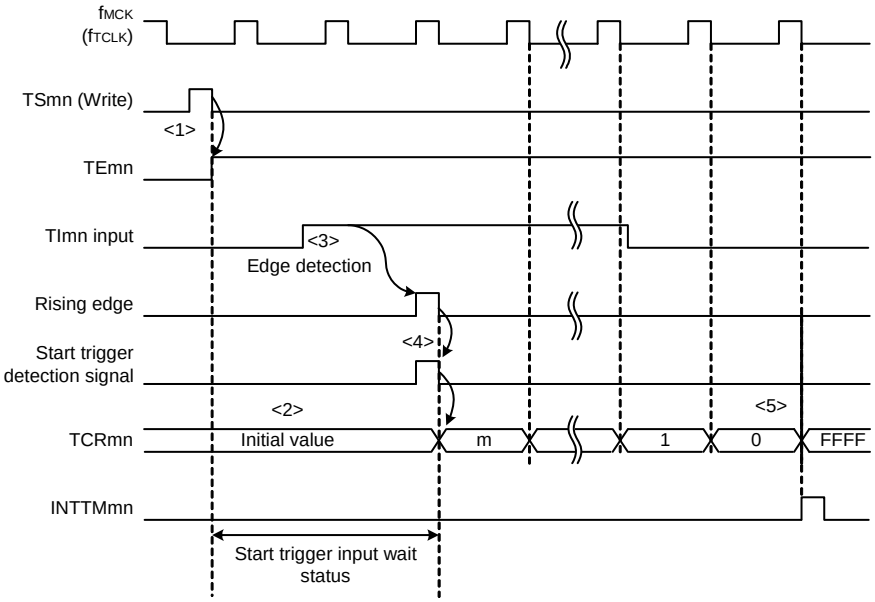
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.

<3> Rising edge of the TImn input is detected.

<4> On start trigger detection, the value of timer data register mn (TDRmn) is loaded to the TCRmn register and count starts.

<5> When the TCRmn register counts down and its count value is 0000H, INTTMmn is generated and the value of the TCRmn register becomes FFFFH and counting stops.

Figure 6 - 35 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 TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (fMCK).

(5) Start timing in capture & one-count mode (when high-level width is measured)

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit of timer channel start register m (TSm).

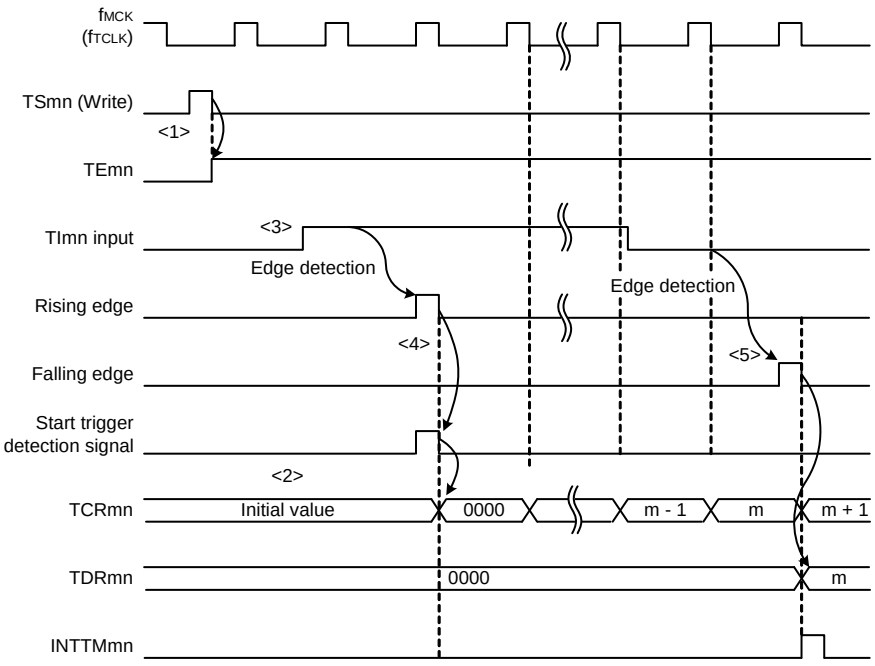
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.

<3> Rising edge of the TImn input is detected.

<4> On start trigger detection, the value of 0000H is loaded to the TCRmn register and count starts.

<5> On detection of the falling edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated.

Figure 6 - 36 Operation Timing (In Capture & One-count Mode: High-level Width Measurement)

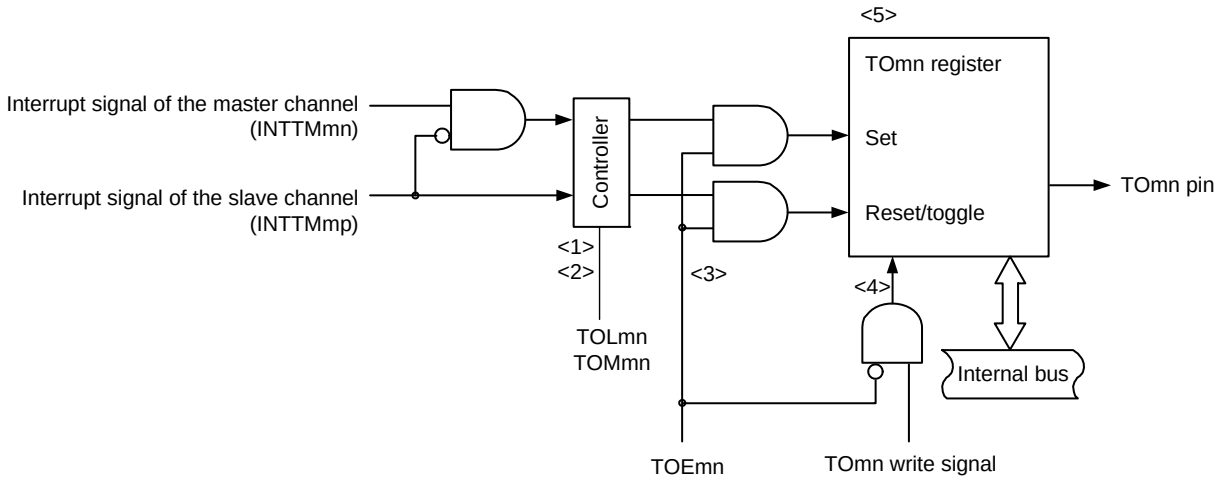


Remark The above figure shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (fMCK).

6.6 Channel Output (TOmn pin) Control

6.6.1 TOmn pin output circuit configuration

Figure 6 - 37 Output Circuit Configuration



The following describes the TOmn pin output circuit.

<1> When TOMmn = 0 (master channel output mode), the set value of timer output level register m (TOLm) is ignored and only INTTM0p (slave channel timer interrupt) is transmitted to timer output register m (TOm).

<2> When TOMmn = 1 (slave channel output mode), both INTTMmn (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TOm register.
At this time, the TOLm register becomes valid and the signals are controlled as follows:

When TOLmn = 0: Positive logic output (INTTMmn → set, INTTM0p → reset)
When TOLmn = 1: Negative logic output (INTTMmn → reset, INTTM0p → set)

When INTTMmn and INTTM0p are simultaneously generated, (0% output of PWM), INTTM0p (reset signal) takes priority, and INTTMmn (set signal) is masked.

<3> While timer output is enabled (TOEmn = 1), INTTMmn (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TOm register. Writing to the TOm register (TOmn write signal) becomes invalid.
When TOEmn = 1, the TOmn pin output never changes with signals other than interrupt signals.
To initialize the TOmn pin output level, it is necessary to set timer operation is stopped (TOEmn = 0) and to write a value to the TOm register.

<4> While timer output is disabled (TOEmn = 0), writing to the TOmn bit to the target channel (TOmn write signal) becomes valid. When timer output is disabled (TOEmn = 0), neither INTTMmn (master channel timer interrupt) nor INTTM0p (slave channel timer interrupt) is transmitted to the TOm register.

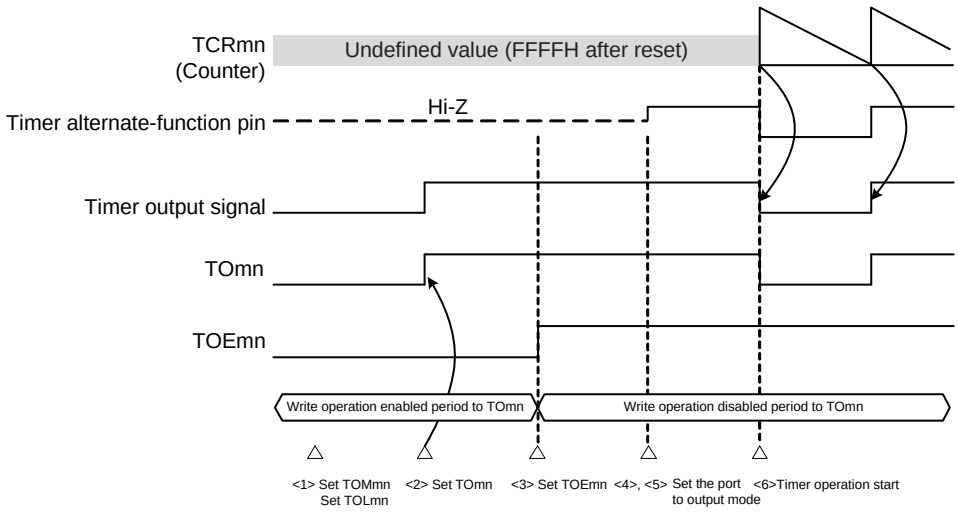
<5> The TOm register can always be read, and the TOmn pin output level can be checked.

Remark m: Unit number (m = 0)
n: Channel number
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
p: Slave channel number
n < p ≤ 7

6.6.2 TOmn Pin Output Setting

The following figure shows the procedure and status transition of the TOmn output pin from initial setting to timer operation start.

Figure 6 - 38 Status Transition from Timer Output Setting to Operation Start



<1> The operation mode of timer output is set.

- TOMmn bit (0: Master channel output mode, 1: Slave channel output mode)
- TOLmn bit (0: Positive logic output, 1: Negative logic output)

<2> The timer output signal is set to the initial status by setting timer output register m (TOm).
<3> The timer output operation is enabled by writing 1 to the TOEmn bit (writing to the TOm register is disabled).
<4> The port is set to digital I/O by port mode control register (PMCxx).
<5> The port I/O setting is set to output (see **6.3.16 Registers that control port functions of timer input/output pins**).
<6> The timer operation is enabled (TSmn = 1).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.6.3 Cautions on Channel Output Operation

(1) Changing values set in the registers TOm, TOEm, and TOLm during timer operation

Since the timer operations (operations of timer count register mn (TCRmn) and timer data register mn (TDRmn)) are independent of the TOmn output circuit and changing the values set in timer output register m (TOm), timer output enable register m (TOEm), and timer output level register m (TOLm) does not affect the timer operation, the values can be changed during timer operation. To output an expected waveform from the TOmn pin by timer operation, however, set the TOm, TOEm, TOLm, and TOMm registers to the values stated in the register setting example of each operation shown by 6.7 and 6.8.

When the values set to the TOEm, and TOMm registers (but not the TOm register) are changed close to the occurrence of the timer interrupt (INTTMmn) of each channel, the waveform output to the TOmn pin might differ, depending on whether the values are changed immediately before or immediately after the timer interrupt (INTTMmn) occurs.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

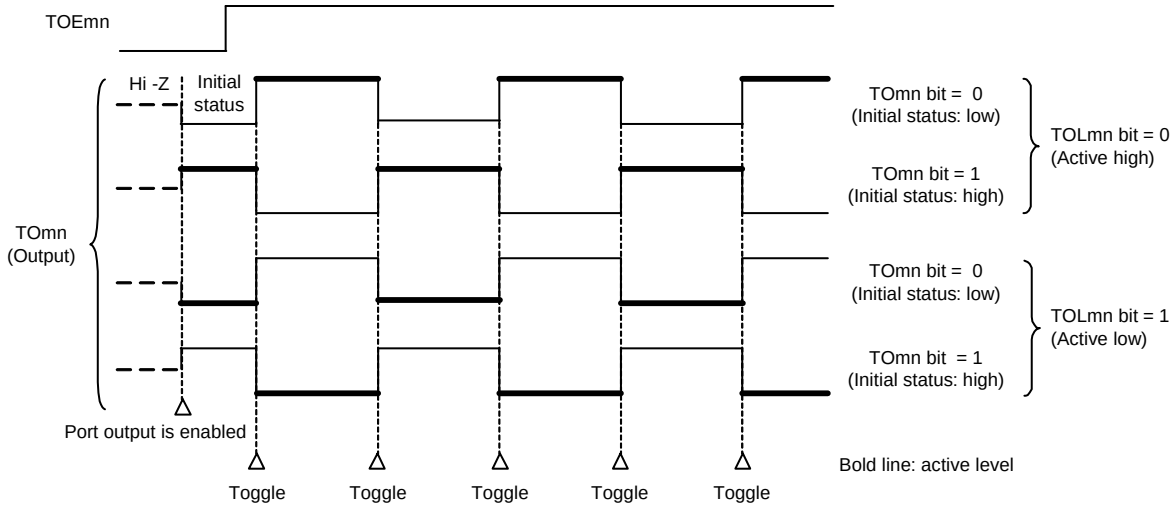
(2) Default level of TOmn pin and output level after timer operation start

The change in the output level of the TOmn pin when timer output register m (TOm) is written while timer output is disabled (TOEmn = 0), the initial level is changed, and then timer output is enabled (TOEmn = 1) before port output is enabled, is shown below.

(a) When operation starts with master channel output mode (TOMmn = 0) setting

The setting of timer output level register m (TOLm) is invalid when master channel output mode (TOMmn = 0). When the timer operation starts after setting the default level, the toggle signal is generated and the output level of the TOmn pin is reversed.

Figure 6 - 39 TOmn Pin Output Status at Toggle Output (TOMmn = 0)



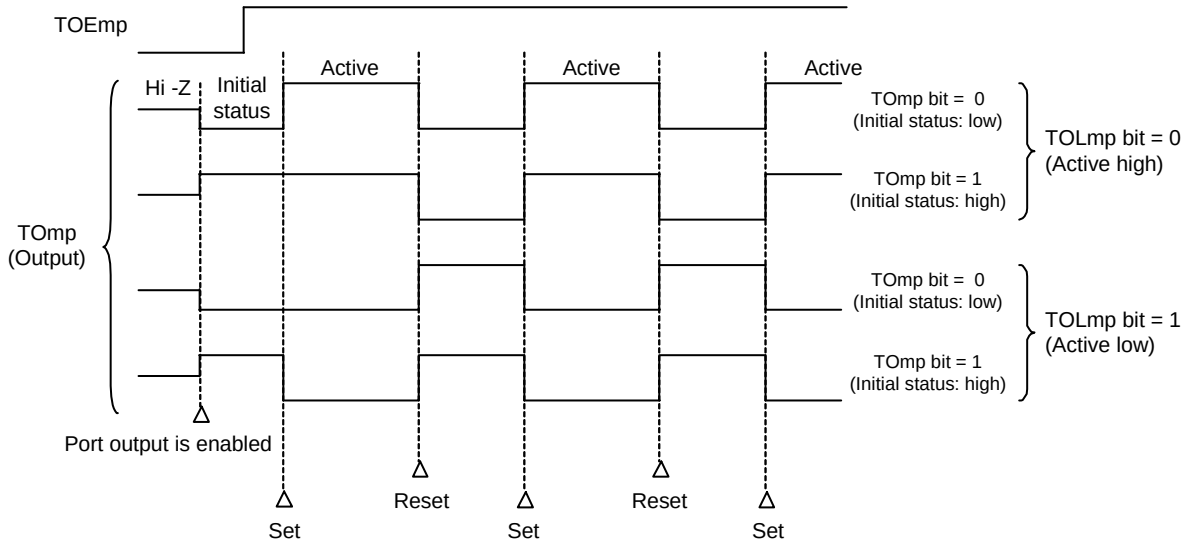
Remark 1. Toggle: Reverse TOmn pin output status

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

(b) When operation starts with slave channel output mode (TOMmp = 1) setting (PWM output))

When slave channel output mode (TOMmp = 1), the active level is determined by timer output level register m (TOLm) setting.

Figure 6 - 40 TOmp Pin Output Status at PWM Output (TOMmp = 1)



Remark 1. Set: The output signal of the TOmp pin changes from inactive level to active level.
Reset: The output signal of the TOmp pin changes from active level to inactive level.

Remark 2. m: Unit number (m = 0), p: Channel number (p = 1 to 7)

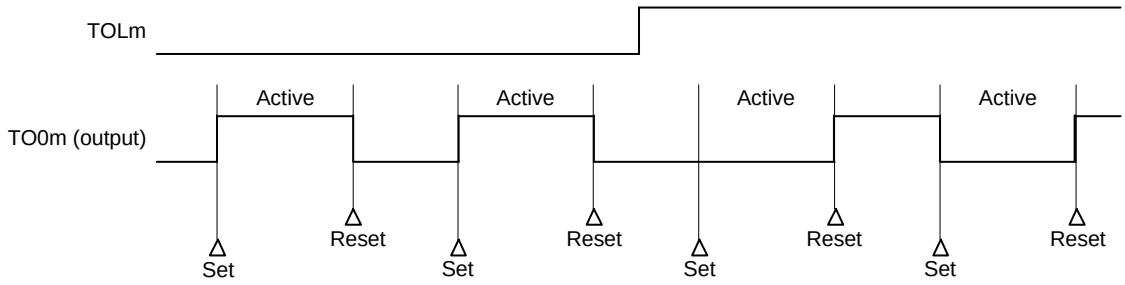
(3) Operation of TOmn pin in slave channel output mode (TOMmn = 1)

(a) When timer output level register m (TOLm) setting has been changed during timer operation

When the TOLm register setting has been changed during timer operation, the setting becomes valid at the generation timing of the TOmn pin change condition. Rewriting the TOLm register does not change the output level of the TOmn pin.

The operation when TOMmn is set to 1 and the value of the TOLm register is changed while the timer is operating (TEmn = 1) is shown below.

Figure 6 - 41 Operation when TOLm Register Has Been Changed Contents during Timer Operation



Remark 1. Set: The output signal of the TOmn pin changes from inactive level to active level.
Reset: The output signal of the TOmn pin changes from active level to inactive level.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

(b) Set/reset timing

To realize 0%/100% output at PWM output, the TOmn pin/TOmn bit set timing at master channel timer interrupt (INTTMmn) generation is delayed by 1 count clock by the slave channel.

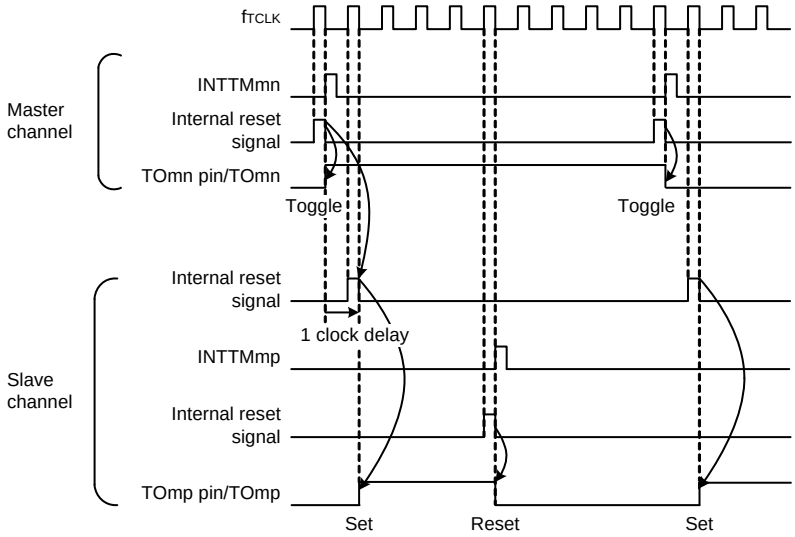
If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

Figure 6 - 42 shows the set/reset operating statuses where the master/slave channels are set as follows.

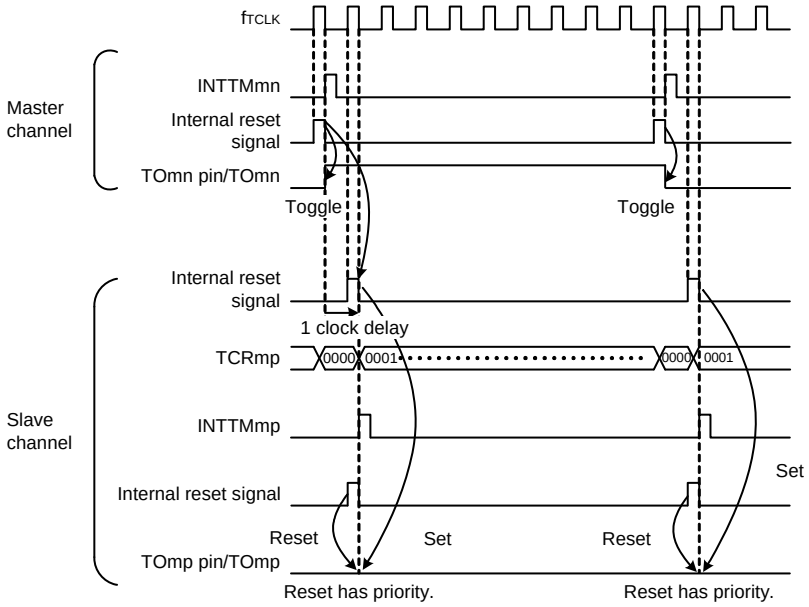
Master channel: TOEmn = 1, TOMmn = 0, TOLmn = 0
Slave channel: TOEmp = 1, TOMmp = 1, TOLmp = 0

Figure 6 - 42 Set/Reset Timing Operating Statuses

(1) Basic operation timing



(2) Operation timing when 0% duty



Remark 1. Internal reset signal: TOmn pin reset/toggle signal
Internal set signal: TOmn pin set signal

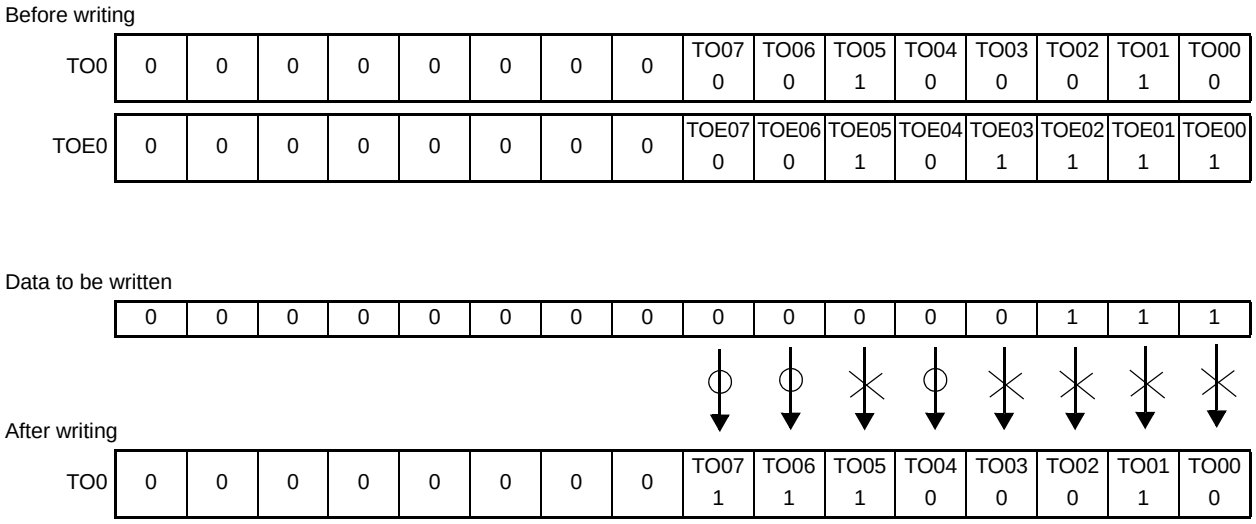
Remark 2. m: Unit number (m = 0)
n: Channel number
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
p: Slave channel number
n < p ≤ 7

6.6.4 Collective manipulation of TOmn bit

In timer output register m (TOm), the setting bits for all the channels are located in one register in the same way as timer channel start register m (TSm). Therefore, the TOmn bit of all the channels can be manipulated collectively.

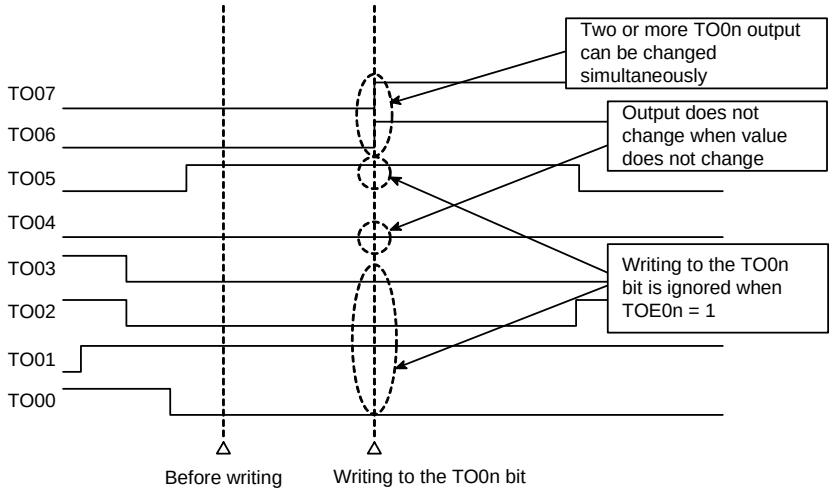
Only the desired bits can also be manipulated by enabling writing only to the TOmn bits (TOEmn = 0) that correspond to the relevant bits of the channel used to perform output (TOmn).

Figure 6 - 43 Example of TO0n Bit Collective Manipulation



Writing is done only to the TOmn bit with TOEmn = 0, and writing to the TOmn bit with TOEmn = 1 is ignored. TOmn (channel output) to which TOEmn = 1 is set is not affected by the write operation. Even if the write operation is done to the TOmn bit, it is ignored and the output change by timer operation is normally done.

Figure 6 - 44 TO0n Pin Statuses by Collective Manipulation of TO0n Bit



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.6.5 Timer Interrupt and TOmn Pin Output at Operation Start

In the interval timer mode or capture mode, the MDmn0 bit in timer mode register mn (TMRmn) sets whether or not to generate a timer interrupt at count start.

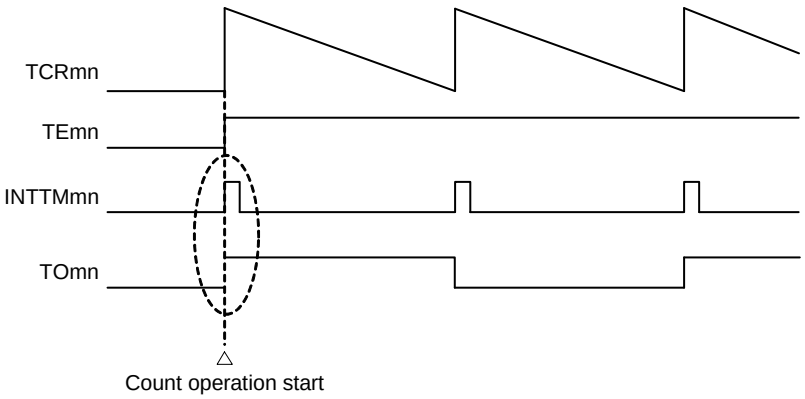
When MDmn0 is set to 1, the count operation start timing can be known by the timer interrupt (INTTMmn) generation.

In the other modes, neither timer interrupt at count operation start nor TOmn output is controlled.

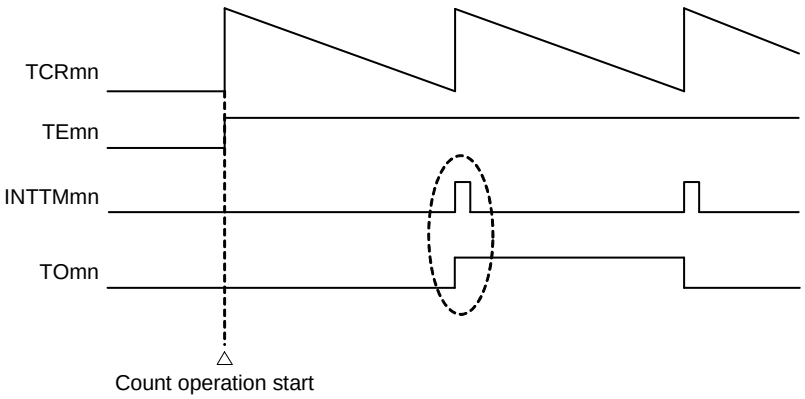
Figure 6 - 45 shows operation examples when the interval timer mode (TOEmn = 1, TOMmn = 0) is set.

Figure 6 - 45 Operation examples of timer interrupt at count operation start and TOmn output

(a) When MDmn0 is set to 1



(b) When MDmn0 is set to 0



When MDmn0 is set to 1, a timer interrupt (INTTMmn) is output at count operation start, and TOmn performs a toggle operation.

When MDmn0 is set to 0, a timer interrupt (INTTMmn) is not output at count operation start, and TOmn does not change either. After counting one cycle, INTTMmn is output and TOmn performs a toggle operation.

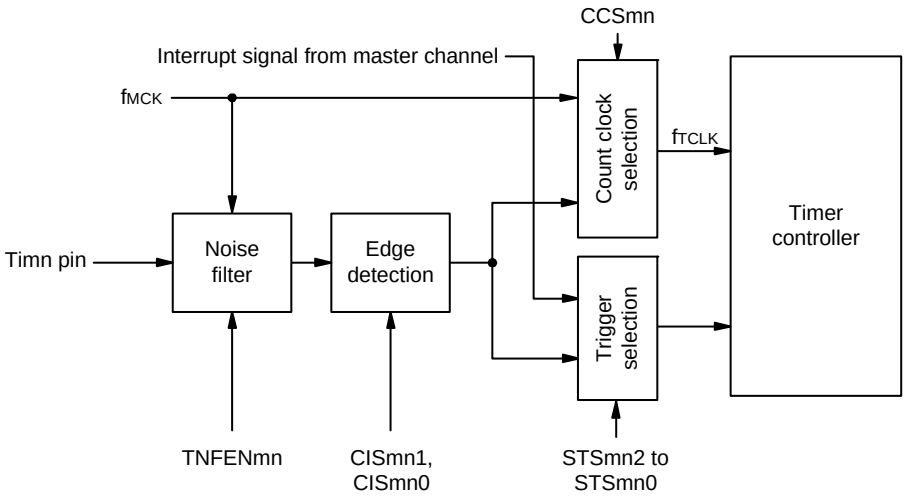
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.7 Timer Input (TImn) Control

6.7.1 TImn input circuit configuration

A signal is input from a timer input pin, goes through a noise filter and an edge detector, and is sent to a timer controller. Enable the noise filter for the pin in need of noise removal. The following shows the configuration of the input circuit.

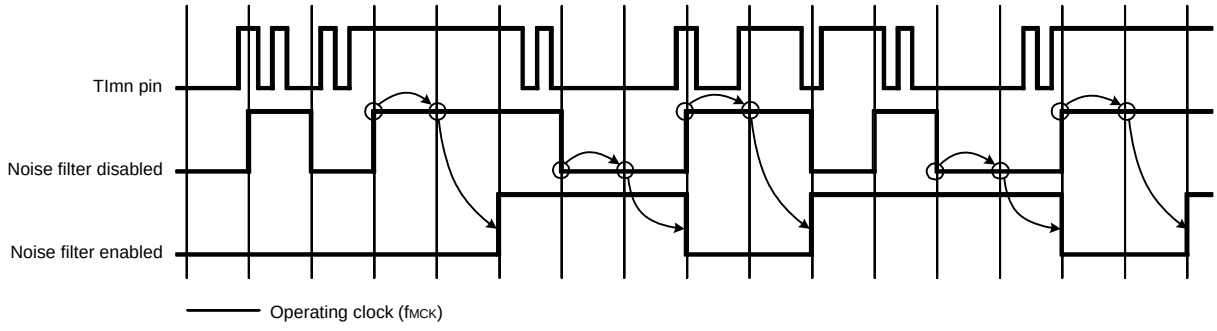
Figure 6 - 46 Input Circuit Configuration



6.7.2 Noise filter

When the noise filter is disabled, the input signal is only synchronized with the operating clock (fMCK) for channel n. When the noise filter is enabled, after synchronization with the operating clock (fMCK) for channel n, whether the signal keeps the same value for two clock cycles is detected. The following shows differences in waveforms output from the noise filter between when the noise filter is enabled and disabled.

Figure 6 - 47 Sampling Waveforms through TImn Input Pin with Noise Filter Enabled and Disabled



Caution The input waveforms to the TImn pin are shown to explain the operation when the noise filter is enabled or disabled. When actually inputting waveforms, input them according to the TImn input high-level and low-level widths listed in 34.4 or 35.4 AC Characteristics.

6.7.3 Cautions on channel input operation

When a timer input pin is set as unused, the operating clock is not supplied to the noise filter. Therefore, after settings are made to use the timer input pin, the following wait time is necessary before a trigger is specified to enable operation of the channel corresponding to the timer input pin.

(1) Noise filter is disabled

When bits 12 (CCSmn), 9 (STSmn1), and 8 (STSmn0) in the timer mode register mn (TMRmn) are 0 and then one of them is set to 1, wait for at least two cycles of the operating clock (f_{MCK}), and then set the operation enable trigger bit in the timer channel start register (TSm).

(2) Noise filter is enabled

When bits 12 (CCSmn), 9 (STSmn1), and 8 (STSmn0) in the timer mode register mn (TMRmn) are all 0 and then one of them is set to 1, wait for at least four cycles of the operating clock (f_{MCK}), and then set the operation enable trigger bit in the timer channel start register (TSm).

6.8 Independent Channel Operation Function of Timer Array Unit

6.8.1 Operation as interval timer/square wave output

(1) Interval timer

The timer array unit can be used as a reference timer that generates INTTMmn (timer interrupt) at fixed intervals.

The interrupt generation period can be calculated by the following expression.

Generation period of INTTMmn (timer interrupt) = Period of count clock × (Set value of TDRmn + 1)

(2) Operation as square wave output

TOmn performs a toggle operation as soon as INTTMmn has been generated, and outputs a square wave with a duty factor of 50%.

The period and frequency for outputting a square wave from TOmn can be calculated by the following expressions.

- Period of square wave output from TOmn = Period of count clock × (Set value of TDRmn + 1) × 2
- Frequency of square wave output from TOmn = Frequency of count clock/{(Set value of TDRmn + 1) × 2}

Timer count register mn (TCRmn) operates as a down counter in the interval timer mode.

The TCRmn register loads the value of timer data register mn (TDRmn) at the first count clock after the channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) is set to 1. If the MDmn0 bit of timer mode register mn (TMRmn) is 0 at this time, INTTMmn is not output and TOmn is not toggled. If the MDmn0 bit of the TMRmn register is 1, INTTMmn is output and TOmn is toggled.

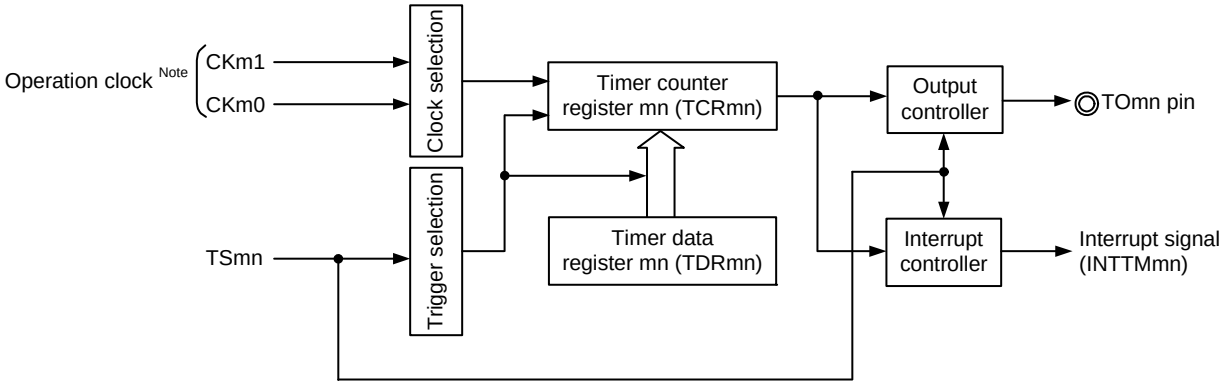
After that, the TCRmn register count down in synchronization with the count clock.

When TCRmn = 0000H, INTTMmn is output and TOmn is toggled at the next count clock. At the same time, the TCRmn register loads the value of the TDRmn register again. After that, the same operation is repeated.

The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid from the next period.

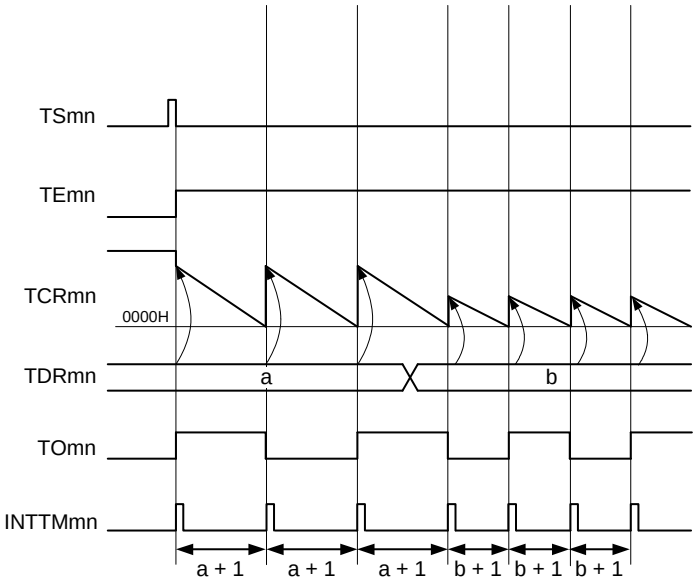
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6 - 48 Block Diagram of Operation as Interval Timer/Square Wave Output



Note When channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Figure 6 - 49 Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MDmn0 = 1)

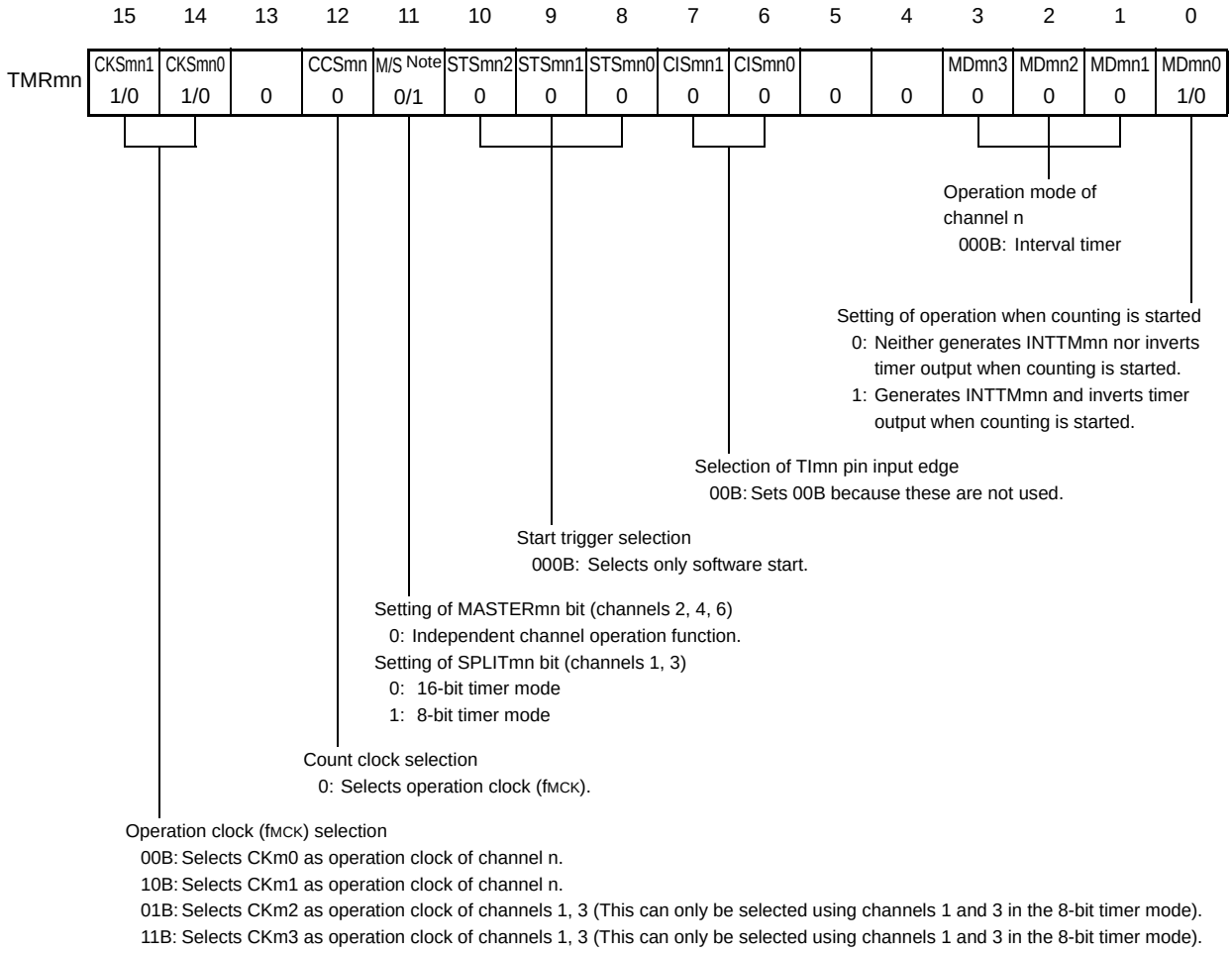


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

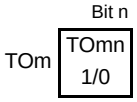
Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
TOmn: TOmn pin output signal

Figure 6 - 50 Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output

(a) Timer mode register mn (TMRmn)

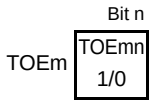


(b) Timer output register m (TOm)



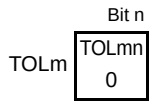
0: Outputs 0 from TOmn.
1: Outputs 1 from TOmn.

(c) Timer output enable register m (TOEm)



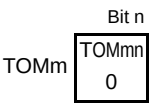
0: Stops the TOmn output operation by counting operation.
1: Enables the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)



0: Cleared to 0 when TOMmn = 0 (master channel output mode)

(e) Timer output mode register m (TOMm)



0: Sets master channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6 - 51 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 TAUmEN 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 m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets timer mode register mn (TMRmn) (determines operation mode of channel). Sets interval (period) value to timer data register mn (TDRmn).	Channel stops operating. (Clock is supplied and some power is consumed.)
	To use the TOmn output Clears the TOMmn bit of timer output mode register m (TOMm) to 0 (master channel output mode). Clears the TOLmn bit to 0. Sets the TOmn bit and determines default level of the TOmn output. →	The TOmn pin goes into Hi-Z output state. The TOmn default setting level is output when the port mode register is in the output mode and the port register is 0.
	Sets the TOEmn bit to 1 and enables operation of TOmn. →	TOmn does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmn pin outputs the TOmn set level.
Operation start	(Sets the TOEmn bit to 1 only if using TOmn output and resuming operation.). Sets the TSmn (TSHm1, TSHm3) bit to 1. → The TSmn (TSHm1, TSHm3) bit automatically returns to 0 because it is a trigger bit.	TEmn (TEHm1, TEHm3) = 1, and count operation starts. Value of the TDRmn register is loaded to timer count register mn (TCRmn). INTTMmn is generated and TOmn performs toggle operation if the MDmn0 bit of the TMRmn register is 1.
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TOm and TOEm registers can be changed. Set values of the TMRmn register, TOMmn, and TOLmn bits cannot be changed.	Counter (TCRmn) counts down. When count value reaches 0000H, the value of the TDRmn register is loaded to the TCRmn register again and the count operation is continued. By detecting TCRmn = 0000H, INTTMmn is generated and TOmn performs toggle operation. After that, the above operation is repeated.
Operation stop	The TTmn (TTHm1, TTHm3) bit is set to 1. → The TTmn (TTHm1, TTHm3) bit automatically returns to 0 because it is a trigger bit.	TEmn (TEHm1, TEHm3), and count operation stops. The TCRmn register holds count value and stops. The TOmn output is not initialized but holds current status.
	The TOEmn bit is cleared to 0 and value is set to the TOmn bit. →	The TOmn pin outputs the TOmn bit set level.

Operation is resumed.

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

Figure 6 - 52 Operation Procedure of Interval Timer/Square Wave Output Function (2/2)

	Software Operation	Hardware Status
TAU stop	To hold the TOmn pin output level Clears the TOmn bit to 0 after the value to be held is set to the port register. →	The TOmn pin output level is held by port function.
	When holding the TOmn pin output level is not necessary Setting not required.	
	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmn bit is cleared to 0 and the TOmn pin is set to port mode.)

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 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 TImn pin. When a specified count value is reached, the event counter generates an interrupt. The specified number of counts can be calculated by the following expression.

Specified number of counts = Set value of TDRmn + 1

Timer count register mn (TCRmn) operates as a down counter in the event counter mode.

The TCRmn register loads the value of timer data register mn (TDRmn) by setting any channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) to 1.

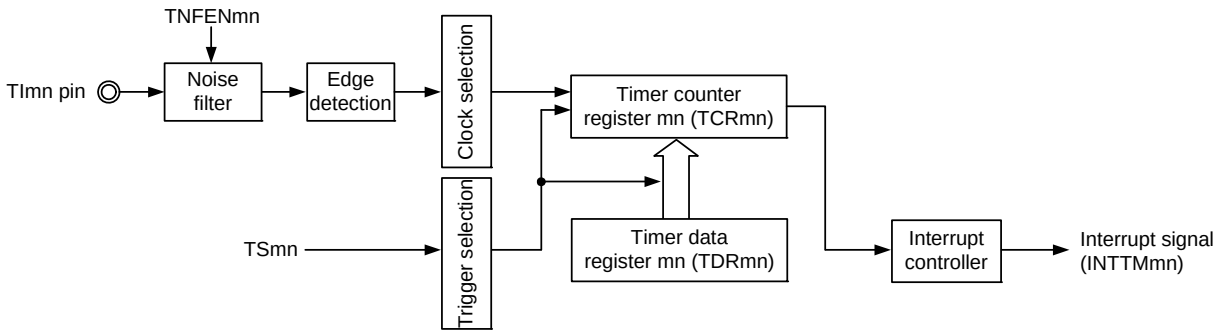
The TCRmn register counts down each time the valid input edge of the TImn pin has been detected. When TCRmn = 0000H, the TCRmn register loads the value of the TDRmn register again, and outputs INTTMmn.

After that, the above operation is repeated.

An irregular waveform that depends on external events is output from the TOmn pin. Stop the output by setting the TOEmn bit of timer output enable register m (TOEm) to 0.

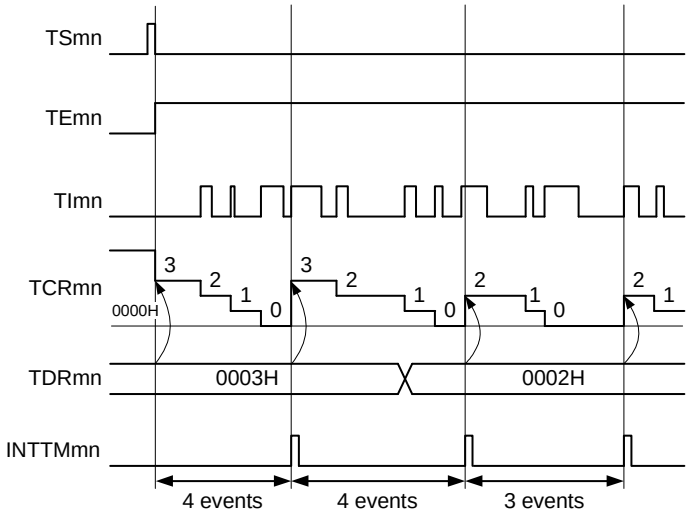
The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid during the next count period.

Figure 6 - 53 Block Diagram of Operation as External Event Counter



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6 - 54 Example of Basic Timing of Operation as External Event Counter

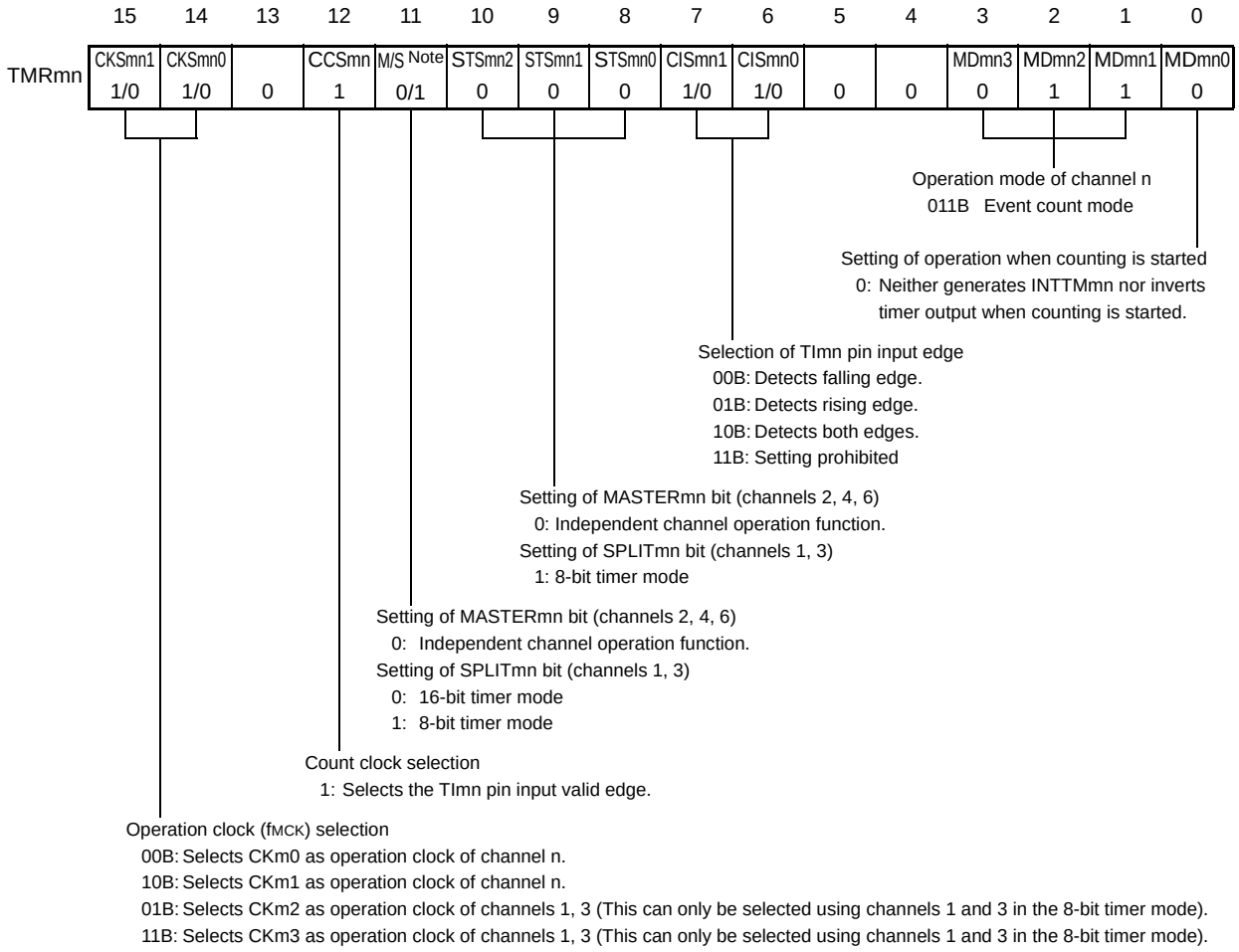


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

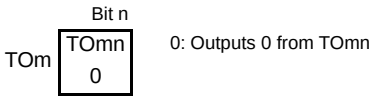
Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)

Figure 6 - 55 Example of Set Contents of Registers in External Event Counter Mode

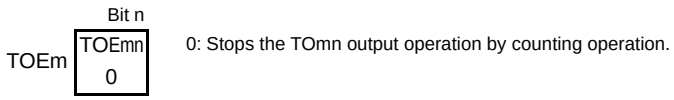
(a) Timer mode register mn (TMRmn)



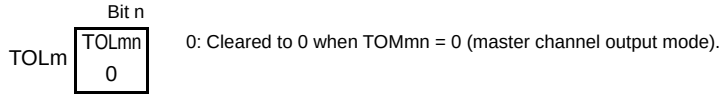
(b) Timer output register m (TOm)



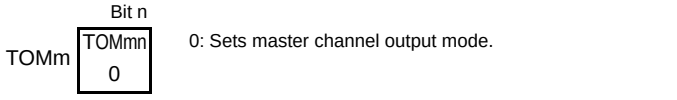
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



Note TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6 - 56 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 TAUmEN 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 m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). Sets number of counts to timer data register mn (TDRmn). Clears the TOEmn bit of timer output enable register m (TOEm) to 0.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. Value of the TDRmn register is loaded to timer count register mn (TCRmn) and detection of the TImn pin input edge is awaited.
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TMRmn register, TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed.	Counter (TCRmn) counts down each time input edge of the TImn pin has been detected. When count value reaches 0000H, the value of the TDRmn register is loaded to the TCRmn register again, and the count operation is continued. By detecting TCRmn = 0000H, the INTTMmn output is generated. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.8.3 Operation as input pulse interval measurement

The count value can be captured at the TImn valid edge and the interval of the pulse input to TImn can be measured. In addition, the count value can be captured by using software operation (TSmn = 1) as a capture trigger while the TEmn bit is set to 1.

The pulse interval can be calculated by the following expression.

TImn input pulse interval = Period of count clock × ((10000H × TSRmn: OVF) + (Capture value of TDRmn + 1))

Caution The TImn pin input is sampled using the operating clock selected with the CKSmn bit of timer mode register mn (TMRmn), so an error of up to one operating clock cycle occurs.

Timer count register mn (TCRmn) operates as an up counter in the capture mode.

When the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, the TCRmn register counts up from 0000H in synchronization with the count clock.

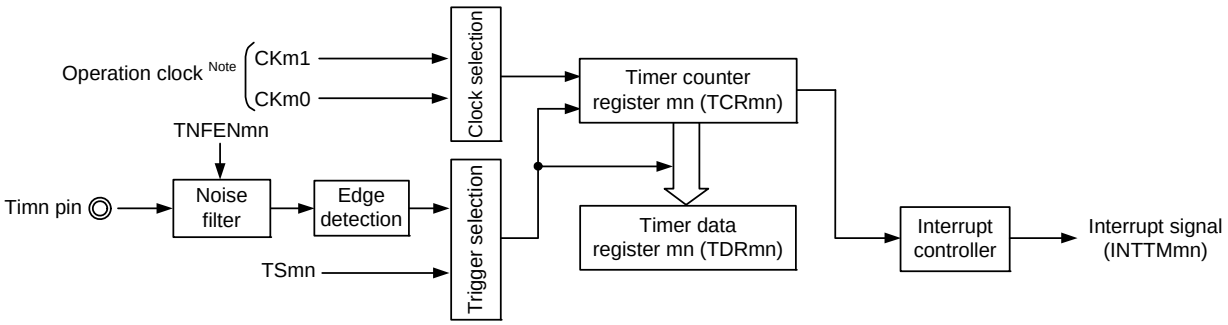
When the TImn pin input valid edge is detected, the count value of the TCRmn register is transferred (captured) to timer data register mn (TDRmn) and, at the same time, the TCRmn register is cleared to 0000H, and the INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status register mn (TSRmn) is set to 1. If the counter does not overflow, the OVF bit is cleared. After that, the above operation is repeated.

As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSRmn register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

Set the STSmn2 to STSmn0 bits of the TMRmn register to 001B to use the valid edges of TImn as a start trigger and a capture trigger.

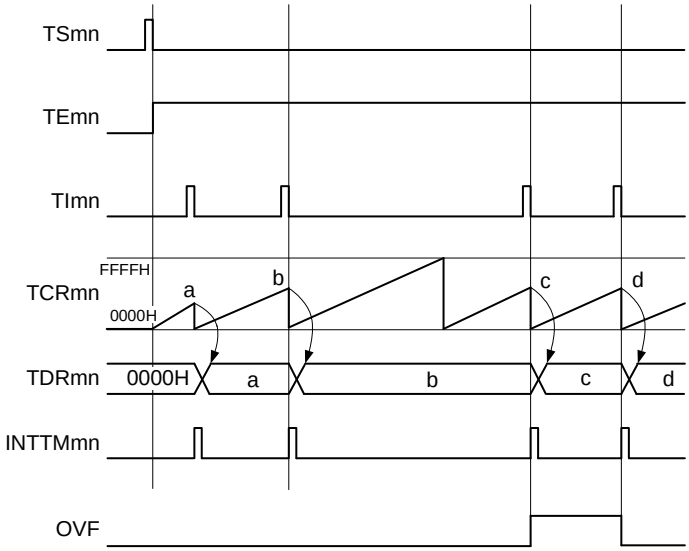
Figure 6 - 57 Block Diagram of Operation as Input Pulse Interval Measurement



Note When channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6 - 58 Example of Basic Timing of Operation as Input Pulse Interval Measurement (MDmn0 = 0)

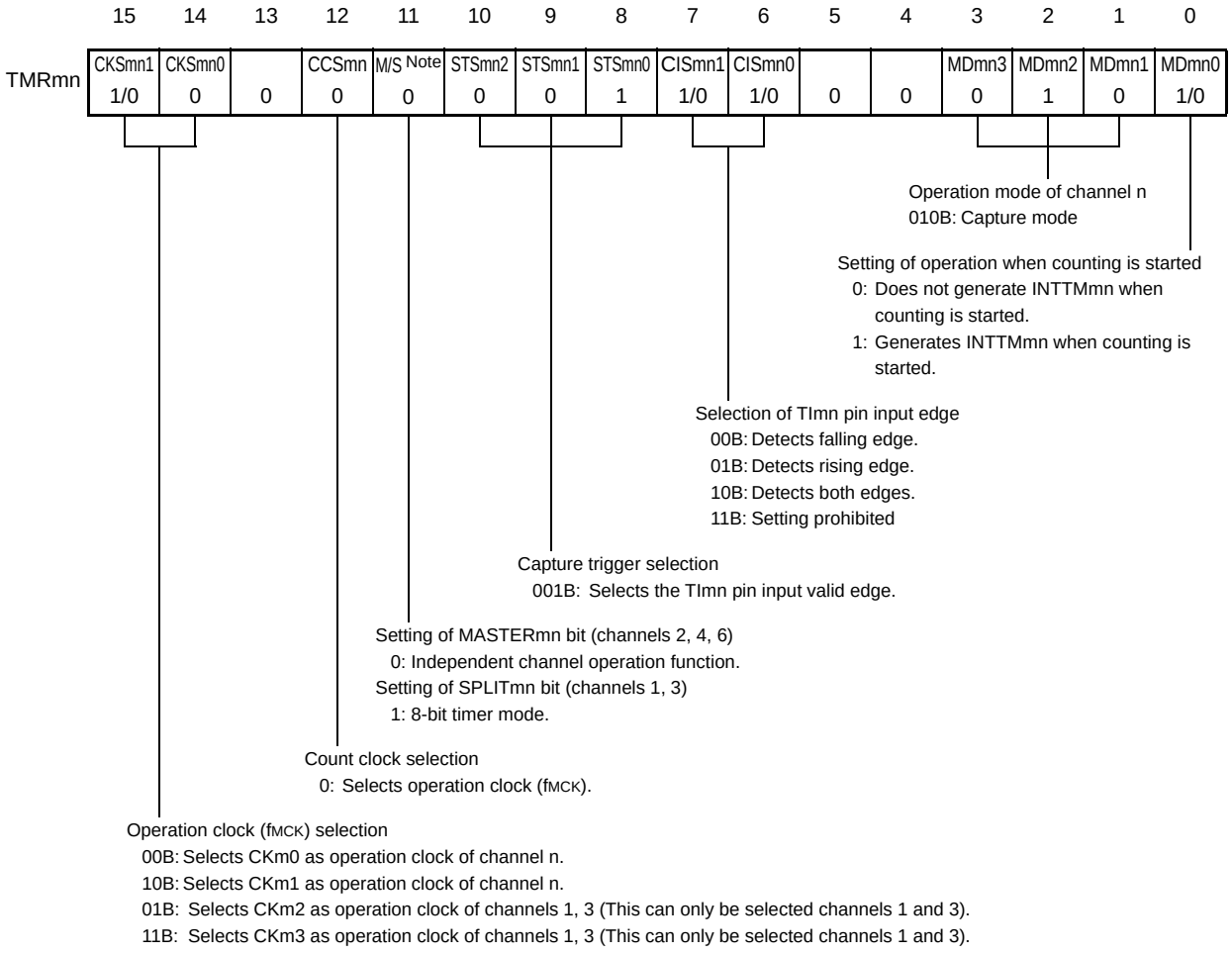


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

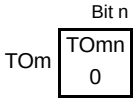
Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
OVF: Bit 0 of timer status register mn (TSRmn)

Figure 6 - 59 Example of Set Contents of Registers to Measure Input Pulse Interval

(a) Timer mode register mn (TMRmn)

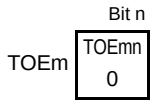


(b) Timer output register m (TOm)



0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

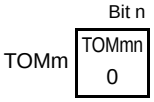


0: Stops TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)



0: Sets master channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6 - 60 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 TAUmEN 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 m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel).	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. Timer count register mn (TCRmn) is cleared to 0000H at the count clock input. When the MDmn0 bit of the TMRmn register is 1, INTTMmn is generated.
During operation	Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed. The TDRmn register can always be read. The TCRmn register can always be read. The TSRmn register can always be read. Set values of the TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed.	Counter (TCRmn) counts up from 0000H. When the valid edge of the TImn pin input is detected or the TSmn bit is set to 1, the count value is transferred (captured) to timer data register mn (TDRmn). At the same time, the TCRmn register is cleared to 0000H, and the INTTMmn signal is generated. If an overflow occurs at this time, the OVF bit of timer status register mn (TSRmn) is set; if an overflow does not occur, the OVF bit is cleared. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.8.4 Operation as input signal high-/low-level width measurement

Caution When using a channel to implement the LIN-bus, set bit 1 (ISC1) of the input switch control register (ISC) to 1. In the following descriptions, read TImn as RxD0.

By starting counting at one edge of the TImn pin input and capturing the number of counts at another edge, the signal width (high-level width/low-level width) of TImn can be measured. The signal width of TImn can be calculated by the following expression.

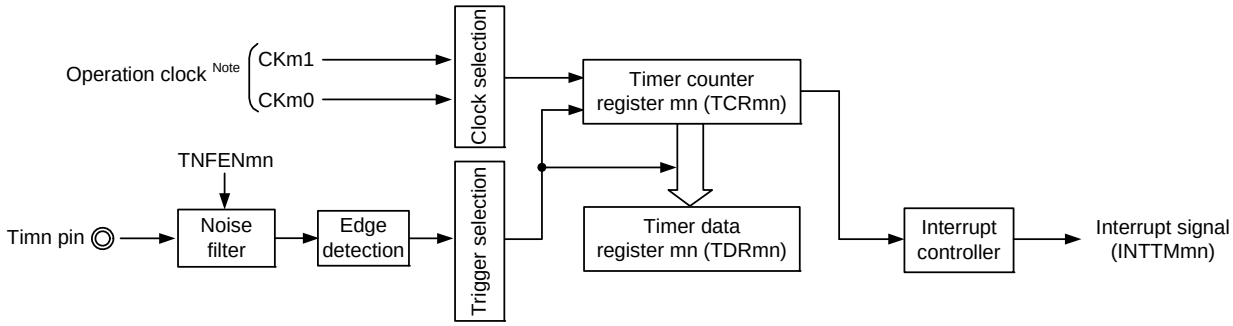
Signal width of TImn input = Period of count clock × ((10000H × TSRmn: OVF) + (Capture value of TDRmn + 1))

Caution The TImn pin input is sampled using the operating clock selected with the CKSmn bit of timer mode register mn (TMRmn), so an error equivalent to one operation clock occurs.

Timer count register mn (TCRmn) operates as an up counter in the capture & one-count mode.
When the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, the TEmn bit is set to 1 and the TImn pin start edge detection wait status is set.
When the TImn pin input start edge (rising edge of the TImn pin input when the high-level width is to be measured) is detected, the counter counts up from 0000H in synchronization with the count clock. When the valid capture edge (falling edge of the TImn pin input when the high-level width is to be measured) is detected later, the count value is transferred to timer data register mn (TDRmn) and, at the same time, INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status register mn (TSRmn) is set to 1. If the counter does not overflow, the OVF bit is cleared. The TCRmn register stops at the value “value transferred to the TDRmn register + 1”, and the TImn pin start edge detection wait status is set. After that, the above operation is repeated.
As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.
If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSRmn register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.
Whether the high-level width or low-level width of the TImn pin is to be measured can be selected by using the CISmn1 and CISmn0 bits of the TMRmn register.
Because this function is used to measure the signal width of the TImn pin input, the TSmn bit cannot be set to 1 while the TEmn bit is 1.

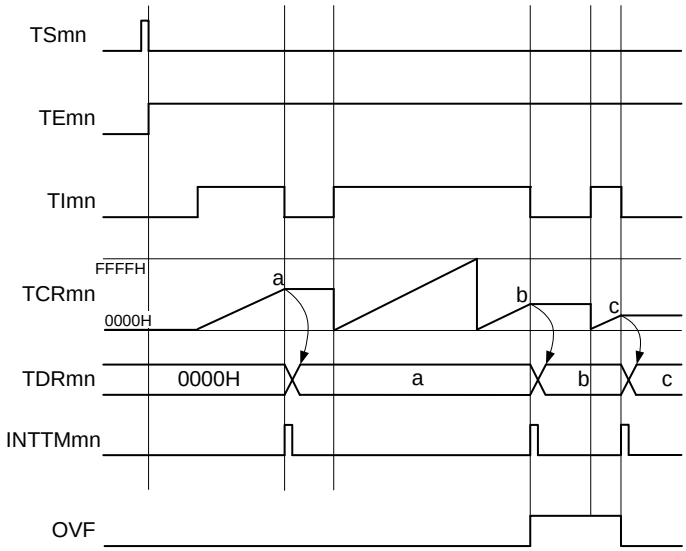
CISmn1, CISmn0 of TMRmn register = 10B: Low-level width is measured.
CISmn1, CISmn0 of TMRmn register = 11B: High-level width is measured.

Figure 6 - 61 Block Diagram of Operation as Input Signal High-/Low-Level Width Measurement



Note For channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Figure 6 - 62 Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement

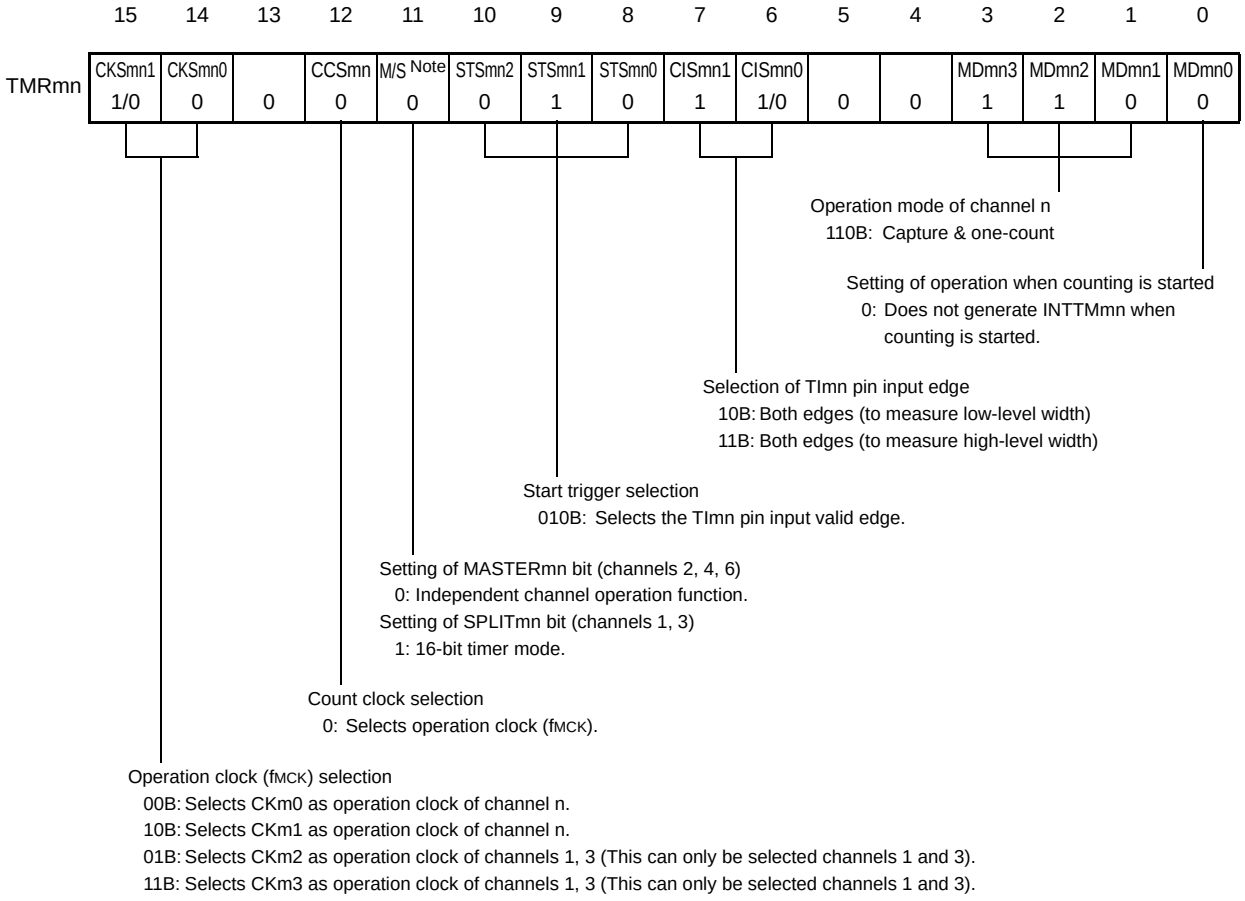


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
OVF: Bit 0 of timer status register mn (TSRmn)

Figure 6 - 63 Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6 - 64 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 TAUmEN 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 m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). Clears the TOEmn bit to 0 and stops operation of TOmn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and the TImn pin start edge detection wait status is set.
	Detects the TImn pin input count start valid edge. →	Clears timer count register mn (TCRmn) to 0000H and starts counting up.
During operation	Set value of the TDRmn register can always be read. The TCRmn register can always be read. The TSRmn register can always be read. Set values of the TMRmn register, TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed.	When the TImn pin start edge is detected, the counter (TCRmn) counts up from 0000H. If a capture edge of the TImn pin is detected, the count value is transferred to timer data register mn (TDRmn) and INTTMmn is generated. If an overflow occurs at this time, the OVF bit of timer status register mn (TSRmn) is set; if an overflow does not occur, the OVF bit is cleared. The TCRmn register stops the count operation until the next TImn pin start edge is detected.
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.8.5 Operation as delay counter

It is possible to start counting down when the valid edge of the TImn pin input is detected (an external event), and then generate INTTMmn (a timer interrupt) after any specified interval.

It is also possible to start counting down and generate INTTMmn (timer interrupt) at any interval by setting TSmn to 1 by software while TEmn = 1.

The interrupt generation period can be calculated by the following expression.

Generation period of INTTMmn (timer interrupt) = Period of count clock × (Set value of TDRmn + 1)

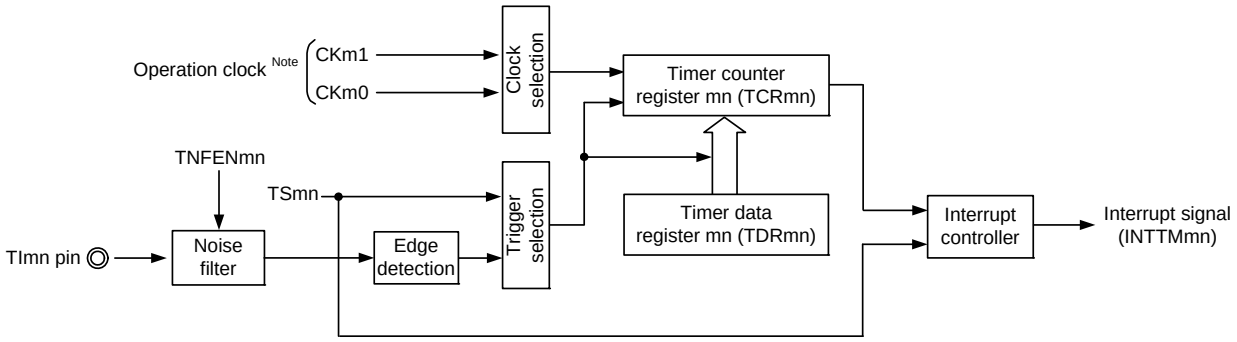
Timer count register mn (TCRmn) operates as a down counter in the one-count mode.

When the channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) is set to 1, the TEmn, TEHm1, TEHm3 bits are set to 1 and the TImn pin input valid edge detection wait status is set.

Timer count register mn (TCRmn) starts operating upon TImn pin input valid edge detection and loads the value of timer data register mn (TDRmn). The TCRmn register counts down from the value of the TDRmn register it has loaded, in synchronization with the count clock. When TCRmn = 0000H, it outputs INTTMmn and stops counting until the next TImn pin input valid edge is detected.

The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid from the next period.

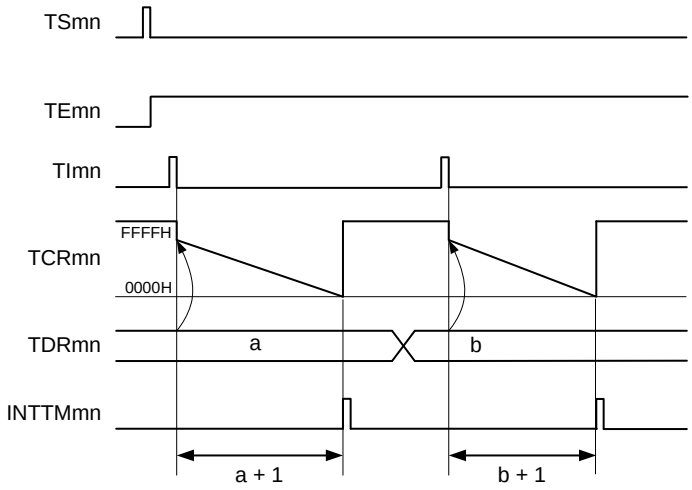
Figure 6 - 65 Block Diagram of Operation as Delay Counter



Note For using channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6 - 66 Example of Basic Timing of Operation as Delay Counter

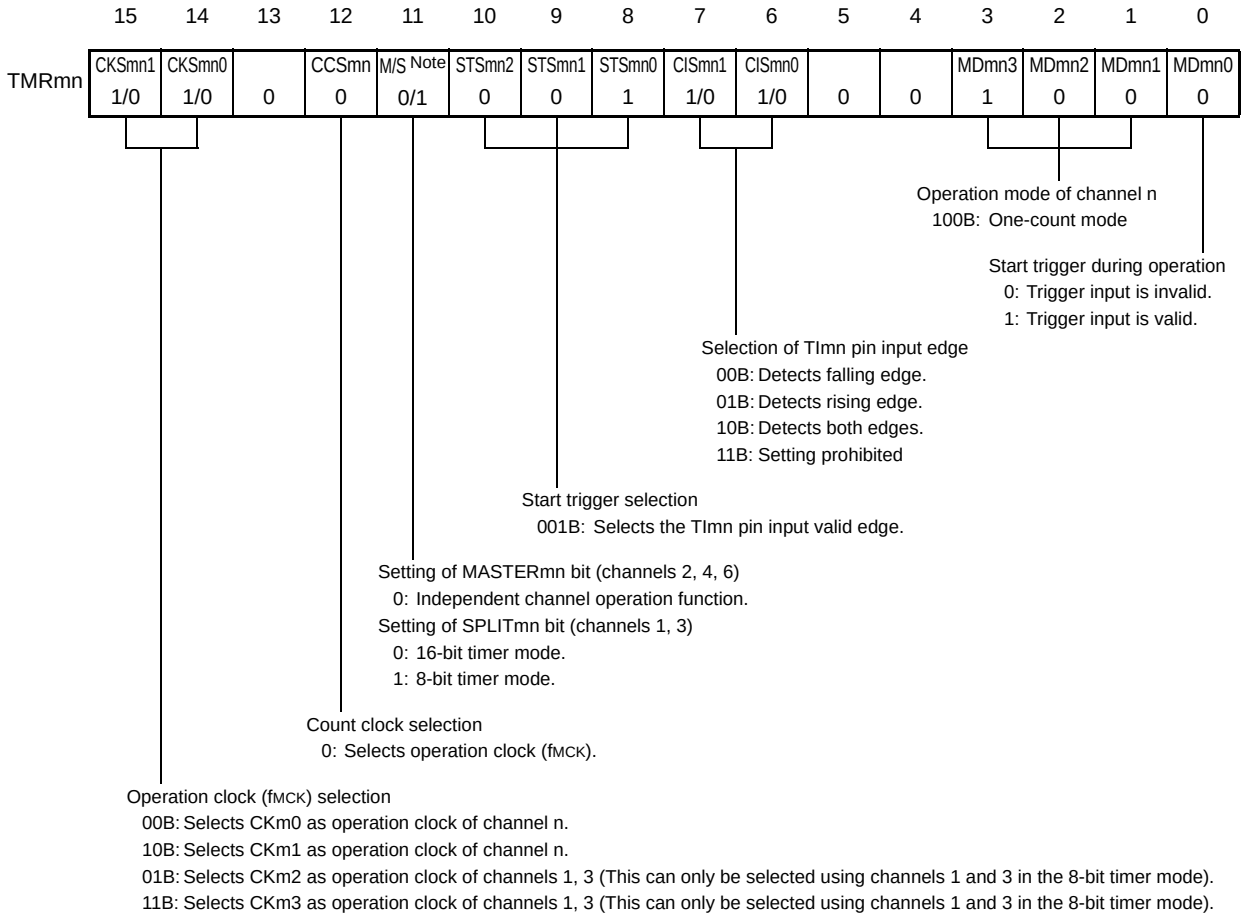


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

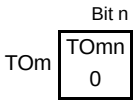
Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)

Figure 6 - 67 Example of Set Contents of Registers to Delay Counter

(a) Timer mode register mn (TMRmn)

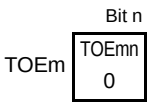


(b) Timer output register m (TOm)



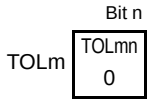
0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)



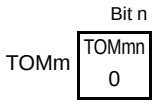
0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)



0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)



0: Sets master channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6 - 68 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 TAUmEN 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 m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). INTTMmn output delay is set to timer data register mn (TDRmn). Clears the TOEmn bit to 0 and stops operation of TOmn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit is set to 1) wait status is set.
	Detects the TImn pin input valid edge. →	Value of the TDRmn register is loaded to the timer count register mn (TCRmn).
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used.	The counter (TCRmn) counts down. When the count value of TCRmn reaches 0000H, the INTTMmn output is generated, and the count operation stops until the next start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit is set to 1).
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.9 Simultaneous Channel Operation Function of Timer Array Unit

6.9.1 Operation as one-shot pulse output function

By using two channels as a set, a one-shot pulse having any delay pulse width can be generated from the signal input to the TImn pin.

The delay time and pulse width can be calculated by the following expressions.

Delay time = {Set value of TDRmn (master) + 2} × Count clock period
Pulse width = {Set value of TDRmp (slave)} × Count clock period

The master channel operates in the one-count mode and counts the delays. Timer count register mn (TCRmn) of the master channel starts operating upon start trigger detection and loads the value of timer data register mn (TDRmn).

The TCRmn register counts down from the value of the TDRmn register it has loaded, in synchronization with the count clock. When TCRmn = 0000H, it outputs INTTMmn and stops counting until the next start trigger is detected.

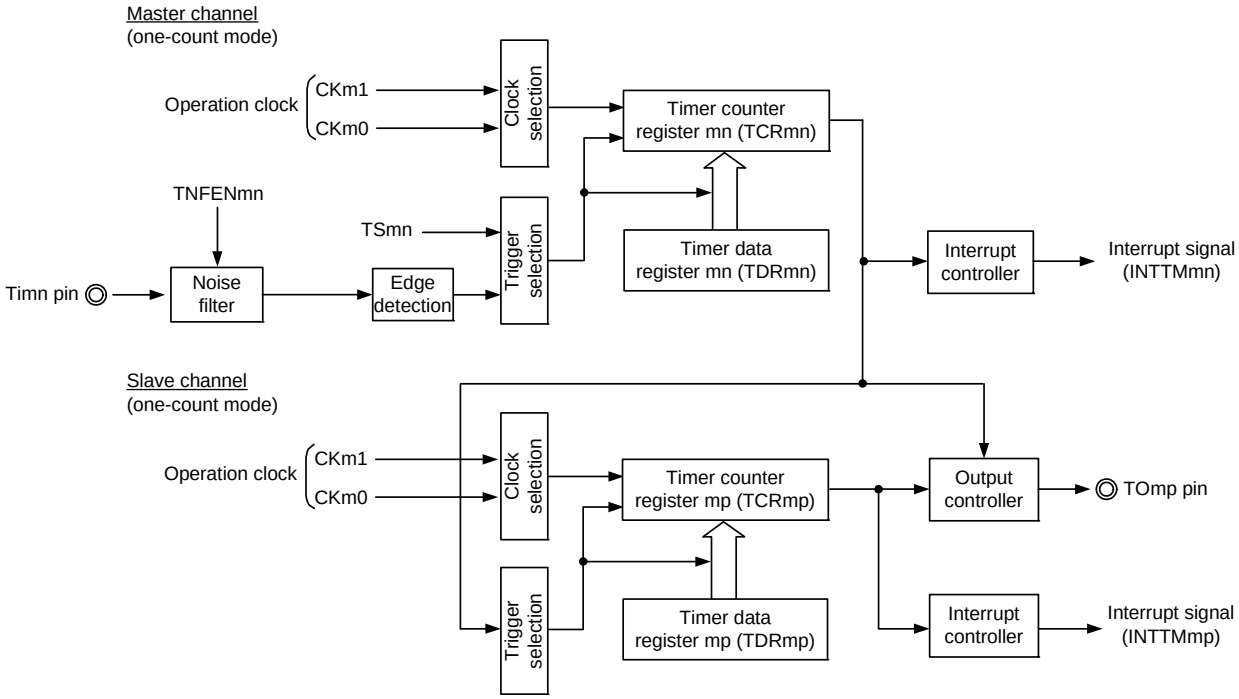
The slave channel operates in the one-count mode and counts the pulse width. The TCRmp register of the slave channel starts operation using INTTMmn of the master channel as a start trigger, and loads the value of the TDRmp register. The TCRmp register counts down from the value of The TDRmp register it has loaded, in synchronization with the count value. When count value = 0000H, it outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) is detected. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

Instead of using the TImn pin input, a one-shot pulse can also be output using the software operation (TSmn = 1) as a start trigger.

Caution **The timing of loading of timer data register mn (TDRmn) of the master channel is different from that of the TDRmp register of the slave channel. If the TDRmn and TDRmp registers are rewritten during counting, therefore, an illegal waveform may be output in conflict with the timing of loading. Rewrite the TDRmn register after INTTMmn is generated and the TDRmp register after INTTMmp is generated.**

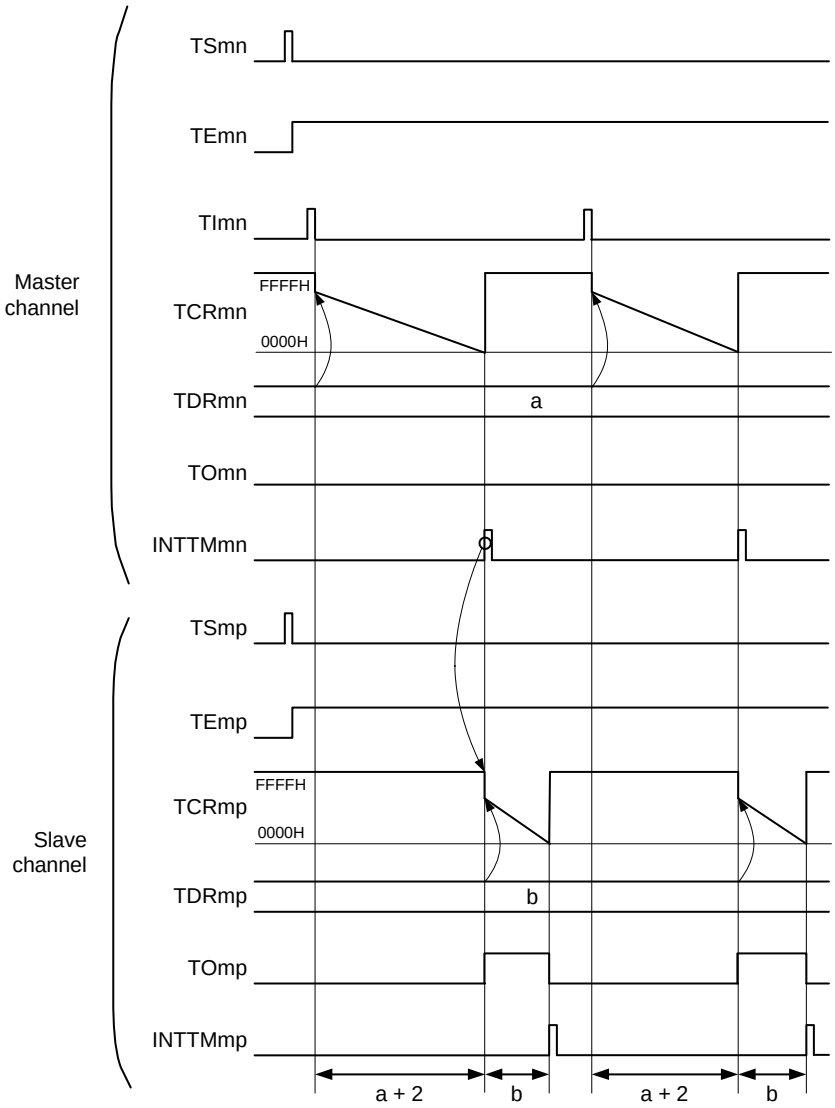
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6 - 69 Block Diagram of Operation as One-Shot Pulse Output Function



Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6))
p: Slave channel number (n < p ≤ 7)

Figure 6 - 70 Example of Basic Timing of Operation as One-Shot Pulse Output Function

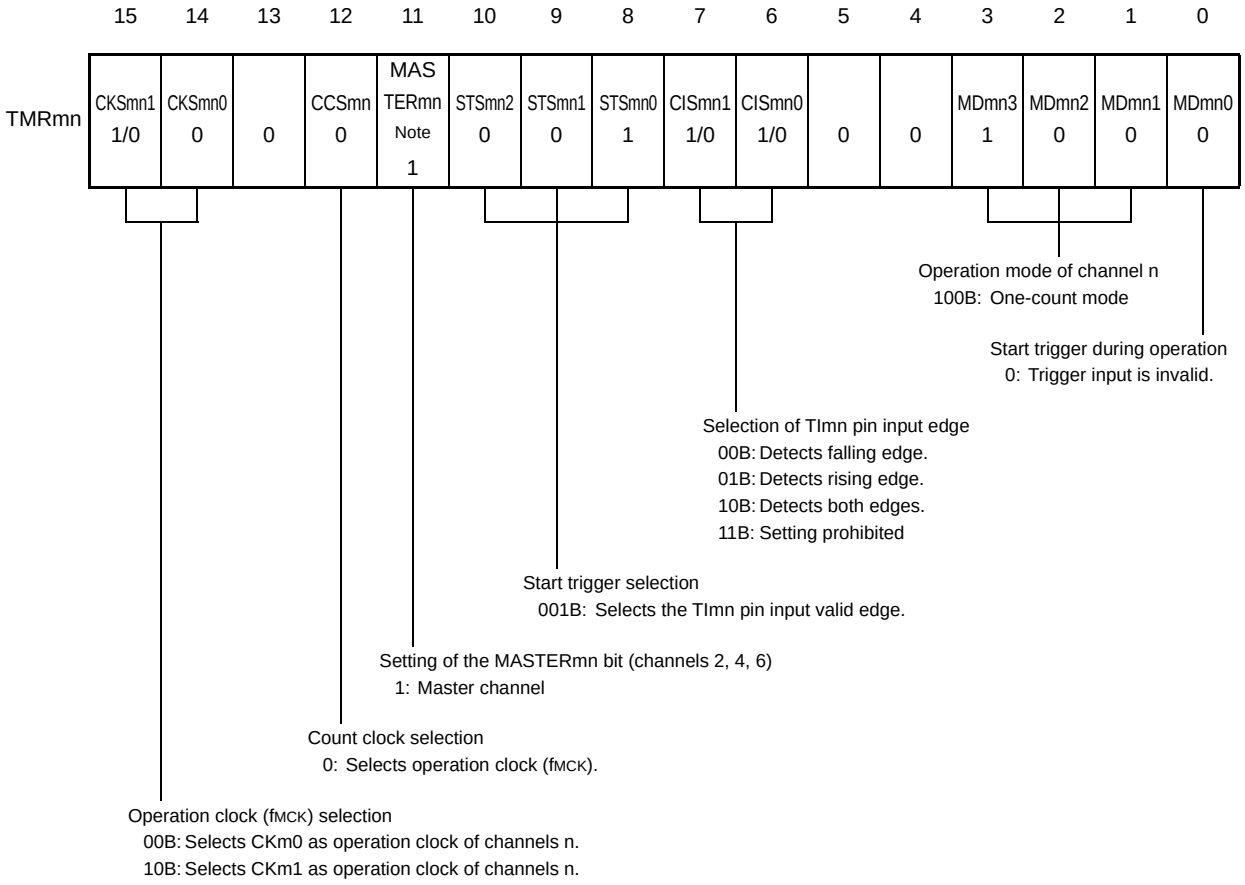


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Remark 2. TSmn, TSmp: Bit n, p of timer channel start register m (TSm)
TEmn, TEmp: Bit n, p of timer channel enable status register m (TEm)
TImn, TImp: TImn and TImp pins input signal
TCRmn, TCRmp: Timer count registers mn, mp (TCRmn, TCRmp)
TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)
TOmn, TOmp: TOmn and TOmp pins output signal

Figure 6 - 71 Example of Set Contents of Registers
When One-Shot Pulse Output Function Is Used (Master Channel)

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

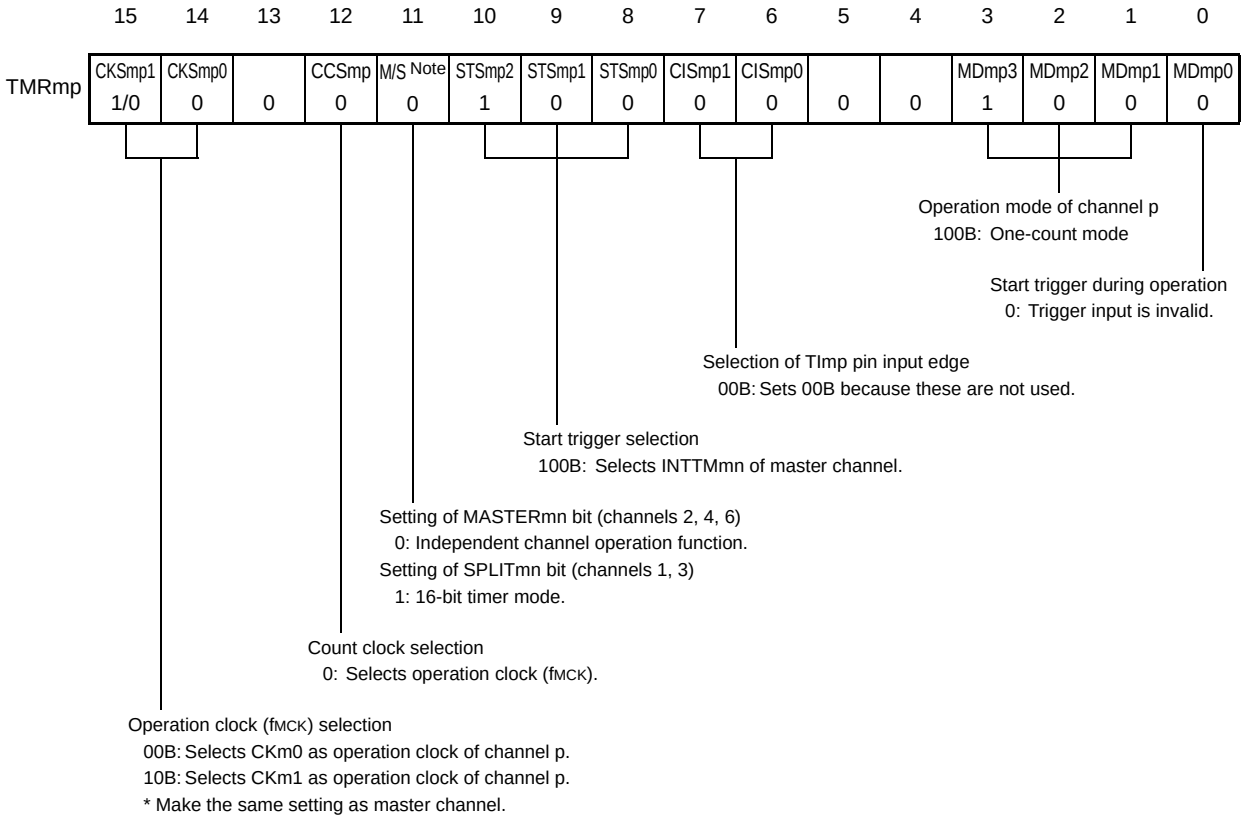
	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn = 1
TMRm0: Fixed to 0

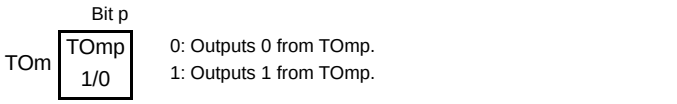
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)

Figure 6 - 72 Example of Set Contents of Registers
When One-Shot Pulse Output Function Is Used (Slave Channel)

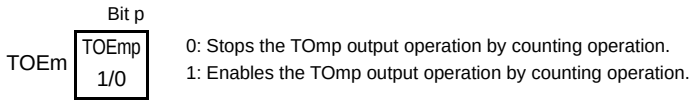
(a) Timer mode register mp (TMRmp)



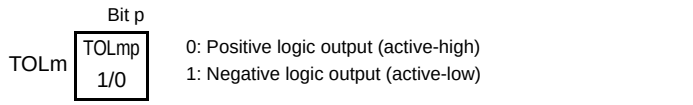
(b) Timer output register m (TOm)



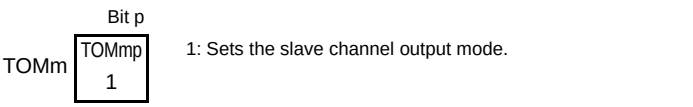
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



Note TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmp bit
TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6 - 73 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 TAUmEN 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 m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets the corresponding bit of the noise filter enable registers 1 (NFEN1) to 1. Sets timer mode register mn, mp (TMRmn, TMRmp) of two channels to be used (determines operation mode of channels). An output delay is set to timer data register mn (TDRmn) of the master channel, and a pulse width is set to the TDRmp register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMmp bit of timer output mode register m (TOMm) is set to 1 (slave channel output mode). Sets the TOLmp bit. Sets the TOmp bit and determines default level of the TOmp output. →	The TOmp pin goes into Hi-Z output state. The TOmp default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmp bit to 1 and enables operation of TOmp. →	TOmp does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmp pin outputs the TOmp set level.

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

Figure 6 - 74 Operation Procedure of One-Shot Pulse Output Function (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets the TOEmp bit (slave) to 1 (only when operation is resumed). →	The TEmn and TEmp bits are set to 1 and the master channel enters the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit of the master channel is set to 1) wait status. Counter stops operating.
	The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	
	Count operation of the master channel is started by start trigger detection of the master channel. • Detects the TImn pin input valid edge. • Sets the TSmn bit of the master channel to 1 by software Note. Note Do not set the TSmn bit of the slave channel to 1. →	Master channel starts counting.
During operation	Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed. Set values of the TMRmp, TDRmn, TDRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used. Set values of the TOm and TOEm registers by slave channel can be changed.	Master channel loads the value of the TDRmn register to timer count register mn (TCRmn) by the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit of the master channel is set to 1), and the counter starts counting down. When the count value reaches TCRmn = 0000H, the INTTMmn output is generated, and the counter stops until the next start trigger detection. The slave channel, triggered by INTTMmn of the master channel, loads the value of the TDRmp register to the TCRmp register, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. →	TEmn, TEmp = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized but holds current status.
	The TOEmp bit of slave channel is cleared to 0 and value is set to the TOmp bit. →	The TOmp pin outputs the TOmp set level.
TAU stop	To hold the TOmp pin output level Clears the TOmp bit to 0 after the value to be held is set to the port register. →	The TOmp pin output level is held by port function.
	When holding the TOmp pin output level is not necessary Setting not required.	
	The TAUmEN 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 TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

6.9.2 Operation as PWM function

Two channels can be used as a set to generate a pulse of any period and duty factor.
The period and duty factor of the output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDRmn (master) + 1} × Count clock period
Duty factor [%] = {Set value of TDRmp (slave)}/{Set value of TDRmn (master) + 1} × 100
0% output: Set value of TDRmp (slave) = 0000H
100% output: Set value of TDRmp (slave) ≥ {Set value of TDRmn (master) + 1}

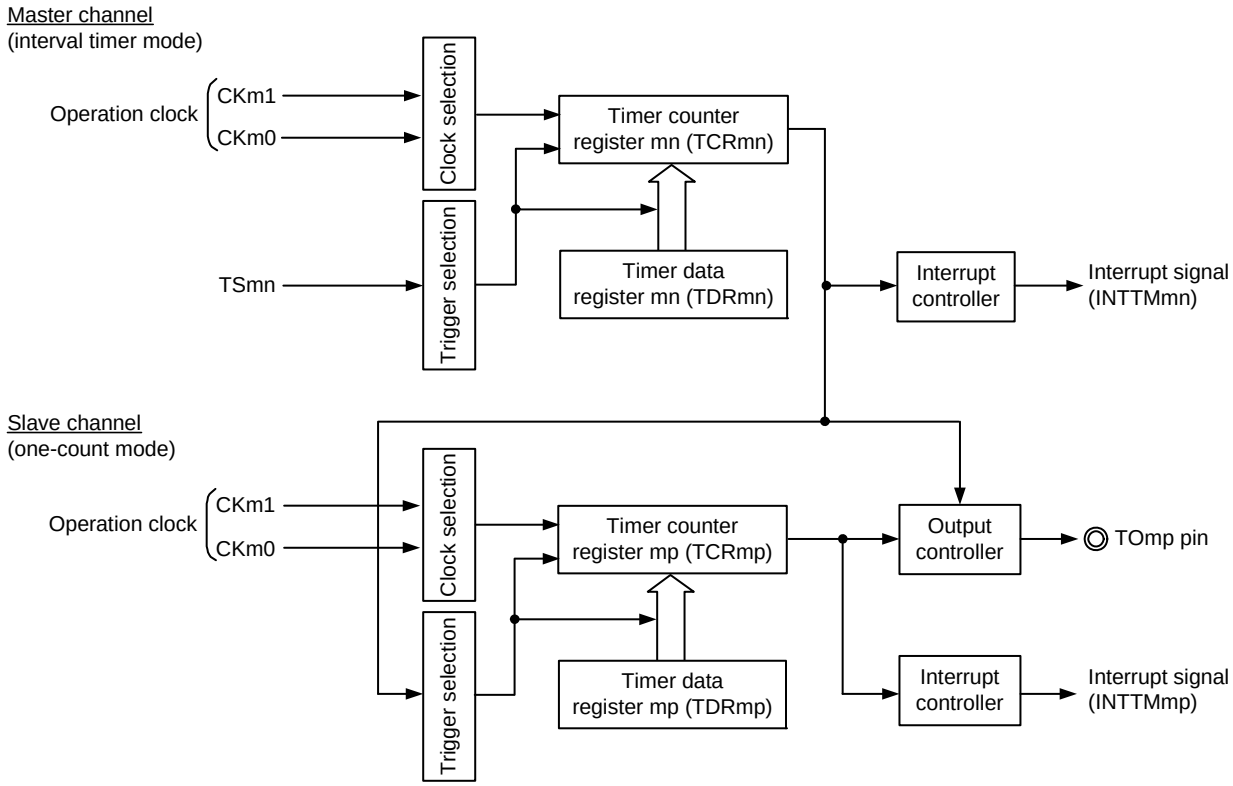
Remark The duty factor exceeds 100% if the set value of TDRmp (slave) > (set value of TDRmn (master) + 1), it summarizes to 100% output.

The master channel operates in the interval timer mode. If the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, an interrupt (INTTMmn) is output, the value set to timer data register mn (TDRmn) is loaded to timer count register mn (TCRmn), and the counter counts down in synchronization with the count clock. When the counter reaches 0000H, INTTMmn is output, the value of the TDRmn register is loaded again to the TCRmn register, and the counter counts down. This operation is repeated until the channel stop trigger bit (TTmn) of timer channel stop register m (TTm) is set to 1.
If two channels are used to output a PWM waveform, the period until the master channel counts down to 0000H is the PWM output (TOmp) cycle.
The slave channel operates in one-count mode. By using INTTMmn from the master channel as a start trigger, the TCRmp register loads the value of the TDRmp register and the counter counts down to 0000H. When the counter reaches 0000H, it outputs INTTMmp and waits until the next start trigger (INTTMmn from the master channel) is generated.
If two channels are used to output a PWM waveform, the period until the slave channel counts down to 0000H is the PWM output (TOmp) duty.
PWM output (TOmp) goes to the active level one clock after the master channel generates INTTMmn and goes to the inactive level when the TCRmp register of the slave channel becomes 0000H.

Caution To rewrite both timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel, a write access is necessary two times. The timing at which the values of the TDRmn and TDRmp registers are loaded to the TCRmn and TCRmp registers is upon occurrence of INTTMmn of the master channel. Thus, when rewriting is performed split before and after occurrence of INTTMmn of the master channel, the TOmp pin cannot output the expected waveform. To rewrite both the TDRmn register of the master and the TDRmp register of the slave, therefore, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel.

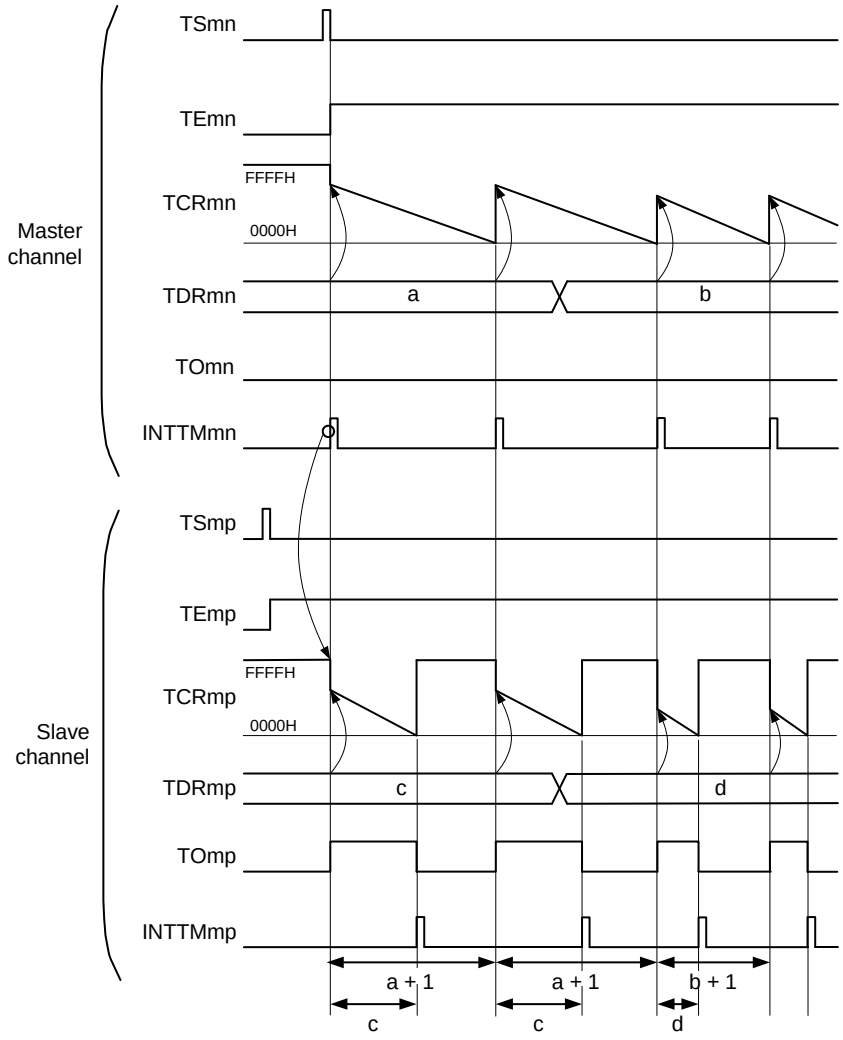
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6 - 75 Block Diagram of Operation as PWM Function



Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6 - 76 Example of Basic Timing of Operation as PWM Function

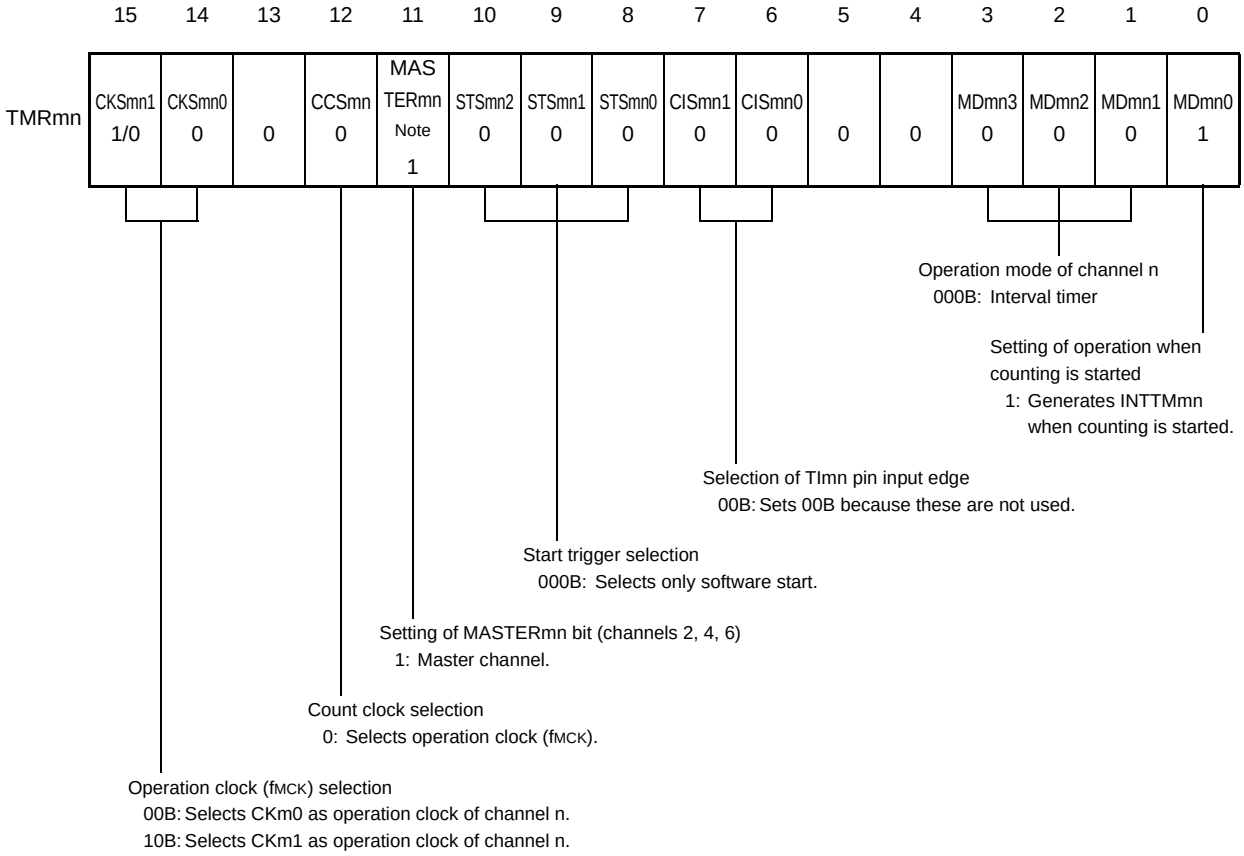


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

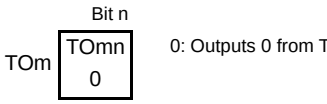
Remark 2. TSmn, TSmp: Bit n, p of timer channel start register m (TSm)
TEmn, TEmp: Bit n, p of timer channel enable status register m (TEm)
TCRmn, TCRmp: Timer count registers mn, mp (TCRmn, TCRmp)
TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)
TOmn, TOmp: TOmn and TOmp pins output signal

Figure 6 - 77 Example of Set Contents of Registers When PWM Function (Master Channel) Is Used

(a) Timer mode register mn (TMRmn)

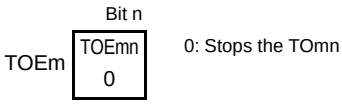


(b) Timer output register m (TOm)



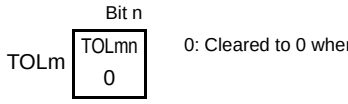
0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)



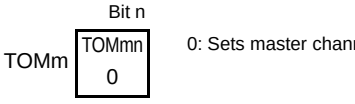
0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)



0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)



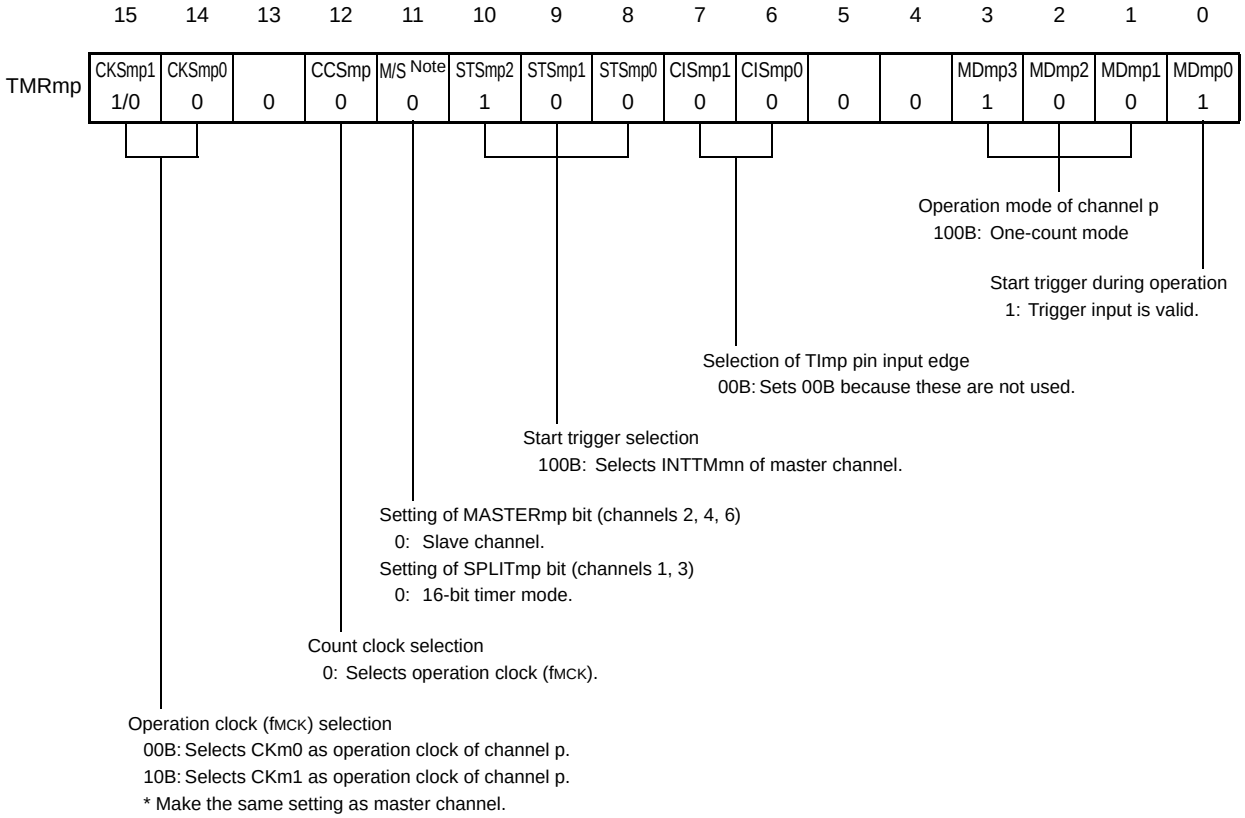
0: Sets master channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)

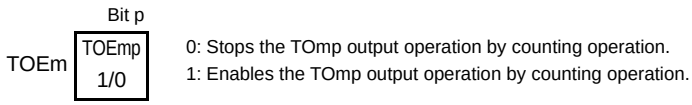
Figure 6 - 78 Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used

(a) Timer mode register mp (TMRmp)

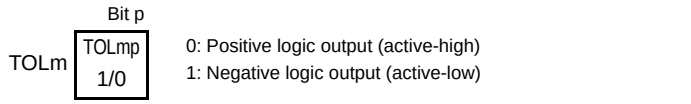


(b) Timer output register m (TOm)

(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)

Note TMRm2, TMRm4 TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmp bit
TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6 - 79 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 TAUmEN 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 m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets timer mode registers mn, mp (TMRmn, TMRmp) of two channels to be used (determines operation mode of channels). An interval (period) value is set to timer data register mn (TDRmn) of the master channel, and a duty factor is set to the TDRmp register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMmp bit of timer output mode register m (TOMm) is set to 1 (slave channel output mode). Sets the TOLmp bit. Sets the TOmp bit and determines default level of the TOmp output. →	The TOmp pin goes into Hi-Z output state. The TOmp default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmp bit to 1 and enables operation of TOmp. →	TOmp does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmp pin outputs the TOmp set level.

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

Figure 6 - 80 Operation Procedure When PWM Function Is Used (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets the TOEmp bit (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. → The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	TEmn = 1, TEmp = 1 → When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	Set values of the TMRmn and TMRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. Set values of the TDRmn and TDRmp registers can be changed after INTTMmn of the master channel is generated. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used.	The counter of the master channel loads the TDRmn register value to timer count register mn (TCRmn), and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to the TCRmn register, and the counter starts counting down again. At the slave channel, the value of the TDRmp register is loaded to the TCRmp register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. → The TTmn and TTmp bits automatically return to 0 because they are trigger bits.	TEmn, TEmp = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized but holds current status.
	The TOEmp bit of slave channel is cleared to 0 and value is set to the TOmp bit. →	The TOmp pin outputs the TOmp set level.
TAU stop	To hold the TOmp pin output level Clears the TOmp bit to 0 after the value to be held is set to the port register. →	The TOmp pin output level is held by port function.
	When holding the TOmp pin output level is not necessary Setting not required.	
	The TAUmEN 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 TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

6.9.3 Operation as multiple PWM output function

By extending the PWM function and using multiple slave channels, many PWM waveforms with different duty values can be output.

For example, when using two slave channels, the period and duty factor of an output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDRmn (master) + 1} × Count clock period
Duty factor 1 [%] = {Set value of TDRmp (slave 1)}/{Set value of TDRmn (master) + 1} × 100
Duty factor 2 [%] = {Set value of TDRmq (slave 2)}/{Set value of TDRmn (master) + 1} × 100

Remark Although the duty factor exceeds 100% if the set value of TDRmp (slave 1) > {set value of TDRmn (master) + 1} or if the {set value of TDRmq (slave 2)} > {set value of TDRmn (master) + 1}, it is summarized into 100% output.

Timer count register mn (TCRmn) of the master channel operates in the interval timer mode and counts the periods.

The TCRmp register of the slave channel 1 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmp pin. The TCRmp register loads the value of timer data register mp (TDRmp), using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmp = 0000H, TCRmp outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

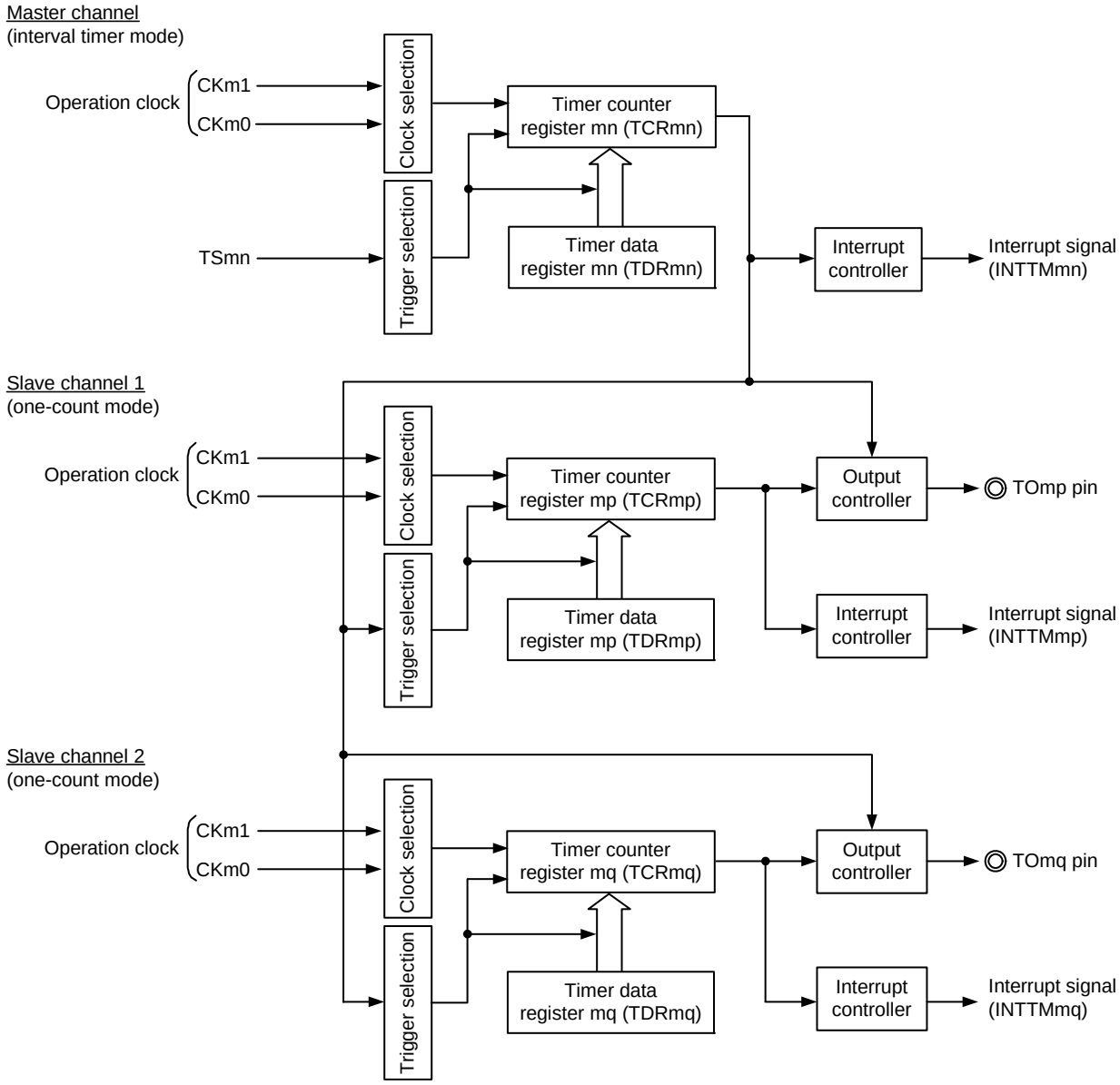
In the same way as the TCRmp register of the slave channel 1, the TCRmq register of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmq pin. The TCRmq register loads the value of the TDRmq register, using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmq = 0000H, the TCRmq register outputs INTTMmq and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmq becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmq = 0000H.

When channel 0 is used as the master channel as above, up to three types of PWM signals can be output at the same time.

Caution **To rewrite both timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel 1, write access is necessary at least twice. Since the values of the TDRmn and TDRmp registers are loaded to the TCRmn and TCRmp registers after INTTMmn is generated from the master channel, if rewriting is performed separately before and after generation of INTTMmn from the master channel, the TOmp pin cannot output the expected waveform. To rewrite both the TDRmn register of the master and the TDRmp register of the slave, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel (This applies also to the TDRmq register of the slave channel 2).**

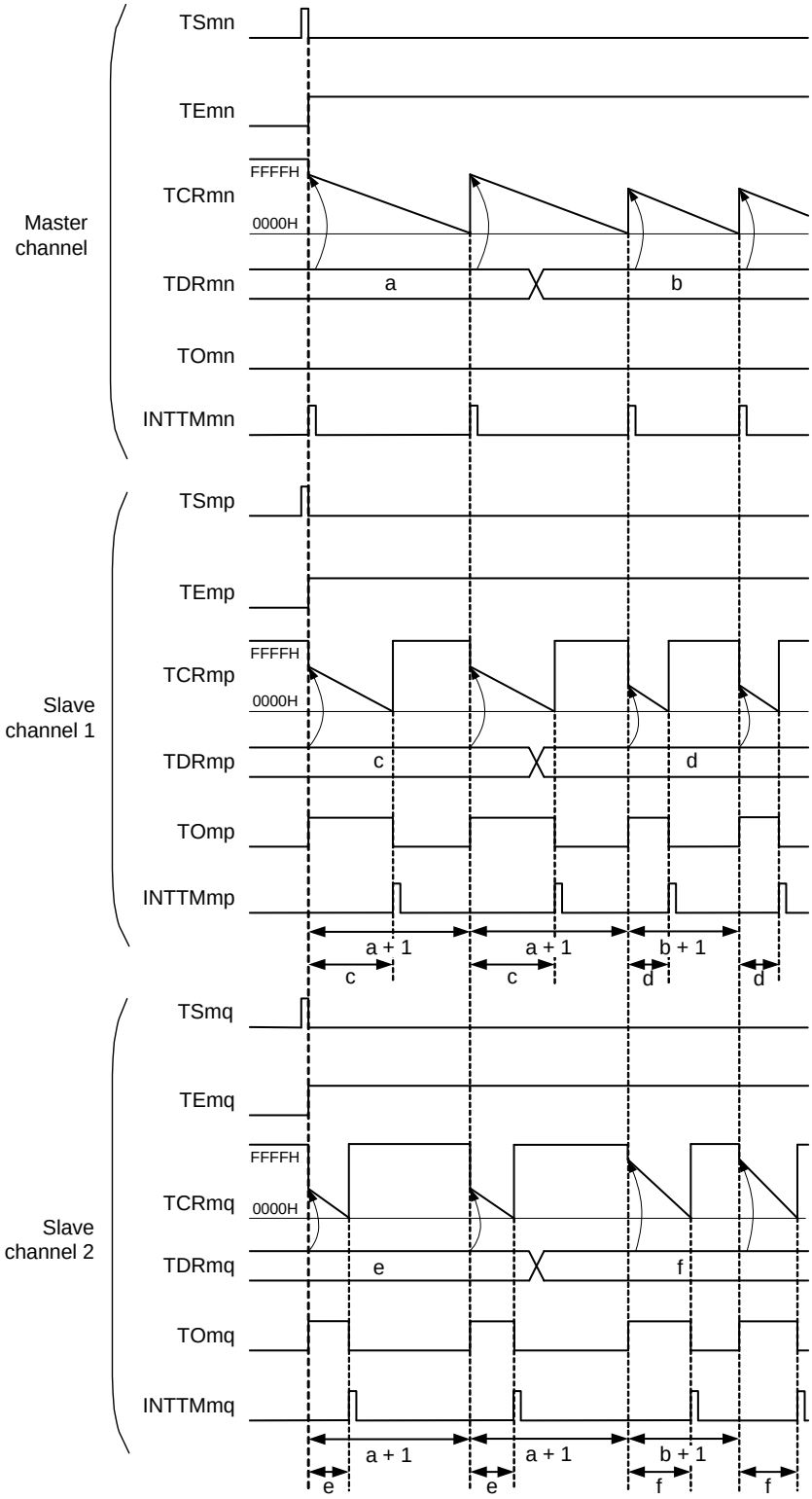
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 7 (Where p and q are integers greater than n)

Figure 6 - 81 Block Diagram of Operation as Multiple PWM Output Function (output two types of PWMs)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 7 (Where p and q are integers greater than n)

Figure 6 - 82 Example of Basic Timing of Operation as Multiple PWM Output Function (Output two types of PWMs) (1/2)



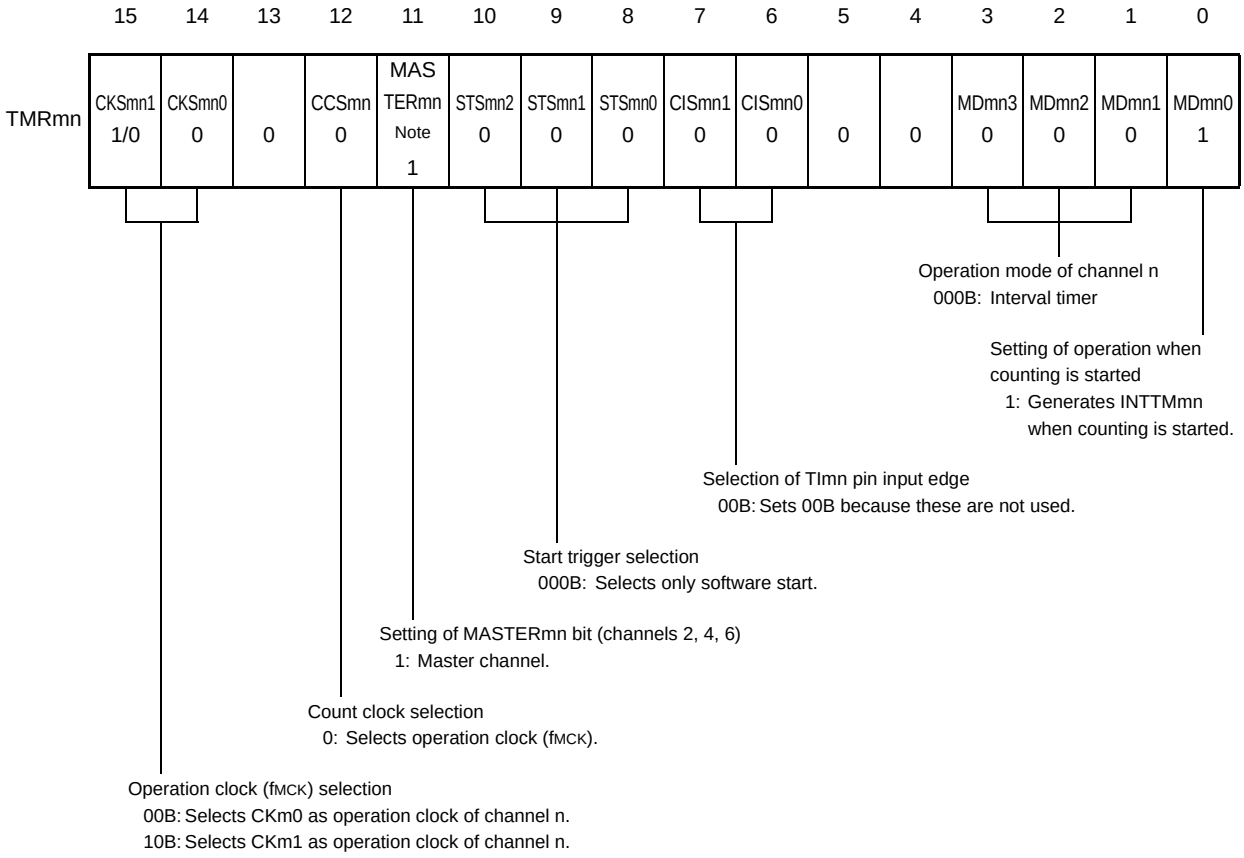
(**Remark** is listed on the next page.)

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 7 (Where p and q are integers greater than n)

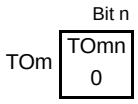
Remark 2. TSmn, TSmp, TSmq: Bit n, p, q of timer channel start register m (TSm)
TEmn, TEmp, TEmq: Bit n, p, q of timer channel enable status register m (TEm)
TCRmn, TCRmp, TCRmq: Timer count registers mn, mp, mq (TCRmn, TCRmp, TCRmq)
TDRmn, TDRmp, TDRmq: Timer data registers mn, mp, mq (TDRmn, TDRmp, TDRmq)
TOmn, TOmp, TOmq: TOmn, TOmp, and TOmq pins output signal

Figure 6 - 83 Example of Set Contents of Registers
When Multiple PWM Output Function (Master Channel) Is Used

(a) Timer mode register mn (TMRmn)

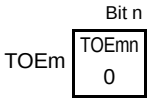


(b) Timer output register m (TOm)



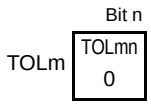
0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)



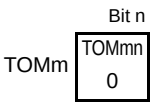
0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)



0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)



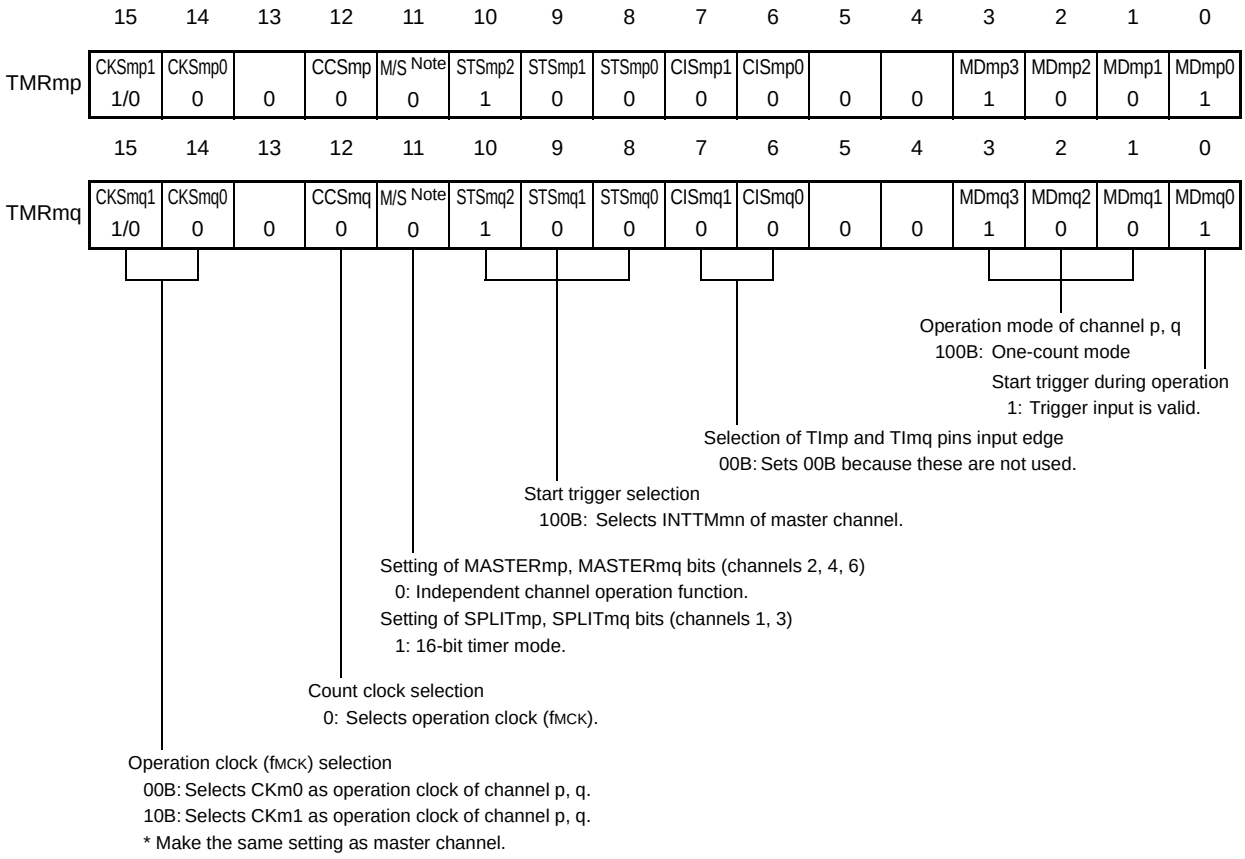
0: Sets master channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4)

Figure 6 - 84 Example of Set Contents of Registers
When Multiple PWM Output Function (Slave Channel) Is Used (output two types of PWMs)

(a) Timer mode register mp, mq (TMRmp, TMRmq)



(b) Timer output register m (TOm)

Bit q Bit p

TOm	TOmq	TOmp
	1/0	1/0

0: Outputs 0 from TOmp or TOmq.
1: Outputs 1 from TOmp or TOmq.

(c) Timer output enable register m (TOEm)

Bit q Bit p

TOEm	TOEmq	TOEmp
	1/0	1/0

0: Stops the TOmp or TOmq output operation by counting operation.
1: Enables the TOmp or TOmq output operation by counting operation.

(d) Timer output level register m (TOLm)

Bit q Bit p

TOLm	TOLmq	TOLmp
	1/0	1/0

0: Positive logic output (active-high)
1: Negative logic output (active-low)

(e) Timer output mode register m (TOMm)

Bit q Bit p

TOMm	TOMmq	TOMmp
	1	1

1: Sets the slave channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmp, MASTERmq bit
TMRm1, TMRm3: SPLITmp, SPLIT0q bit
TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 7 (Where p and q are integers greater than n)

Figure 6 - 85 Operation Procedure When Multiple PWM Output Function Is Used (output two types of PWMs) (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 TAUmEN 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 m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets timer mode registers mn, mp, mq (TMRmn, TMRmp, TMRmq) of each channel to be used (determines operation mode of channels). An interval (period) value is set to timer data register mn (TDRmn) of the master channel, and a duty factor is set to the TDRmp and TDRmq registers of the slave channels.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channels. The TOMmp and TOMmq bits of timer output mode register m (TOMm) are set to 1 (slave channel output mode). Clears the TOLmp and TOLmq bits to 0. Sets the TOmp and TOmq bits and determines default level of the TOmp and TOmq outputs. →	The TOmp and TOmq pins go into Hi-Z output state. The TOmp and TOmq default setting levels are output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmp and TOEmq bits to 1 and enables operation of TOmp and TOmq. →	TOmp and TOmq do not change because channels stop operating.
	Clears the port register and port mode register to 0. →	The TOmp and TOmq pins output the TOmp and TOmq set levels.

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

Figure 6 - 86 Operation Procedure When Multiple PWM Output Function Is Used (output two types of PWMs) (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	(Sets the TOEmp and TOEmq (slave) bits to 1 only when resuming operation.) The TSmn bit (master), and TSmp and TSmq (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. → The TSmn, TSmp, and TSmq bits automatically return to 0 because they are trigger bits.	TEmn = 1, TEmp, TEmq = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	Set values of the TMRmn, TMRmp, TMRmq registers, TOMmn, TOMmp, TOMmq, TOLmn, TOLmp, and TOLmq bits cannot be changed. Set values of the TDRmn, TDRmp, and TDRmq registers can be changed after INTTMmn of the master channel is generated. The TCRmn, TCRmp, and TCRmq registers can always be read. The TSRmn, TSRmp, and TSRmq registers are not used.	The counter of the master channel loads the TDRmn register value to timer count register mn (TCRmn) and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to the TCRmn register, and the counter starts counting down again. At the slave channel 1, the values of the TDRmp register are transferred to the TCRmp register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOmp become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. At the slave channel 2, the values of the TDRmq register are transferred to TCRmq register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOmq become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmq = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn bit (master), TTmp, and TTmq (slave) bits are set to 1 at the same time. → The TTmn, TTmp, and TTmq bits automatically return to 0 because they are trigger bits.	TEmn, TEmp, TEmq = 0, and count operation stops. The TCRmn, TCRmp, and TCRmq registers hold count value and stop. The TOmp and TOmq output are not initialized but hold current status.
	The TOEmp and TOEmq bits of slave channels are cleared to 0 and value is set to the TOmp and TOmq bits. →	The TOmp and TOmq pins output the TOmp and TOmq set levels.
TAU stop	To hold the TOmp and TOmq pin output levels Clears the TOmp and TOmq bits to 0 after the value to be held is set to the port register. → When holding the TOmp and TOmq pin output levels are not necessary Setting not required	The TOmp and TOmq pin output levels are held by port function.
	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp and TOmq bits are cleared to 0 and the TOmp and TOmq pins are set to port mode.)

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 7 (Where p and q are integer greater than n)

6.9.4 Remote control output function

The PWM output function is applied to the remote control output function.

The pairings of channels 2 and 3 and channels 4 and 5 are used to output the PWM signal (see **6.9.2 Operation as PWM function** for how to set up each channel). The PWM signal output from channel 3 is used as a mask wave, the PWM signal output from channel 5 is used as a carrier waves, and the logical products of these signals are output as remote control output.

The high level width output part of the remote control output is composed of a 20 to 60 kHz carrier signal.

Figure 6 - 87 Remote Control Output

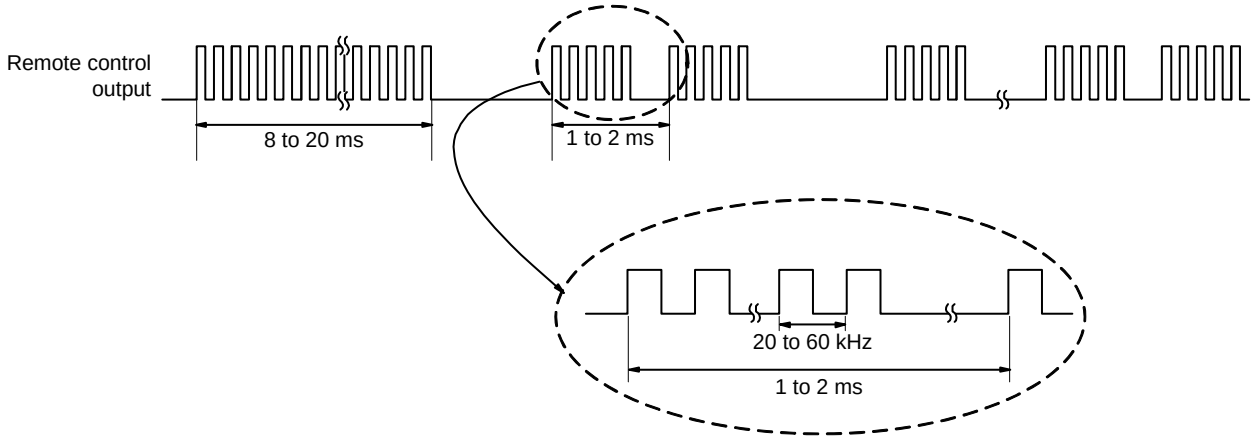
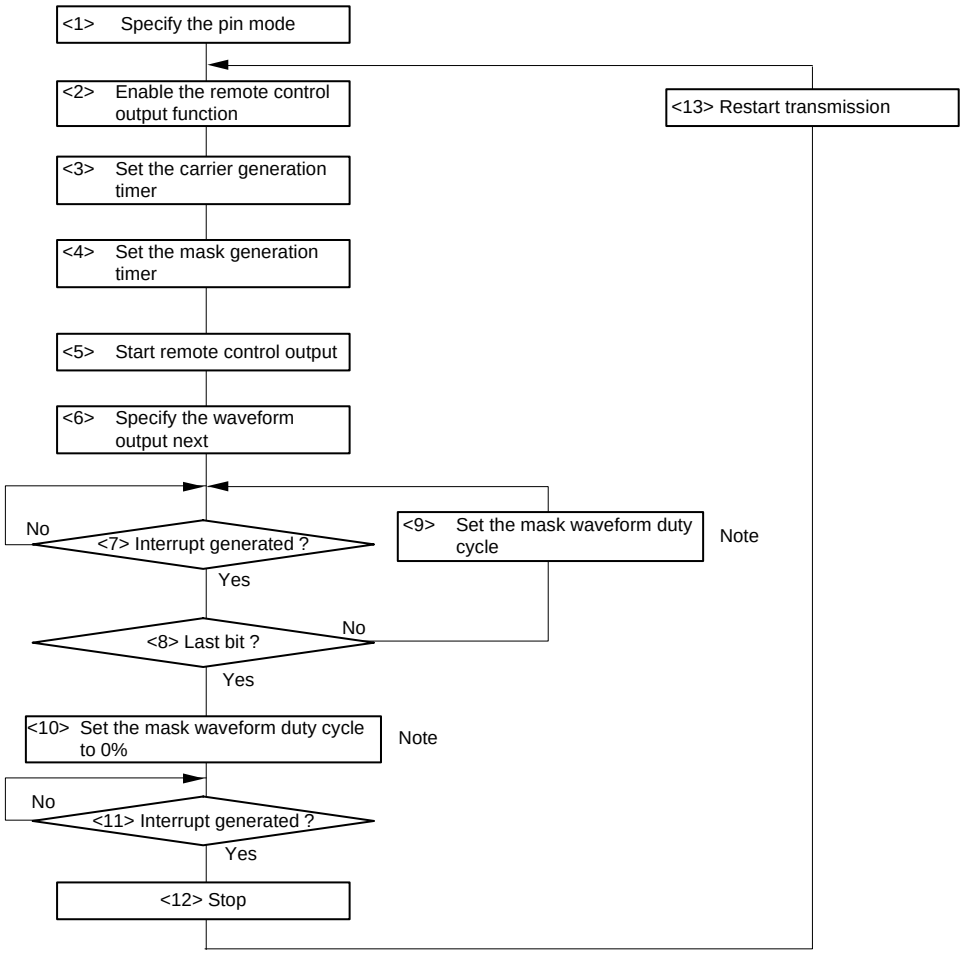


Figure 6 - 88 shows the steps for setting the remote control output.

Figure 6 - 88 Procedure for Setting Remote control Output (1/2)



<1> Specify the pin mode.
Clear the PFSEG20 bit of the PFSEG2 register to 0, the PM30 bit of the PM3 register to 0, the PU30 bit of the PU3 register to 0, and the P30 bit of the P3 register to 0.

<2> Set the TOS0 bit of the timer output select register (TOS) to 1.

<3> Specify the carrier waveform by using the PWM function for channel 4 (master) and channel 5 (slave).
TDR04 register value = Carrier waveform cycle - 1
TDR05 register value = Carrier waveform high-level width

<4> Generate the mask waveform by using the PWM function for channel 2 (master) and channel 3 (slave).
Specify the mask waveform cycle and high-level width (header code)
TDR02 register value = Mask waveform cycle - 1
TDR03 register value = Mask waveform high-level width

<5> Start output operation.
Set the TS02, TS03, TS04, and TS05 bits of timer channel start register 0 (TS0) to 1 at the same time.

Figure 6 - 89 Procedure for Setting Remote control Output (2/2)

<6> Specify the waveform to be output next.
TDR02 register value = Mask waveform cycle - 1
TDR03 register value = Mask waveform high-level width
<7> Wait for an interrupt signal (INTTM02) to be generated at the rise of the mask waveform.
<8> Jump to step <10> for processing that stops output if the bit is the final code bit of the remote control carrier waveform.
If not the final code bit, set the next mask waveform in step <9> and repeat steps <7> to <9> until the final code bit is received.

[Consecutive transmission]
<9> Specify the cycle and duty (cycle: TDR02; high level width: TDR03).

[Stopping remote control output]
<10> Set the mask waveform duty to 0%. (Set TDR03 to 0000H.)
<11> Wait for an interrupt signal (INTTM02) to be generated at the rise of the mask waveform.
<12> Stop the timer.
Set the TT02, TT03, TT04, and TT05 bits of timer channel stop register 0 (TT0) to 1 at the same time, clear the TOE02, TOE03, TOE04, and TOE05 bits of timer output enable register 0 (TOE0) to 0, and then clear the TO02, TO03, TO04, and TO05 bits of timer output register 0 (TO0) to 0.

[Restarting remote control output]
<13> To restart transmission, set the TOE03 and TOE05 bits to 1 and re-specify the settings from step <2>. (It is not necessary to overwrite the same value.)

Note Setting values are applied at the rise of the mask waveform.
The mask waveform cycle and high level width can only be specified as an integral multiple of the carrier cycle.

Caution 1. During the period between <3> and <11>, do not stop supplying a clock to TAU (by using a STOP instruction, etc.) or change the value of registers other than TDR02 and TDR03.
Caution 2. The system must be in the normal operation mode or the HALT mode during steps <3> to <11>.
Caution 3. Select the same operation clock for channels 2, 3, 4, and 5.
Caution 4. Be sure to perform steps <7> to <9> to stop remote control output.
The following errors might occur if stopped using a different method:
- A waveform with a cycle that is not the same as the carrier is output.
- The timer output is fixed to a high level after the operation stops.
Caution 5. When performing remote control output, be sure to set channels 2, 3, 4, and 5 to PWM output mode.

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 TIMER KB20, KB21, KB22

16-bit timers KB20, KB21, and KB22 are timers that can generate PWM output which is suitable to control power sources and lighting. 16-bit timer KB20 in the RL78/L1C does not have a PWM output function for IH control.

	80/85-pin	100-pin
16-bit timer KB20	√	√
16-bit timer KB21	√	√
16-bit timer KB22	√	√

Caution Most of the following descriptions in this chapter use the 100-pin products as an example.

7.1 Functions of 16-bit Timers KB20, KB21, and KB22

16-bit timers KB20, KB21, and KB22 are dedicated PWM output timers and have two outputs each. These timers are provided with the following functions.

(1) PWM output
- A variable PWM with any duty or cycle can be output while the timer is operating.
- The default timer output level (high or low level) can be set.

(2) Trigger output (ELC event generation signal output)
Can be output to the ELC event generation source using the 16-bit timer KB2 trigger compare register (TKBTGCRn).

(3) Simultaneous start / stop mode
Combining two or three timer units (16-bit timers KB20, KB21 and KB22) which start/stop simultaneously can start up to six PWM outputs.

(4) Timer start / clear mode
Start, clear, and stop of PWM output can be synchronized.

(5) Timer restart function
Timer output can be restarted directly (not via the CPU) when a trigger source (counter restart trigger source 0 to 2) occurs.

(6) Forced output stop function 1 (by interlocking with the comparator, INTPiNF Note, ELC)
Timer output can be fixed to high impedance, high, or low level directly (not via the CPU) when a trigger source (comparator 0, 1, INTPiNF Note via the ELC) occurs. The stop function is cancelled by the stop trigger setting of forced output stop function 1.

Note INTPiNF is output that is not passed through the noise filter.

Remark i = 0 to 7

(7) Forced output stop function 2 (by interlocking with the comparator, INTPiNF Note, ELC)
Timer output can be fixed to high or low level directly (not via the CPU) when a trigger source occurs (comparator 0, 1, INTPiNF Note via the ELC). When the following counter period is started or when a trigger source is eliminated, the stop function is cancelled directly (not via the CPU).

Note INTPiNF is output that is not passed through the noise filter.

Remark i = 0 to 7

(8) Dithering function
The “set duty + 1” waveform in each 16-period cycle can be output in the range of periods 0 to 15.

(9) Smooth start function
It is possible to make a smooth start that automatically increases the duty after PWM output starts until it reaches the configured duty value.
It is possible to configure the initial duty and duty plus one incremental period.

(10) Maximum frequency setting function
With the timer restart function, restart can be held pending until the set period.

(11) Interleave function
With the timer restart function, it is possible to use external sources to automatically alternate restart output between two outputs. It is possible to make interleaved PFC control with critical conduction mode.

Remark 1. Critical conduction mode is a PFC control method that activates a switching FET by detecting zero level of inductor current.
Remark 2. i = 0 to 7

7.2 Configuration of 16-bit Timers KB0, KB1, and KB2

16-bit timers KB0, KB1, and KB22 include the following hardware.

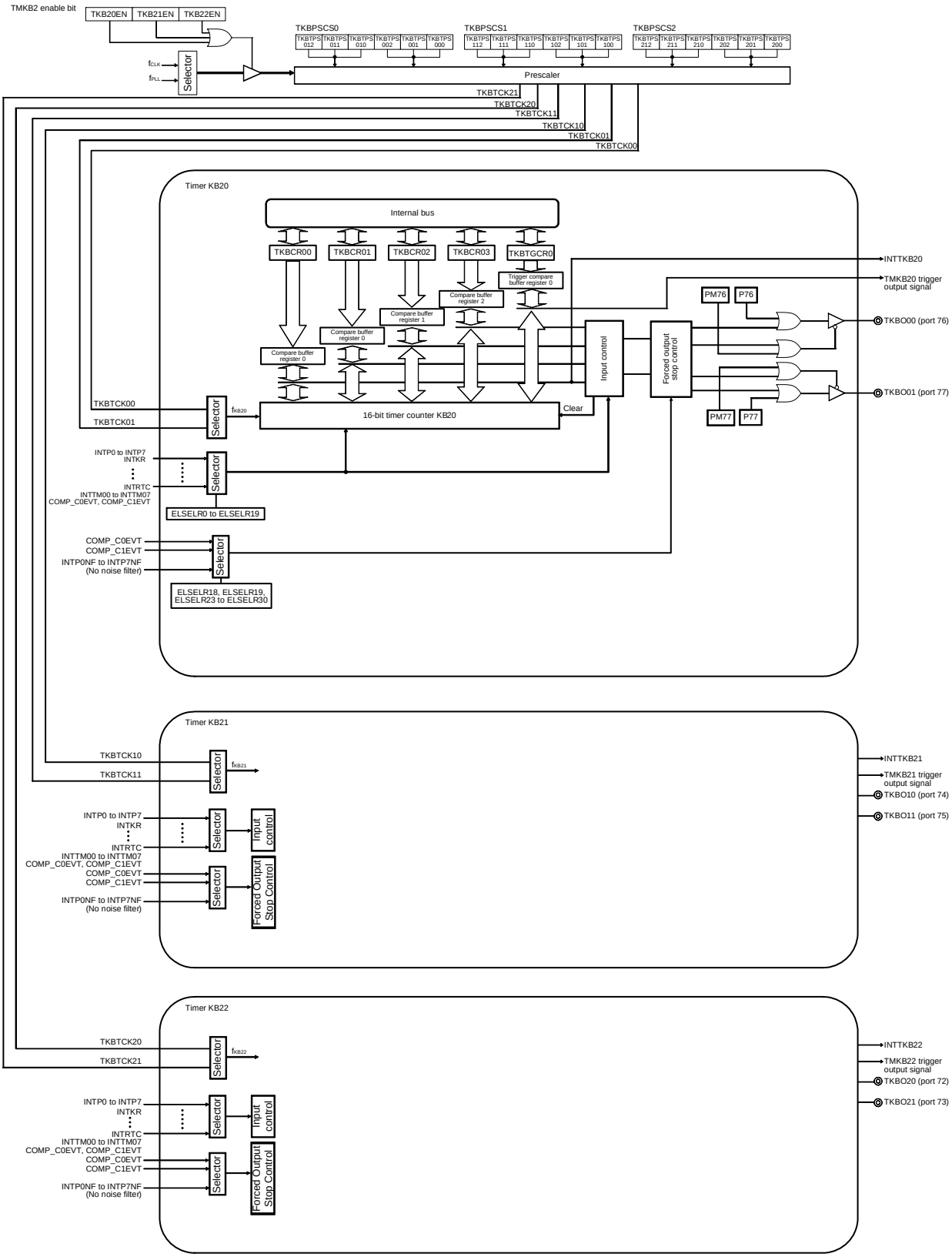
Table 7 - 1 Configuration of 16-bit Timer KB0, KB1, and KB22

Item	Configuration
Timer counter register	16-bit timer counter n (TKBCNTn)
Compare registers	16-bit timer KB2 compare registers n0 to n3 (TKBCRn0 to TKBCRn3) 16-bit timer KB2 trigger compare register n (TKBTGCRn)
Timer output	TKBO00, TKBO01, TKBO10, TKBO11, TKBO20, TKBO21
Control registers	Peripheral enable registers 1, 2 (PER1, PER2) 16-bit timer KB2 clock division ratio select register n (TKBPSCSn) 16-bit timer KB2 operation control register n0 (TKBCTLn0) 16-bit timer KB2 operation control register n1 (TKBCTLn1) 16-bit timer KB2 output control register n0 (TKBIOCn0) 16-bit timer KB2 output control register n1 (TKBIOCn1) 16-bit timer KB2 flag register n (TKBFLGn) 16-bit timer KB2 trigger register n (TKBTRGn) 16-bit timer KB2 flag clear trigger register n (TKBCLRn) 16-bit timer KB2 dithering count registers n0, n1 (TKBDNRn0, TKBDNRn1) 16-bit timer KB2 compare 1L & dithering count register n0 (TKBCRLDn0) 16-bit timer KB2 compare 1L & dithering count register n1 (TKBCRLDn1) 16-bit timer KB2 smooth start initial duty registers n0, n1 (TKBSIRn0, TKBSIRn1) 16-bit timer KB2 smooth start step width registers n0, n1 (TKBSSRn0, TKBSSRn1) 16-bit timer KB2 maximum frequency limit setting register n (TKBMFRn) Forced output stop function control register 0p (TKBPACTL0p) Forced output stop function control register 1p (TKBPACTL1p) Forced output stop function control register 2p (TKBPACTL2p) 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 1 start trigger register (TKBPAHFTn) Port mode register 7 (PM7) Port register 7 (P7)

Remark n = 0, 1, 2; p = 0, 1

Figure 7 - 1 shows a block diagram.

Figure 7 - 1 Block Diagram of 16-bit Timer KB20, KB21, KB22



Remark f_{KB20}: Count clock of 16-bit timer KB20
f_{KB21}: Count clock of 16-bit timer KB21
f_{KB22}: Count clock of 16-bit timer KB22

(1) 16-bit timer counter register n (TKBCNTn)

TKBCNTn performs up-counting synchronously with the clock selected by TKBCKSn. The value of the TKBCNTn changes to 0000H at the following timings, and operation continues.

- When the values of TKBCNTn and TKBCRn0 match
- When the external trigger input selected by the ELC and TKBSTSnm is detected
- When the counter input from 16-bit timer KB20 is detected by 16-bit timers KB21 and KB22 in synchronous start/clear mode

This register can be set in 16-bit units.

Reset signal generation clears this register to 0000H

Figure 7 - 2 Format of 16-bit timer counter register n (TKBCNTn)

Address: F0520H (TKBCNT0), F0260H (TKBCNT1), F02A0H (TKBCNT2) After reset: 0000H R

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

Remark n = 0 to 2

(2) 16-bit timer KB2 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 the counter (TKBCEn = 0)
- When a batch overwrite trigger occurs (TKBRDTn = 1 and TKBTSEn = 1)

This register can be set in 16-bit units.

Reset signal generation clears this register to 0000H.

Figure 7 - 3 Format of 16-bit timer KB2 compare registers n0 to n3 (TKBCRn0 to TKBCRn3)

Address: F0500H (TKBCR00), F0502H (TKBCR01), F0504H (TKBCR02), F0506H (TKBCR03),
F0240H (TKBCR10), F0242H (TKBCR11), F0244H (TKBCR12), F0246H (TKBCR13),
F0280H (TKBCR20), F0282H (TKBCR21), F0284H (TKBCR22), F0286H (TKBCR23),
After reset: 0000H R/W

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

Remark n = 0 to 2; m = 0 to 3

(3) 16-bit timer KB2 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 the counter (from TKBCEn = 0 to TKBCEn = 1)
- When a batch overwrite trigger occurs (TKBRDTn = 1 or TKBTSEn = 1)

Periodic signals from this register can be used as an event generation source for the ELC.

ELC event generation sources correspond to ELSELR20 to ELSELR22 (TMKB2n trigger output).

This register can be read or written in 16-bit units.

Reset signal generation clears this register to 0000H.

Figure 7 - 4 Format of 16-bit timer KB2 trigger compare register n (TKBTGCRn)

Address: F0508H (TKBTGCR0), F0248H (TKBTGCR1), F0288H (TKBTGCR2) After reset: 0000H R/W

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

Remark n = 0 to 2

7.3 Registers Controlling 16-bit Timer KB0, KB1, and KB22

16-bit timers KB20, KB21, and KB22 are controlled by the following registers.

- Peripheral enable registers 1, 2 (PER1, PER2)
- 16-bit timer KB2 clock division ratio select register n (TKBPSCSn)
- 16-bit timer KB2 operation control register n0 (TKBCTLn0)
- 16-bit timer KB2 operation control register n1 (TKBCTLn1)
- 16-bit timer KB2 output control register n0 (TKBIOCn0)
- 16-bit timer KB2 output control register n1 (TKBIOCn1)
- 16-bit timer KB2 flag register n (TKBFLGn)
- 16-bit timer KB2 trigger register n (TKBTRGn)
- 16-bit timer KB2 flag clear trigger register n (TKBCLRn)
- 16-bit timer KB2 dithering count registers n0, n1 (TKBDNRn0, TKBDNRn1)
- 16-bit timer KB2 compare 1L & dithering count register n0 (TKBCRLDn0)
- 16-bit timer KB2 compare 1L & dithering count register n1 (TKBCRLDn1)
- 16-bit timer KB2 smooth start initial duty registers n0, n1 (TKBSIRn0, TKBSIRn1)
- 16-bit timer KB2 smooth start step width registers n0, n1 (TKBSSRn0, TKBSSRn1)
- 16-bit timer KB2 maximum frequency limit setting register n (TKBMFRn)
- Forced output stop function control register 0p (TKBPACTL0p)
- Forced output stop function control register 1p (TKBPACTL1p)
- Forced output stop function control register 2p (TKBPACTL2p)
- 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 1 start trigger register (TKBPAHFTn)
- Port mode register 7 (PM7)
- Port register 7 (P7)

Remark n = 0, 1, 2; p = 0, 1

7.3.1 Peripheral enable register 1 (PER1)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.
When timer KB20 is used, be sure to set bit 4 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 these registers to 00H.

Figure 7 - 5 Format of Peripheral enable register 1 (PER1)

Address: F007AH After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	2	1	<0>
PER1	TMKAEN	0	CMPEN	TKB20EN	DTCEN	0	0	DACEN

TKB20EN	Control of timer KB20 input clock
0	Stops supply of input clock. • SFR used by timer KB20 cannot be written. • Timer KB20 is in the reset status.
1	Supplies input clock. • SFR used by timer KB20 can be read/written. • Timer KB20 can be operated.

Caution 1. When setting timer KB20, be sure to set the TKB20EN bit to 1 first.
If TKB20EN = 0, writing to a control register of timer KB20 is ignored, and all read values are default values (except for port mode register 7 (PM7)).

Caution 2. Be sure to clear bits 1, 2, and 6 to 0.

7.3.2 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 KB21 and KB2 are used, be sure to set bit 0 (TKB21EN) and bit 1 (TKB22EN) 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 these registers to 00H.

Figure 7 - 6 Format of Peripheral enable register 2 (PER2)

Address: F00FDH After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
PER2	0	0	0	0	0	0	TKB22EN	TKB21EN

TKB22EN	Control of timer KB22 input clock
0	Stops supply of input clock. • SFR used by timer KB22 cannot be written. • Timer KB22 is in the reset status.
1	Supplies input clock. • SFR used by timer KB22 can be read/written. • Timer KB22 can be operated.

TKB21EN	Control of timer KB21 input clock
0	Stops supply of input clock. • SFR used by timer KB21 cannot be written. • Timer KB21 is in the reset status.
1	Supplies input clock. • SFR used by timer KB21 can be read/written. • Timer KB21 can be operated.

Caution 1. When setting timer KB22, be sure to set the TKB22EN bit to 1 first.
If TKB22EN = 0, writing to a control register of timer KB22 is ignored, and all read values are default values (except for port mode register 7 (PM7)).

Caution 2. When setting timer KB21, be sure to set the TKB21EN bit to 1 first.
If TKB21EN = 0, writing to a control register of timer KB21 is ignored, and all read values are default values (except for port mode register 7 (PM7)).

Caution 3. Be sure to clear bits 7 to 2 to 0.

7.3.3 16-bit timer KB2 clock division ratio select register n (TKBPSCSn)

The TKBPSCSn register is a register that is used to select the division ratio of TKBTCKn0/TKBTCKn1.
Rewriting of the TKBPSCSn register is possible only in the following cases.

If TKBTPSn00 to TKBTPSn02 can be rewritten:
All channels for which TKBTCKn0 is selected as the operation clock (TKBCKSn = 0) are stopped (TKBCEn = 0).

If the TKBTPSn10 to TKBTPSn12 bits can be rewritten:
All channels for which TKBTCKn1 is selected as the operation clock (TKBCKSn = 1) are stopped (TKBCEn = 0).

The TKBPSCSn register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 7 - 7 Format of 16-bit timer KB2 clock division ratio select register n (TKBPSCSn)

Address: F052AH (TKBPSCS0), F026AH (TKBPSCS1), F02AAH (TKBPSCS2), After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TKBPSCSn	0	TKBTPSn12	TKBTPSn11	TKBTPSn10	0	TKBTPSn02	TKBTPSn01	TKBTPSn00

TKBTPSni2	TKBTPSni1	TKBTPSni0	Selection of operation clock Note 1
0	0	0	f_{CLK} or f_{HOCO} with no division is selected for TKBTCKi Note 2 or f_{PLL} with no division is selected Notes 4, 5
0	0	1	f_{CLK} divided by 2 is selected for TKBTCKi Note 3 or f_{PLL} divided by 2 is selected Notes 4, 5
0	1	0	f_{CLK} divided by 4 is selected for TKBTCKi Note 3 or f_{PLL} divided by 4 is selected Notes 4, 5
0	1	1	f_{CLK} divided by 8 is selected for TKBTCKi Note 3 or f_{PLL} divided by 8 is selected Notes 4, 5
1	0	0	f_{CLK} divided by 16 is selected as TKBTCKi Note 3 or f_{PLL} divided by 16 is selected Notes 4, 5
1	0	1	f_{CLK} divided by 32 is selected as TKBTCKi Note 3 or f_{PLL} divided by 32 is selected Notes 4, 5
Other than above			Setting prohibited

Note 1. When changing the clock selected for f_{CLK}, stop timer KB2n (TKBCEn = 0).

Note 2. f_{CLK} is selected when FRQSEL4 = 0 and f_{HOCO} is selected when FRQSEL4 = 1 in the user option byte (000C2H/010C2H). When selecting f_{HOCO} for the operating clock, set f_{CLK} to f_{IH} before setting bit 4 (TKB20EN) in peripheral enable register 1 (PER1) and bit 1 (TKB22EN) and bit 0 (TKB21EN) in peripheral enable register 2 (PER2). When changing f_{CLK} to a clock other than f_{IH}, clear bit 4 (TKB20EN) in peripheral enable register 1 (PER1) and bit 1 (TKB22EN) and bit 0 (TKB21EN) in peripheral enable register 2 (PER2) before changing.

Note 3. Do not set this value when FRQSEL4 = 1 in the user option byte (000C2H/010C2H).

Note 4. When the PLL oscillates and the CKSELR in the MCKC register is set to 1, f_{PLL} is selected as the source clock of 16-bit timers KB20, KB21, and KB22. When selecting f_{PLL} as the operating clock, set f_{CLK} to f_{PLL} before setting bit 4 (TKB20EN) in peripheral enable register 1 (PER1) and bit 1 (TKB22EN) and bit 0 (TKB21EN) in peripheral enable register 2. When changing f_{CLK} to a clock other than f_{PLL}, clear bit 4 (TKB20EN) in peripheral enable register 1 (PER1) and bit 1 (TKB22EN) and bit 0 (TKB21EN) in peripheral enable register 2 (PER2) before changing.

Note 5. The f_{PLL} clock can be selected only in products with USB when the CKSELR bit in the MCKC register is set to 1 while the PLL oscillates. For details, refer to **CHAPTER 5 CLOCK GENERATOR**.

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

Remark 1. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 2. n = 0 to 2; i = 0, 1

Remark 3. f_{PLL} is set to 48 MHz (TYP.)

7.3.4 16-bit timer KB2 operation control register n0 (TKBCTLn0)

TKBCTLn0 is a register that is used to select the smooth start function, dithering function, maximum frequency limit function, interleaved PFC1 output, batch overwrite function for compare registers using external triggers, and counter triggers.

TKBCTLn0 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 7 - 8 Format of 16-bit timer KB2 operation control register n0 (TKBCTLn0) (1/2)

Address: F0522H (TKBCTL00), F0262H (TKBCTL10), F02A2H (TKBCTL20) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
TKBCTLn0	0	0	TKBSSEn1	TKBDIEn1	0	0	TKBSSEn0	TKBDIEn0

7	6	5	4	3	2	1	0
TKBMEFn	0	TKBIRSn1	TKBIRSn0	0	TKBTSEn	TKBSTSn1	TKBSTSn0

TKBSSEnp	Control of PWM output smooth start function of TKBOnp
0	PWM output smooth start function not used
1	PWM output smooth start function used

TKBDIEnp	Control of PWM output dithering function of TKBOnp
0	PWM output dithering function not used
1	PWM output dithering function used

TKBMEFn	Control of maximum frequency limit function of TKBOnp
0	Maximum frequency limit function not used
1	Maximum frequency limit function used

TKBIRSn1	TKBIRSn0	Acceptable range setting of restart trigger source input for immediately outputting TKBOnp in interleave 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	Compare register batch overwrite function set by external trigger not used
1	Compare register batch overwrite function set by external trigger used

Remark n = 0 to 2; p = 0, 1

Figure 7 - 9 Format of 16-bit timer KB2 operation control register n0 (TKBCTLn0) (2/2)

Address: F0522H (TKBCTL00), F0262H (TKBCTL10), F02A2H (TKBCTL20) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
TKBCTLn0	0	0	TKBSSEn1	TKBDIEn1	0	0	TKBSSEn0	TKBDIEn0

7	6	5	4	3	2	1	0
TKBMEFn	0	TKBIRSn1	TKBIRSn0	0	TKBTSEn	TKBSTSn1	TKBSTSn0

TKBSTSn1	TKBSTSn0	Selection for timer KB2n restart trigger
0	0	Trigger input not used
0	1	Count restart trigger source 0 selected Note
1	0	Count restart trigger source 1 selected Note
1	1	Count restart trigger source 2 selected Note

Note The corresponding event generation source is selected by ELSELRm (m = 00 to 19). For the event link destination, select the same source as that set by TKBSTSn1 and TKBSTSn0.

Caution 1. Do not rewrite the TKBCTLn0 register during timer operation. However, the TKBCTLn0 register can be refreshed (the same value can be written).

Caution 2. Be sure to clear bits 15, 14, 11, 10, 6, and 3 to 0.

Remark n = 0 to 2

7.3.5 16-bit timer KB2 operation control register n1 (TKBCTLn1)

TKBCTLn1 is a register that controls the count operation and sets the count clock of the 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 - 10 Format of 16-bit timer KB2 operation control register n1 (TKBCTLn1)

Address: F0529H (TKBCTL01) After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
TKBCTL01	TKBCE0	0	0	TKBCKS0	TKBSCM0	0	TKBMD01	TKBMD00

Address: F0269H (TKBCTL11), F02A9H (TKBCTL21) After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
TKBCTLm1	TKBCEm	0	0	TKBCKSm	TKBSCMm	0	TKBMDm1	TKBMDm0

• TKBCTL01

TKBMD01	TKBMD00	Selection of timer KB20 operation mode
0	0	Standalone mode
1	1	Interleave PFC output mode
Other than above		Setting prohibited

• TKBCTL11, TKBCTL21

TKBMDm1	TKBMDm0	Selection of timer KB2m operation mode
0	0	Standalone mode
0	1	Simultaneous start/stop mode (slave used)
1	0	Synchronous start/clear mode (slave used)
1	1	Interleave PFC output mode

• TKBCTLn1

TKBCEn	Control of timer KB2n operation
0	Timer operation stopped (counter is set to FFFF).
1	Timer count operation enabled

TKBCKSn	Selection of timer KB2n clock
0	TKBTCKn0 selected
1	TKBTCKn1 selected

TKBSCMn	Timer KB2n start operation control
0	Operates using clock selected by TKBCKSn bit
1	The count start timing per se is to start when the TKBTCKn0 and TKBTCKn1 clocks 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 TKBSCM0 bit, start timing of the slave and master can be matched.**

Caution 1. Do not rewrite the TKBCTLn1 register during timer operation. However, the TKBCTLn1 register can be refreshed (the same value can be written).

Caution 2. In TKBCTLn1, be sure to clear bits 6, 5, and 2 to 0.

Remark n = 0 to 2; m = 1, 2

7.3.6 16-bit timer KB2 output control register n0 (TKBIOCn0)

TKBIOCn0 is a register that is used to set the default level/active level in 16-bit timer KB2n 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 - 11 Format of 16-bit timer KB2 output control register n0 (TKBIOCn0)

Address: F0526H (TKBIOC00), F0266H (TKBIOC10), F02A6H (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	Active level is set to high level
1	Active level is set to low level

TKBTODnp	Default level setting of timer output TKBOnp
0	Default level is set to low level
1	Default level is set to high level

Caution 1. Do not rewrite the TKBIOCn0 register during timer operation. However, the TKBIOCn0 register can be refreshed (the same value can be written).

Caution 2. Be sure to clear bits 7 to 4 to 0.

Caution 3. Actual output of the TKBOnp pins is set not only by TKBOnp output but by the port mode register (PM7) and port register (P7) for the shared ports.

Remark n = 0 to 2; p = 0, 1

7.3.7 16-bit timer KB2 output control register n1 (TKBIOCn1)

TKBIOCn1 is a register that controls output disabled/enabled of 16-bit timer KB2n 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 - 12 Format of 16-bit timer KB2 output control register n1 (TKBIOCn1)

Address: F0528H (TKBIOC01), F0268H (TKBIOC11), F02A8H (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	Output enabled/disabled of timer output TKBOnp
0	Timer output disabled (Low-level output when TKBTODnp = 0. High-level output when TKBTODnp = 1.)
1	Timer output enabled

Caution 1. Bits 1 and 0 (TKBTOEn1, TKBTOEn0) in the TKBIOCn1 register can be rewritten during timer operation.

Caution 2. Be sure to clear bit 7 to 2 to 0.

Caution 3. Actual output of the TKBOnp pins is set not only by TKBOn0, TKBOnp output but by the port mode register (PM7) and the port register (P7) for the shared ports.

Remark n = 0 to 2

7.3.8 16-bit timer KB2 flag register n (TKBFLGn)

TKBFLGn is a register with status flags for 16-bit timer KB2n.
TKBFLGn can be read by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 7 - 13 Format of 16-bit timer KB2 flag register n (TKBFLGn)

Address: F0513H (TKBFLG0), F0253H (TKBFLG1), F0293H (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 pins
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 pins
0	No error, or completion of clearing by TKBCLSEn
1	Error (TKBSSFnp = 1 occurred during PWM output smooth start execution (TKBRDTn = 1))

TKBIRFn	Undetected restart trigger source 1 trigger error flag for interleave PFC mode
0	No error, or completion of clearing by TKBCLIRn
1	Error (Includes a period that restart trigger source 1 trigger is not detected in the range of T/2 + T/n (n = 8, 16, 32, 64))

TKBIEFn	Restart trigger source 1 trigger multiplex detection error flag for interleave PFC mode
0	No error, or completion of clearing by TKBCLIE0
1	Error (Another count start trigger was detected during counting of the TKBOn1 width)

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	Batch overwrite trigger pending status flag
0	Batch overwrite enabled status or completion of batch overwrite caused by batch overwrite trigger
1	On hold (waiting for completion) status of batch overwrite due to writing to batch overwrite trigger bit TKBRDTn

Remark 1. n = 0 to 2; p = 0, 1

Remark 2. T is the period of the last restart

7.3.9 16-bit timer KB2 trigger register n (TKBTRGn)

TKBTRGn is a trigger register used for batch overwriting of the compare register for 16-bit timer KB2n.
TKBTRGn can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 7 - 14 Format of 16-bit timer KB2 trigger register n (TKBTRGn)

Address: F0512H (TKBTRG0), F0252H (TKBTRG1), F0292H (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 Set the TKBRSFn flag to 1.

Remark The read value of TKBTRGn is always 0.

7.3.10 16-bit timer KB2 flag clear trigger register n (TKBCLRn)

TKBCLRn is a register used to clear flags in the 16-bit timer KB2 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 - 15 Format of 16-bit timer KB2 flag clear trigger register n (TKBCLRn)

Address: F0527H (TKBCLR0), F0267H (TKBCLR1), F02A7H (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 TKBOn0TKBOn1-0, TKBOn1-1, TKBOn1-2 pins
0	Invalid setting
1	Clear the TKBSEFnp flag to 0.

TKBCLIRn	Trigger for clearing undetected restart trigger source 1 trigger error flag for interleave PFC mode
0	Invalid setting
1	Clear the TKBIRFn flag to 0.

TKBCLIEn	Trigger for clearing restart trigger source 1 trigger multiplex detection 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.

Caution Be sure to clear bits 7, 6, and 0 to 0.

Remark 1. n = 0 to 2; p = 0, 1
Remark 2. The read value of TKBTRGn is always 0.

7.3.11 16-bit timer KB2 dithering count registers n0, n1 (TKBDNRn0, TKBDNRn1)

TKBDNRnp is a register that is used by the PWM dithering function for TKBOn0, TKBOn1-0, TKBOn1-1, TKBOn1-2 output.

When the values of the higher 4 bits of this register are N (N = 0H to FH), the active period of N period cycles during the 16-period cycle of PWM output is output by extending one clock.

Figure 7 - 17 shows the relation among the TKBDNRnp setting and the repetitions (N) of the period cycle extending the active period by one clock and the ordinal of the period (the kth period) during the 16-period cycle to be extended.

TKBDNRnp can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7 - 16 Format of 16-bit timer KB2 dithering count registers n0, n1 (TKBDNRn0, TKBDNRn1)

Address: F050EH (TKBDNR00), F0510H (TKBDNR01), F024EH (TKBDNR10), F0250H (TKBDNR11), F028EH (TKBDNR20), F0290H (TKBDNR21)

After reset: 00H R/W

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. The TKBDNRnp register can be rewritten during timer operation.

Remark n = 0 to 2; p = 0, 1

Figure 7 - 17 16-bit Timer KB2 Dithering Count Register 0p (TKBDNRnp) Setting

kth Period / Repetitions (N)	k															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0																
1	■															
2	■								■							
3	■				■				■							
4	■				■				■				■			
5	■		■		■				■				■			
6	■		■		■				■		■		■			
7	■		■		■		■		■		■		■			
8	■		■		■		■		■		■		■		■	
9	■	■	■		■		■		■		■		■		■	
10	■	■	■		■		■		■	■	■		■		■	
11	■	■	■		■	■	■		■	■	■		■		■	
12	■	■	■		■	■	■		■	■	■		■	■	■	
13	■	■	■	■	■	■	■		■	■	■		■	■	■	
14	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
15	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	

Remark 1. cell: Set to active period according to settings in TKB0CR1 and TKBCRn3 registers

■ cell: Set to active period according to “settings + 1” in TKB0CR1 and TKBCRn3 registers

Remark 2. n = 0 to 2; p = 0, 1

7.3.12 16-bit timer KB2 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 - 18 Format of 16-bit timer KB2 compare 1L & dithering count register n0 (TKBCRLDn0)

Address: F0514H (TKBCRLD00), F0524H (TKBCRLD10), F0294H (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. The TKBDNRnp register can be rewritten during timer operation.

7.3.13 16-bit timer KB2 compare 1L & 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 - 19 Format of 16-bit timer KB2 compare 1L & dithering count register n1 (TKBCRLDn1)

Address: F0516H (TKBCRLD01), F0526H (TKBCRLD11), F0296H (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. The TKBDNRnp register can be rewritten during timer operation.

7.3.14 16-bit timer KB2 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 - 20 Format of 16-bit timer KB2 smooth start initial duty registers n0, n1 (TKBSIRn0, TKBSIRn1)

Address: F050AH (TKBSIR00), F050CH (TKBSIR01), F024AH (TKBSIR10), F024CH (TKBSIR11), F028AH (TKBSIR20), F028CH (TKBSIR21) After reset: 0000H R/W

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

Caution The TKBSIRnp can be rewritten during timer operation.

Remark n = 0 to 2; p = 0, 1

7.3.15 16-bit timer KB2 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 - 21 Format of 16-bit timer KB2 smooth start step width registers n0, n1 (TKBSSRn0, TKBSSRn1)

Address: F050FH (TKBSSR00), F0511H (TKBSSR01), F024FH (TKBSSR10), F0251H (TKBSSR11), F028FH (TKBSSR20), F0291H (TKBSSR21) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TKBSSRnp	0	0	0	0				

Caution The TKBSSRnp can be rewritten during timer operation. Be sure to clear bits 7 to 4 to 0.

Remark n = 0 to 2; p = 0, 1

7.3.16 16-bit timer KB2 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 - 22 Format of 16-bit timer KB2 maximum frequency limit setting register n (TKBMFRn)

Address: F0524H (TKBMFR0), F0264H (TKBMFR1), F02A4H (TKBMFR2) After reset: 0000H R/W

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

Do not rewrite the TKBMFRn register during timer operation. However, the TKBMFRn register can be refreshed (the same value can be written).

7.3.17 Forced output stop function control register 0p (TKBPACTL0p)

TKBPACTL0p is a register that selects the signal to be used as the trigger to control the forced output stop function of the TKBO0p pin.

TKBPACTL0p can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 7 - 23 Format of Forced output stop function control register 0p (TKBPACTL0p) (1/3)

Address: F0530H (TKBPACTL00), F0532H (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	Input of forced output stop function 2 for timer KB20 (3)
0	Timer KB2 forced output stop source 2 is not used
1	Timer KB2 forced output stop source 2 is used (corresponding to ELC link destination No.23)

TKBPAFXS0p2	Input of forced output stop function 2 for timer KB20 (2)
0	Timer KB20 forced output stop source is not used
1	Timer KB20 forced output stop source is used (corresponding to ELC link destination No.20)

TKBPAFXS0p1	Input of forced output stop function 2 for timer KB20 (1)
0	Timer KB2 forced output stop source 1 is not used
1	Timer KB2 forced output stop source 1 is used (corresponding to ELC link destination No.19)

TKBPAFXS0p0	Input of forced output stop function 2 for timer KB20 (0)
0	Timer KB2 forced output stop source 0 is not used
1	Timer KB2 forced output stop source 0 is used (corresponding to ELC link destination No.18)

TKBPAFCM0p	Operation mode selection for timer KB20 forced output stop function 2
0	Forced output stop function 2 is started when forced output stop input 2 is detected. Forced output stop function 2 is cancelled in synchronization with the next restart of the counter.
1	Forced output stop function 2 is started when forced output stop input 2 is detected. After cancellation of the trigger is detected, forced output stop function is cancelled in synchronization with the next restart of the counter.

Remark p = 0, 1

Figure 7 - 24 Format of Forced output stop function control register 0p (TKBPACTL0p) (2/3)

Address: F0530H (TKBPACTL00), F0532H (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

TKBPAFXS0p2	Input of forced output stop function 1 for timer KB20 (2)
0	Timer KB20 forced output stop source is not used
1	Timer KB20 forced output stop source is used (corresponding to ELC link destination No.20)

TKBPAHZS0p1	Input of forced output stop function 1 for timer KB20 (1)
0	Timer KB2 forced output stop source 1 is not used
1	Timer KB2 forced output stop source 1 is used (corresponding to ELC link destination No.19)

TKBPAHZS0p0	Input of forced output stop function 1 for timer KB20 (0)
0	Timer KB2 forced output stop source 0, external interrupts, and INTP1 are not used
1	Timer KB2 forced output stop source 0 is used (corresponding to ELC link destination No.18)

TKBPAHCM0p1	TKBPAHCM0p0	Operation mode selection for forced output stop function 1
0	0	Forced output stop function 1 is started when forced output stop input 1 is detected. Forced output stop function 1 is cancelled when TKBPAHTT0p is set to 1, regardless of the level of the input.
0	1	Forced output stop function 1 is started when forced output stop input 1 is detected. After the input is cancelled, forced output stop function 1 is cancelled when TKBPAHTT0p is set to 1. Setting TKBPAHTT0p = 1 is invalid during the active period of the input.
1	0	Forced output stop function 1 is started when forced output stop input 1 is detected. After TKBPAHTT0p is set to 1, forced output stop function 1 is cancelled in synchronization with the next restart of the counter, regardless of the level of the input.
1	1	Forced output stop function 1 is started when forced output stop input 1 is detected. After the input is cancelled, forced output stop function 1 is cancelled in synchronization with the next restart of the counter after TKBPAHTT0p is set to 1. Setting TKBPAHTT0p = 1 is invalid during the active period of the input.

Remark p = 0, 1

Figure 7 - 25 Format of Forced output stop function control register 0p (TKBPACTL0p) (3/3)

Address: F0530H (TKBPACTL00), F0532H (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

TKBPAMD0p1	TKBPAMD0p0	Output status selection when executing forced output stop function	
		Forced output stop function 1	Forced output stop function 2
0	0	High-impedance output	Output fixed at low level
0	1	High-impedance 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

Caution 1. Be sure to clear bits 11 to 9 and 7 to 0.

Caution 2. Do not rewrite the TKBPACTL0p register during timer operation. However, the TKBPACTL0p register can be refreshed (the same value can be written).

Caution 3. When using comparator 0 or 1 detection as the trigger source while the C1EDG and C0EDG bits in the comparator filter control register (COMPFIR) are 1 (both-edge detection), make sure that TKBPAFCM0p and TKMPAHCMnp0 are set to 0.

Remark p = 0, 1; n = 0 to 2

7.3.18 Forced output stop function control register 1p (TKBPACTL1p)

TKBPACTL1p is a register that selects the signal to be used as the trigger to control the forced output stop function of the TKBO1p pin.

TKBPACTL1p can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 7 - 26 Format of Forced output stop function control register 1p (TKBPACTL1p) (1/3)

Address: F0270H (TKBPACTL10), F0272H (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	Input of forced output stop function 2 for timer KB21 (3)
0	Timer KB2 forced output stop source 2 is not used
1	Timer KB2 forced output stop source 2 is used (corresponding to ELC link destination No.23)

TKBPAFXS1p2	Input of forced output stop function 2 for timer KB21 (2)
0	Timer KB21 forced output stop source is not used
1	Timer KB21 forced output stop source is used (corresponding to ELC link destination No.21)

TKBPAFXS1p1	Input of forced output stop function 2 for timer KB21 (1)
0	Timer KB2 forced output stop source 1 is not used
1	Timer KB2 forced output stop source 1 is used (corresponding to ELC link destination No.19)

TKBPAFXS1p0	Input of forced output stop function 2 for timer KB21 (0)
0	Timer KB2 forced output stop source 0 is not used
1	Timer KB2 forced output stop source 0 is used (corresponding to ELC link destination No.18)

TKBPAFCM1p	Operation mode selection for timer KB21 forced output stop function 2
0	Forced output stop function 2 is started when forced output stop input 2 is detected. Forced output stop function 2 is cancelled in synchronization with the next restart of the counter.
1	Forced output stop function 2 is started when forced output stop input 2 is detected. After the reverse edge of the trigger is detected, forced output stop function 2 is cancelled in synchronization with the next restart of the counter.

Remark p = 0, 1

Figure 7 - 27 Format of Forced output stop function control register 1p (TKBPACTL1p) (2/3)

Address: F0270H (TKBPACTL10), F0272H (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

TKBPAHZS1p2	Input of forced output stop function 1 for timer KB21 (2)
0	Timer KB21 forced output stop source is not used
1	Timer KB21 forced output stop source is used (corresponding to ELC link destination No.21)

TKBPAHZS1p1	Input of forced output stop function 1 for timer KB21 (1)
0	Timer KB2 forced output stop source 1 is not used
1	Timer KB2 forced output stop source 1 is used (corresponding to ELC link destination No.19)

TKBPAHZS1p0	Input of forced output stop function 1 for timer KB21 (0)
0	Timer KB2 forced output stop source 0 is not used
1	Timer KB2 forced output stop source 0 is used (corresponding to ELC link destination No.18)

TKBPAHCM1p1	TKBPAHCM1p0	Operation mode selection for forced output stop function 1
0	—	Forced output stop function 1 is started when forced output stop input 1 is detected. Forced output stop function 1 is cancelled when TKBPAHTT1p is set to 1, regardless of the level of the input.
1	0	Forced output stop function 1 is started when forced output stop input 1 is detected. After the input is cancelled, forced output stop function 1 is cancelled when TKBPAHTT1p is set to 1. Setting TKBPAHTT1p = 1 invalid during the active period of the input.
1	1	Forced output stop function 1 is started when forced output stop input 1 is detected. After TKBPAHTT1p is set to 1, forced output stop function 1 is cancelled with the next restart of the counter, regardless of the level of the input.
1	1	Forced output stop function 1 is started when forced output stop input 1 is detected. After the input is cancelled, forced output stop function 1 is cancelled with the next restart of the counter after TKBPAHTT1p is set to 1. Setting TKBPAHTT1p = 1 is invalid during the active period of the input.

Remark p = 0, 1

Figure 7 - 28 Format of Forced output stop function control register 1p (TKBPACTL1p) (3/3)

Address: F0270H (TKBPACTL10), F0272H (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

TKBPAMD1p1	TKBPAMD1p0	Output status selection when executing forced output stop function	
		Forced output stop function 1	Forced output stop function 2
0	0	High-impedance output	Output fixed at low level
0	1	High-impedance 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

Caution 1. Be sure to clear bits 11 to 9 and 7 to 0.

Caution 2. Do not rewrite the TKBPACTL1p register during timer operation. However, the TKBPACTL1p register can be refreshed (the same value can be written).

Caution 3. When using comparator 0 or 1 detection as the trigger source while the C1EDG and C0EDG bits in the comparator filter control register (COMPFIR) are 1 (both-edge detection), make sure that TKBPAFCMP1p and TKBPAHCM1p0 are set to 0.

Remark p = 0, 1; n = 0 to 2

7.3.19 Forced output stop function control register 2p (TKBPACTL2p)

TKBPACTL2p is a register that selects the signal to be used as the trigger to control the forced output stop function of the TKBO2p pin.

TKBPACTL2p can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 7 - 29 Format of Forced output stop function control register 2p (TKBPACTL2p) (1/3)

Address: F02B0H (TKBPACTL20), F02B2H (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	Input of forced output stop function 2 for timer KB22 (3)
0	Timer KB2 forced output stop source 2 is not used
1	Timer KB2 forced output stop source 2 is used (corresponding to ELC link destination No.23)

TKBPAFXS2p2	Input of forced output stop function 2 for timer KB22 (2)
0	Timer KB22 forced output stop source is not used
1	Timer KB22 forced output stop source is used (corresponding to ELC link destination No.22)

TKBPAFXS2p1	Input of forced output stop function 2 for timer KB22 (1)
0	Timer KB2 forced output stop source 1 is not used
1	Timer KB2 forced output stop source 1 is used (corresponding to ELC link destination No.19)

TKBPAFXS2p0	Input of forced output stop function 2 for timer KB22 (0)
0	Timer KB2 forced output stop source 0 is not used
1	Timer KB2 forced output stop source 0 is used (corresponding to ELC link destination No.18)

TKBPAFCM2p	Operation mode selection for forced output stop function 2
0	Forced output stop function 2 is started when forced output stop input 2 is detected. Forced output stop function 2 is cancelled in synchronization with the next restart of the counter.
1	Forced output stop function 2 is started when forced output stop input 2 is detected. After the reverse edge of the trigger is detected, forced output stop function 2 is cancelled in synchronization with the next restart of the counter.

Remark p = 0, 1

Figure 7 - 30 Format of Forced output stop function control register 2p (TKBPACTL2p) (2/3)

Address: F02B0H (TKBPACTL20), F02B2H (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

TKBPAHZS2p2	Input of forced output stop function 1 for timer KB22 (2)
0	Timer KB22 forced output stop source is not used
1	Timer KB22 forced output stop source is used (corresponding to ELC link destination No.22)

TKBPAHZS2p1	Input of forced output stop function 1 for timer KB22 (1)
0	Timer KB2 forced output stop source 1 is not used
1	Timer KB2 forced output stop source 1 is used (corresponding to ELC link destination No.19)

TKBPAHZS2p0	Input of forced output stop function 1 for timer KB22 (0)
0	Timer KB2 forced output stop source 0 is not used
1	Timer KB2 forced output stop source 0 is used (corresponding to ELC link destination No.18)

TKBPAHCM2p1	TKBPAHCM2p0	Operation mode selection for forced output stop function 1
0	—	Forced output stop function 1 is started when forced output stop input 1 is detected. Forced output stop function 1 is cancelled when TKBPAHTT2p is set to 1, regardless of the level of the input.
1	0	Forced output stop function 1 is started when forced output stop input 1 is detected. After the input is cancelled, forced output stop function 1 is cancelled when TKBPAHTT2p is set to 1. Setting TKBPAHTT2p = 1 invalid during the active period of the input.
1	1	Forced output stop function 1 is started when forced output stop input 1 is detected. After TKBPAHTT2p is set to 1, forced output stop function 1 is cancelled with the next restart of the counter, regardless of the level of the input.
1	1	Forced output stop function 1 is started when forced output stop input 1 is detected. After the input is cancelled, forced output stop function 1 is cancelled with the next restart of the counter after TKBPAHTT2p is set to 1. Setting TKBPAHTT2p = 1 is invalid during the active period of the input.

Remark p = 0, 1

Figure 7 - 31 Format of Forced output stop function control register 2p (TKBPACTL2p) (3/3)

Address: F02B0H (TKBPACTL20), F02B2H (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

TKBPAMD2p1	TKBPAMD2p0	Output status selection when executing forced output stop function	
		Forced output stop function 1	Forced output stop function 2
0	0	High-impedance output	Output fixed at low level
0	1	High-impedance 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

Caution 1. Be sure to clear bits 11 to 9 and 7 to 0.

Caution 2. Do not rewrite the TKBPACTL2p register during timer operation. However, the TKBPACTL2p register can be refreshed (the same value can be written).

Caution 3. When using comparator 0 or 1 detection as the trigger source while the C1EDG and C0EDG bits in the comparator filter control register (COMPFIR) are 1 (both-edge detection), make sure that TKBPAFCMP2p and TKBPAHCM2p0 are set to 0.

Remark p = 0, 1

7.3.20 Forced output stop function control register n2 (TKBPACTLn2)

TKBPACTLn2 is a register that enables or disables the forced output stop function.

TKBPACTLn2 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7 - 32 Format of Forced output stop function control register n2 (TKBPACTLn2)

Address: F0537H (TKBPACTL02), F0277H (TKBPACTL12), F02B7H (TKBPACTL22) 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
0	Operation of forced output stop function disabled
1	Operation of forced output stop function enabled

Caution 1. The TKBPACTLn2 register can be rewritten during timer operation.

Caution 2. Be sure to clear bits 7 to 2 to 0.

Remark n = 0 to 2; p = 0, 1

7.3.21 Forced output stop function flag register n (TKBPAFLGn)

TKBPAFLGn is a register with status flags for the forced output stop function.
TKBPAFLGn can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 7 - 33 Format of Forced output stop function flag register n (TKBPAFLGn)

Address: F0536H (TKBPAFLG0), F0276H (TKBPAFLG1), F02B6H (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 cancel status
1	Forced output stop status

TKBPAHSFnp	Status flag of forced output stop function 1 for TKBOnp pin
0	Forced output stop cancel status
1	Forced output stop status

TKBPAFIFnp	Monitor bit of forced output stop input 2 input for TKBOnp pin
0	Input of forced output stop function 2 is inactive level
1	Input of forced output stop function 2 is active level

TKBPAHIFnp	Monitor bit of forced output stop input 1 input for TKBOnp pin
0	Input of forced output stop function 1 is inactive level
1	Input of forced output stop function 1 is active level

Caution **Status flags TKBPAHIFnp and TKBPAFIFnp cannot be used when using comparator 0 or 1 detection as a trigger while the C0EDG or C1EDG bit in the comparator filter control register (COMPFIR) is 1 (both-edge detection).**

Remark n = 0 to 2; p = 0, 1

7.3.22 Forced output stop function 1 start trigger register n (TKBPAHFSn)

TKBPAHFSn is the start trigger register used by forced output stop function 1.
TKBPAHFSn can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 7 - 34 Format of Forced output stop function 1 start trigger register n (TKBPAHFSn)

Address: F0534H (TKBPAHFS0), F0274H (TKBPAHFS1), F02B4H (TKBPAHFS2) After reset: 00H W

Symbol	7	6	5	4	3	2	1	0
TKBPAHFSn	0	0	0	0	0	0	TKBPAHTSn1	TKBPAHTSn0

TKBPAHTSnp	Start of forced output stop function 1 for TKBOnp pin
0	Writing 0 is invalid.
1	Starts forced output stop function 1 for TKBOnp pin.

Caution 1. Be sure to clear bits 7 to 2 to 0. The TKBPAHFSn register can be rewritten during timer operation. The read value is 0.

Caution 2. The read value of TKBPAHFSn is always 0.

Remark n = 0 to 2; p = 0, 1

7.3.23 Forced output stop function 1 start trigger register (TKBPAHFTn)

TKBPAHFTn is the stop trigger register used by forced output stop function 1.
TKBPAHFTn can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 7 - 35 Format of Forced output stop function 1 start trigger register (TKBPAHFTn)

Address: F0535H (TKBPAHFT0), F0275H (TKBPAHFT1), F02B5H (TKBPAHFT2) After reset: 00H W

Symbol	7	6	5	4	3	2	1	0
TKBPAHFTn	0	0	0	0	0	0	TKBPAHTTn1	TKBPAHTTn0

TKBPAHTTnp	Stop of forced output stop function 1 for TKBOnp pin
0	Writing 0 is invalid.
1	Stops forced output stop function 1 for TKBOnp pin.

Caution 1. Be sure to clear bits 7 to 2 to 0. The TKBPAHFTn register can be rewritten during timer operation. The read value is 0.

Caution 2. The read value of TKBPAHFTn is always 0.

When the TKBPAHCM0n1 and TKBPAHCM0n0 bits are 10 or 11, forced output function 1 is cancelled when the TMKB2 hardware macro period is generated after TKBPAHTTnp is set to 1.
For details on the operation when forced output stop input is detected or TKBPAHTSnp is set to 1 during the period after TKBPAHTTnp is set to 1 and before the TMKB hardware macro period is generated, see **7.7.3 Notes on Using Forced Output Stop Function 1**.

Remark n = 0 to 2; p = 0, 1

7.3.24 Port mode register 7 (PM7)

This register specifies input or output mode for port 7 in 1-bit units.
When using the P72/TKBO20, P73/TKBO21, P74/TKBO10, P75/TKBO11, P76/TKBO00, and P77/TKBO01 pins for timer output, set PM74 to PM77 and the output latches of PM72 to PM74 and P72 to PM77 to 0.
PM7 can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to FFH.

Figure 7 - 36 Port mode register 7 (PM7)

Address: FFF27H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	P7n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution The above shows the format of port mode register 7 of the 100-pin products. For the format of the port mode registers of other products, see Tables 4 - 2 to 4 - 5 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx Registers and the Bits Mounted on Each Product.

7.4 Operation of 16-bit Timer KB20, KB21, KB22

Operation specifications of 16-bit timer KB2n (n = 0 to 2) are 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 Operation (See **7.4.4**)

There are 5 different operation modes for 16-bit timer KB2n (n = 0 to 2).

- 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 (See **7.4.7**)
- Synchronous Start / Clear Mode (See **7.4.8**)
- Interleave PFC (Power Factor Correction) Output Mode (See **7.4.9**)

7.4.1 Counter Basic Operation

(1) Count start operation

In any mode, the 16-bit counter of timer KB2n 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

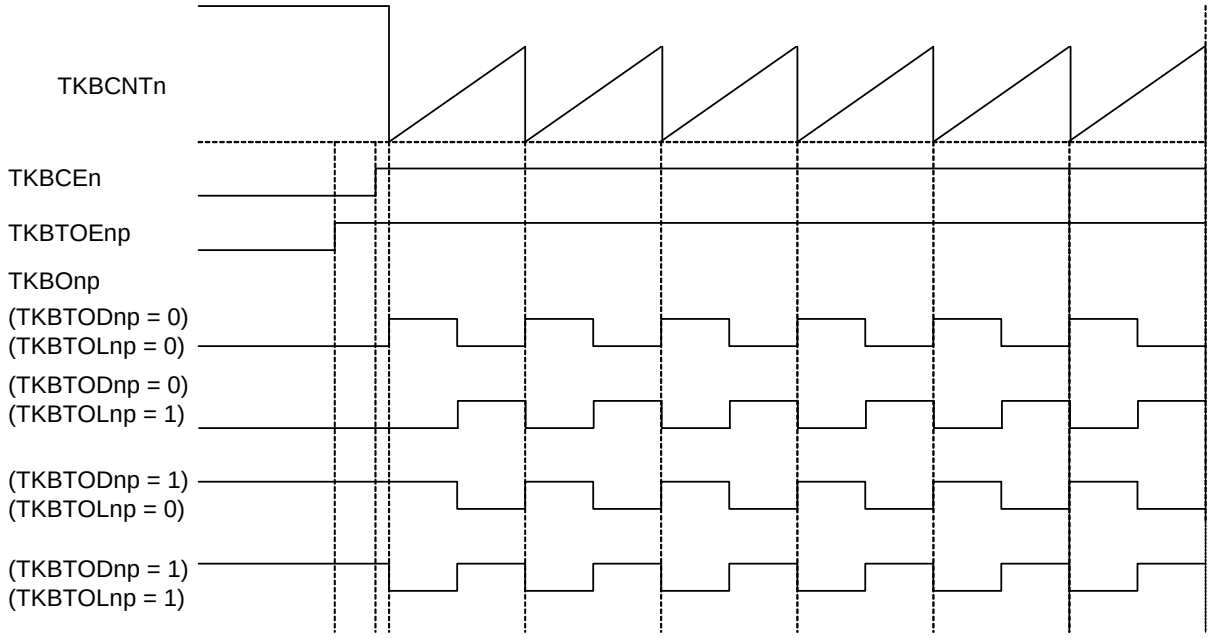
The 16-bit counter is cleared to 0000H when the 16-bit counter value matches the value defined in TKBCRn0 or an external trigger is in effect if the period is determined by external triggers. An INTTKB2n interrupt occurs when the counter is cleared at the time when it matches the value defined in TKBCRn0, 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 KB2n output by 16-bit timer KB2 output control register n0 (TKBIOCn0).

Figure 7 - 37 Figure of Timing of Default and Active Level (Basic operation)



When TKBTOEnp is switched from 0 to 1, TKBOnp output is enabled and PWM waveform is output according to TKBTOLnp setting.

When TKBTOEnp is switched from 1 to 0, TKBOnp output is disabled and default level is output according to TKBTODnp setting.

Remark n = 0 to 2; p = 0, 1

(2) TKBTOEnp switched from 0 to 1

When TKBTOEnp is changed from 0 to 1 before the value of counter TKBCNTn matches 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 the value of compare register (TKBCRnp), the timer output remains its default level until the next restart of the counter.

Figure 7 - 38 Figure of Timing of Default and Active Level
(TKBTOEnp = 0 switched to 1 before matching counter and compare register (TKBCRnp))

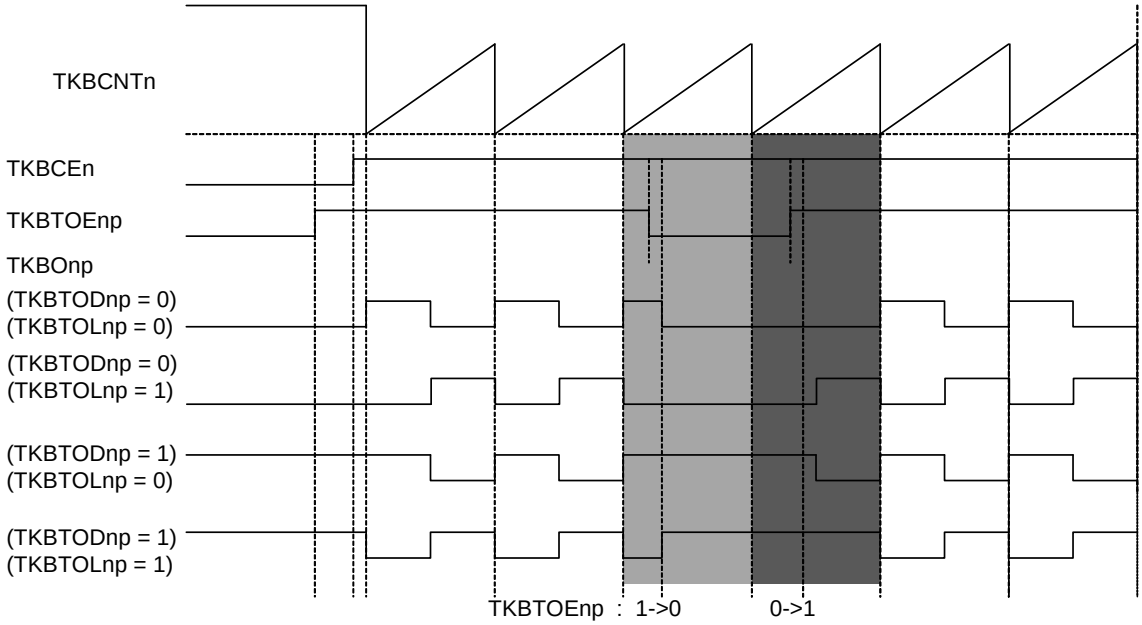
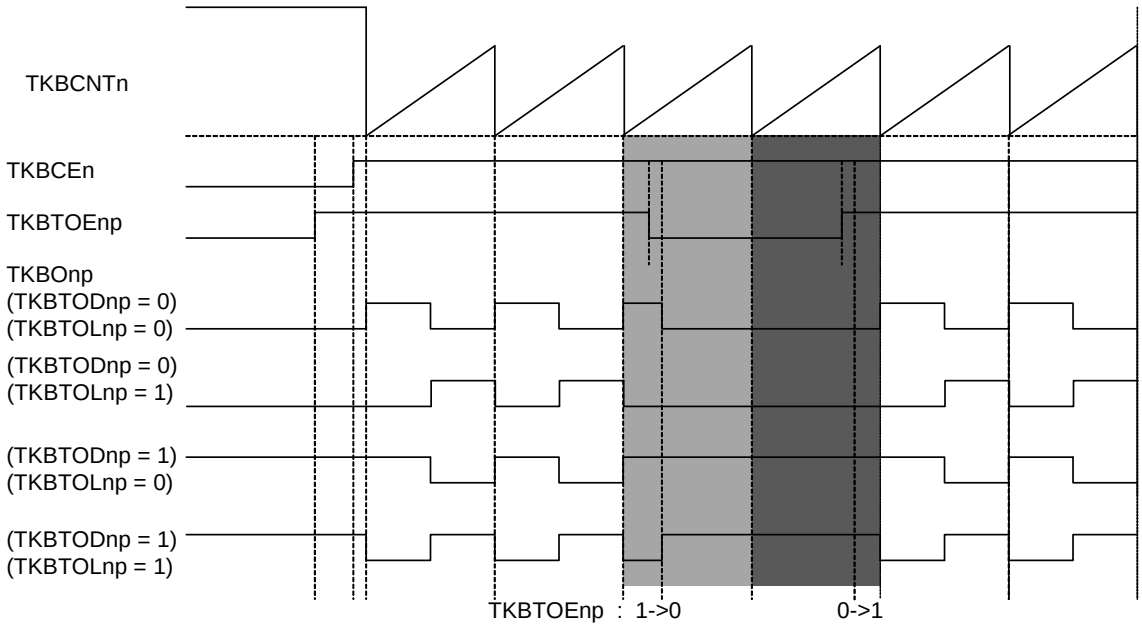


Figure 7 - 39 Figure of Timing of Default and Active Level
(TKBTOEnp = 0 switched to 1 after matching counter and compare register (TKBCRnp))



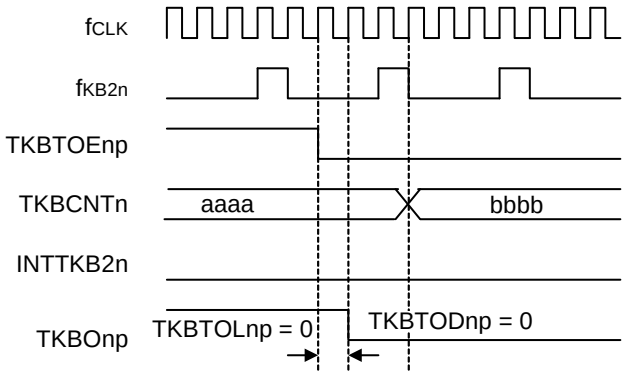
Remark n = 0 to 2; p = 0, 1

(3) TKBTOEnp switched from 1 to 0

(a) Basic timing

TKBOnp is default level set by TKBTODnp after 1 f_{CLK} clock when TKBTOEnp is switched from 1 to 0.

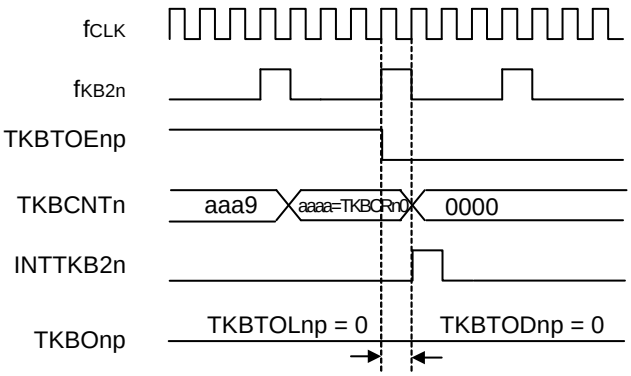
Figure 7 - 40 Figure of Timing of Default and Active Level (TKBTOEnp switched from 1 to 0)



(b) When the setting due to the matched value of TKBCRn0 and the event that TKBTOEnp is cleared occur at the same timing:

When TKBTOEnp switch timing (1 to 0) is simultaneous with the matching between TKBCNTn and TKBCRnm, the change of TKBTOEnp is given priority, and TKBOnp is default level set by TKBTODnp.

Figure 7 - 41 Figure of Timing for Default and Active Level (TKBTOEnp switch timing (1 to 0) is simultaneous with matching between TKBCNTn and TKBCRnm)



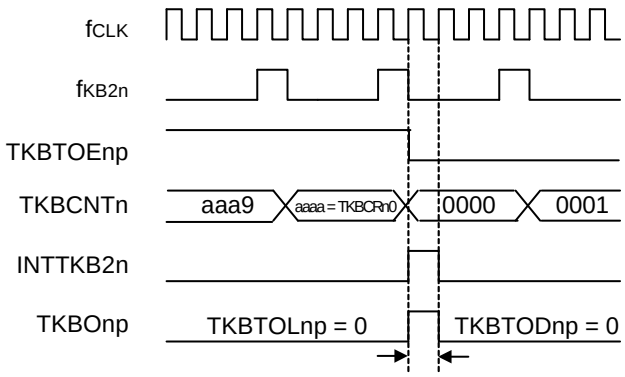
Remark n = 0 to 2; m = 0 to 3; p = 0, 1

(c) When operation of TKBTOEnp is simultaneous with generation of the timer count clock

TKBOnp is set by the matching of TKBCNTn = TKBCRn0 when operation of TKBTOEnp is simultaneous with generation of f_{KB2n}.

After 1 f_{CLK} clock, TKBOnp is default level set by TKBTODnp.

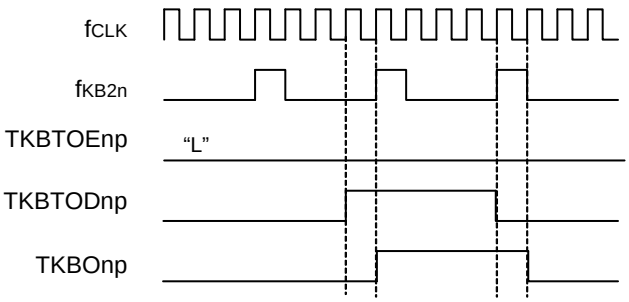
Figure 7 - 42 Figure of Timing of Default and Active Level (Operation of TKBTOEnp is simultaneous with generation of timer count clock)



(4) When TKBTODnp is changed while TKBTOEnp = 0

When TKBTODnp is changed while TKBTOEnp = 0, TKBOnp is default level set by TKBTODnp after 1 f_{CLK} clock.

Figure 7 - 43 Figure of Timing of Default and Active Level (TKBTODnp is changed while TKBTOEnp = 0)



Remark n = 0 to 2; p = 0, 1

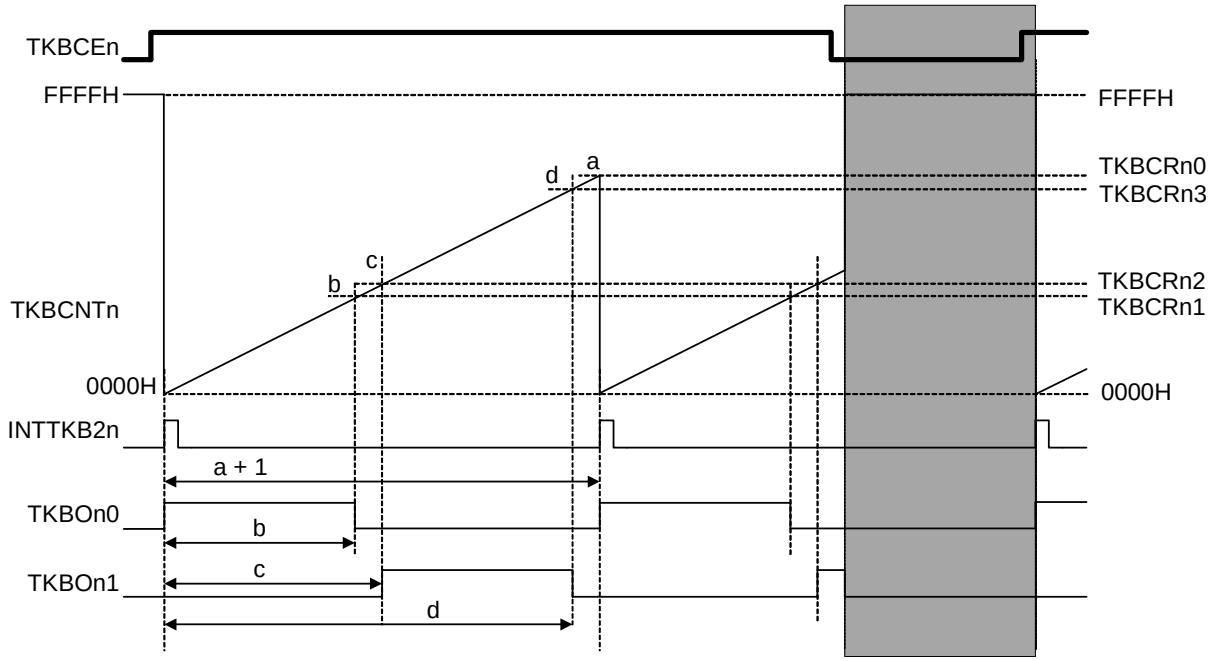
7.4.3 Stop/Restart Operation

Stop and start of operation of 16-bit timer KB2n is available by controlling TKBCEn.
16-bit timer KB2n is reset and stops operation by changing TKBCEn from 1 to 0.

Counter TKBCNTn is reset to FFFFH and stops operation then.
TKBOnp output outputs the default level set by TKBTODnp.

16-bit timer KB2n starts operation by changing TKBCEn from 0 to 1.
Counter TKBCNTn maintains FFFFH when TKBCEn = 0 and starts up counting operation by changing TKBCEn from 0 to 1.

Figure 7 - 44 Figure of Timing of Stop Operation (TKBTOLnp = 0, TKBTODnp = 0)



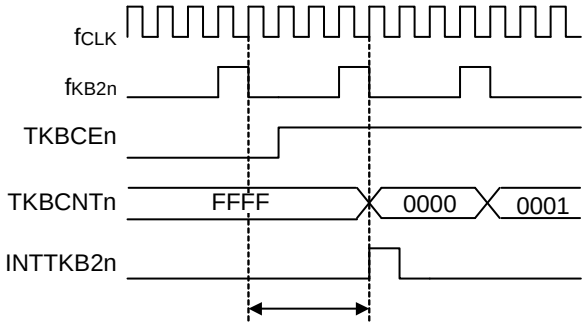
Remark n = 0 to 2; p = 0, 1

(1) Count operation start timing

When TKBCEn is switched from 0 to 1, counting operation starts after the progress of the minimum 1 f_{CLK} clock to the maximum 1 f_{KB2n} clock.

INTTMKB2 is output at counting operation start timing.

Figure 7 - 45 Figure of Timing of Start Operation (TKBCEn switched from 0 to 1)



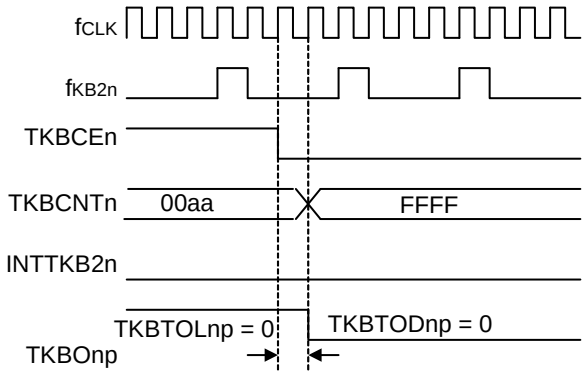
(2) Count operation stop timing

(a) Basic timing

When TKBCEn is switched from 1 to 0, counting operation is stopped after the progress of minimum 1 f_{CLK} clock.

TKBCNTn is reset to FFFFH and TKBOnp is default level set by TKBTODnp.

Figure 7 - 46 Figure of Timing of Stop Operation (TKBCEn switched from 1 to 0)



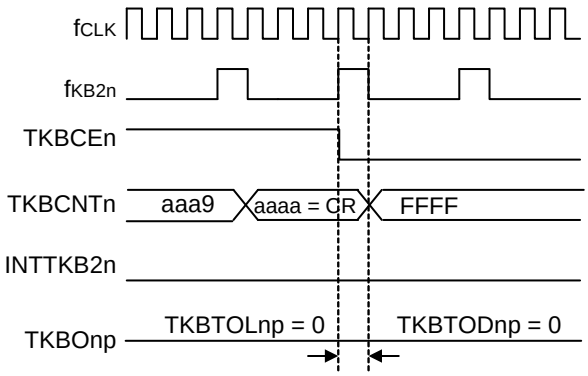
Remark n = 0 to 2; p = 0, 1

(b) When the setting due to the matched value of TKBCRn0 and the event that TKBTOE0 is cleared occur at the same timing:

When TKBCEn switch timing (1 to 0) is simultaneous with the matching between TKBCNTn and TKBCRn0, the change of TKBCNTn is given priority, and TKBOnp is default level set by TKBTODnp.

At this time, INTTKB2n is not generated.

Figure 7 - 47 Figure of Timing of Stop Operation
(When TKBCEn switch timing (1 to 0) is simultaneous with the matching between TKBCNTn and TKBCRn0)

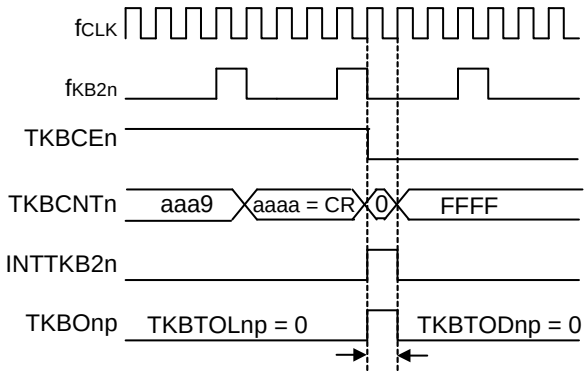


(c) When operation of TKBCEn is simultaneous with generation of the timer count clock

When operation of TKBCEn is simultaneous with generation of 1 fCLK, INTTMKB2 is output by the matching of TKBCNTn = TKBCRn0, and TKBOnp is set.

After 1 fCLK clock, TKBCNTn is reset to FFFFH, and TKBOnp is default level set by TKBTODnp.

Figure 7 - 48 Figure of Timing of Stop Operation
(Operation of TKBCEn is simultaneous with generation of timer count clock)

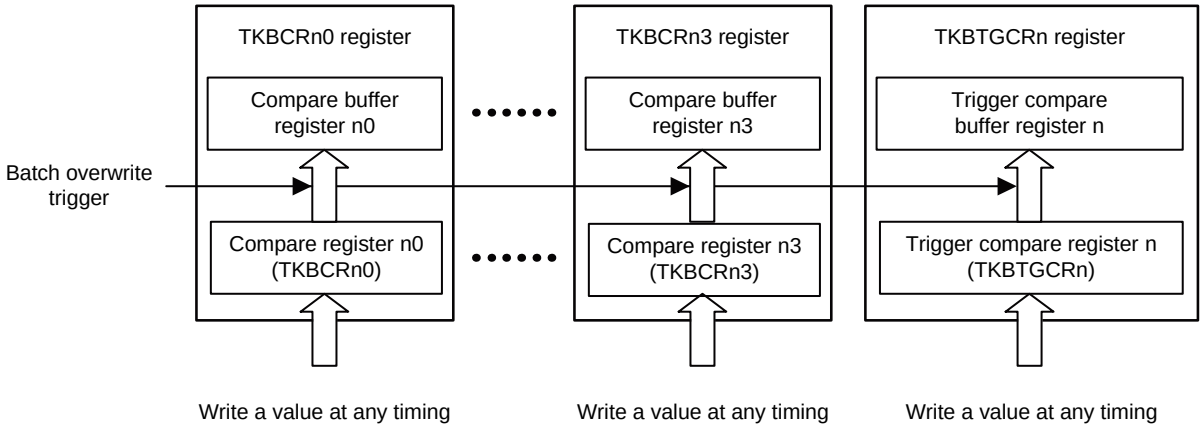


Remark n = 0 to 2; p = 0, 1

7.4.4 Batch Overwrite Operation

As shown in Figure 7 - 49, 16-bit timer KB2 compare register np (TKBCRnm) for 16-bit timer KB2n has two stages. Therefore, its value does not become effective immediately even if any value is set to TKBCRnm by a program. The value set to TKBCRnm 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 - 49 Compare Register Batch Overwrite Function



Remark As shown above, 16-bit timer KB2 compare register nm (TKBCRnm) has two stages and is treated as a single register except when values are written to.

(1) Timing of batch overwrite

There are three cases when the compare registers are written all together. Among these, (c) cannot be controlled by configuration of the register.

(a) When starting count operation of 16-bit timer KB2n

(b) The count value of the 16-bit counter and the value that is set to 16-bit timer KB2 compare register n0 (TKBCRn0) match.

(c) An external trigger occurs while batch overwrite with an external trigger is enabled.

Remark n = 0 to 2; m = 0 to 3

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 TKBOn1 are generated by TKBCRn2 and TKBCRn3.

The duty can be set within a 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) × Count 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

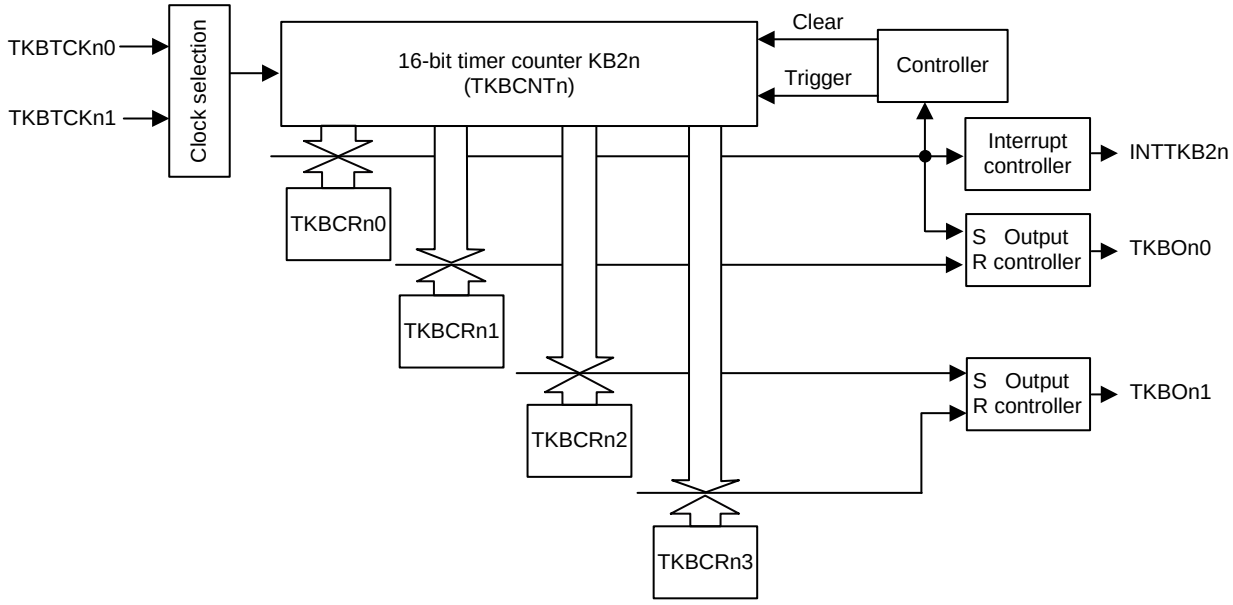
100% output: TKBCRn2 setting = 0000H

TKBCRn3 setting ≥ TKBCRn0 setting + 1

Caution It should always be: TKBCRn2 setting ≤ TKBCRn3 setting.

Figure 7 - 50 shows the Configuration of Standalone Mode (Period Controlled by TKBCRn0).

Figure 7 - 50 Configuration of Standalone Mode (Period Controlled by TKBCRn0)

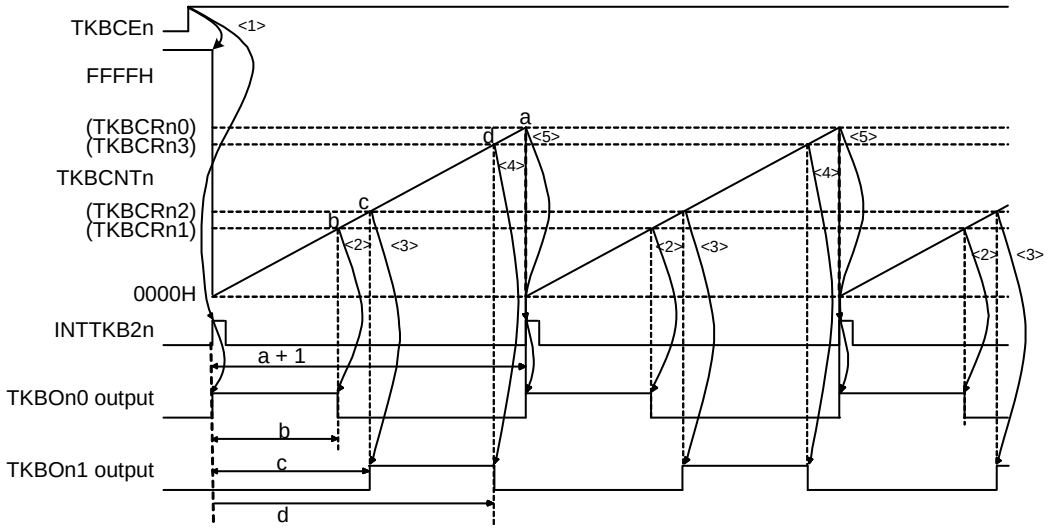


Remark n = 0 to 2

(2) Outline of Operation

Figure 7 - 51 shows the timing sample for standalone mode.

Figure 7 - 51 Timing Sample for Standalone Mode (Period Controlled by TKBCRn0)
(When 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 controlled by TKBCRn0). The following descriptions are linked with <1> to <5> in Figure 7 - 51.

<1> When TKBCEn is set to 1, 16-bit timer counter KB2 (TKBCNTn) changes from FFFFH to 0000H in synchronization with the count clock, then it starts counting up. At the same time, INTTMKB2 output is generated and TKBOn0 output changes from its default value specified with the TKB0TOD00 bit in TKB0IOC00 register to its active value (high level in this example) specified with the TKB0TOL00 bit (TKBOn1 output holds its default value specified with the TKB0TOD01 bit).

<2> When TKBCNTn is counted up and its value matches the value specified in 16-bit timer KB2 compare register n1 (TKBCRn1), TKBOn0 output becomes inactive level.

<3> When TKBCNTn is counted up and its value matches the value specified in 16-bit timer KB2 compare register n2 (TKBCRn2), TKBOn1 output becomes active level.

<4> When TKBCNTn is counted up and its value matches the value specified in 16-bit timer KB2 compare register n3 (TKBCRn3), TKBOn1 output becomes inactive level.

<5> When TKBCNTn is counted up and its value matches the value specified in 16-bit timer KB2 compare register n0 (TKBCRn0), INTTMKB2 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>.

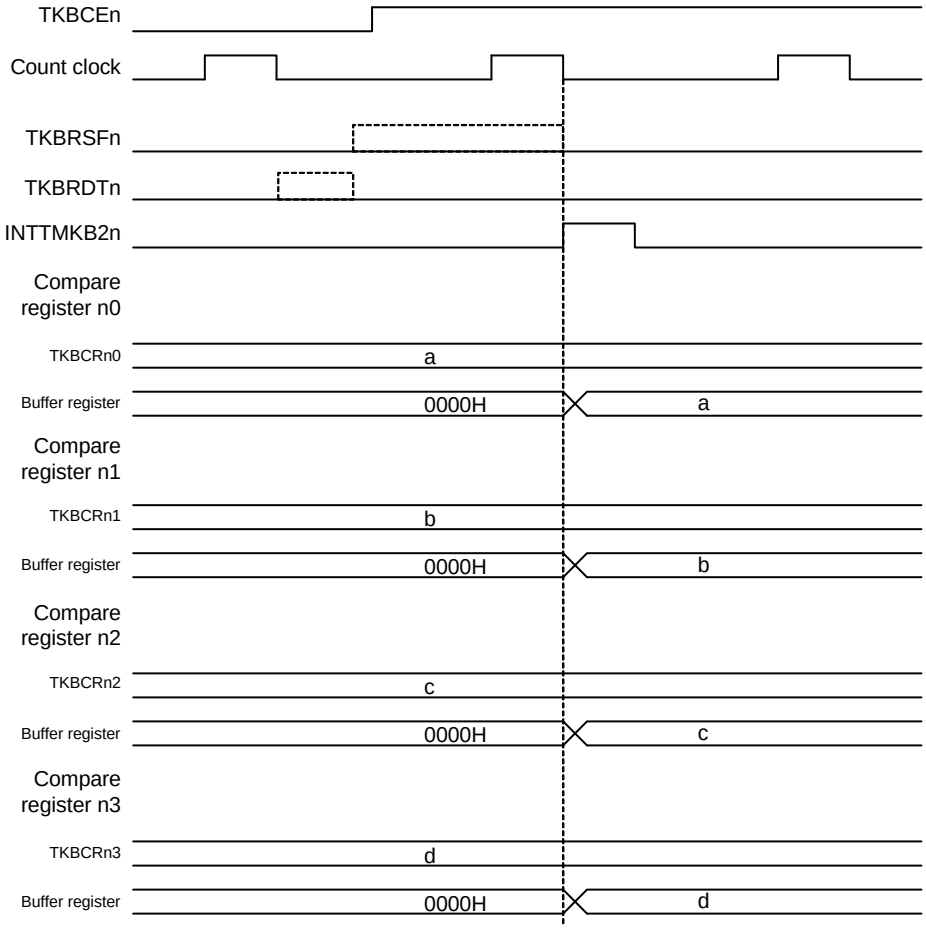
Remark n = 0 to 2; p = 0, 1

(3) Operation of Batch Overwrite (at Starting Counting Operation)

The compare register of 16-bit timer KB2n has a function which updates the internal buffer registers simultaneously at the starting of counter operation caused by the count clock which is generated after writing 1 to the TKBCEn bit in the TKBCTLn1 register.

Batch overwrite is not generated even 1 is not written to the TKBRDTn bit in the TKBTRGn register only in the case of counting operation start timing (see **Figure 7 - 52 Batch Overwrite Function: Figure 1 of Buffer Updating Timing at Counting Operation Start**).

Figure 7 - 52 Batch Overwrite Function: Figure 1 of Buffer Updating Timing at Counting Operation Start



Remark 1. When TKBCEn = 0, TKBRSFn is set to 1 by writing 1 to TKBRDTn. TKBRSFn is cleared to 0 at counting operation start timing (counter start trigger generated).

Remark 2. n = 0 to 2

(4) Batch Overwrite Operation (Buffer Updating during Counting Operation)

The compare register of 16-bit timer KB2n has a function which updates the internal buffer register simultaneously at the next counter clear (TKBCNTn and TKBCRn0 matched), identifying writing 1 to the TKBRDTn bit as batch overwriting trigger. The batch overwrite trigger pending status flag (TKBRSFn) is set during the period from writing 1 to the TKBRDTn bit until completion of batch overwrite (see **Figure 7 - 53 Batch Overwrite Function: Figure 2 of Timing of Buffer Updating during Counting Operation**).

<1> Compare register setting is transferred to the buffer register at the timing when the TKBCEn bit is set from 0 to 1 and TKBCNTn starts counting operation.

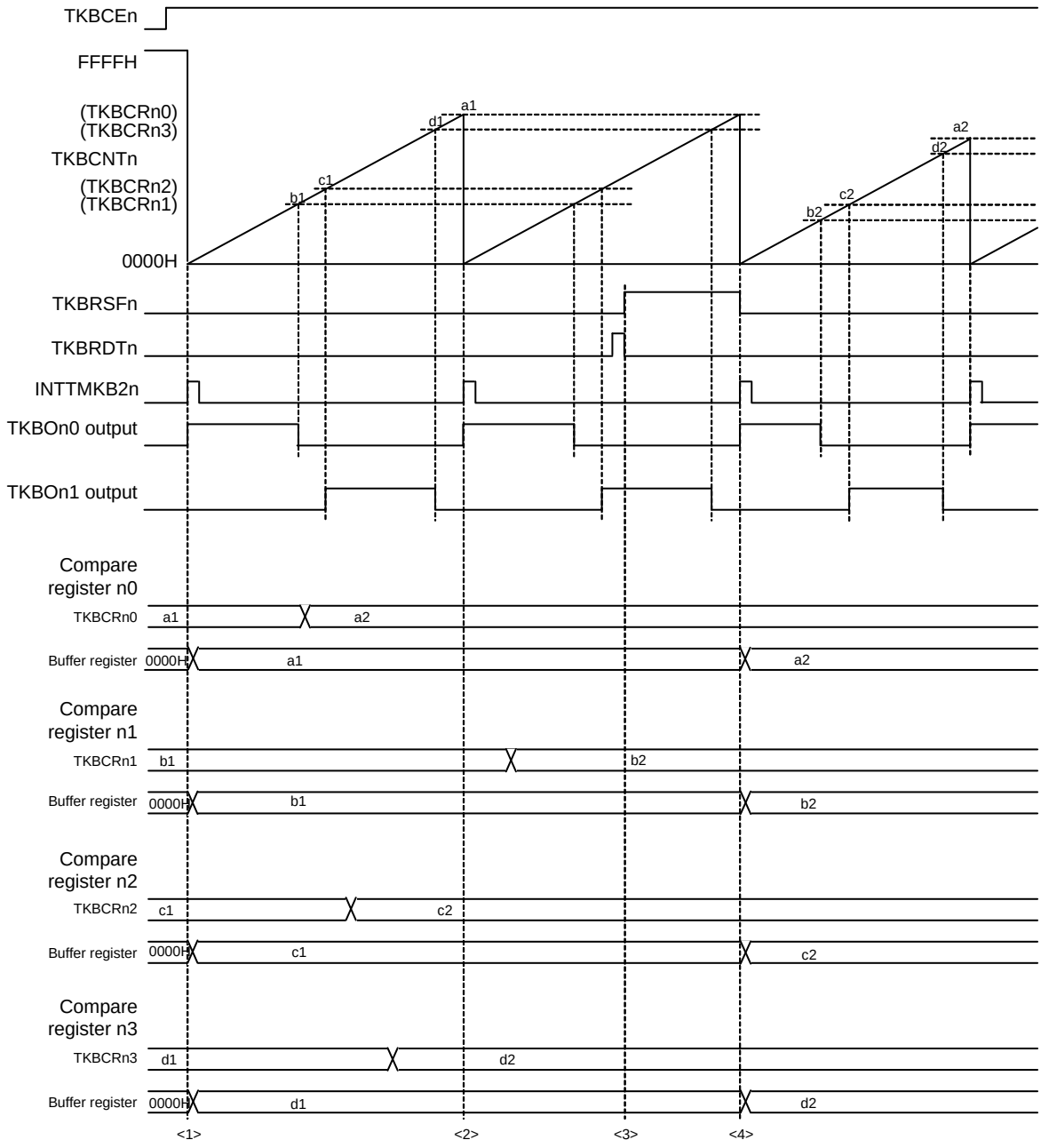
<2> After the TKBCRn0 to TKBCRn3 registers are overwritten, even when counter clear is generated, batch overwrite is not generated if 1 is not written to the TKBRDTn bit.

<3> The batch overwrite trigger pending status flag (TKBRSFn bit) is set to 1 by writing 1 to the TKBRDTn bit.

<4> Compare register setting is transferred to the buffer register by the counter clear generated when the TKBRSFn bit is 1. The TKBRSFn bit is set to 0 simultaneously.

Remark n = 0 to 2

Figure 7 - 53 Batch Overwrite Function: Figure 2 of Timing of Buffer Updating during Counting Operation



Remark n = 0 to 2

(5) Sample of Register Setting Details at Standalone Mode (Period Controlled by TKBCRn0)

	15	14	13	12	11	10	9	8
TKBCTLn0	— 0	— 0	TKBSSEn1 1/0	TKBDIEn1 1/0	— 0	— 0	TKBSSEn0 1/0	TKBDIEn0 1/0
	7	6	5	4	3	2	1	0
	TKBMFE0 0	— 0	TKBIRSn1 0	TKBIRSn0 0	— 0	TKBTSEn 0	TKBSTSn1 0	TKBSTSn0 0
	7	6	5	4	3	2	1	0
TKBCTLn1	TKBCEn 1	— 0	— 0	TKBCKSn0 1/0	TKBSCMn 0	— 0	TKBMDn1 0	TKBMDn0 0
	7	6	5	4	3	2	1	0
TKBIOCn0	— 0	— 0	— 0	— 0	TKBTOLn1 1/0	TKBTOLn0 1/0	TKBTODn1 1/0	TKBTODn0 1/0
	7	6	5	4	3	2	1	0
TKBIOCn1	— 0	— 0	— 0	— 0	— 0	— 0	TKBTOEn1 1/0	TKBTOEn0 1/0
	7	6	5	4	3	2	1	0
TKBPSCSn	— 0	TKBTPSn12 1/0	TKBTPSn11 1/0	TKBTPSn10 1/0	— 0	TKBTPSn02 1/0	TKBTPSn01 1/0	TKBTPSn00 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)

Remark n = 0 to 2

7.4.6 Standalone Mode (Period Controlled by External Trigger Input)

(1) Outline of functions

In standalone mode, the period can be controlled not only by TKBCRn0 but also by external trigger input.

The input signals selected by event link setting register (ELSLER00 to ELSELR22) and the TKBSTSn1 and TKBSTSn0 bits in 16-bit timer KB2n operation control register n0 (TKBCTLn0) are used to detect external trigger input.

When the external trigger input is detected, counter TKBCNTn is cleared to 0000H and TKBOn0/TKBOn1 output is respectively set to active level and inactive level. When the setting values of TKBCRn0 and the counter (TKBCNTn) match before detection of external trigger input, the counter is cleared to 0000H and operation is continued.

For the formula to calculate TKBOn0/TKBOn1 output when external trigger input is not yet detected and the period is controlled by TKBCRn0, see **7.4.5 Standalone Mode (Period Controlled by TKBCRn0)**.

The calculation formula for TTKBOn0/TKBOn1 output when the period is 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 at 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 at external trigger input detection + 1)) × 100

0% output: TKBCRn3 setting = TKBCRn2 setting

100% output: TKBCRn2 setting = 0000H

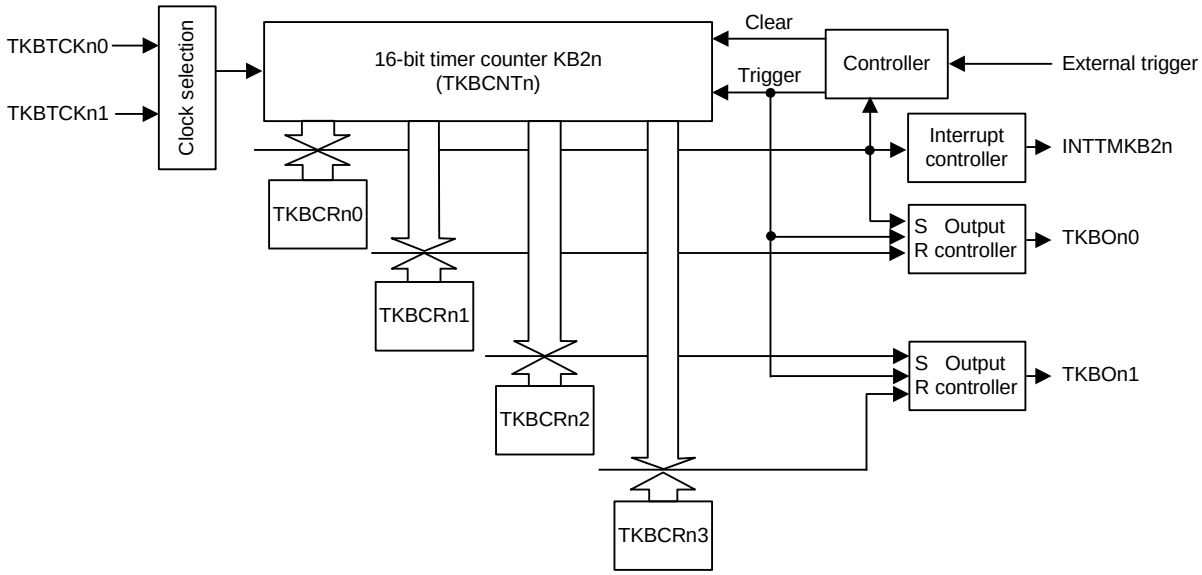
TKBCRn3 setting ≥ Counter value at external trigger input detection + 1

Caution It should always be: TKBCRn2 setting ≤ KBCR03 setting.

Figure 7 - 54 shows the Configuration of Standalone Mode (Period Controlled by External Trigger Input).

Remark n = 0 to 2

Figure 7 - 54 Configuration of Standalone Mode (Period Controlled by External Trigger Input)



Remark n = 0 to 2

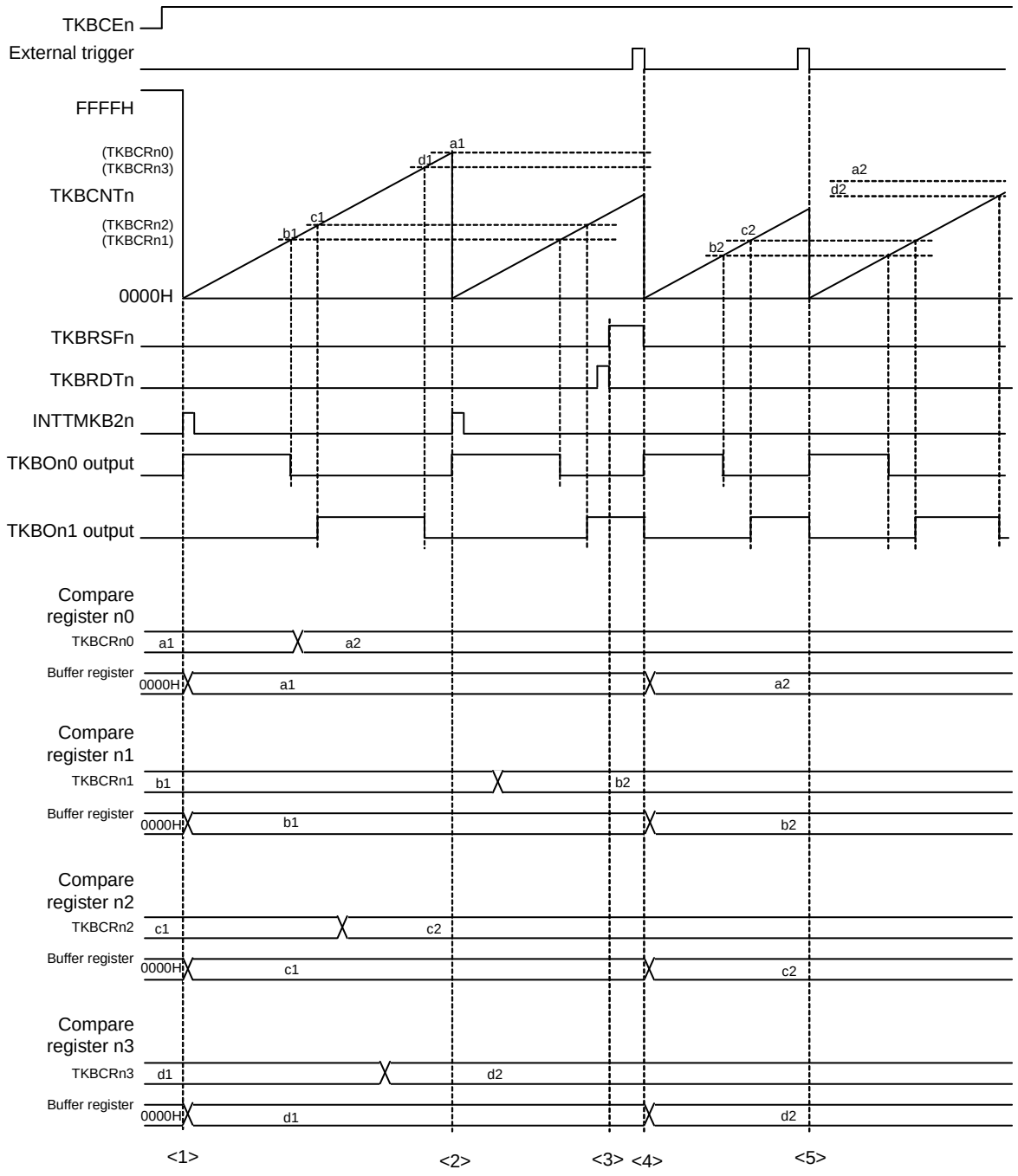
(2) Batch Overwrite Function (In Standalone Operation during Period Controlled by External Trigger Input, Buffer Updating during Counting Operation (TKBTSEn bit set to 1))

In standalone operation during the period controlled by external trigger input, counter clear and compare register batch overwrite are implemented at the timing when external trigger input is detected after writing 1 to the TKBRDTn bit and by setting the TKBTSEn bit in TKBCTLn0 register to 1. Same as in counter clear, batch overwrite is also implemented when TKBCRn0 and counter (TKBCNTn) match before detection of external trigger input after writing 1 to the TKBRDTn bit. The source of external trigger input is selected by the ELSLER00 to ELSELR22 registers and the TKBSTSn1 and TKBSTSn0 bits in the TKBCTLn0 register. Figure 7 - 55 shows an example of the timing of batch overwrite operation when TKBTSEn bit is set to 1.

<1> Compare register setting is transferred to the buffer register at the timing when TKBCEn bit is set from 0 to 1 and TKBCNTn starts counting operation.
<2> After the TKBCRn0 to TKBCRn3 registers are overwritten, even when counter clear is generated, batch overwrite is not generated if 1 is not written to the TKBRDTn bit.
<3> The batch overwrite trigger pending status flag (TKBRSFn bit) is set to 1 by writing 1 to the TKBRDTn bit.
<4> When counter clear is generated by external trigger input while TKBTSEn bit is set to 1 and TKBRSFn bit is 1, the setting value of the compare register is transferred to the buffer register. At the same time, the TKBRSFn bit becomes 0.
<5> Even if the counter clear event is generated by external trigger input, batch overwrite does not occur unless 1 is written to the TKBRDTn bit.

Remark n = 0 to 2

Figure 7 - 55 Batch Overwrite Function: Figure of Standalone Operation during Period Controlled by External Trigger Input and Timing of Buffer Updating during Counting Operation (TKBTSEn bit set to 1)



Remark n = 0 to 2

(3) Batch Overwrite Function (Standalone Operation during Period Controlled by External Trigger Input, Buffer Updating during Counting Operation (TKBTSEn bit clear to 0))

This is an example when the TKBTSEn bit in the TKBCTLn0 register is set to 0 in standalone operation during the period controlled by external trigger input. In this case, the counter is cleared when 1 is written to the TKBRDTn bit and the external trigger input is detected while the batch overwrite trigger pending status flag (TKBRSFn bit) is 1. However, batch overwrite of the compare register is not implemented.

The source of external trigger input is selected by the ELSLER00 to ELSELR22 registers and the TKBSTSn1 and TKBSTSn0 bits in the TKBCTLn0 register. Figure 7 - 56 shows an example of the batch overwrite operation timing when the TKBTSEn bit is set to 0.

<1> Compare register setting is transferred to the buffer register at the timing when the TKBCEn bit is set from 0 to 1 and TKBCNTn starts counting operation.

<2> After the TKBCRn0 to TKBCRn3 registers are overwritten, the batch overwrite trigger pending status flag (TKBRSFn bit) is set to 1 by writing 1 to the TKBRDTn bit.

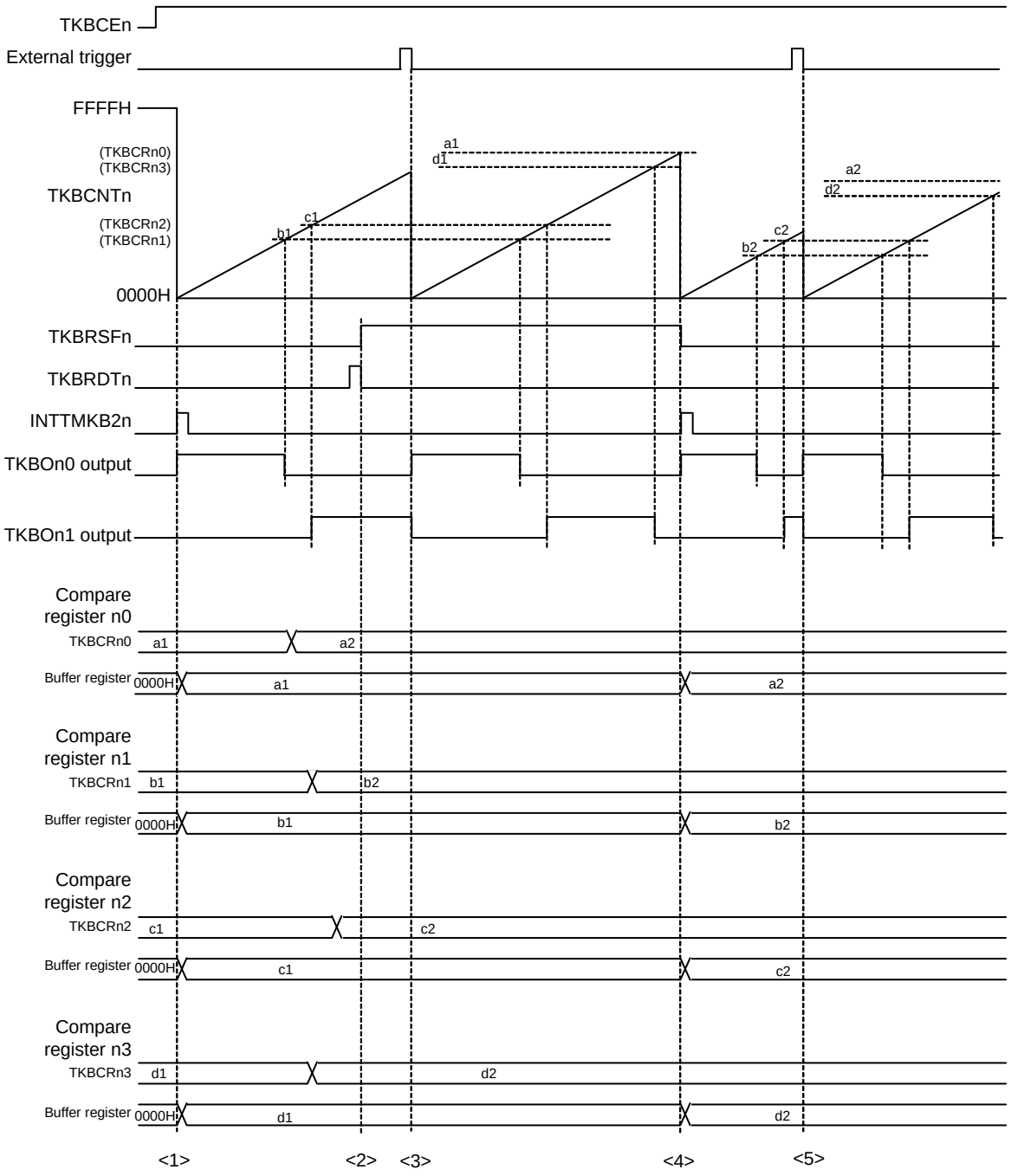
<3> Even if the counter clear event is generated by external trigger input, batch overwrite does not occur unless the TKBTSEn bit is set to 1.

<4> When the counter clear event (TKBCNTn and TKBCRn0 match) is generated while TKBRSFn bit is 1, the setting value of the compare register is transferred to the buffer register. At the same time, the TKBRSFn bit becomes 0.

<5> Even if the counter clear event is generated by external trigger input, batch overwrite does not occur unless the TKBTSEn and TKBRSFn bits are both 1.

Remark n = 0 to 2

Figure 7 - 56 Batch Overwrite Function: Figure of Standalone Operation during Period Controlled by External Trigger Input and Timing of Buffer Updating during Counting Operation (TKBTSEn bit cleared to 0)



Remark n = 0 to 2

(4) Sample of Register Setting Details at Standalone Mode (Period Controlled by External Trigger Input)

	15	14	13	12	11	10	9	8
TKBCTLn0	0	— 0	TKBSSEn1 0	TKBDIEn1 0	— 0	— 0	TKBSSEn0 0	TKBDIEn0 0
	7	6	5	4	3	2	1	0
	TKBMFE0 1/0	— 0	TKBIRSn1 0	TKBIRSn0 0	— 0	TKBTSEn 1/0	TKBSTSn1 1/0	TKBSTSn0 1/0
	7	6	5	4	3	2	1	0
TKBCTLn1	TKBCEn 1	— 0	— 0	TKBCKSn0 1/0	TKBSCMn 0	— 0	TKBMDn1 0	TKBMDn0 0
	7	6	5	4	3	2	1	0
TKBIOCn0	— 0	— 0	— 0	— 0	TKBTOLn1 1/0	TKBTOLn0 1/0	TKBTODn1 1/0	TKBTODn0 1/0
	7	6	5	4	3	2	1	0
TKBIOCn1	— 0	— 0	— 0	— 0	— 0	— 0	TKBTOEn1 1/0	TKBTOEn0 1/0
	7	6	5	4	3	2	1	0
TKBPSCSn	— 0	TKBTPSn12 1/0	TKBTPSn11 1/0	TKBTPSn10 1/0	— 0	TKBTPSn02 1/0	TKBTPSn01 1/0	TKBTPSn00 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)

Remark n = 0 to 2

7.4.7 Simultaneous Start / Stop Mode

(1) Outline of functions

Slave timer KB2n can be start/stop simultaneously by synchronization with count start/stop of Master timer KB20 when Master/Slave is configured using multiple KB2m 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 the count clocks (TKBTCKn0/TKBTCKn1) selected for the master and slave differ, counting operation start timing for the master and slave can be arranged by setting the TKBSCMn bits of the master and slave to all 1.

Each timer operates separately after the timing for counting operation to be started.

Caution 1. Master is timer KB20 only.

Caution 2. Master selecting clock must be faster or with the same speed as Slave selecting clock.

Caution 3. Be sure to set the TKBPSCSn register to the same value, and use TKBCKSn to set the count clock for timer KB2n.

Relationship of selected clock between Master and Slave	Relationship between TKBCK00/TKBCK01 and TKBCKM0/TKBCKm1	TKBSCM0 bit of timer KB20	TKBSCM1 bit of timer KB21	TKBSCM2 bit of timer KB22	Available
Selecting the same clock for Master and Slave	—	0	0	0	√
Selecting different clocks for Master and Slave	When Master selected clock is faster than Slave selected clock	1	1	1	√
Selecting different clocks for Master and Slave	When Master selected clock is faster than Slave selected clock	Other than above			×
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, 7.4.5 Standalone Mode (Period Controlled by TKBCRn0) and 7.4.6 Standalone Mode (Period Controlled by External Trigger Input).

Remark n = 0 to 2; m = 1, 2

(2) Operation Mode Combination Available for Simultaneous Start / Stop Mode

The following shows the operation mode available for simultaneous start / stop mode.

Master:

Operation Mode	TKBMD01, TKBMD00	TKBSTS01, TKBSTS00	Setting Available
Standalone Mode (Period controlled by TKBCR00)	00B	00B	√
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	—	×
Interleave PFC Output Mode	11B	—	×

Remark m = 1, 2

(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
TKBCTL00	— 0	— 0	TKBSSE01 1/0	TKBDIE01 1/0	— 0	— 0	TKBSSE00 1/0	TKBDIE00 1/0
	7	6	5	4	3	2	1	0
	TKBMFE0 0	— 0	TKBIRS01 0	TKBIRS00 0	— 0	TKBTSE0 0	TKBSTS01 0	TKBSTS00 0
	7	6	5	4	3	2	1	0
TKBCTL01	TKBCE0 1	— 0	— 0	TKBCKS0 1/0	TKBSCM0 1	— 0	TKBMD01 0	TKBMD00 0
	7	6	5	4	3	2	1	0
TKBIOC00	— 0	— 0	— 0	— 0	TKBTOL01 1/0	TKBTOL00 1/0	TKBTOD01 1/0	TKBTOD00 1/0
	7	6	5	4	3	2	1	0
TKBIOC01	— 0	— 0	— 0	— 0	— 0	— 0	TKBTOE01 1/0	TKBTOE00 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)

(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	— 0	— 0	TKBSSEn1 1/0	TKBDIEn1 1/0	— 0	— 0	TKBSSEn0 1/0	TKBDIEn0 1/0
	7	6	5	4	3	2	1	0
	TKBMFEn 0	— 0	TKBIRSn1 0	TKBIRSn0 0	— 0	TKBTSEn 0	TKBSTSn1 0	TKBSTSn0 0
	7	6	5	4	3	2	1	0
TKBCTLn1	TKBCEn 1	— 0	— 0	TKBCKSn 1/0	TKBSCMn 1	— 0	TKBMDn1 0	TKBMDn0 1
	7	6	5	4	3	2	1	0
TKBIOCn0	— 0	— 0	— 0	— 0	TKBTOLn1 1/0	TKBTOLn0 1/0	TKBTODn1 1/0	TKBTODn0 1/0
	7	6	5	4	3	2	1	0
TKBIOCn1	— 0	— 0	— 0	— 0	— 0	— 0	TKBTOEn1 1/0	TKBTOEn0 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)

Remark n = 1, 2

(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
TKBCTL00	— 0	— 0	TKBSSE01 0	TKBDIE01 0	— 0	— 0	TKBSSE00 0	TKBDIE00 0
	7	6	5	4	3	2	1	0
	TKBMFE0 1/0	— 0	TKBIRS01 0	TKBIRS00 0	— 0	TKBTSE0 1/0	TKBSTS01 1/0	TKBSTS00 1/0
	7	6	5	4	3	2	1	0
TKBCTL01	TKBCE0 1	— 0	— 0	TKBCKS0 1/0	TKBSCM0 1	— 0	TKBMD01 0	TKBMD00 0
	7	6	5	4	3	2	1	0
TKBIOC00	— 0	— 0	— 0	— 0	TKBTOL01 1/0	TKBTOL00 1/0	TKBTOD01 1/0	TKBTOD00 1/0
	7	6	5	4	3	2	1	0
TKBIOC01	— 0	— 0	— 0	— 0	— 0	— 0	TKBTOE01 1/0	TKBTOE00 1/0
TKBCR00	0000H to FFFFH							
TKBCR01	0000H to FFFFH							
TKBCR02	0000H to FFFFH							
TKBCR03	0000H to FFFFH							
TKBTGCR0	0000H to FFFFH							
TKBSIR00	0000H							
TKBSIR01	0000H							
TKBSSR00	00H							
TKBSSR01	00H							
TKBDNR00	00H							
TKBDNR01	00H							
TKBMFR0	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	— 0	— 0	TKBSSEn1 0	TKBDIEn1 0	— 0	— 0	TKBSSEn0 0	TKBDIEn0 0
	7	6	5	4	3	2	1	0
	TKBMFEn 1/0	— 0	TKBIRSn1 0	TKBIRSn0 0	— 0	TKBTSEn 1/0	TKBSTSn1 1/0	TKBSTSn0 1/0
	7	6	5	4	3	2	1	0
TKBCTLn1	TKBCEn 1	— 0	— 0	TKBCKSn 1/0	TKBSCM0 1	— 0	TKBMDn1 0	TKBMDn0 1
	7	6	5	4	3	2	1	0
TKBIOCn0	— 0	— 0	— 0	— 0	TKBTOLn1 1/0	TKBTOLn0 1/0	TKBTODn1 1/0	TKBTODn0 1/0
	7	6	5	4	3	2	1	0
TKBIOCn1	— 0	— 0	— 0	— 0	— 0	— 0	TKBTOEn1 1/0	TKBTOEn0 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							

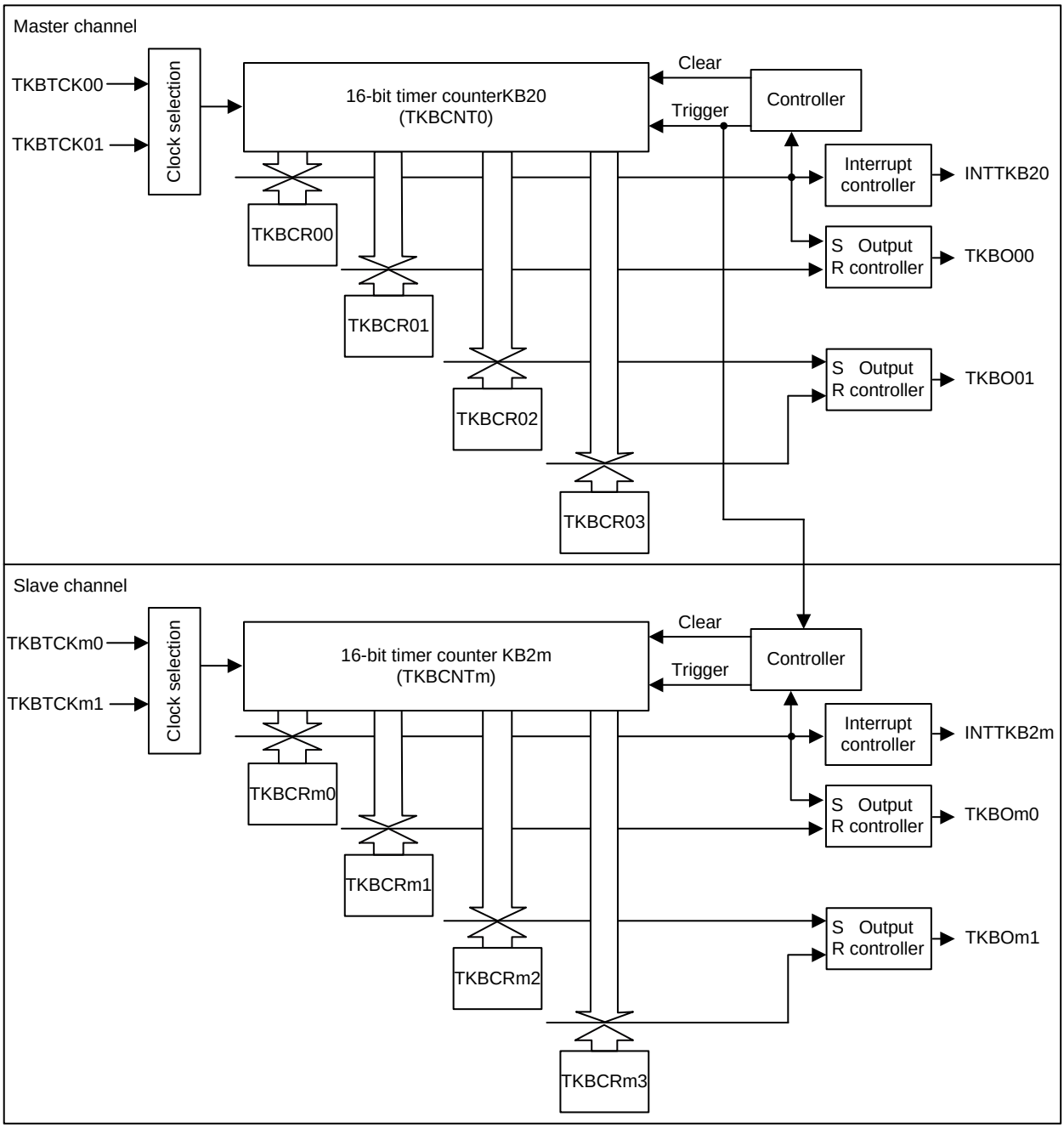
: Setting is fixed for this mode : Setting is not needed (default setting)

Remark n = 1, 2

(7) Configuration of Simultaneous Start / Stop Mode (Period controlled by TKBCRn0)

Figure 7 - 57 shows the configuration of simultaneous start / stop mode.

Figure 7 - 57 Configuration of Simultaneous Start / Stop Mode (Period controlled by TKBCRn0)

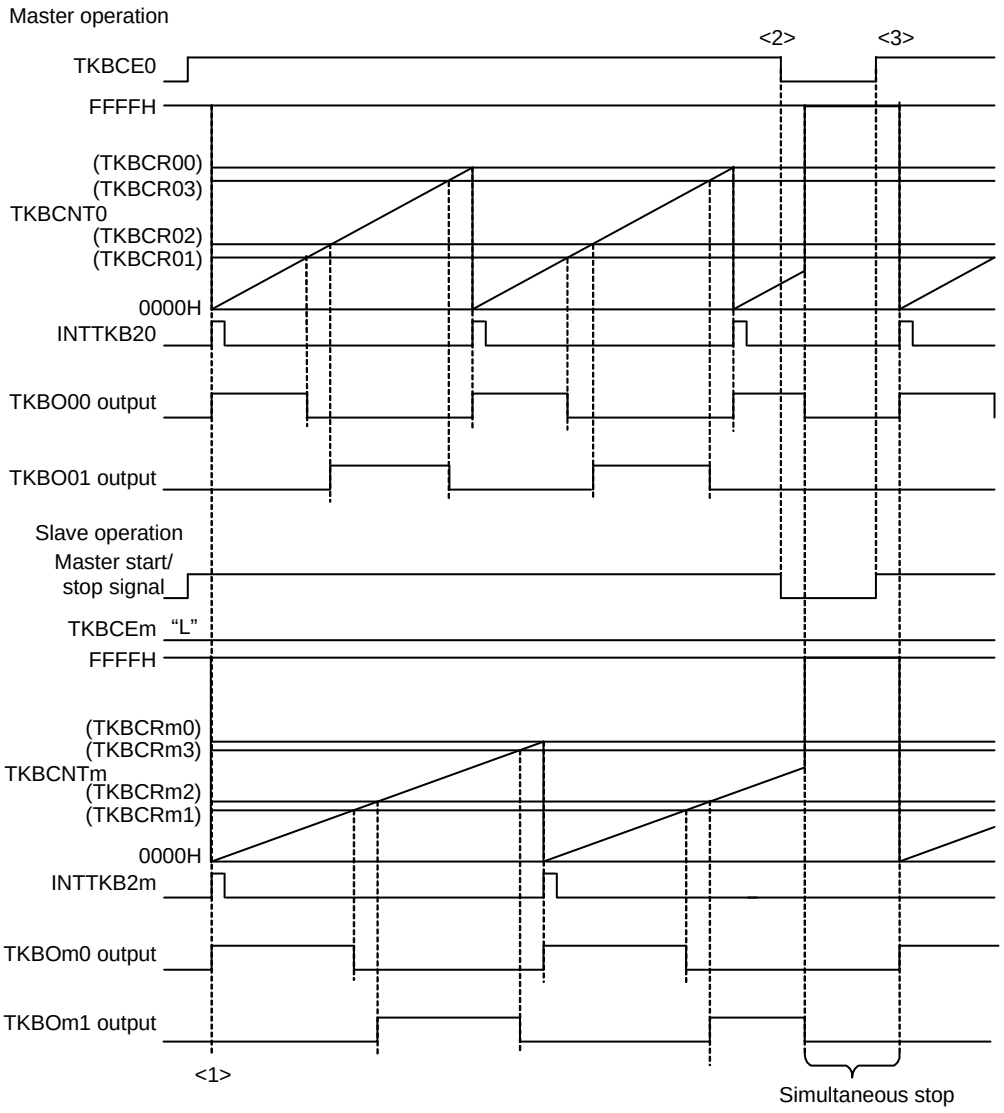


Remark n = 0 to 2; m = 1, 2

(8) Outline of operation

Figure 7 - 58 shows the timing sample for simultaneous start / stop mode.

Figure 7 - 58 Timing Sample for Simultaneous Start / Stop Mode (Period controlled by TKBCRn0)
(at default value of output is low level (TKBTODnp = 0) and active level is high level (TKBTOLnp = 0))



Remark n = 0 to 2; p = 0, 1

The following describes an operational example of simultaneous start / stop mode. The following descriptions are linked with <1> to <3> in Figure 7 - 58.

<1> When the master TKBCE0 is set to 1, the master 16-bit timer counter KB20 (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 INTTKB20 and INTTKB2m 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, refer to **Figure 7 - 51 Timing Sample for Standalone Mode (Period Controlled by TKBCRn0) (When default value of output is low level (TKBTODnp = 0) and active level is high level (TKBTOLnp = 0))**.

<2> If TKBCEn is set to 0, synching with the input clock of timer KB20, 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.

Remark n = 0 to 2; m = 1, 2; p = 0, 1

7.4.8 Synchronous Start / Clear Mode

Slave timer KB2m can be started and cleared simultaneously by synchronization with timings of start/stop of counting by Master timer KB20, counter clearing and batch overwriting when Master/Slave is configured using multiple KB timers.
Operate by “Standalone Mode (TKBMD01, TKBMD00 = 00)” for Master and “Synchronous Start / Clear Mode (TKBMDm1, TKBMDm0 = 10)” for Slave in such case.
Make sure that the same division clock is selected for the master/slave.
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 TKBRSF0 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 n = 0 to 2; m = 1, 2

(1) Operation Mode Combination Available for Synchronous Start / Clear Mode

The following 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	√
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	—	√
Interleave PFC Output Mode	11B	—	×

Remark m = 1, 2

(2) Synchronous Start / Clear Mode: List of register setting by Master

	15	14	13	12	11	10	9	8
TKBCTL00	— 0	— 0	TKBSSE01 1/0	TKBDIE01 1/0	— 0	— 0	TKBSSE00 1/0	TKBDIE00 1/0
	7	6	5	4	3	2	1	0
	TKBMFE0 0	— 0	TKBIRS01 0	TKBIRS00 0	— 0	TKBTSE0 0	TKBSTS01 0	TKBSTS00 0
	7	6	5	4	3	2	1	0
TKBCTL01	TKBCE0 1	— 0	— 0	TKBCKS0 1/0	TKBSCM0 0	— 0	TKBMD01 0	TKBMD00 0
	7	6	5	4	3	2	1	0
TKBIOC00	— 0	— 0	— 0	— 0	TKBTOL01 1/0	TKBTOL00 1/0	TKBTOD01 1/0	TKBTOD00 1/0
	7	6	5	4	3	2	1	0
TKBIOC01	— 0	— 0	— 0	— 0	— 0	— 0	TKBTOE01 1/0	TKBTOE00 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) Synchronous Start / Clear Mode: List of register setting by Slave

	15	14	13	12	11	10	9	8
TKBCTLn0	— 0	— 0	TKBSSEn1 1/0	TKBDIEn1 1/0	— 0	— 0	TKBSSEn0 1/0	TKBDIEn0 1/0
	7	6	5	4	3	2	1	0
	TKBMFEn 0	— 0	TKBIRSn1 0	TKBIRSn0 0	— 0	TKBTSEn 0	TKBSTSn1 0	TKBSTSn0 0
	7	6	5	4	3	2	1	0
TKBCTLn1	TKBCEn 1	— 0	— 0	TKBCKSn 1/0	TKBSCMn 0	— 0	TKBMDn1 1	TKBMDn0 0
	7	6	5	4	3	2	1	0
TKBIOCn0	— 0	— 0	— 0	— 0	TKBTOLn1 1/0	TKBTOLn0 1/0	TKBTODn1 1/0	TKBTODn0 1/0
	7	6	5	4	3	2	1	0
TKBIOCn1	— 0	— 0	— 0	— 0	— 0	— 0	TKBTOEn1 1/0	TKBTOEn0 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							

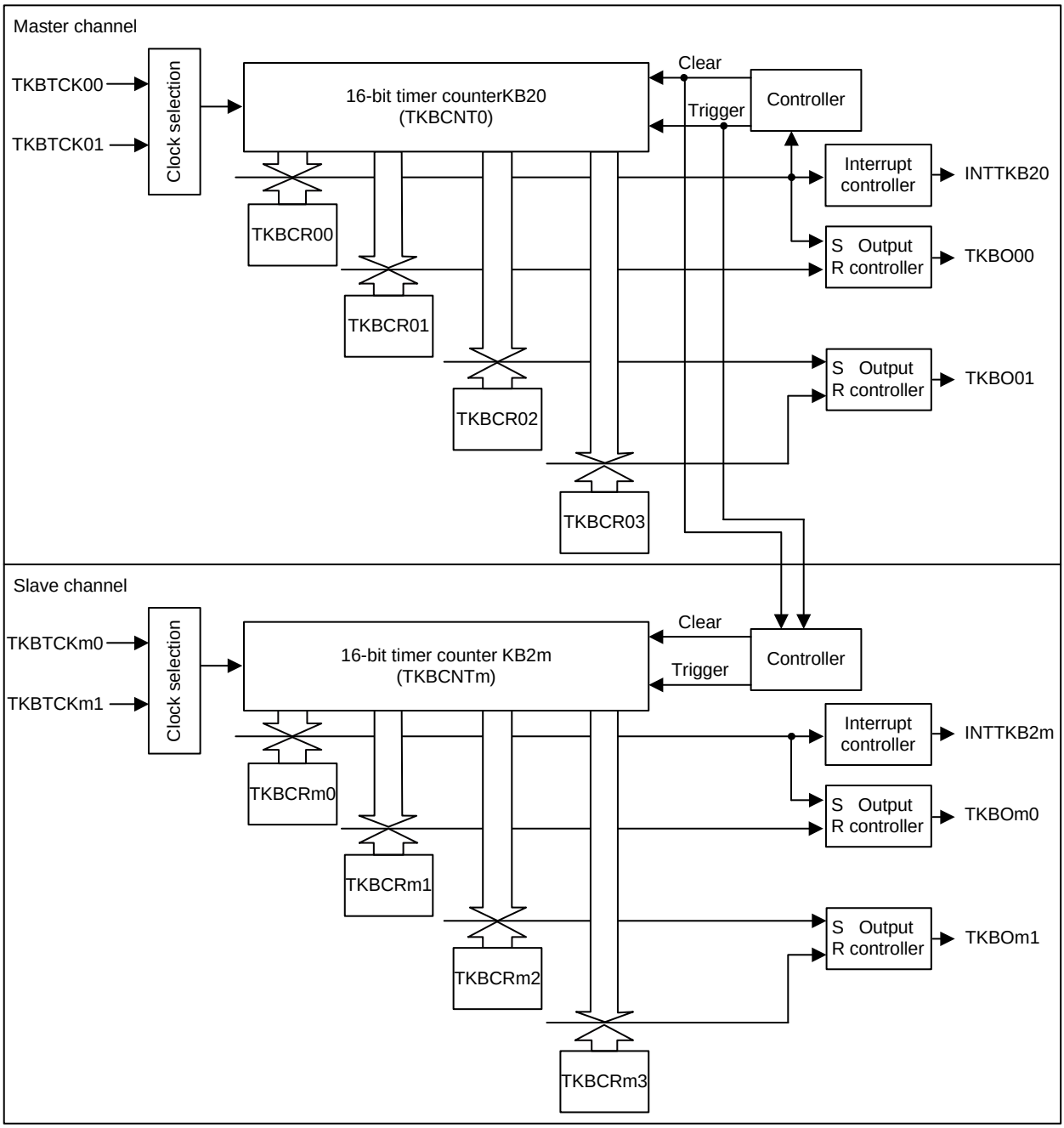
: Setting is fixed for this mode : Setting is not needed (default setting)

Remark n = 1, 2

(4) Configuration of Synchronous Start / Clear Mode (Period Controlled by Master)

Figure 7 - 59 shows the configuration of synchronous start / clear mode.

Figure 7 - 59 Configuration of Synchronous Start / Clear Mode (Period Controlled by Master)

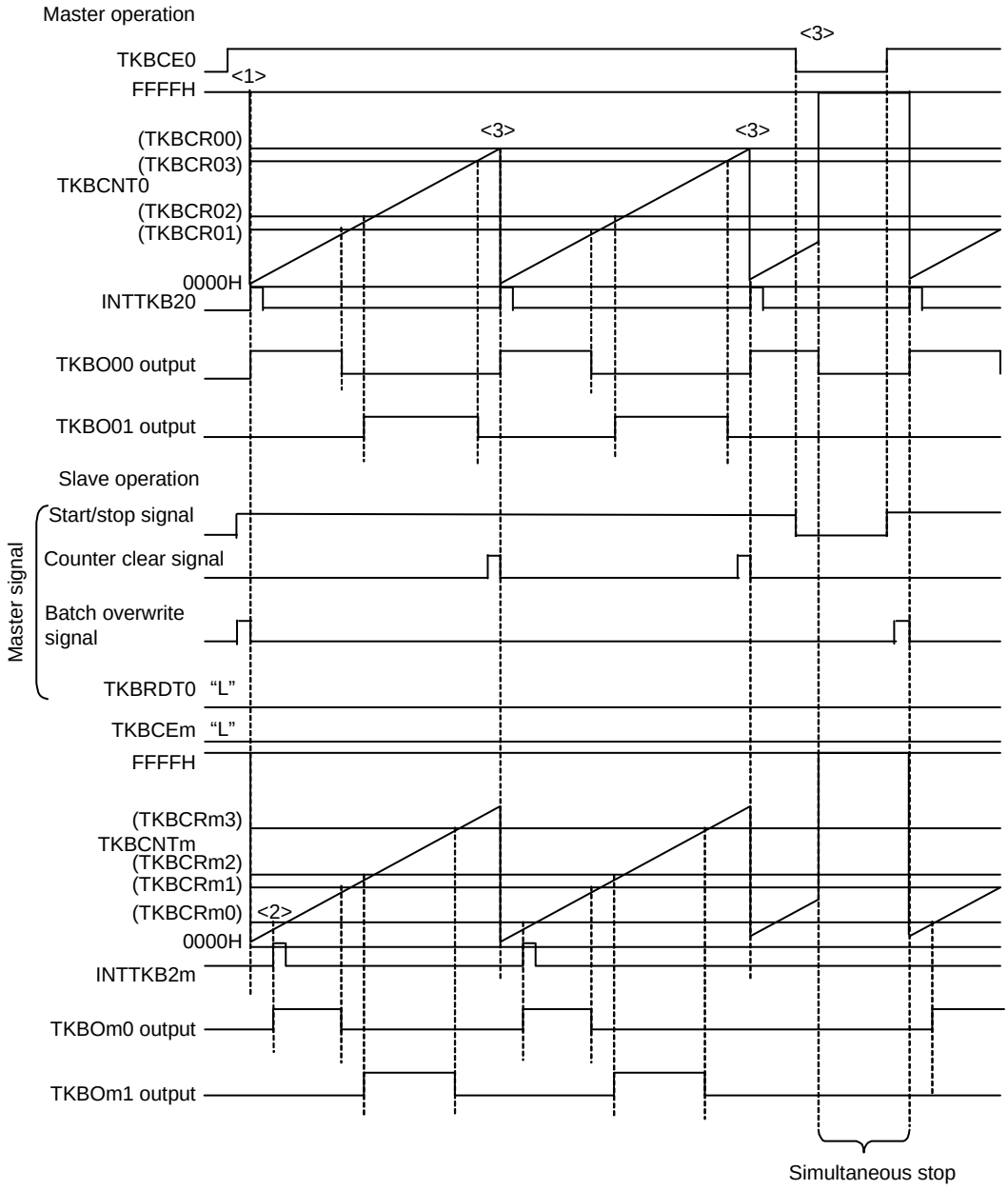


Remark m = 1, 2

(5) Outline of operation

Figure 7 - 60 shows the timing sample for synchronous start / clear mode.

Figure 7 - 60 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))



Remark 0 to 2; m = 1, 2; p = 0, 1

The following describes an operational example of synchronous start / clear mode. The following descriptions are linked with <1> to <4> in Figure 7 - 60.

<1> When the master TKBCE0 is set to 1, the master 16-bit timer counter KB20 (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 INTTKB20 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 TKBCRm0, TKBOm0 output of slave becomes active level. For further detailed operation, refer to **Figure 7 - 51 Timing Sample for Standalone Mode (Period Controlled by TKBCRn0) (When default value of output is low level (TKBTODnp = 0) and active level is high level (TKBTOLnp = 0))**.

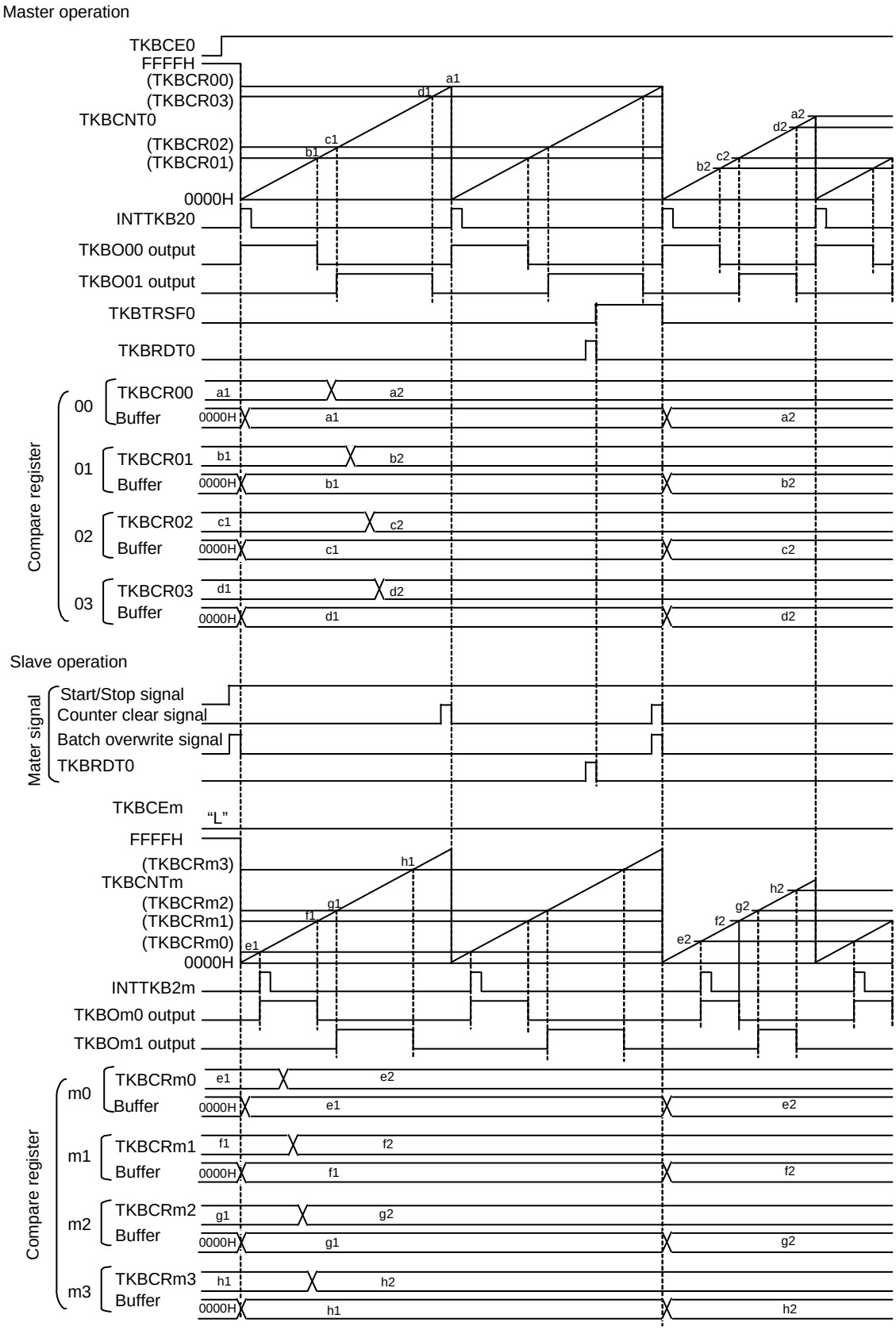
<3> When count value of TKBCNT0 matches with the value specified in TKBCR00, clear signal of master is output. At the same time, 16-bit timer counter (TKBCCNT0, TKBCNTm) for master and slave is cleared.

<4> If TKBCE0 is set to 0, synching with the input clock of timer KB20, 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 - 61 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.

Remark n = 0 to 2; m = 1, 2

Figure 7 - 61 Timing Sample for Synchronous Start / Clear Mode (Period Controlled by Master) (at batch overwrite)



Remark m = 1, 2

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.
Interleaved PFC output mode is implemented by a combination of external interrupt input selected for counter restart trigger source 0 and TKBOn0, and external interrupt input selected for counter restart trigger source 1 and TKBOn1.
TKBOn1 phased shifted by 180 degrees by external interrupt input selected for counter restart trigger source 1 is output based on the TKBOn0 output controlled by external interrupt input selected for counter restart trigger source 0.

Remark 1. Single PFC control can be implemented in standalone mode (period controlled by external input trigger). For more detail, refer to **7.4.6 Standalone Mode (Period Controlled by External Trigger Input)**.

The counter restart period is set by TKBCRn0 if external interrupt input selected for counter restart trigger source 0 is not detected.
The active width of TKBOn0 output is set by TKBCRn1.
The active width of TKBOn1 output is set by TKBCRn3.

Remark 2. Interleave PFC (Power Factor Correction) output mode does not use TKBCRn2.

The setting value of the TKBTOLn0 bit and the TKBTODn0 bit, and the TKBTOLn1 bit and the 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 and TKBOn1 Output]

Pulse period (MAX) Note = (TKBCRn0 setting + 1) × Count clock period
Active width of TKBOn0 output = TKBCRn1 setting × Count clock period
Active width of TKBOn1 output = TKBCRn3 setting × Count clock period

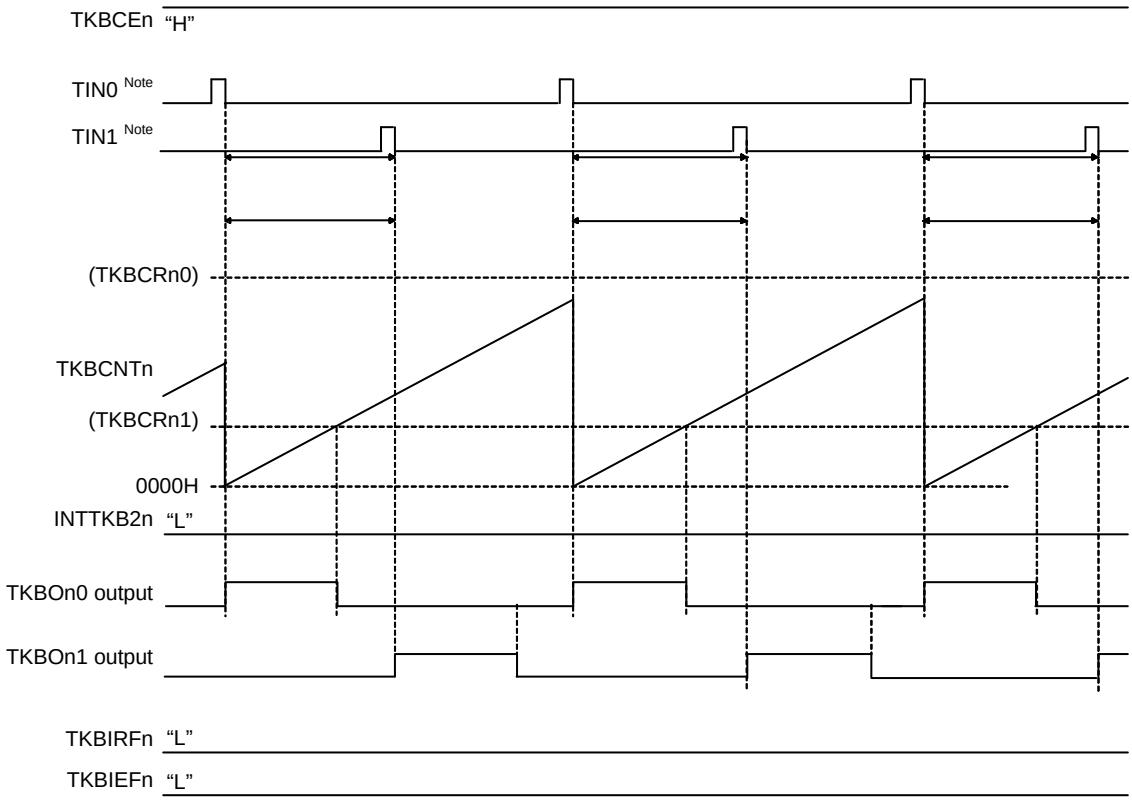
Note This is the counter restart period in case when external interrupt input selected for counter restart trigger source 0 not being detected.

Figure 7 - 62 shows the overview of basic operation of interleave PFC mode. In basic operation of interleave PFC mode, TKBCNTn is incremented from 0000H by external interrupt input selected for counter restart trigger source 0 as a trigger. In this case, TKBOn0 becomes active level, and then becomes inactive level when it matches the setting value of the TKBCRn1 register.
TKBOn1 becomes active level by being triggered by external interrupt input selected for counter restart trigger source 0, and becomes inactive level when it matches the setting value of TKBCRn3 register.

Remark n = 0 to 2

Another external interrupt input selected for counter restart trigger source 0 comes in before TKBCNTn matches the setting value of the TKBCRn0 register, and then the above operation is repeated.

Figure 7 - 62 Overview of Basic Operation of Interleave PFC Mode
(When default value of output is low level (TKBTODnp = 0) and active level is high level (TKBTOLnp = 0))



Note TIN0 indicates an external interrupt assigned to counter restart trigger source 0. TIN1 indicates an external interrupt assigned to counter restart trigger source 1.

Remark n = 0 to 2; p = 0, 1

(1) Output Conditions of TKBOn1 at Interleave PFC

There are output conditions for TKBOn1 output which are controlled according to the table below.

Status No.	Output Conditions for TKBOn1 Output			TKBOn1 Start at High Level
	TIN0 Note 1 Input	Matching with CR00/TIN1 Note 1	Period Width	
1	First period	—	—	Output start by T/2 Note 4
2	TIN0 input not detected	Matching of TKBCNTn and CR00 (Ignore TIN1 input detection)	Subsequent period (CR00 value) is over 1/2 the previous period	Output start by T/2
3	↑	↑	Subsequent period (CR00 value) is below 1/2 the previous period	Maintain the status
4	Subsequent period of No.3	—	—	Output start by T/2
5	TIN0 input detected (for the first time) Note 2	—	—	Output start by T/2
6	TIN0 input detected (from the second time) Note 3	TIN1 detected (within the range from previous TOUT1 falling edge to T/2)	—	Output start by T/2
7	TIN0 input detected (from the second time) Note 3	TIN1 detected (T/2 to T/2 + T/(TKBIRSn1 to TKBIRSn0 setting) range)	—	Output start by trigger input
8	TIN0 input detected (from the second time) Note 3	TIN1 detected (after the range T/(TKBIRSn1 to TKBIRSn0 setting)	—	Maintain the status
9	Subsequent period of No.8	—	—	Output start by T/2
10	TIN0 input detected	—	Subsequent period is below T/2	Maintain the status
11	Subsequent period of No.10	—	—	Output start by T/2

Note 1. TIN0 indicates an external interrupt assigned to counter restart trigger source 0. TIN1 indicates an external interrupt assigned to counter restart trigger source 1.

Note 2. TIN0 input detected (for the first time) means that the previous period was not cleared for TIN0 input being detected.

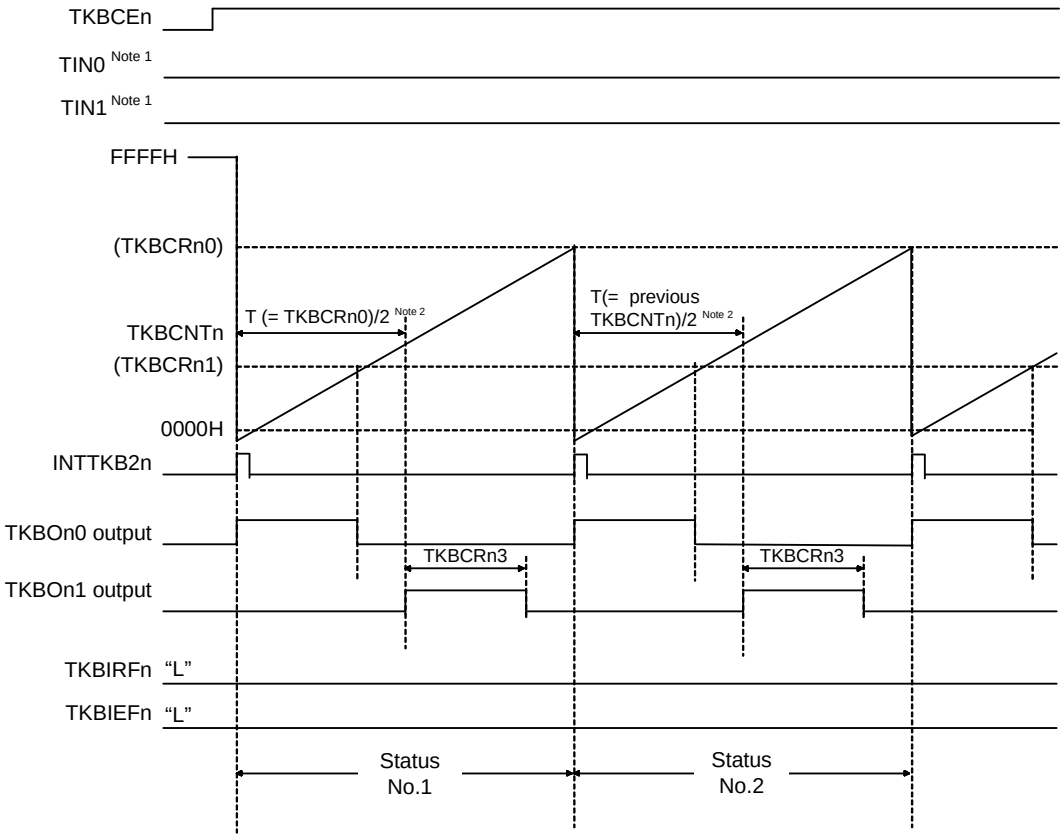
Note 3. TIN0 input detected (from the second time) means that the previous period being cleared for TIN0 input being detected.

Note 4. T is calculated by the following formula.
When the setting value of TKBCRn0 is even, T = TKBCRn0 + 2
When the setting value of TKBCRn0 is odd, T = TKBCRn0 + 1

Remark n = 0 to 2

See the following figures of the waveform corresponding to each "Condition No."

Figure 7 - 63 Figure of Timing of Interleave PFC Mode (Operation for Status No. 1 to 2)



Note 1. TIN0 indicates an external interrupt assigned to counter restart trigger source 0. TIN1 indicates an external interrupt assigned to counter restart trigger source 1.

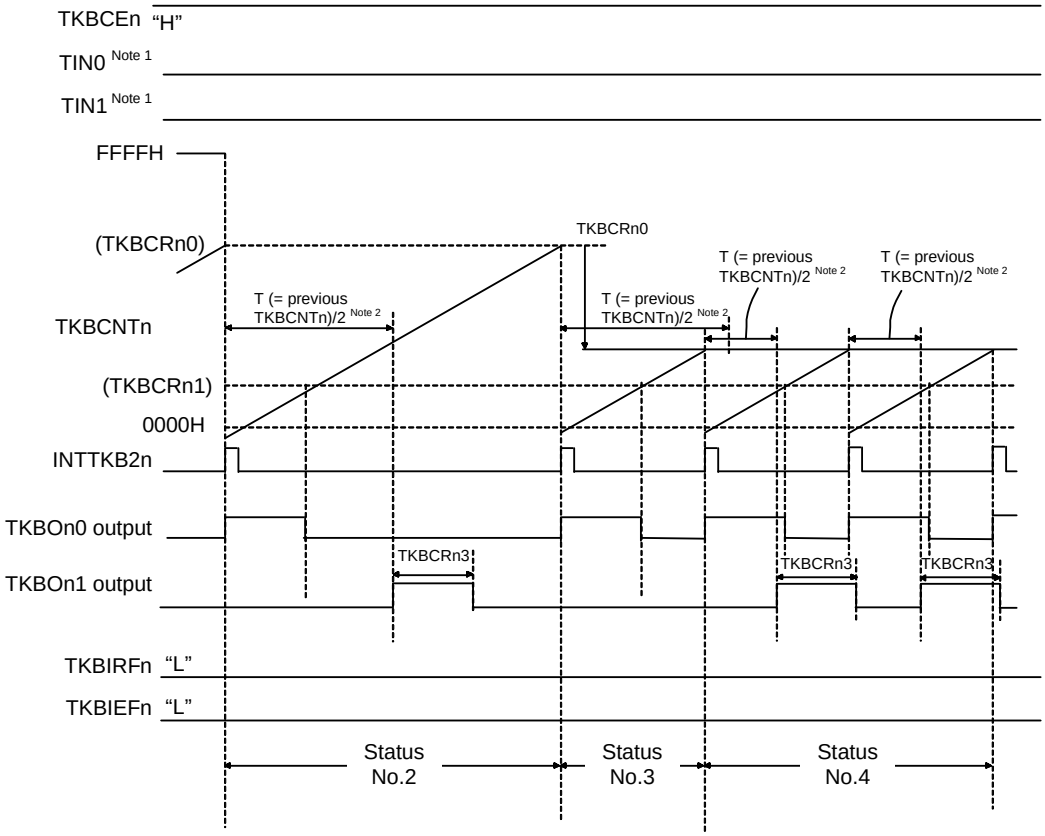
Note 2. T is calculated by the following formula.
When the setting value of TKBCRn0 is even, T = TKBCRn0 + 2
When the setting value of TKBCRn0 is odd, T = TKBCRn0 + 1

Status No.1: Only for the first period after TKBCEn = 1 setting, TKBOn1 with setting width of TKBCRn3 is output setting "T" as TKBCRn0.

Status No.2: In the second period, TKBOn1 with setting width of TKBCRn3 is output at T/2 of the previous period.

Remark n = 0 to 2

Figure 7 - 64 Figure of Timing of Interleave PFC Mode (Operation for Status No. 3 to 4)
(When default value of output is low level (TKBTODnp = 0) and active level is high level (TKBTOLnp = 0)



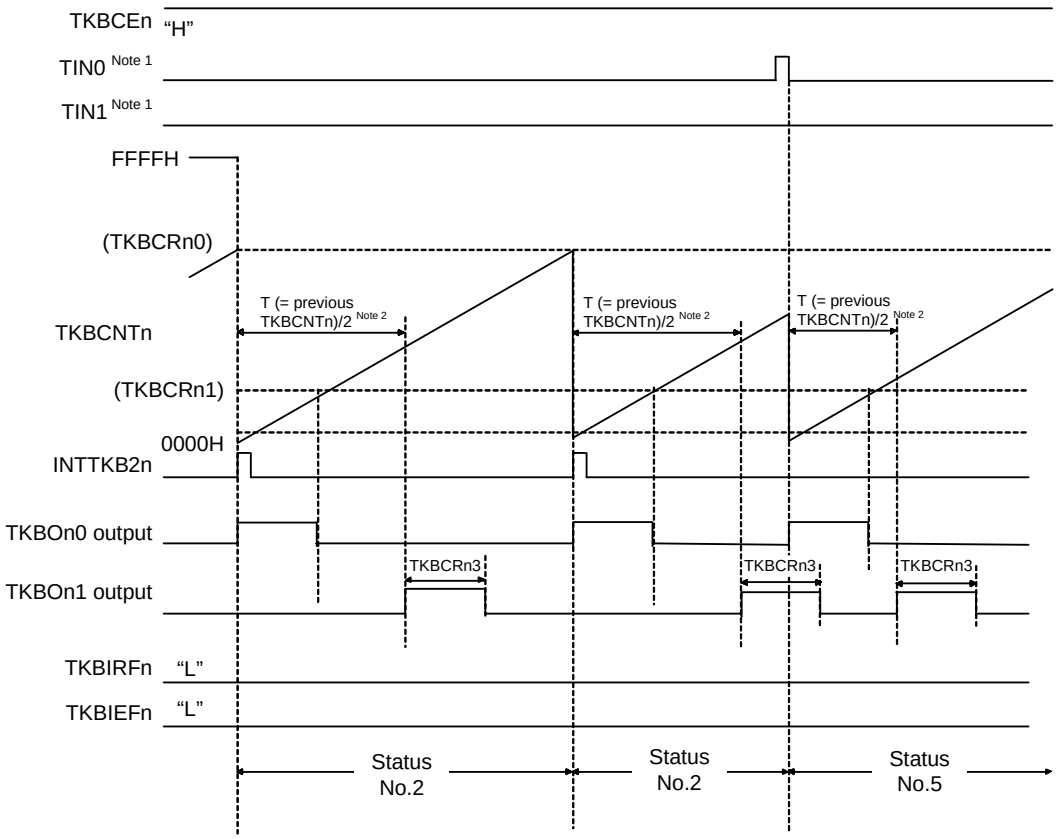
Note 1. TIN0 indicates an external interrupt assigned to counter restart trigger source 0. TIN1 indicates an external interrupt assigned to counter restart trigger source 1.

Note 2. T is calculated by the following formula.
When the setting value of TKBCRn0 is even, T = TKBCRn0 + 2
When the setting value of TKBCRn0 is odd, T = TKBCRn0 + 1

Remark n = 0 to 2; p = 0, 1

Status No.3: TKBOn1 maintains the status and T/2 of the previous period is not ensured.
Status No.4: TKBOn1 with setting width of TKBCRn3 is output at T/2 of the previous period.

Figure 7 - 65 Figure of Timing of Interleave PFC Mode
(Operation for Status No. 5: INT0 Input Detected (for the first time))



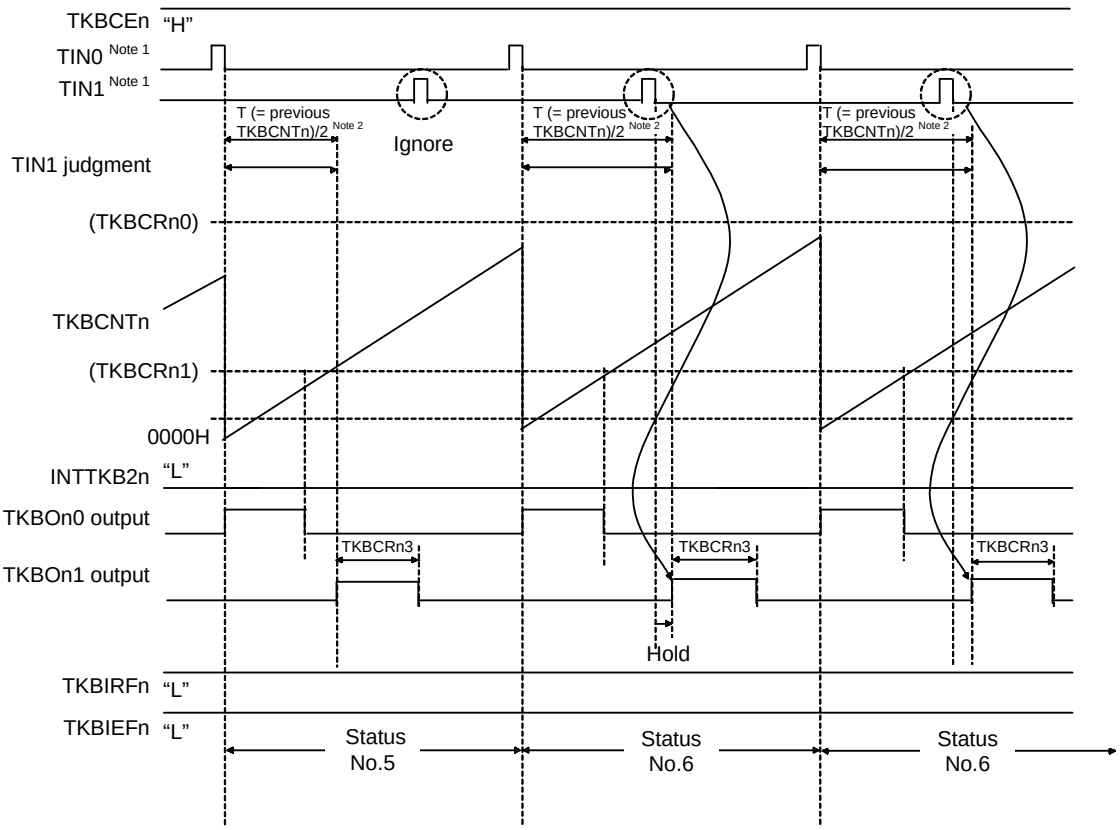
Note 1. TIN0 indicates an external interrupt assigned to counter restart trigger source 0. TIN1 indicates an external interrupt assigned to counter restart trigger source 1.

Note 2. T is calculated by the following formula.
When the setting value of TKBCRn0 is even, T = TKBCRn0 + 2
When the setting value of TKBCRn0 is odd, T = TKBCRn0 + 1

Status No.5: TIN0 which was first detected after setting TKBCEn = 1 outputs TKBOn1 with setting width of TKBCRn3 at T/2 of the previous period. It does not depend on whether TIN1 is detected or not detected.

Remark n = 0 to 2; p = 0, 1

Figure 7 - 66 Figure of Timing of Interleave PFC Mode (Operation for Status No. 6)
(When default value of output is low level (TKBTODnp = 0) and active level is high level (TKBTOLnp = 0))



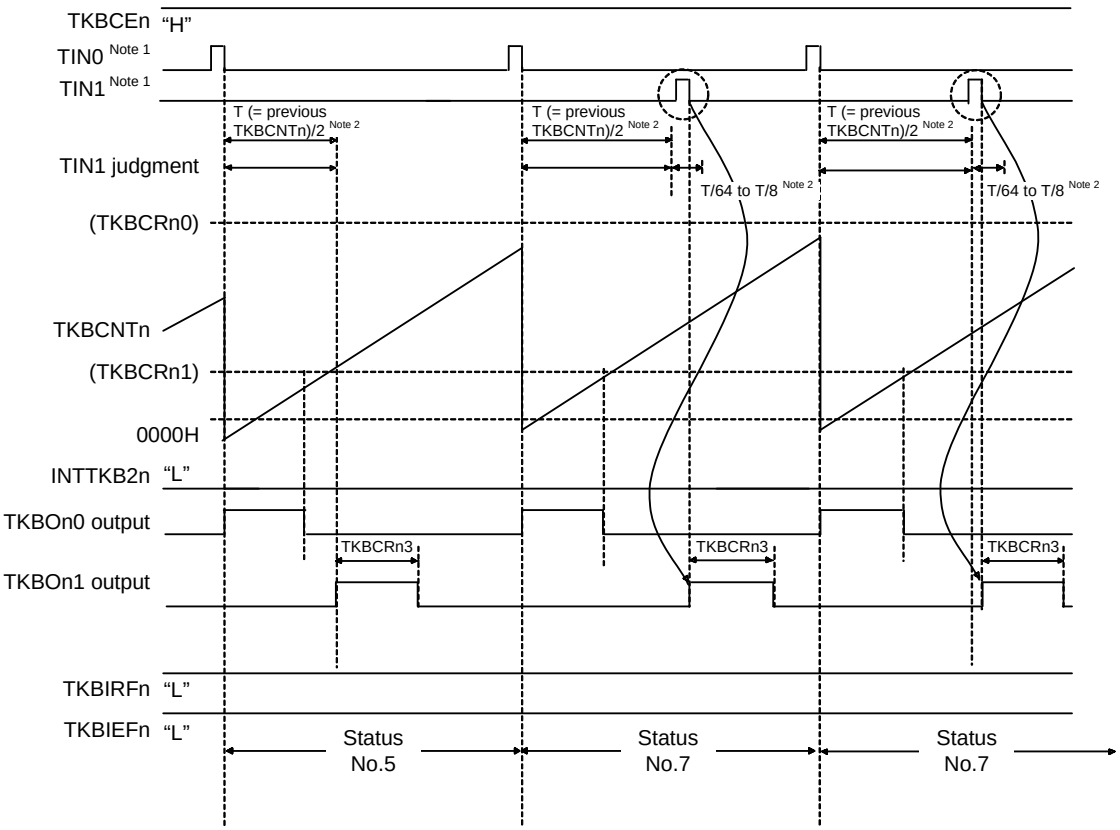
Note 1. TIN0 indicates an external interrupt assigned to counter restart trigger source 0. TIN1 indicates an external interrupt assigned to counter restart trigger source 1.

Note 2. T is calculated by the following formula.
When the setting value of TKBCRn0 is even, T = TKBCRn0 + 2
When the setting value of TKBCRn0 is odd, T = TKBCRn0 + 1

Remark n = 0 to 2; p = 0, 1

Status No.6: TKBOn1 with setting width of TKBCRn3 is output at T/2 of the previous period as TIN1 input is below T/2 of the previous period.

Figure 7 - 67 Figure of Timing of Interleave PFC Output Mode (Operation for Status No. 7)
(When default value of output is low level (TKBTODnp = 0) and active level is high level (TKBTOLnp = 0)



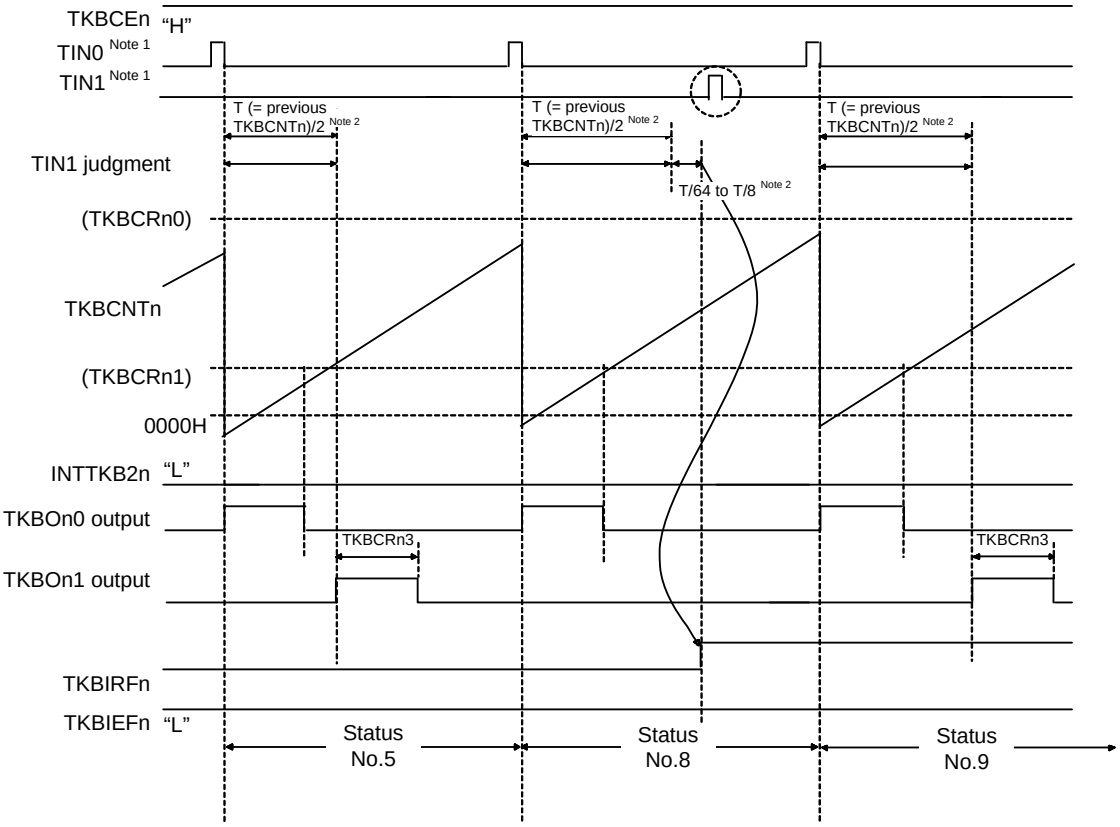
Note 1. TIN0 indicates an external interrupt assigned to counter restart trigger source 0. TIN1 indicates an external interrupt assigned to counter restart trigger source 1.

Note 2. T is calculated by the following formula.
When the setting value of TKBCRn0 is even, T = TKBCRn0 + 2
When the setting value of TKBCRn0 is odd, T = TKBCRn0 + 1

Remark n = 0 to 2; p = 0, 1

Status No.7: After the detection of TIN0 when TIN1 is detected over T/2 of the previous period and within T/2 + T/m (m = 8, 16, 32, 64: set by TKBIRSn1 and TKBIRSn0), TKBOn1 with setting width of TKBCRn3 is output at the detection of TIN1.

Figure 7 - 68 Figure of Timing of Interleave PFC Output Mode (Operation for Status No. 8 to 9)
(When default value of output is low level (TKBTODnp = 0) and active level is high level (TKBTOLnp = 0)



Note 1. TIN0 indicates an external interrupt assigned to counter restart trigger source 0. TIN1 indicates an external interrupt assigned to counter restart trigger source 1.

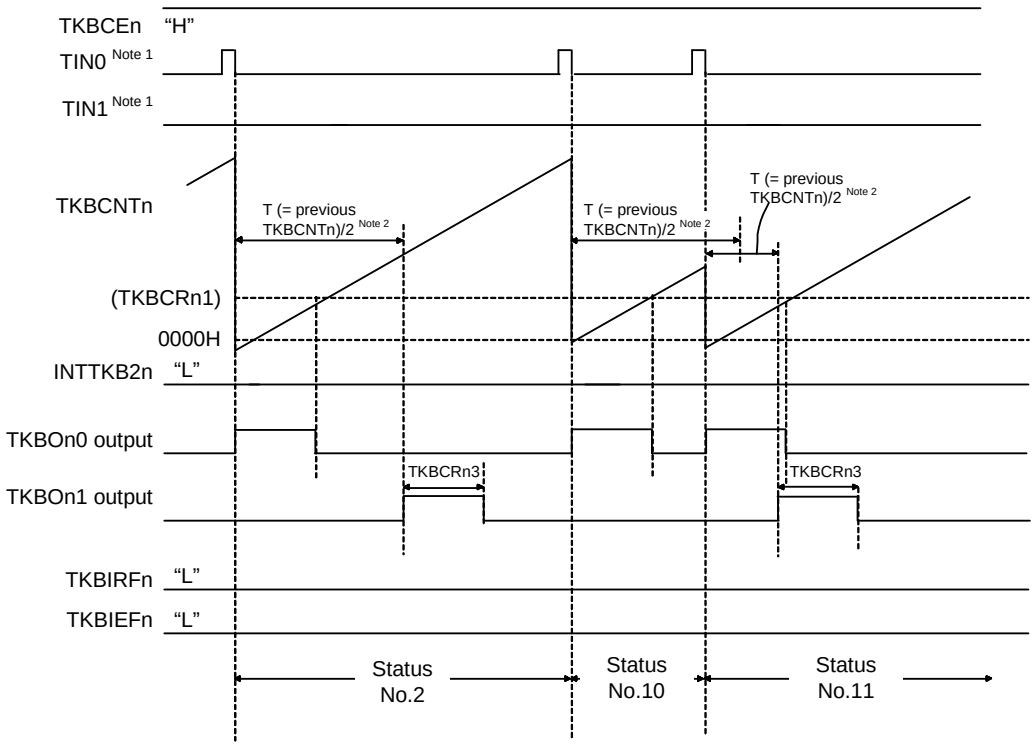
Note 2. T is calculated by the following formula.
When the setting value of TKBCRn0 is even, T = TKBCRn0 + 2
When the setting value of TKBCRn0 is odd, T = TKBCRn0 + 1

Remark n = 0 to 2; p = 0, 1

Status No.8: If TIN1 is not detected within T/2 + T/m (m = 8, 16, 32, 64: set by TKBIRSn1 and TKBIRSn0) of the previous period, TKBOn1 maintains the status. TKBIRFn is set to 1 at this time.

Status No.9: TKBOn1 with setting width of TKBCRn3 is output at T/2 of the previous period.

Figure 7 - 69 Figure of Timing of Interleave PFC Output Mode (Operation for Status No. 10 to 11)
(When default value of output is low level (TKBTODnp = 0) and active level is high level (TKBTOLnp = 0)



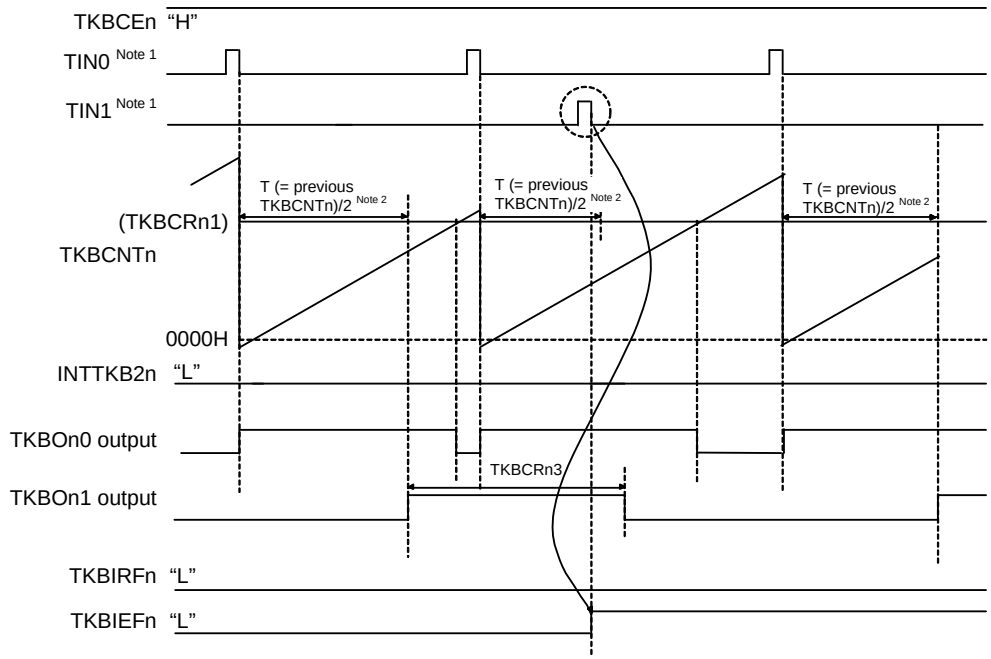
Note 1. TIN0 indicates an external interrupt assigned to counter restart trigger source 0. TIN1 indicates an external interrupt assigned to counter restart trigger source 1.

Note 2. T is calculated by the following formula.
When the setting value of TKBCRn0 is even, T = TKBCRn0 + 2
When the setting value of TKBCRn0 is odd, T = TKBCRn0 + 1

Remark n = 0 to 2; p = 0, 1

Status No.10: TKBOn1 maintains the status and T/2 of the previous period is not ensured.
Status No.11: TKBOn1 with setting width of TKBCRn3 is output at T/2 of the previous period.

Figure 7 - 70 Figure of Timing of Interleave PFC Output Mode (When trigger is generated again during TKBOn1)



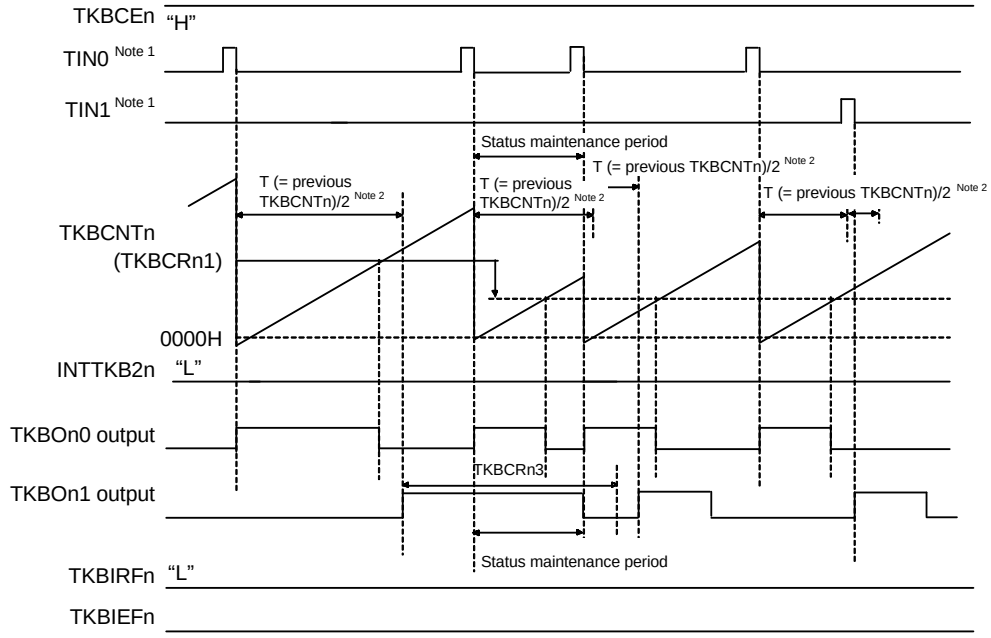
Note 1. TIN0 indicates an external interrupt assigned to counter restart trigger source 0. TIN1 indicates an external interrupt assigned to counter restart trigger source 1.

Note 2. T is calculated by the following formula.
When the setting value of TKBCRn0 is even, T = TKBCRn0 + 2
When the setting value of TKBCRn0 is odd, T = TKBCRn0 + 1

The trigger is ignored when the subsequent TKBOn1 output trigger is generated while outputting TKBOn1 of the previous period. TKBIEFn is set to 1 at this time.

Remark n = 0 to 2

Figure 7 - 71 Figure of Timing of Interleave PFC Output Mode
(TKBOn1 output is at the width of the previous output width and exceeds status maintenance period)



Note 1. TIN0 indicates an external interrupt assigned to counter restart trigger source 0. TIN1 indicates an external interrupt assigned to counter restart trigger source 1.

Note 2. T is calculated by the following formula.
When the setting value of TKBCRn0 is even, T = TKBCRn0 + 2
When the setting value of TKBCRn0 is odd, T = TKBCRn0 + 1

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.

Remark n = 0 to 2

(2) List of Register Setting at Interleave PFC Output Mode

	15	14	13	12	11	10	9	8
TKBCTLn0	— 0	— 0	TKBSSEn1 0	TKBDIEn1 0	— 0	— 0	TKBSSEn0 0	TKBDIEn0 0
	7	6	5	4	3	2	1	0
	TKBMFE0 1/0	— 0	TKBIRSn1 1/0	TKBIRSn0 1/0	— 0	TKBTSEn 1	TKBSTSn1 0	TKBSTSn0 0
	7	6	5	4	3	2	1	0
TKBCTLn1	TKBCEn 1	— 0	— 0	TKBCKSn0 1/0	TKBSCMn 0	— 0	TKBMDn1 1	TKBMDn0 1
	7	6	5	4	3	2	1	0
TKBIOCn0	— 0	— 0	— 0	— 0	TKBTOLn1 1/0	TKBTOLn0 1/0	TKBTODn1 1/0	TKBTODn0 1/0
	7	6	5	4	3	2	1	0
TKBIOCn1	— 0	— 0	— 0	— 0	— 0	— 0	TKBTOEn1 1/0	TKBTOEn0 1/0
	7	6	5	4	3	2	1	0
TKBPSCSn	— 0	TKBTPS012 1/0	TKBTPS011 1/0	TKBTPS010 1/0	— 0	TKBTPS002 1/0	TKBTPS001 1/0	TKBTPS000 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)

Remark n = 0 to 2

7.5 Option Functions of 16-bit Timers KB20, KB2, and KB22

Option functions can be added to timers KB20, KB2, and KB22.

The following table shows available options for each operation mode for timers KB20, KB2, and KB22.

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 CR00	Period Controlled by Trigger	Period Controlled by CR00	Period Controlled by Trigger	Period Controlled by Master	Period Controlled by Restart Trigger Source 0/CR00
Optional Functions	Trigger output function	√	√	√	√	√	√
	PWM output dithering function	√	—	√	—	√	—
	PWM output smooth start 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/Restart Operation**.

7.5.1 Trigger Output Function

Timer KB2n trigger output signal can be generated by setting 16-bit timer KB2 trigger compare register n (TKBTGCRn). This trigger output signal can be used as an ELC event input signal (corresponding to ELSELR20 to ELSELR22).

Timer KB2n 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 KB2n 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 KB2n trigger output signal is not output when TKBCRn0 < TKBTGCRn.

Figure 7 - 72 Trigger Output Function for Standalone Mode (Period Controlled by TKB0CR0)

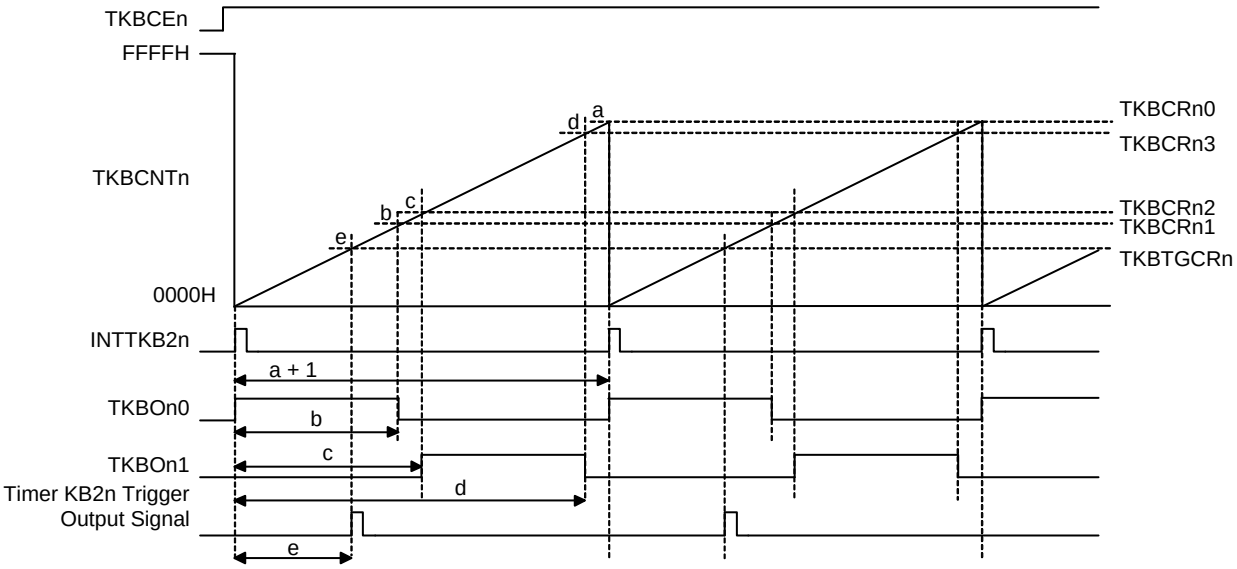
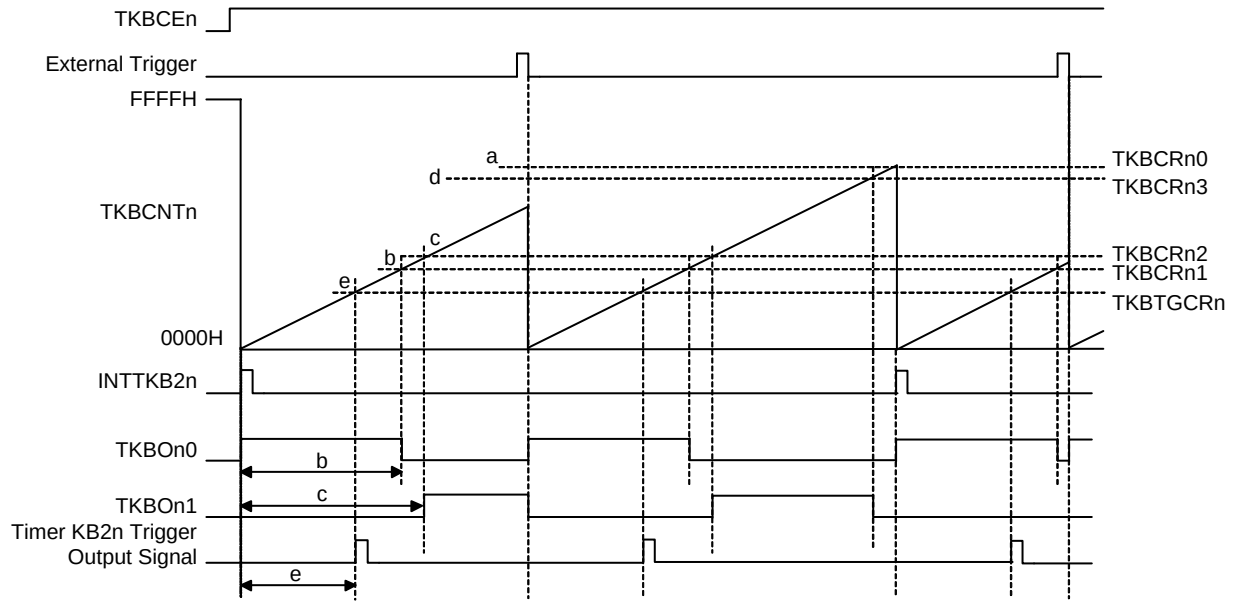


Figure 7 - 73 Trigger Output Function for Standalone Mode (Period Controlled by External Trigger Input)



Remark n = 0 to 2

7.5.2 PWM Output Dithering Function

16-bit timer KB2n is available for high resolution PWM using PWM output dithering function.

Taking 16 periods of the PWM period as one unit, 16 times higher PWM is available for average resolution by extending the active period of N times (N = 0 to 15) by one count clock during one unit.

The number of repetitions (N) extending the active period by one count clock during one unit is set to the TKBDNRnp register. The following figure shows the ordinal of the period (kth period) extending the active period (N times) by one count clock during one unit.

For example, when N = 3, the PWM active period is extended by one clock at the first, fifth, and ninth periods during one unit.

Figure 7 - 74 16-bit Timer KB2 Dithering Count Register 0p (TKBDNRnp) Setting

kth period / Repetitions (N)	k															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0																
1	■															
2	■								■							
3	■				■				■							
4	■				■				■				■			
5	■		■		■				■				■			
6	■		■		■				■		■		■			
7	■		■		■		■		■		■		■			
8	■		■		■		■		■		■		■		■	
9	■	■	■		■		■		■		■		■		■	
10	■	■	■		■		■		■	■	■		■		■	
11	■	■	■		■	■	■		■	■	■		■		■	
12	■	■	■		■	■	■		■	■	■		■	■	■	
13	■	■	■	■	■	■	■		■	■	■		■	■	■	
14	■	■	■	■	■	■	■		■	■	■	■	■	■	■	
15	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	

Remark 1. ☐ cell: Set to active period according to settings in TKBCRn1 and TKBCRn3 registers

■ cell: Set to active period according to “settings + 1” in TKB0CR1 and TKBCRn3 registers

Remark 2. n = 0 to 2; p = 0, 1

Figure 7 - 75 Figure of Waveform at Dithering Operation

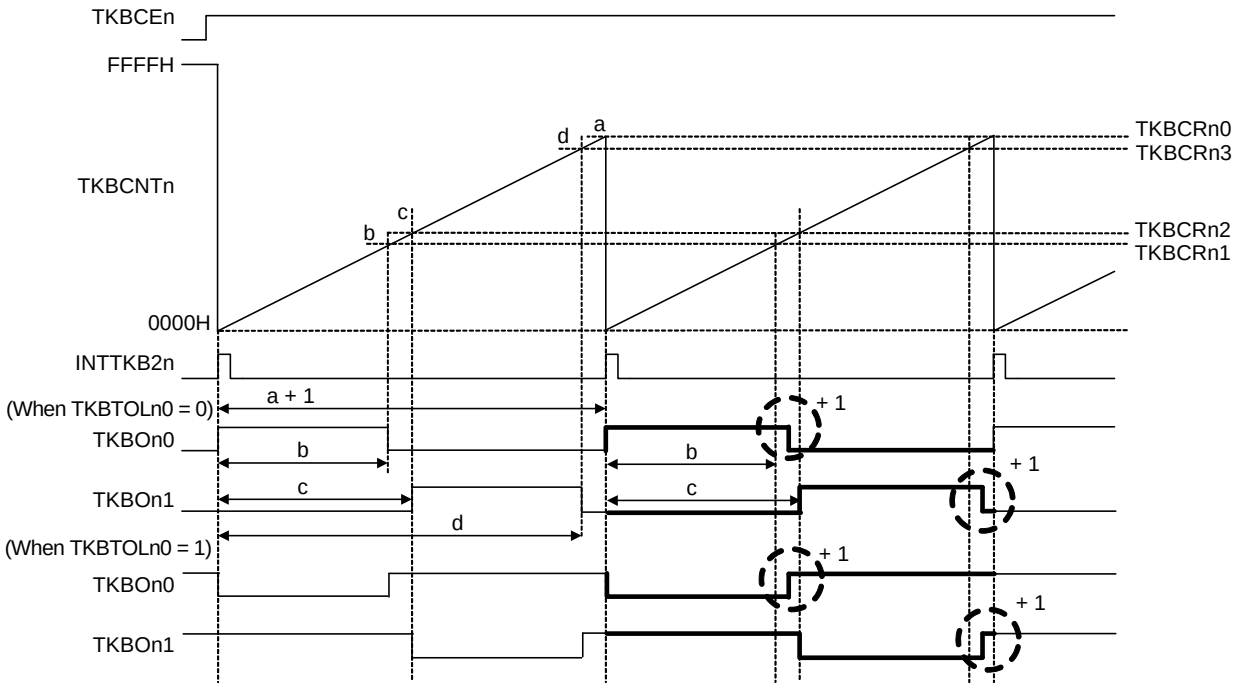
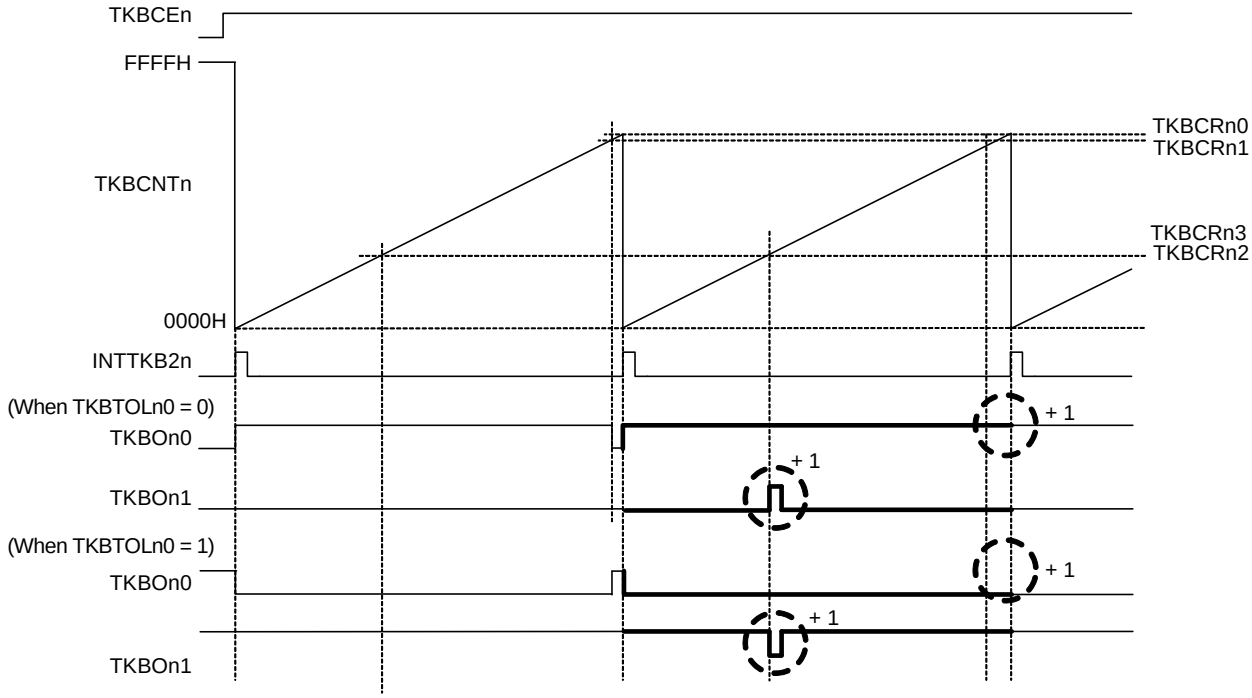
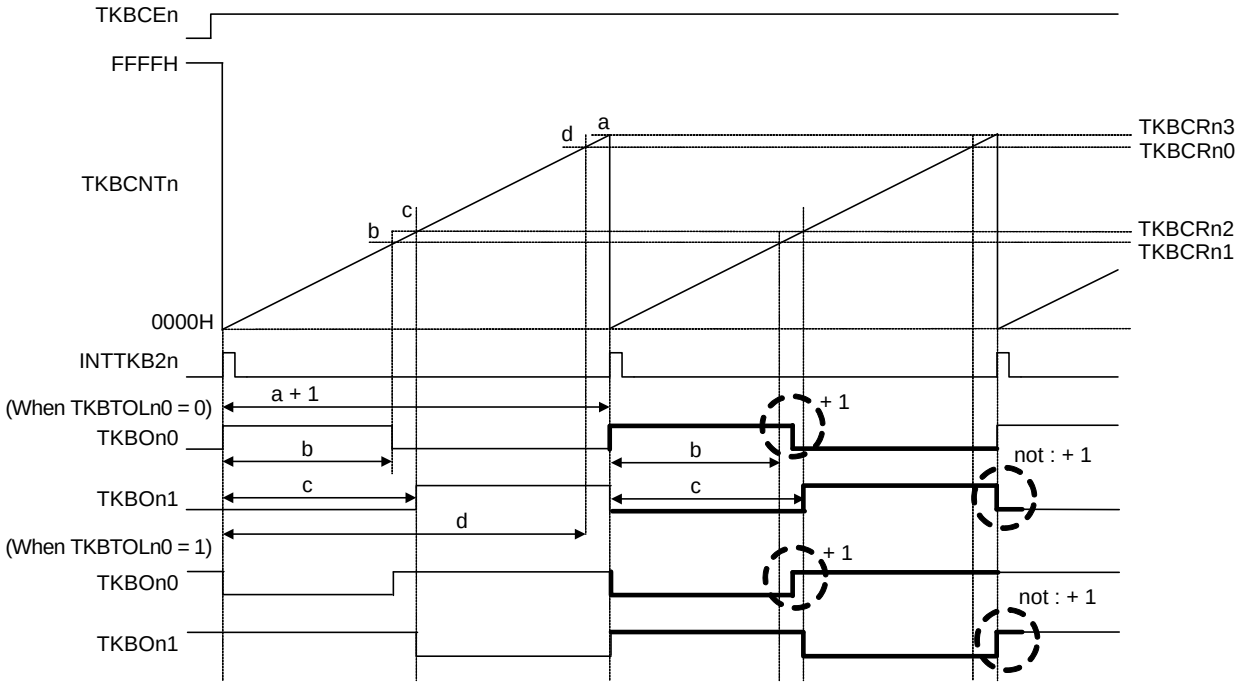


Figure 7 - 76 Figure of Waveform at Dithering Operation
(When TKBCRn1 = TKBCRn0 (100% nearest neighbor), TKBCRn2 = TKBCRn3 (0% nearest neighbor)



Remark n = 0 to 2

Figure 7 - 77 Figure of Waveform at Dithering Operation (When TKBCRn3 = TKBCRn0 + 1)



Remark n = 0 to 2

(1) Available Operation Mode

The following shows enable or disable status under each mode that is specified by the TKBCTLn0 register (TKBSTSn1 and TKBSTSn0 bits) and the TKBCTLn1 register (TKBMDn1 and TKBMDn0 bits).

Operation Mode	TKBMDn1, TKBMDn0	TKBSTSn1, TKBSTSn0	Setting
Standalone Mode (Period controlled by TKBCRn0)	00B	00B	Available
Standalone Mode (Period controlled by external trigger input)	00B	01B/10B/11B	Not available
Simultaneous Start / Stop Mode (Period controlled by TKBCRn0)	01B	00B	Available
Simultaneous Start / Stop Mode (Period controlled by external trigger input)	01B	01B/10B/11B	Not available
Synchronous Start / Clear Mode (Period controlled by Master)	10B	—	Available
Interleave PFC Output Mode	11B	—	Not available

PWM output dithering function is available when external trigger input is not used and the period being controlled by TKBCRn0.

TKBDNRn0/TKBDNRn1 controls PWM output dithering function of respective TKBOn0/TKBOn1.

Caution 1. [Overwrite during Operation (TKBCEn = 1) of TKBDNRn0/TKBDNRn1 Register]
Since TKBDNRn0/TKBDNRn1 owns the buffer, overwrite during the operation (TKBCEn = 1) is available.
At this time, batch overwriting is available via writing 1 to the TKBRDTn bit.

Caution 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.
The value of TKBDNRn0/TKBDNRn1 is changed even when the TKBCRLDn0/TKBCRLDn1 register is accessed.
The value of TKBCRn1/TKBCRn3 is changed even when the TKBCRLDn0/TKBCRLDn1 register is accessed.
Only the lower 8 bits of TKBCRn1/TKBCRn3 are changed when the TKBCRLDn0/TKBCRLDn1 register is accessed.

Caution 3. [To Combine PWM Output Smooth Start Function with PWM Output Dithering Function]
PWM output dithering function is invalid during the execution of PWM output smooth start function (TKBSSFnp = 1).
PWM output dithering function is valid when PWM output smooth start function is stopped (TKBSSFnp = 0).

Remark n = 0 to 2; p = 0, 1

7.5.3 PWM Output Smooth Start Function

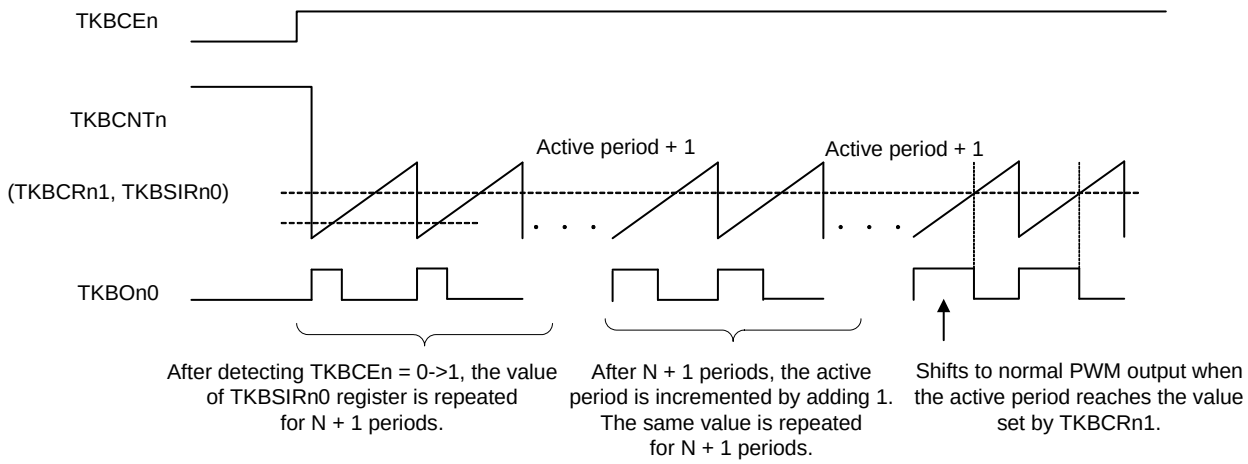
Timer KB2n has PMW output smooth start function corresponding to rush current control and over-voltage prevention. PWM output smooth start function begins at timer start timing. The process that a user has performed with software in the past can be easily accomplished with the optional function of the hardware.

Operation starts with the value set by the 16-bit timer KB2 smooth start default duty register (TKBSIRnp) and the PWM active period is sequentially incremented by one clock. The rate to increment the active period is specified by the 16-bit timer KB2 smooth start step width register (TKBSSRnp). When the value set by the TKBSSRnp register is N, after the currently set active period is output for N + 1 times, the active period is incremented by adding 1 and then the new active period is output for N + 1 times. After repeating this operation to increment the active period, PWM output smooth start function is cancelled when the same active period specified by the TKBCRn1 and TKBCRn3 registers is reached.

The 16-bit timer KB2 smooth start default duty register should be set according to the following condition;
0000H ≤ TKBSIRn0 < TKBCRn1 ≤ TKBCRn0 + 1
TKBCRn2 ≤ TKBSIRn1 < TKBCRn3 ≤ TKBCRn0 + 1

When using simultaneous start/clear mode, the setting should be made according to the following condition;
TKBCRm0 ≤ TKBSIRn0 < TKBCRn1 ≤ TKBCR00 of master + 1

Figure 7 - 78 Example of TKBOn0 Output Using PWM Output Smooth Start Function



Remark 1. N: Value set by TKBSSRnp register
Remark 2. n = 0 to 2; p = 0, 1; m = 1, 2

(1) Operation Mode Available for PMW Output Smooth Start Function

Operation Mode	TKBMDn1, TKBMDn0	TKBSTSn1, TKBSTSn0	Setting
Standalone Mode (Period controlled by TKBCRn0)	00B	00B	Available
Standalone Mode (Period controlled by external trigger input)	00B	01B/10B/11B	Not Available
Simultaneous Start / Stop Mode (Period controlled by TKBCRn0)	01B	00B	Available
Simultaneous Start / Stop Mode (Period controlled by external trigger input)	01B	01B/10B/11B	Not available
Synchronous Start / Clear Mode (Period controlled by Master)	10B	—	Available
Interleave PFC Output Mode	11B	—	Not Available

(2) Overwrite during Operation (TKBCEn = 1) of TKBSIRn0/TKBSIRn1/TKBSSRn0/TKBSSRn1 Registers

Overwrite during normal operation (TKBCEn = 1) is available for TKBSIRn0/TKBSIRn1/TKBSSRn0/TKBSSRn1, but batch overwrite is not available during the period of PWM output software start.

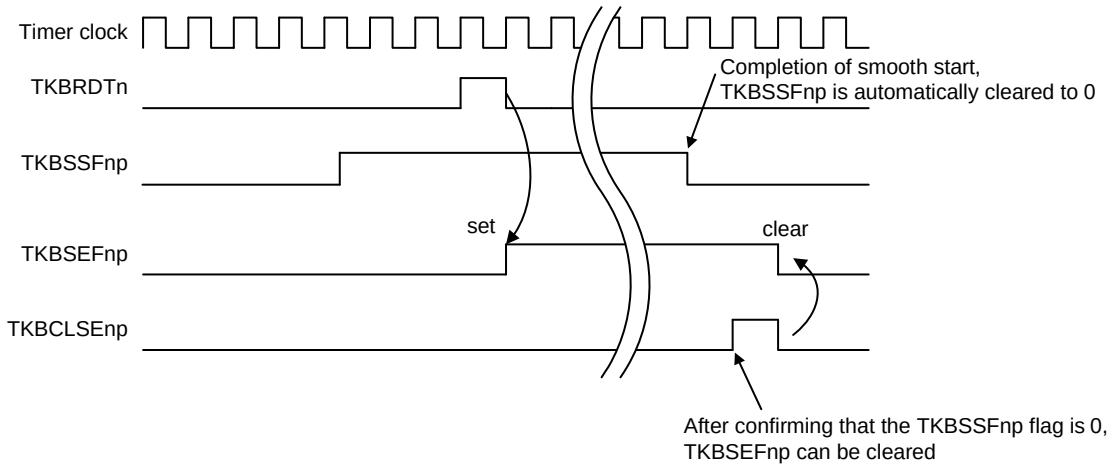
TKBSIRn0/TKBSIRn1/TKBSSRn0/TKBSSRn1 own the buffer and batch overwriting is available via writing 1 to the TKBRDTn bit. In TKBSIRn0/TKBSIRn1, the value of the buffer at the timing when the PWM output smooth start function starts is the initial value of the duty. TKBSSRn0/TKBSSRn1 are set to the comparison value of the internal 4-bit counter.

The internal 4-bit counter counts up the TKBCNTn cycle as the count clock, becomes 0H when it matches TKBSSRn0/TKBSSRn1, and continues count operation.

(3) Overwrite during Operation (TKBCEn = 1) of TKBCRn0/TKBCRn1/TKBCRn2/TKBCRn3/TKBSIRn0/TKBSIRn1/TKBSSRn0/TKBSSRn1 Registers

During the period of PWM output smooth start (TKBSSFn0 = 1, TKBSSFn1 = 1), batch overwrite is masked. When TKBRDTn is set to 1, batch overwrite is masked and the TKBSEFnp flag is set. At this time the TKBSEFnp flag cannot be cleared. After confirming that TKBSSFnp = 0 (completion of PWM output smooth start), the TKBSEFnp flag is cleared by setting the TKBSEFnp bit = 1.

Figure 7 - 79 Flag Clearing Timing When Overwrite is Not Available during Smooth Start Function Operation (TKBSSFnp = 1) of TKBCRn0/TKBCRn1/TKBCRn2/TKBCRn3/TKBSIRn0/TKBSIRn1/TKBSSRn0/TKBSSRn1 Registers



Remark n = 0 to 2; p = 0, 1

(4) To Combine PWM Output Smooth Start Function with PWM Output Dithering Function

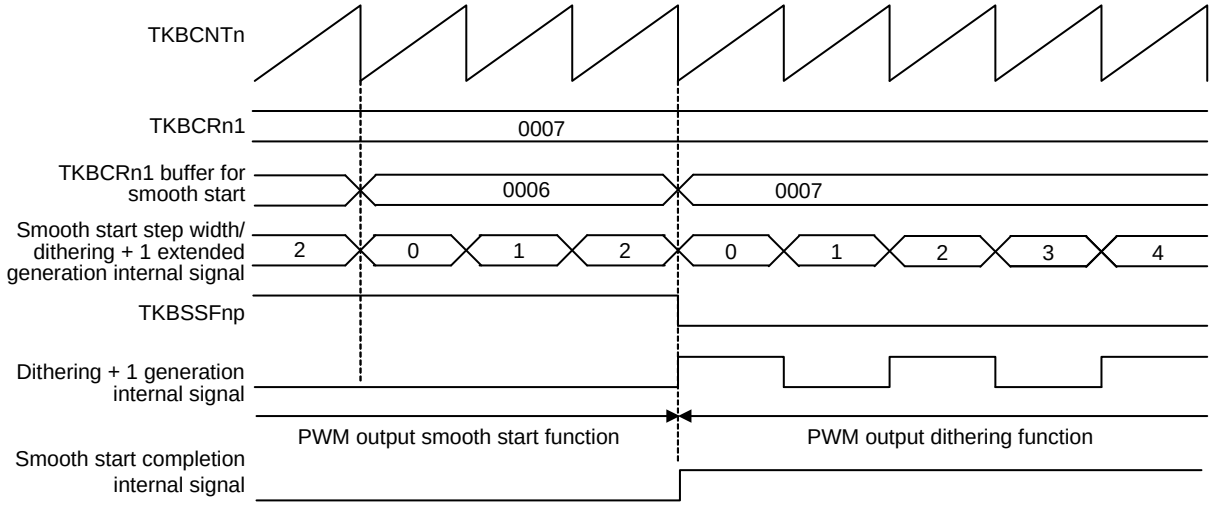
PWM output dithering function is invalid during the execution of PWM output smooth start function (TKBSSFnp = 1).

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

The following figure 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 smooth start matches, TKBSSFnp is cleared, and then dithering function begins.

Figure 7 - 80 Completion of PWM Output Smooth Start Function and Operation of TKBSSFnp



Remark n = 0 to 2; p = 0, 1

7.5.4 Maximum Frequency Limit Function

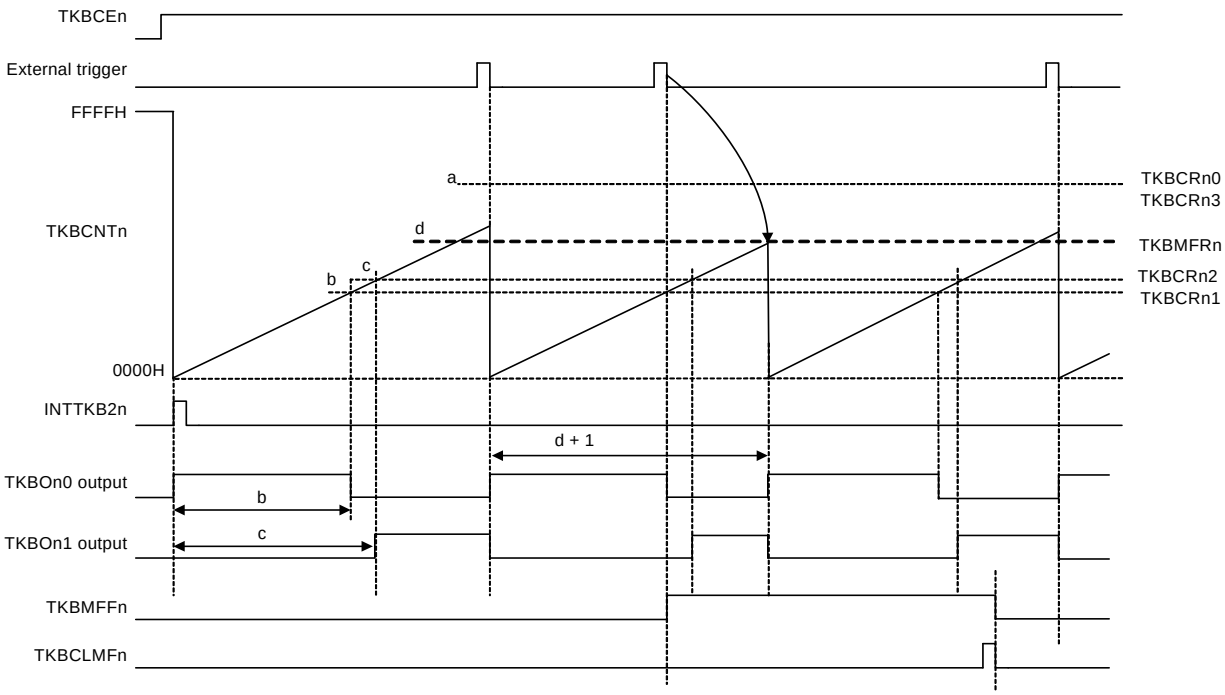
Timer KB2n has a function that regulates the minimum period of the counter clear (maximum frequency) in the periodic control by external trigger or interleave PFC output mode.
When this function is used, if external trigger input which performs the counter clear occurs while the counter value is less than the setting value of the maximum frequency limit register (TKBMFRn), it performs the counter clear after it continues counting until it reaches the setting value of the TKBMFRn register.

(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 must be satisfied: TKBMFRn setting ≤ TKBCRn0 setting

When the counter value is smaller than TKBMFRn at the timing for external trigger input detection, the TKBMFFn flag is set to 1. The TKBMFFn flag is cleared to 0 by writing 1 to the TKBCLMFn bit.

Figure 7 - 81 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
Standalone Mode (Period controlled by TKBCRn0)	00B	00B	Not available
Standalone Mode (Period controlled by external trigger input)	00B	01B/10B/11B	Available
Simultaneous Start / Stop Mode (Period controlled by TKBCRn0)	01B	00B	Not available
Simultaneous Start / Stop Mode (Period controlled by external trigger input)	01B	01/10/11B	Available
Synchronous Start / Clear Mode (Period controlled by Master)	10B	—	Not available
Interleave PFC Output Mode	11B	—	Available

Remark 1. Available when the period is controlled by external trigger input.

Remark 2. n = 0 to 2

7.6 Forced Output Stop Function

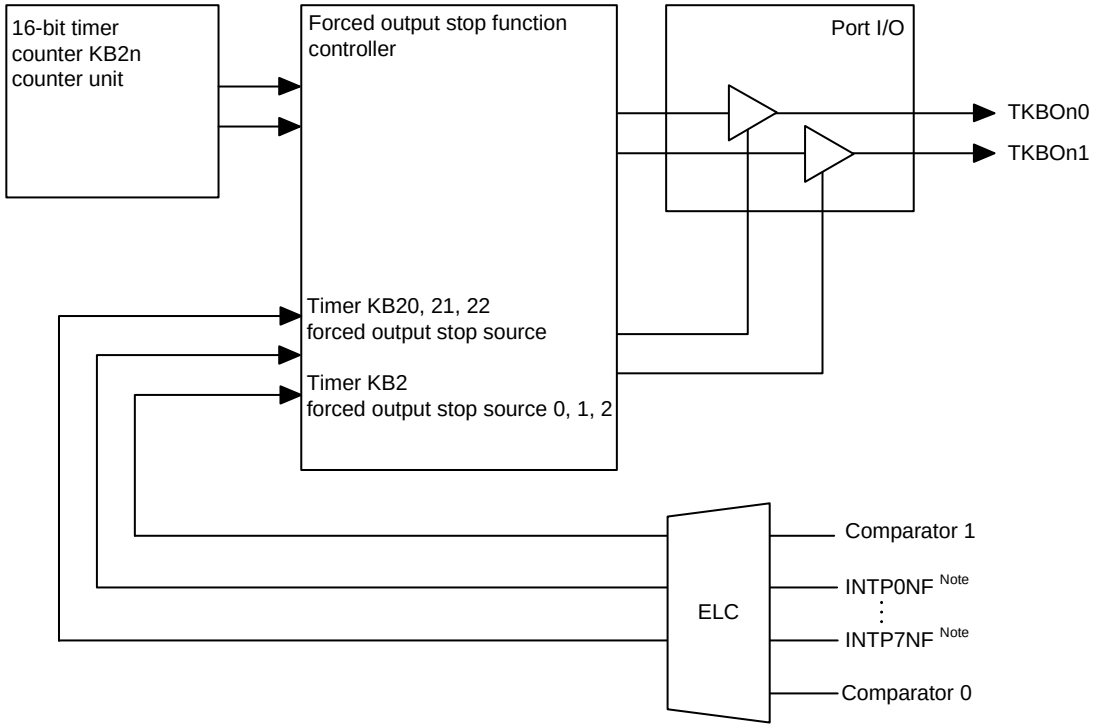
Forced output stop function is a function to protect power supply, etc.

If any abnormal situation that occurs in a power circuit configured outside of a microcomputer leads to over-voltage of over-current, making voltage or current sense signal into INTPiNF/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 an abnormality.

The following figure shows the system structure of forced output stop function.

Figure 7 - 82 System Structure of Forced Output Stop Function



Note INTP0NF to INTP7NF are signals that are not passed through the noise filter (signals corresponding to ELSELR23 to ELSELR30).

Remark n = 0 to 2

7.6.1 Forced Output Stop Function 1 and 2

There are two control methods for the forced output stop function. Forced output stop function 1 can be set to fixed level output or high-impedance output, and forced output stop function 2 can set to fixed level output only. The differences of the control methods are shown below.

(1) Selectable Output Levels for Forced Output Stop Function 1 and 2

Selectable Output Levels	Forced Output Stop Function	
	Function 1	Function 2
High-impedance output	√	—
Low-level fixed output	√	√
High-level fixed output	√	√

(2) Start/Cancel Conditions for Forced Output Stop Function 1 and 2

Function/Operation Details (Start of Forced Output Stop)	Forced Output Stop Function	
	Function 1	Function 2
TMKB2n forced output stop source, TMKB2 forced output stop source 0, 1 forced output stop is started by detection (via the ELC)	√	√
TMKB2n forced output stop source 2 forced output stop is started by detection (via the ELC)	—	√
Forced output stop is stopped by software bit (TKBPAHTSnp) setting.	√	—

Function/Operation Details (Cancel of Forced Output Stop)	Forced Output Stop Function	
	Function 1	Function 2
Forced output stop cancelled by software bit (TKBPAHTTnp) setting.	√	—
Forced output stop cancelled in synchronization with TMKB period after the software bit (TKBPAHTTnp) setting.	√	—
Forced output stop cancelled at the next TMKB period after the start of forced output stop.	—	√
Forced output stop cancelled in synchronization with TMKB period after the detection of the start edge and reverse edge.	—	√

(3) Conditions of Selectable Input Pins and Available Trigger Bits for Forced Output Stop Function 1 and 2

Selectable Input Pins	Forced Output Stop Function	
	Function 1	Function 2
External interrupt (INTPiNF) (via the ELC, corresponding to ELSELR23 to ELSELR30)	√	√
Converter 0/1 (via the ELC, corresponding to ELSELR18 to ELSELR19)	√	√

Available Trigger Bits for Start/Cancel of Forced Output Stop	Forced Output Stop Function	
	Function 1	Function 2
TKBPAHTSnp	√	—
TKBPAHTTnp	√	—

Remark n = 0 to 2

7.7 Operation of Forced Output Stop Function 1

7.7.1 I/O Setting for Forced Output Stop Function 1

In forced output stop function 1, TMKB2n forced output stop source or TMKB2n forced output stop source 0, 1 is used as the trigger signal of forced output stop.
Selectable output status for forced output stop is high impedance or high/low-level fixed. The tables below show trigger signal selection and output status settings.

(1) TKBOnp Output Control

• Input selection

TKBPACTLnp	Input Selection
TKBPAHZSnp2	External interrupt detection (INTPiNF) or comparator 0/1 Note
TKBPAHZSnp1	External interrupt detection (INTPiNF) or comparator 0/1 Note
TKBPAHZSnp0	External interrupt detection (INTPiNF) or comparator 0/1 Note

• Output selection

TKBPACTLnp		Output Status
TKBPAMDnp1	TKBPAMDnp0	
0	0	High-impedance output
0	1	High-impedance output
1	0	Low-level fixed output
1	1	High-level fixed output

• Start of forced output stop function 1

TKBPACTLn0		Start Condition Selection for Forced Output Stop Function 1
TKBPAHCMnp1	TKBPAHCMnp0	
0	0	The forced output stop function is started when input of the forced output stop function trigger is detected or when 1 is written to the TKBPAHTnp bit.
0	1	
1	0	
1	1	

• Cancel of forced output stop function 1

TKBPACTLn0		Cancel Condition Selection for Output of Forced Output Stop Function 1
TKBPAHCMnp1	TKBPAHCMnp0	
0	0	Forced output stop function 1 is cancelled by setting the TKBPAHTTn0 bit to 1, regardless of the input level of forced output function stop input 1.
0	1	After the input of forced output stop function input 1 is cancelled, forced output stop function 1 is cancelled by setting the TKBPAHTTn0 bit to 1. Setting the TKBPAHTTn0 bit to 1 is invalid during the active period of the input.
1	0	After the TKBPAHTTn0 bit is set to 1, forced output stop function 1 is cancelled in synchronization with the next restart of the counter, regardless of the input level of forced output stop function input 1.
1	1	After the input of forced output stop function input 1 is cancelled, forced output stop function 1 is cancelled in synchronization with the next restart of the counter after the TKBPAHTTn0 bit is set to 1. Writing 1 to the TKBPAHTTn0 bit is invalid during the active period of the input.

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

Note For details on trigger source settings, refer to **CHAPTER 20 EVENT LINK CONTROLLER (ELC)**.

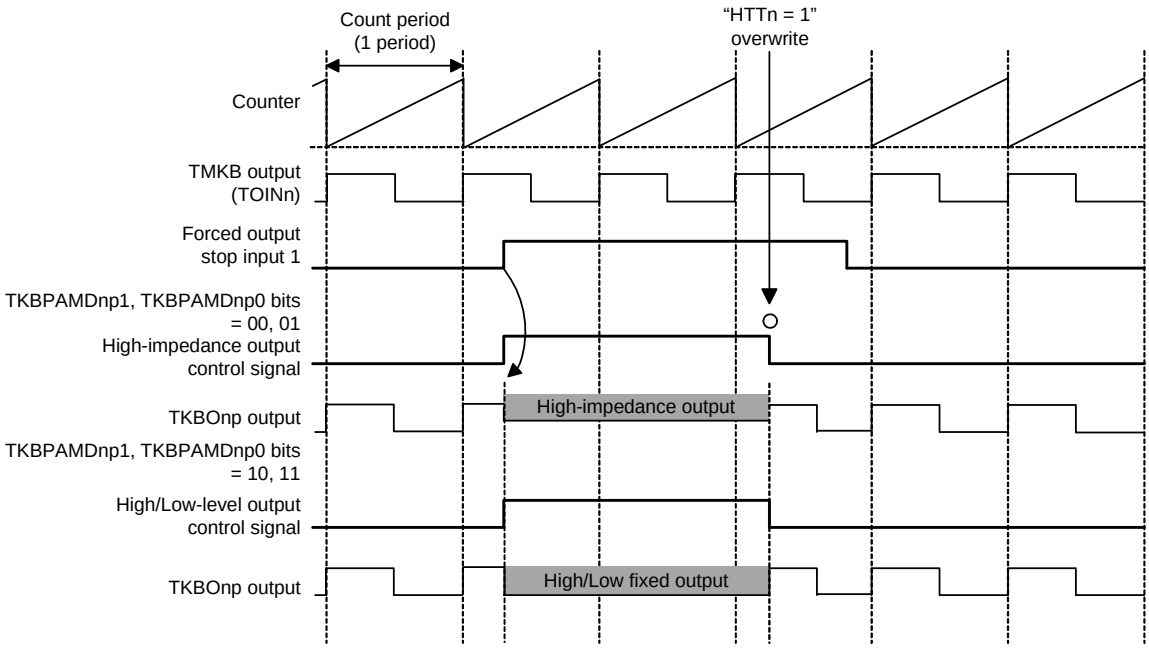
Caution 1. If comparator 0 or 1 detection is selected as a trigger input for forced output stop function 1 while the C0EDG or C1EDG bit in the comparator filter control register (COMPFIR) is 1 (both-edge detection), forced output stop function 1 cannot be cancelled by setting TKBPAHCMnp0 to 1. To cancel forced output stop function 1, be sure to set TKBPAHCMnp0 to 0.

Caution 2. Trigger input INTPiNF (event source corresponding to ELSELR23 to ELSELR30) which is used for forced output stop functions 1 and 2, are not affected by the setting of the external interrupt rising edge enable register (EGP0) and external interrupt falling edge enable register (EGN0). Only the rising edge is valid.

Remark i = 0 to 7; n = 0 to 2; P = 0, 1

7.7.2 Basic Operation of Forced Output Stop Function 1

(1) TKBPAHCMnp1, TKBPAHCMnp0 Bits = 00



- TKBPAMDnp1, TKBPAMDnp0 bits = 00, 01

  When the rising edge of forced output stop input 1 is detected, the high-impedance output control signal is set to high level and TKBOnp output becomes high impedance.

  Regardless of the level of forced output stop input 1, the high-impedance output control signal is set to low level by writing 1 to the TKBPAHTTnp bit in the TKBPAHFTn register, and TKBOnp returns to PWM output.

  The high-level period of the high-impedance output control signal is the period for forced output stop 1 (high-impedance output).

- TKBPAMDnp1, TKBPAMDnp0 bits = 10, 11

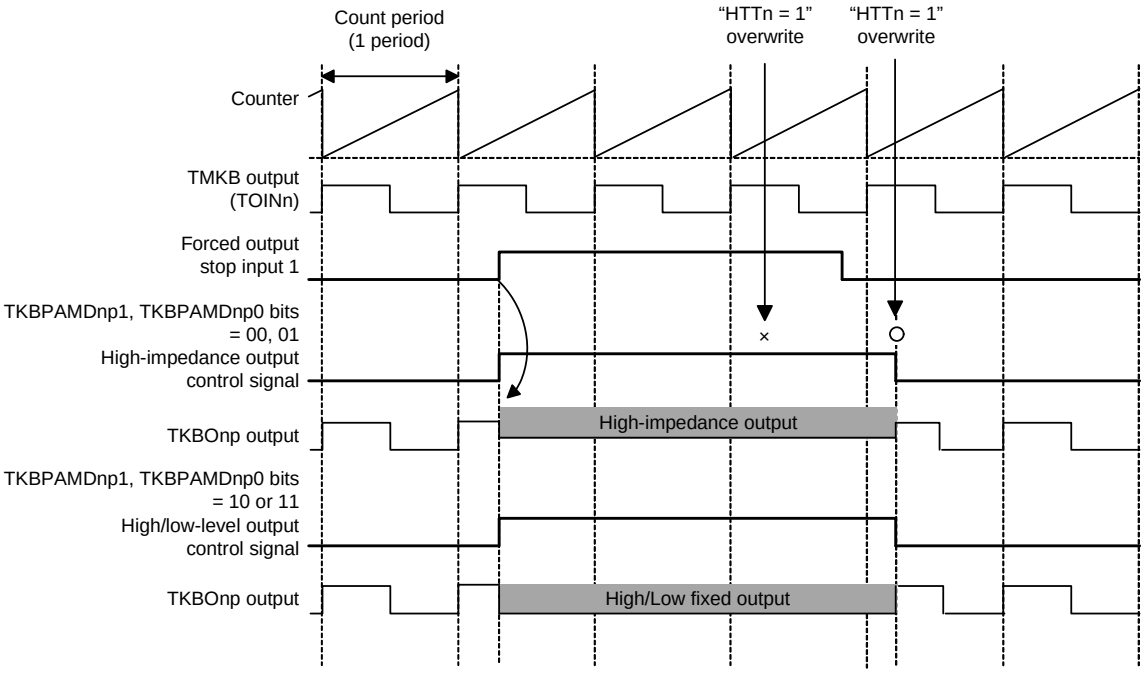
  When the rising edge of forced output stop input 1 is detected, the high/low-level output control signal is set to high level and TKBOnp output is fixed to high/low level according to the setting of the TKBPAMDnp0 bit.

  Regardless of the level of forced output stop input 1, the high/low-level output control signal is set to low level by writing 1 to the TKBPAHTTnp bit in the TKBPAHFTn register, and TKBOnp returns to PWM output.

  The high-level period of the high/low-level output control signal is the period for forced output stop 1 (high/low-level fixed output).

Remark n = 0 to 2; p = 0, 1

(2) TKBPAHCMnp1, TKBPAHCMnp0 Bits = 01



- TKBPAMDnp1, TKBPAMDnp0 bits = 00, 01

  When the rising edge of forced output stop input 1 is detected, the high-impedance output control signal is set to high level and TKBOnp output becomes high impedance.

  Writing 1 to the TKBPAHTTnp bit in the TKBPAHFTn register is invalid during the high-level period of the input of forced output stop input 1. After forced output stop input 1 changes to low level, the high-impedance output control signal is set to low level by writing 1 to the TKBPAHTTnp bit, and TKBOnp returns to PWM output.

  The high-level period of the high-impedance output control signal is the period for forced output stop 1 (high-impedance output).

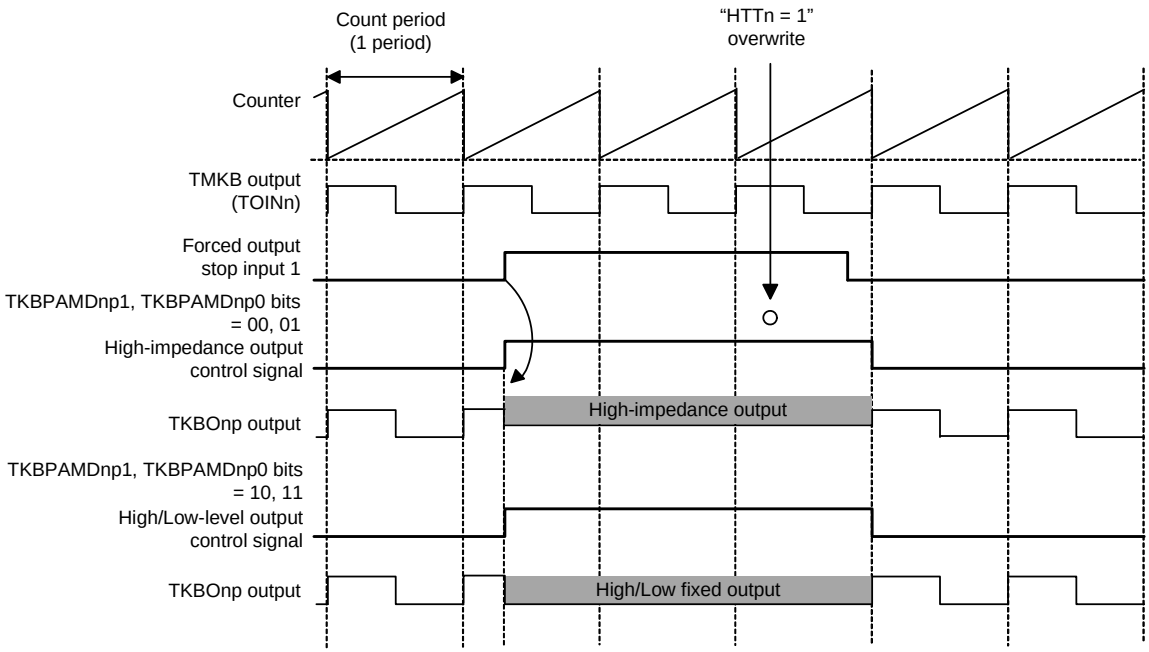
- TKBPAMDnp1, TKBPAMDnp0 bits = 10, 11

  When the rising edge of forced output stop input 1 is detected, the high/low-level output control signal is set to high level and TKBOnp output is fixed to high/low level according to the setting of the TKBPAMDnp0 bit.

  Writing 1 to the TKBPAHTTnp bit in the TKBPAHFTn register is invalid during the high-level period of the input of forced output stop input 1. After input of forced output stop input 1 changes to low level, the fixed low-level/high-level of TKBPATFOUTn is cancelled by writing 1 to the TKBPAHTTnp bit, and PWM is output from TKBOnp.

Remark n = 0 to 2; p = 0, 1

(3) TKBPAHCMnp1, TKBPAHCMnp0 Bits = 10



- TKBPAMDnp1, TKBPAMDnp0 bits = 00, 01

  When the rising edge of forced output stop input 1 is detected, the high-impedance output control signal is set to high level and TKBOnp output becomes high impedance.

  Regardless of the level of forced output stop input 1, the high-impedance output control signal is set to low level in synchronization with the restart of the TMKB counter by writing 1 to the TKBPAHTTnp bit.

  The high-level period of the high-impedance output control signal is the period for forced output stop 1 (high-impedance output).

- TKBPAMDnp1, TKBPAMDnp0 bits = 10, 11

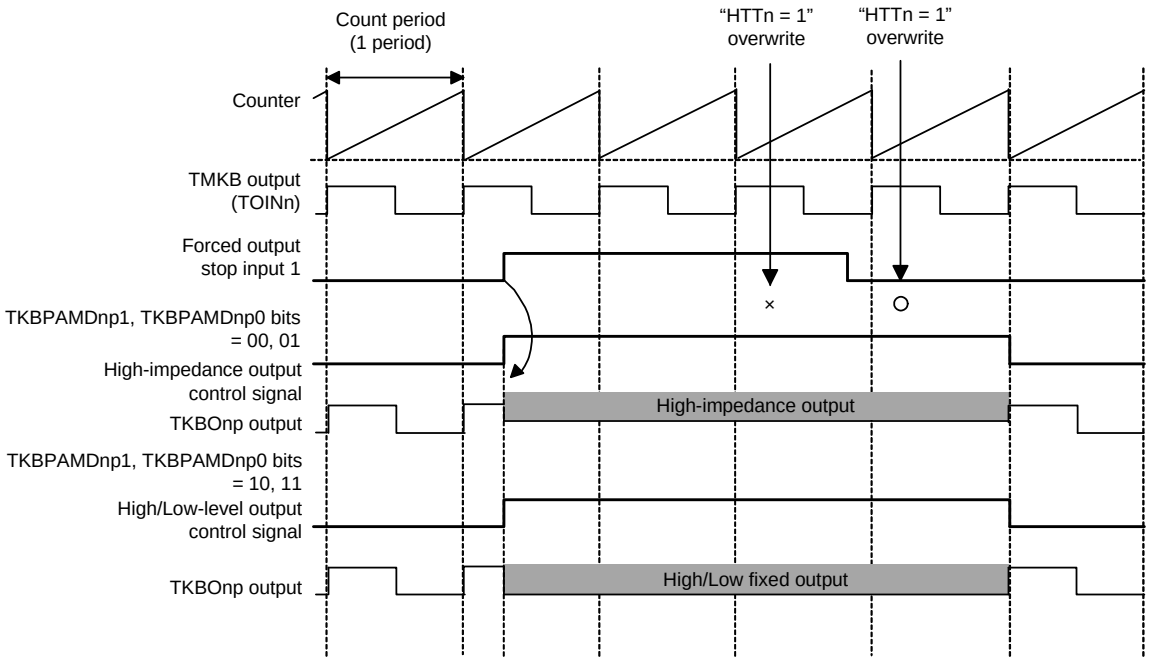
  When the rising edge of forced output stop input 1 is detected, the high/low-level output control signal is set to high level and TKBOnp output is fixed to high/low level according to the setting of the TKBPAMDnp0 bit.

  Regardless of the level of forced output stop input 1, the high/low-level output control signal is set to low level in synchronization with the restart of the TMKB counter by writing 1 to the TKBPAHTTnp bit, and TKBOnp returns to PWM output.

  The high-level period of the high/low-level output control signal is the period for forced output stop 1 (high/low-level fixed output).

Remark n = 0 to 2; p = 0, 1

(4) TKBPAHCMnp1, TKBPAHCMnp0 bits = 11



• TKBPAMDnp1, TKBPAMDnp0 bits = 00, 01

When the rising edge of forced output stop input 1 is detected, the high-impedance output control signal is set to high level and TKBOnp output becomes high impedance.

Writing 1 to the TKBPAHTTnp bit in the TKBPAHFTn register is invalid during the high-level period of the input of forced output stop input 1.

After forced output stop input 1 changes to low level, the high-impedance output control signal is set to low level in synchronization with the restart of the TMKB counter by writing 1 to the TKBPAHTTnp bit.

The high-level period of the high-impedance output control signal is the period for forced output stop 1 (high-impedance output).

• TKBPAMDnp1, TKBPAMDnp0 bits = 10, 11

When the rising edge of forced output stop input 1 is detected, the high/low-level output control signal is set to high level and TKBOnp output is fixed to high/low level according to the setting of the TKBPAMDnp0 bit.

The high/low-level output control signal is set to low level in synchronization with the restart of the TMKB counter by writing 1 to the TKBPAHTTnp bit in the TKBPAHFTn register during the high-level period of forced output stop 1, and TKBOnp returns to PWM output.

The high-level period of the high/low-level output control signal is the period for forced output stop 1 (high/low-level fixed output).

Remark n = 0 to 2; p = 0, 1

7.7.3 Notes on Using Forced Output Stop Function 1

1. When TKBPAHCMnp1, TKBPAHCMnp0 = 10 or 11, forced output stop is cancelled as follows.

(1) Setting TKBPAHCMnp1, TKBPAHCMnp0 = 10

(a) When forced output stop input 1 occurs

When TKBPAHCMnp1, TKBPAHCMnp0 = 10, if forced output stop input 1 is detected after setting TKBPAHTTnp = 1 and before the restart of the counter, stop input is ignored, and forced output stop is cancelled at the next restart of the counter.

(b) When TKBPAHTSnp is set to 1

When TKBPAHCMnp1, TKBPAHCMnp0 = 10, if TKBPAHTSnp is set to 1 after setting TKBPAHTTnp = 1 and before the next restart of the counter, setting TKBPATHTT0n = 1 is invalid, and forced output stop is not cancelled at the next restart of the counter. To cancel forced output stop, set TKBPAHTSnp to 1 again.

(2) Setting TKBPAHCMnp1, TKBPAHCMnp0 = 11

(a) Forced output stop input 1 occurs

When TKBPAHCMnp1, TKBPAHCMnp0 = 11, if forced output stop input 1 is detected after setting TKBPAHTTnp = 1 and before the next counter period, setting TKBPATHTT0n = 1 is invalid, and forced output stop is not cancelled at the next restart of the counter. To cancel forced output stop, set TKBPAHTSnp to 1 again.

(b) TKBPAHTSnp is set to 1

When TKBPAHCMnp1, TKBPAHCMnp = 11, if TKBPAHTSnp is set to 1 after setting TKBPAHTTnp = 1 and before the next counter period, setting TKBPATHTT0n = 1 is invalid and, forced output stop is not cancelled at the next restart of the counter. To cancel forced output stop, set TKBPAHTSnp to 1 again.

2. Timing for setting TKBPAHTSnp and TKBPATHTT0n when TKBPAHCMnp1, TKBPAHCMnp0 = 01, 11

When TKBPAHCMnp1, TKBPAHCMnp0 = 01, 11, wait until 1 f_{CLK} clock elapses before setting TKBPATHTT0n to 1 after setting TKBPAHTSnp = 1.

Remark n = 0 to 2; p = 0, 1

7.8 Operation of Forced Output Stop Function 2

7.8.1 I/O Setting for Forced Output Stop Function 2

In forced output stop function 2, TMKB2n forced output stop source or TMKB2n forced output stop source 0, 1, 2 is used as the trigger signal of forced output stop.
Selectable output status for forced output stop is high/low-level fixed. The tables below show trigger signal selection and output status settings.

(1) TKBOnp Output Control

• Forced output stop function trigger selection

TKBPACTLnp	Input Selection
TKBPAFXSnp3	External interrupt detection (INTPiNF) or comparator 0/1 Note
TKBPAFXSnp2	External interrupt detection (INTPiNF) or comparator 0/1 Note
TKBPAFXSnp1	External interrupt detection (INTPiNF) or comparator 0/1 Note
TKBPAFXSnp0	External interrupt detection (INTPiNF) or comparator 0/1 Note

• Output selection

TKBPACTLnp		Output Status
TKBPAMDnp1	TKBPAMDnp0	
0	0	Low-level fixed output
0	1	High-level fixed output
1	0	Low-level fixed output
1	1	High-level fixed output

Caution Operation of forced output stop function 2 does not affect the high-impedance output control signal. Do not select the high-impedance output control signal using the TKBPACTLnp register.

• Start of forced output stop function 2

TKBPACTLnp TKBPAFCMnp	Start Condition Selection for Forced Output Stop Function 2
0	The forced output stop function is started when the forced output stop function trigger is detected.
1	

• Cancel of forced output stop function 2

TKBPACTLnp TKBPAFCMnp	Cancel Condition Selection for Output of Forced Output Stop Function 2
0	Forced output stop function 2 is started, and cancelled in synchronization of the next restart of the counter.
1	Forced output stop function 2 is started, and cancelled in synchronization of the next restart of the counter after cancellation of the trigger is detected.

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

Note For details on trigger source settings, refer to **CHAPTER 20 EVENT LINK CONTROLLER (ELC)**.

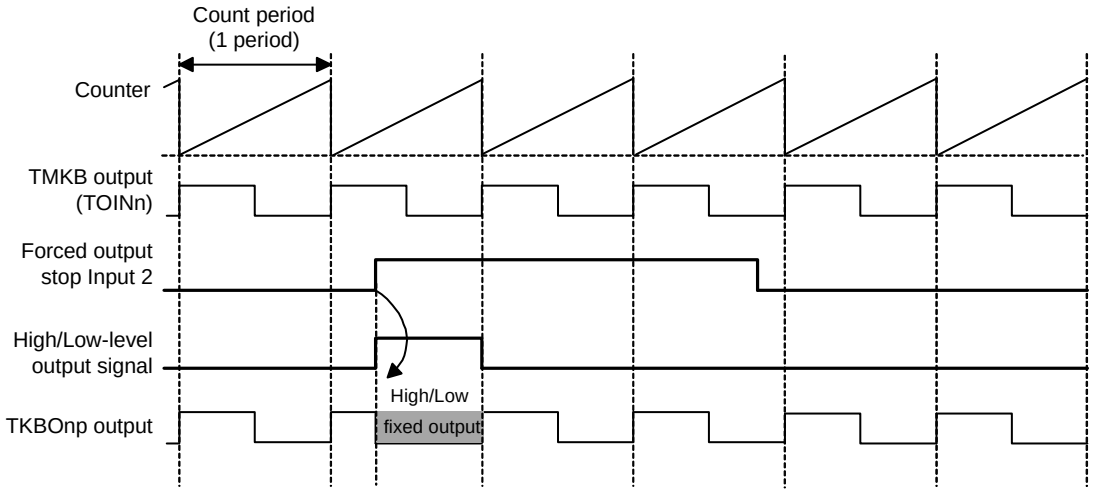
Caution 1. If comparator 0 or 1 detection is selected as a trigger input for forced output stop function 2 while the C0EDG or C1EDG bit in the comparator filter control register (COMPFIR) is 1 (both-edge detection), forced output stop function 2 cannot be cancelled by setting TKBPAFCMnp to 1. To cancel forced output stop function 2, be sure to set TKBPAFCMnp to 0.

Caution 2. Trigger input INTPiNF (event source corresponding to ELSELR23 to ELSELR30) which is used for forced output stop functions 1 and 2, are not affected by the setting of the external interrupt rising edge enable register (EGP0) and external interrupt falling edge enable register (EGN0). Only the rising edge is valid.

Remark i = 0 to 7; n = 0 to 2; p = 0, 1

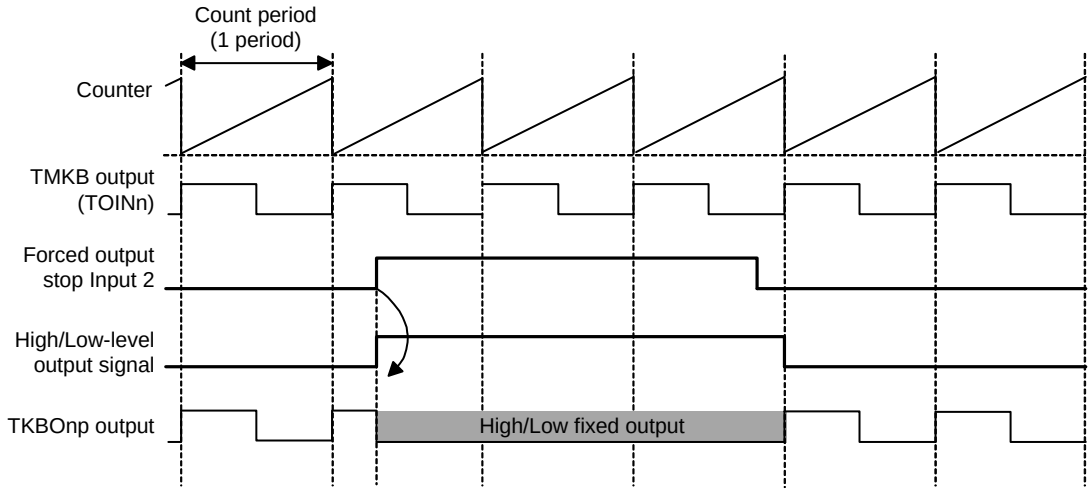
7.8.2 Basic Operation of Forced Output Stop Function 2

(1) Forced Output Stop Function 2 When TKBPAFCMnp = 0



When the rising edge of forced output stop input 2 is detected, TKBPATFOUTn output is fixed to low or high level according to TKBPAMDnp0 setting.
Regardless of the input level of forced output stop input 2, the fixed level of TKBPATFOUTn is cancelled in synchronization of the next restart of the TMKB counter, and PWM is output from TKBOnp.

(2) Forced Output Stop Function 2 When TKBPAFCMnp = 1



When the rising edge of forced output stop input 2 is detected, TKBPATFOUTn output is fixed to low or high level according to TKBPAMDnp0 setting.
After the rising edge of forced output stop input 2 is detected, the fixed level of TKBPATFOUTn is cancelled in synchronization of the next restart of the TMKB counter, and PWM is output from TKBOnp.

Remark n = 0 to 2; p = 0, 1

7.9 Usage Notes

(1) Do not specify multiple restart trigger inputs as one restart trigger source. The SFRs must be changed while a restart trigger for the related peripheral module is not generated.

(2) The LIN-bus function and 16-bit timers KB20, KB21, and KB22 cannot be used together.
When using the LIN-bus function while the input switch control register (ISC) is a value other than the initial value (00H), set bit 4 (TKB20EN) in peripheral enable register 1 (PER1) and bit 0 (TKB21EN) and bit 1 (TKB22EN) in peripheral enable register 2 (PER2) to 0 (timers KB20, KB21, and KB22 are in the reset status).

(3) Tables 7 - 2 to 7 - 4 list the input sources for timer KB2.

Table 7 - 2 Timer KB2n Input Sources Control Register List (Count Restart Trigger)

Input Source	Selection Control Register	Trigger Active Edge Selection Register				ELC Control Register ELSELRn (n = 0 to 19)
	Bits TKBSTSn1, TKBSTSn0 in TKBCTLn0	INTPi	Comparator 0, 1	Key	Others	
Timer KB2n Counter Restart Trigger	01B	EGP0, EGN0	C1EDG, C1EPO, C0EDG, and C0EPO in COMPFIR	KRM0	—	00100B, 01000B, 01100B
	10B					00101B, 01001B, 01101B
	11B					00110B, 01010B, 01110B

Table 7 - 3 Timer KB2n Input Sources Control Register List (Forced Output Stop Function 1)

Input Source	Selection Control Register	Trigger Active Edge Selection Register		ELC Control Register ELSELRn (n = 18,19, 23 to 30)
	TKBPACTLnp	Comparator 0, 1	INTPiNF	
Timer KB2 forced output stop source 0	TKBPAHZSnp0	COMPFIR can be set	Edge cannot be selected, rising edge is always active	10010B
Timer KB2 forced output stop source 1	TKBPAHZSnp1			10011B
Timer KB20 forced output stop source	TKBPAHZS0p2			10100B
Timer KB21 forced output stop source	TKBPAHZS1p2			10101B
Timer KB22 forced output stop source	TKBPAHZS2p2			10110B

Table 7 - 4 Timer KB2n Input Sources Control Register List (Forced Output Stop Function 2)

Input Source	Selection Control Register	Trigger Active Edge Selection Register		ELC Control Register ELSELRn (n = 18,19, 23 to 30)
	TKBPACTLnp	Comparator 0, 1	INTPiNF	
Timer KB2 forced output stop source 0	TKBPAFXSnp0	COMPFIR can be set	Edge cannot be selected, rising edge is always active	10010B
Timer KB2 forced output stop source 1	TKBPAFXSnp1			10011B
Timer KB2 forced output stop source 2	TKBPAFXSnp3			10111B
Timer KB20 forced output stop source	TKBPAFXS0p2			10100B
Timer KB21 forced output stop source	TKBPAFXS1p2			10101B
Timer KB22 forced output stop source	TKBPAFXS2p2			10110B

Remark n = 0 to 2; p = 0, 1

(4) INTP0NF to INTP7NF (via the ELC), which are used for forced output stop functions 1 and 2, are not affected by the setting of the external interrupt rising edge enable register (EGP0) and external interrupt falling edge enable register (EGN0). Only the rising edge is valid.
When using comparator 0 detection or comparator 1 detection (via the ELC) as forced output stop function 1 or 2, the edge can be selected by bit 2, 3, 6, or 7 (C0EPO, C0EDG, C1EPO, C1EDG) in the comparator filter control register (COMPFIR). However, setting TKBPAHCMnm0 = 1 cannot be selected as the condition for cancelling forced output stop function 1 while bit 3 or 7 (C0EDG, C1EDG) in the COMPFIR register is 1 (both-edge detection). Be sure to set TKBPAHCMnm0 to 0. Also, setting TKBPAFCMnm = 1 cannot be selected as the condition for cancelling forced output stop function 2. Be sure to set TKBPAFCMnm to 0.

CHAPTER 8 REAL-TIME CLOCK 2

8.1 Functions of Real-time Clock 2

The real-time clock 2 (RTC2) has the following functions.

- Counters of year, month, day of the week, date, hour, minute, and second, that can count up to 99 years (with leap year correction function)
- Constant-period interrupt function (period: 0.5 seconds, 1 second, 1 minute, 1 hour, 1 day, 1 month)
- Alarm interrupt function (alarm: day of the week, hour, and minute)
- Pin output function of 1 Hz

Caution The year, month, week, day, hour, minute and second can only be counted when a subsystem clock (f_{SUB} = 32.768 kHz) is selected as the operation clock of the real-time clock 2. When the low-speed oscillation clock (f_{IL} = 15 kHz) is selected, only the constant-period interrupt function is available.
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}.

8.2 Configuration of Real-time Clock 2

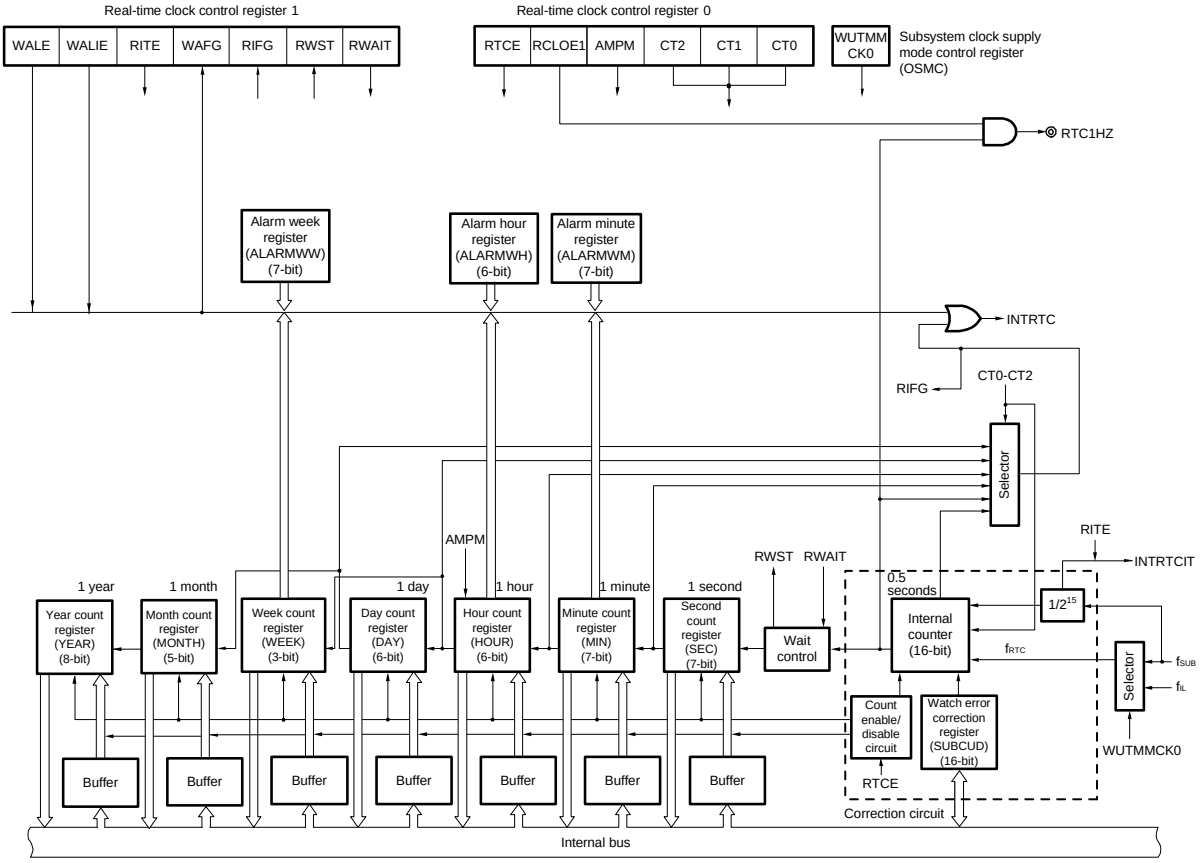
The real-time clock 2 includes the following hardware.

Table 8 - 1 Configuration of Real-time Clock 2

Item	Configuration
Counter	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 8 - 1 shows the Block Diagram of Real-time Clock 2.

Figure 8 - 1 Block Diagram of Real-time Clock 2



Caution The year, month, week, day, hour, minute and second can only be counted when a subsystem clock (f_{SUB} = 32.768 kHz) is selected as the operation clock of the real-time clock 2. When the low-speed oscillation clock (f_{IL} = 15 kHz) is selected, only the constant-period interrupt function is available.
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}.

8.3 Registers Controlling Real-time Clock 2

The real-time clock 2 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)

The following shows the register states depending on reset sources.

Reset Source	System-related Register Note 1	Calendar-related Register Note 2
POR	Reset	Not reset
External reset	Retained	Retained
WDT	Retained	Retained
TRAP	Retained	Retained
LVD	Retained	Retained
Other internal reset	Retained	Retained

Note 1. RTCC0, RTCC1, SUBCUD

Note 2. SEC, MIN, HOUR, DAY, WEEK, MONTH, YEAR, ALARMWM, ALARMWH, ALARMWW, (counter)

Reset generation does not reset the SEC, MIN, HOUR, DAY, WEEK, MONTH, YEAR, ALARMWM, ALARMWH, or ALARMWW register. Initialize all the registers after power on.

8.3.1 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 2 registers are manipulated, be sure to set bit 7 (RTCWEN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Figure 8 - 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	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

RTCWEN	Control of internal clock supply to real-time clock 2
0	Stops input clock supply. (Stops f_{CLK} clock supply) • SFR used by the real-time clock 2 cannot be written. • The real-time clock 2 can operate.
1	Enables input clock supply. • SFR used by the real-time clock 2 can be read/written. • The real-time clock 2 can operate.

Caution 1. When using the real-time clock 2, first set the RTCWEN bit to 1, while oscillation of the input clock (f_{RTC}) is stable. If RTCWEN = 0, writing to a control register of the real-time clock 2 is ignored.

Caution 2. Be sure to set bits 1 and 6 to 0.

8.3.2 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 other than the real-time clock 2, 12-bit interval timer, clock output/buzzer output controller, and LCD controller/driver is stopped in STOP mode or in HALT mode while the subsystem clock is selected as the CPU clock.

In addition, the OSMC register is used to select the operation clock of the real-time clock 2, 12-bit interval timer, clock output/buzzer output, and LCD controller/driver.

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

Reset signal generation clears this register to 00H.

Figure 8 - 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

RTCLPC	In STOP mode and in HALT mode while the CPU operates using the subsystem clock
0	Enables subsystem clock supply to peripheral functions. For peripheral functions for which operation is enabled, refer to **CHAPTER 23 STANDBY FUNCTION**.
1	Stops subsystem clock supply to peripheral functions other than the real-time clock 2, 12-bit interval timer, clock output/buzzer output, and LCD controller/driver.

WUTMMCK0 Notes 1, 2, 3	Selection of operation clock of the real-time clock 2, 12-bit interval timer, and LCD controller/driver	Selection of clock output from PCLBUZn pin of clock output/buzzer output controller
0	Subsystem clock (f_{SUB})	Selecting the subsystem clock (f_{SUB}) is enabled.
1	Low-speed on-chip oscillator clock (f_{IL})	Selecting the subsystem clock (f_{SUB}) is disabled.

Note 1. The f_{IL} clock can be selected (WUTMMCK0 = 1) only when oscillation of the subsystem clock is stopped (the XTSTOP bit in the CSC register = 1).

Note 2. When WUTMMCK0 is set to 1, the low-speed on-chip oscillator clock oscillates.

Note 3. When WUTMMCK0 is set to 1, the 1 Hz output function of the real-time clock 2 cannot be used.

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 2
When the low-speed oscillation clock (f_{IL} = 15 kHz) is selected, only the constant-period interrupt function is available.
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}.

8.3.3 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 2 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.

Internal reset generated by the power-on-reset circuit clears this register to 00H.

Figure 8 - 4 Format of Real-time clock control register 0 (RTCC0) (1/2)

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 Note 1	Real-time clock 2 operation control
0	Stops counter operation.
1	Starts counter operation.

RCLOE1 Note 2	RTC1HZ pin output control
0	Disables output of the RTC1HZ pin (1 Hz).
1	Enables output of the RTC1HZ pin (1 Hz).
Output of 1 Hz is not output because the clock counter does not operate when RTCE = 0.	

Note 1. When shifting to STOP mode immediately after setting RTCE to 1, use the procedure shown in Figure 8 - 21 Procedure for Shifting to HALT/STOP Mode After Setting RTCE bit to 1.

Note 2. When the RCLOE1 bit is set while the clock counter operates (RTCE = 1), a glitch may be output to the 1 Hz output pin (RTC1HZ).

Caution Be sure to clear bits 4 and 6 to “0”.

Figure 8 - 5 Format of Real-time clock control register 0 (RTCC0) (2/2)

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

Table 8 - 2 Relation between RTCE and RCLOE1 Settings and Status

Register Settings		Status	
RTCE	RCLOE1	Real-time clock 2	RTC1HZ pin output
0	×	Counting stopped	No output
1	0	Count operation	No output
	1	Count operation	1 Hz output

AMPM	12-/24-hour system select
0	12-hour system (a.m. and p.m. are displayed.)
1	24-hour system
• When changing the value of the AMPM bit while the clock counter operates (RTCE = 1), set RWAIT (bit 0 of RTCC1) and then set the hour counter (HOUR) again. When the AMPM value is 0, the 12-hour system is displayed. When the value is 1, the 24-hour system is displayed. • Table 8 - 3 shows the displayed time digits.	

CT2	CT1	CT0	Constant-period interrupt (INTRTC) selection
0	0	0	Does not use fixed-cycle interrupt function.
0	0	1	Once per 0.5 s (synchronized with second count up)
0	1	0	Once per 1 s (same time as second count up)
0	1	1	Once per 1 m (second 00 of every minute)
1	0	0	Once per 1 hour (minute 00 and second 00 of every hour)
1	0	1	Once per 1 day (hour 00, minute 00, and second 00 of every day)
1	1	×	Once per 1 month (Day 1, hour 00 a.m., minute 00, and second 00 of every month)
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.			

Caution Be sure to clear bits 4 and 6 to “0”.

Remark ×: don’t care

8.3.4 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.

Internal reset generated by the power-on-reset circuit clears this register to 00H.

Figure 8 - 6 Format of Real-time clock control register 1 (RTCC1) (1/3)

Address: FFF9EH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	RITE	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.

Caution If writing is performed to RTCC1 with a 1-bit manipulation instruction, the RIFG and WAFG flags may be cleared. Therefore, to perform writing to RTCC1, be sure to use an 8- bit manipulation instruction. To prevent the RIFG and WAFG flags from being cleared during writing, set 1 (writing disabled) to the corresponding bit. If the RIFG and WAFG flags are not used and the value may be changed, RTCC1 may be written by using a 1-bit manipulation instruction.

Figure 8 - 7 Format of Real-time clock control register 1 (RTCC1) (2/3)

Address: FFF9EH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	RITE	WAFG	RIFG	0	RWST	RWAIT

RITE	Control of correction timing signal interrupt (INTRTIT) function operation
0	Does not generate interrupt of correction timing signal.
1	Generates interrupt of correction timing signal.

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 clock (32.768 kHz) after matching of the alarm is detected. This flag is cleared when “0” is written to it. Writing “1” to it is invalid. (Writing 1 does not change the value of WAFG.)	

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. (Writing 1 does not change the value of RIFG.)	

Caution If writing is performed to RTCC1 with a 1-bit manipulation instruction, the RIFG and WAFG flags may be cleared. Therefore, to perform writing to RTCC1, be sure to use an 8- bit manipulation instruction. To prevent the RIFG and WAFG flags from being cleared during writing, set 1 (writing disabled) to the corresponding bit. If the RIFG and WAFG flags are not used and the value may be changed, RTCC1 may be written by using a 1-bit manipulation instruction.

Figure 8 - 8 Format of Real-time clock control register 1 (RTCC1) (3/3)

Address: FFF9EH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	RITE	WAFG	RIFG	0	RWST	RWAIT

RWST	Wait status flag of real-time clock 2
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. Even if the RWAIT bit is set to 0, the RWST bit is not set to 0 while writing to the counter. After writing is completed, the RWST bit is set to 0.	

RWAIT	Wait control of real-time clock 2
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 counter (16-bit) is continuing to run, complete reading or writing within one second and turn back to 0. 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 (f_{RTC}) until the counter value can be read or written (RWST = 1).Note1, 2 When the 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.	

Note 1. When setting RWAIT=1 during 1 operating clock (f_{RTC}), after setting RTCE=1, it may take two clock time of the operation clock (f_{RTC}), until RWST bit is set to “1”.

Note 2. When setting RWAIT=1 during 1 operating clock (f_{RTC}), after returning from a stand-by (HALT mode, STOP mode and SNOOZE mode), it many take two clock time of the operation clock (f_{RTC}), until RWST bit is set to “1”.

Caution **If writing is performed to RTCC1 with a 1-bit manipulation instruction, the RIFG and WAFG flags may be cleared. Therefore, to perform writing to RTCC1, be sure to use an 8- bit manipulation instruction. To prevent the RIFG and WAFG flags from being cleared during writing, set 1 (writing disabled) to the corresponding bit. If the RIFG and WAFG flags are not used and the value may be changed, RTCC1 may be written by using a 1-bit manipulation instruction.**

Remark Constant-period 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 constant-period interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

8.3.5 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 is a decimal counter that counts up when the 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 clocks (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 not clears this register to default value.

Figure 8 - 9 Format of Second count register (SEC)

Address: FFF92H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

Caution When reading or writing to SEC while the clock counter operates (RTCE = 1), be sure to use the flows shown in 8.4.3 Reading real-time clock 2 and 8.4.4 Writing to real-time clock 2 counter.

8.3.6 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 is a decimal counter that 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 clocks (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 not clears this register to default value.

Figure 8 - 10 Format of Minute count register (MIN)

Address: FFF93H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

Caution When reading or writing to MIN while the clock counter operates (RTCE = 1), be sure to use the flows shown in 8.4.3 Reading real-time clock 2 and 8.4.4 Writing to real-time clock 2 counter.

8.3.7 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 is a decimal counter that 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 clocks (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 not clears this register to default value.

Figure 8 - 11 Format of Hour count register (HOUR)

Address: FFF94H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

Caution 1. Bit 5 (HOUR20) of the HOUR register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

Caution 2. When reading or writing to HOUR while the clock counter operates (RTCE = 1), be sure to use the flows shown in 8.4.3 Reading real-time clock 2 and 8.4.4 Writing to real-time clock 2 counter.

Table 8 - 3 shows the relationship between the setting value of the AMPM bit, the hour count register (HOUR) value, and time.

Table 8 - 3 Displayed Time Digits

24-Hour Display (AMPM = 1)		12-Hour Display (AMPM = 1)	
Time	HOUR Register	Time	HOUR Register
0	00 H	12 a.m.	12 H
1	01 H	1 a.m.	01 H
2	02 H	2 a.m.	02 H
3	03 H	3 a.m.	03 H
4	04 H	4 a.m.	04 H
5	05 H	5 a.m.	05 H
6	06 H	6 a.m.	06 H
7	07 H	7 a.m.	07 H
8	08 H	8 a.m.	08 H
9	09 H	9 a.m.	09 H
10	10 H	10 a.m.	10 H
11	11 H	11 a.m.	11 H
12	12 H	12 p.m.	32 H
13	13 H	1 p.m.	21 H
14	14 H	2 p.m.	22 H
15	15 H	3 p.m.	23 H
16	16 H	4 p.m.	24 H
17	17 H	5 p.m.	25 H
18	18 H	6 p.m.	26 H
19	19 H	7 p.m.	27 H
20	20 H	8 p.m.	28 H
21	21 H	9 p.m.	29 H
22	22 H	10 p.m.	30 H
23	23 H	11 p.m.	31 H

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.

8.3.8 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 is a decimal counter that count ups when the hour counter overflows.
This counter counts as follows.

[DAY count values]
- 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 clocks (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 not clears this register to default value.

Figure 8 - 12 Format of Day count register (DAY)

Address: FFF96H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

Caution When reading or writing to DAY while the clock counter operates (RTCE = 1), be sure to use the flows shown in 8.4.3 Reading real-time clock 2 and 8.4.4 Writing to real-time clock 2 counter.

8.3.9 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 is a decimal counter that counts up when a carry to the date counter occurs.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (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 not clears this register to default value.

Figure 8 - 13 Format of Week count register (WEEK)

Address: FFF95H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

Caution 1. The value corresponding to the month count register (MONTH) or the day count register (DAY) is not stored in the week count register (WEEK) automatically. After reset release, set the week count register as follow.

Day	WEEK
Sunday	00 H
Monday	01 H
Tuesday	02 H
Wednesday	03 H
Thursday	04 H
Friday	05 H
Saturday	06 H

Caution 2. When reading or writing to WEEK while the clock counter operates (RTCE = 1), be sure to use the flows shown in 8.4.3 Reading real-time clock 2 and 8.4.4 Writing to real-time clock 2 counter.

8.3.10 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 is a decimal counter that count ups when the date counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (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 not clears this register to default value.

Figure 8 - 14 Format of Month count register (MONTH)

Address: FFF97H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

Caution When reading or writing to MONTH while the clock counter operates (RTCE = 1), be sure to use the flows shown in 8.4.3 Reading real-time clock 2 and 8.4.4 Writing to real-time clock 2 counter.

8.3.11 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 is a decimal counter that 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 clocks (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 not clears this register to default value.

Figure 8 - 15 Format of Year count register (YEAR)

Address: FFF98H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

Caution When reading or writing to YEAR while the clock counter operates (RTCE = 1), be sure to use the flows shown in 8.4.3 Reading real-time clock 2 and 8.4.4 Writing to real-time clock 2 counter.

8.3.12 Watch error correction register (SUBCUD)

This register is used to correct the clock with a minimum resolution and accuracy of 0.96 ppm when it is slow or fast by changing the counter value every second.

The SUBCUD register can be set by an 16-bit memory manipulation instruction.

Internal reset generated by the power-on-reset circuit clears this register to 0020H.

Figure 8 - 16 Format of Watch error correction register (SUBCUD)

Address: F0310H After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SUBCUD	F15	0	0	0	0	0	0	F8	F7	F6	F5	F4	F3	F2	F1	F0

F15	Clock error correction enable
0	Stops clock error correction.
1	Enables clock error correction.

The range of value that can be corrected by using the clock error correction register (SUBCUD) is shown in Table 8 - 4.

Table 8 - 4 Correctable Range of Crystal Resonator Oscillation Frequency Deviation

Item	Value
Correctable range	-274.6 ppm to +212.6 ppm
Maximum quantization error	±0.48 ppm
Minimum resolution	0.96 ppm

Table 8 - 5 Clock Error Correction Values

SUBCUD										Target Correction Values
F15	F8	F7	F6	F5	F4	F3	F2	F1	F0	
1	1	0	0	0	0	0	0	0	0	-274.6 ppm
	1	0	0	0	0	0	0	0	1	-273.7 ppm
	1	0	0	0	0	0	0	1	0	-272.7 ppm
	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
	1	1	1	1	1	1	1	0	1	-33.3 ppm
	1	1	1	1	1	1	1	1	0	-32.4 ppm
	1	1	1	1	1	1	1	1	1	-31.4 ppm
	0	0	0	0	0	0	0	0	0	-30.5 ppm
	0	0	0	0	0	0	0	0	1	-29.6 ppm
	0	0	0	0	0	0	0	1	0	-28.6 ppm
	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
	0	0	0	0	1	1	1	1	1	-0.95 ppm
	0	0	0	1	0	0	0	0	0	0 ppm
	0	0	0	1	0	0	0	0	1	0.95 ppm
	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
	0	1	1	1	1	1	1	0	1	210.7 ppm
	0	1	1	1	1	1	1	1	0	211.7 ppm
	0	1	1	1	1	1	1	1	1	212.6 ppm
0	×	×	×	×	×	×	×	×	×	Clock error correction stopped

The F8 to F0 value of the SUBCUD register is calculated from the target correction value by using the following expression.

$$\text{SUBCUD[8:0]} = \left\{ \frac{\text{Target correction value [ppm]} \times 2^{20}}{10^6} \right\}_{\text{Binary (9 digits)}} + 0\ 0010\ 0000\ \text{B}$$

Caution The target correction value is the oscillation frequency deviation (unit: [ppm]) of the crystal resonator. For calculating the correction value, see 8.4.8 Example of watch error correction of real-time clock 2.

Examples 1. When target correction value = 18.3 [ppm]

SUBCUD[8:0] = $(18.3 \times 2^{20} / 10^6)_{\text{Binary (9 digits)}}$ + 000100000B
= $(19.1889408)_{\text{Binary (9 digits)}}$ + 000100000B
= 000010011B + 000100000B
= 000110011B

Examples 2. When target correction value = -18.3 [ppm]

SUBCUD[8:0] = $(-18.3 \times 2^{20} / 10^{6})$ Binary (9 digits) + 000100000B
= (-19.1889408) Binary (9 digits) + 000100000B
= (000010011B) two's complement + 000100000B
= 111101101B + 000100000B
= 000001101B

8.3.13 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 not clears this register to default value.

Figure 8 - 17 Format of Alarm minute register (ALARMWM)

Address: FFF9AH After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

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.

8.3.14 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 not clears this register to default value.

Figure 8 - 18 Format of Alarm hour register (ALARMWH)

Address: FFF9BH After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

Caution 1. Set a decimal value of 00 to 23 or 01 to 12 and 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Caution 2. Bit 5 (WH20) of the ALARMWH register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

8.3.15 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 not clears this register to default value.

Figure 8 - 19 Format of Alarm week register (ALARMWW)

Address: FFF9CH After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

Table 8 - 6 shows an example of setting the alarm.

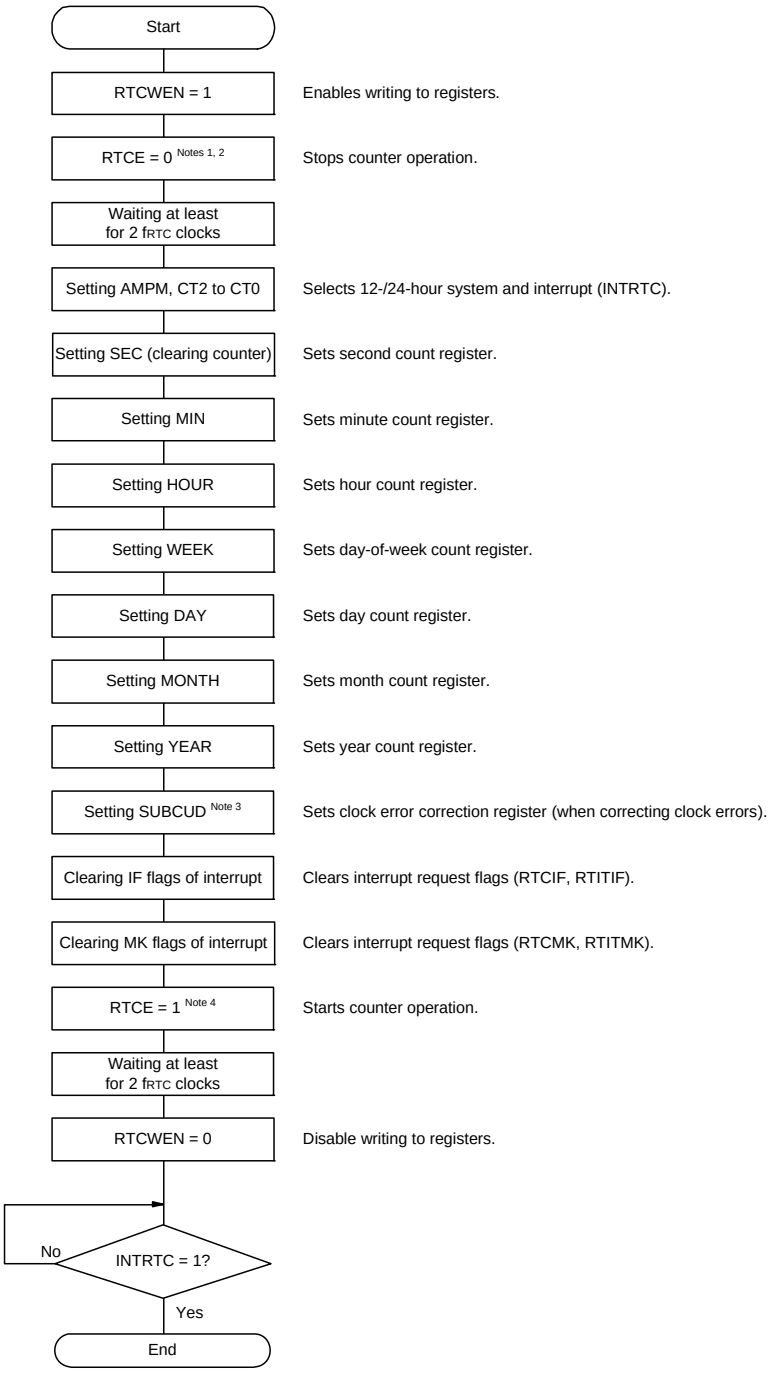
Table 8 - 6 Setting Alarm

Time of Alarm	Day							12-Hour Display				24-Hour Display			
	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Hour 10	Hour 1	Minute 10	Minute 1	Hour 10	Hour 1	Minute 10	Minute 1
	WW0	WW1	WW2	WW3	WW4	WW5	WW6								
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

8.4 Real-time Clock 2 Operation

8.4.1 Starting operation of real-time clock 2

Figure 8 - 20 Procedure for Starting Operation of Real-time Clock 2



Note 1. Set RTCWEN to 0 except when accessing the RTC register.

Note 2. First set the RTCWEN bit to 1, while oscillation of the input clock (f_{RTC}) is stable.

Note 3. Set up the SUBCUD register only if the watch error must be corrected. For details about how to calculate the correction value, see **8.4.8 Example of watch error correction of real-time clock 2**.

Note 4. Confirm the procedure described in **8.4.2 Shifting to HALT/STOP mode after starting operation** when shifting to HALT/STOP mode without waiting for INTRTC = 1 after RTCE = 1.

8.4.2 Shifting to HALT/STOP mode after starting operation

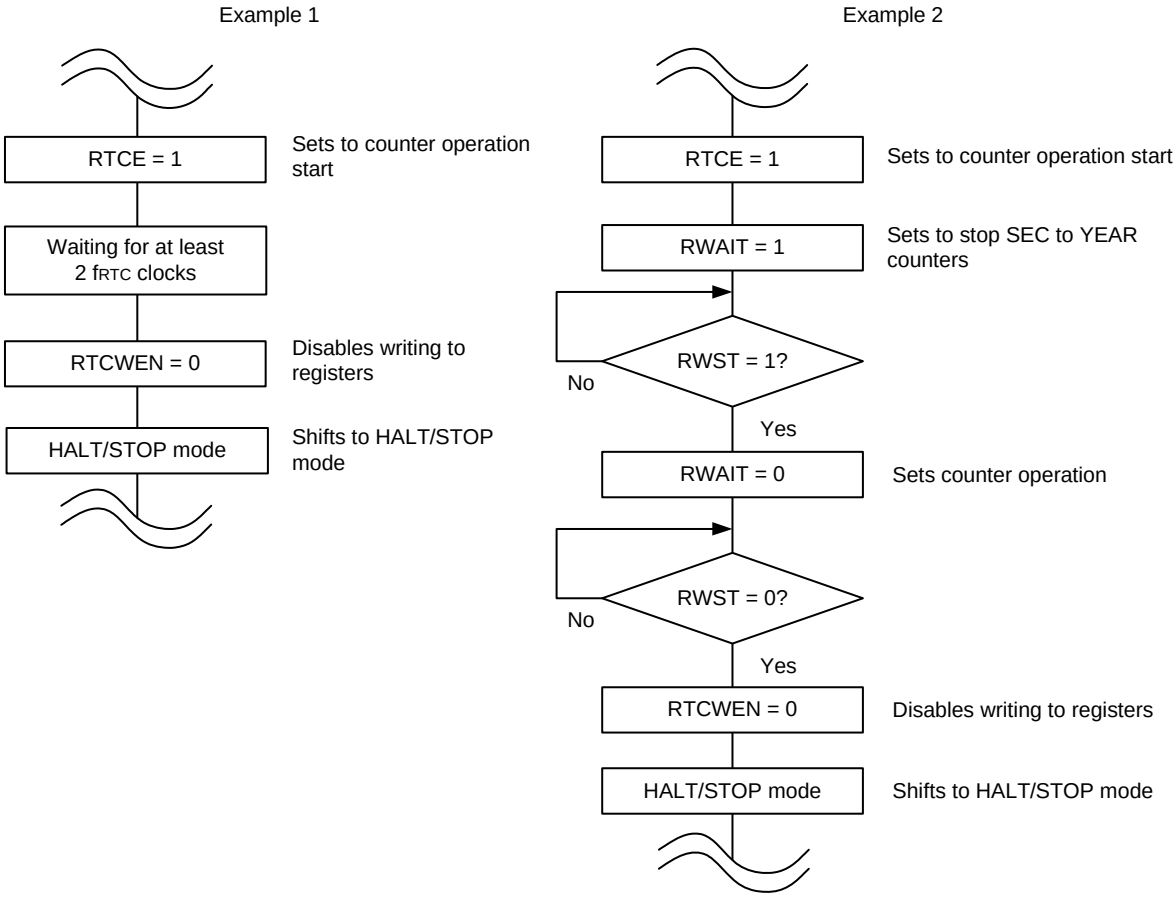
Perform one of the following processing when shifting to HALT/STOP mode immediately after setting the RTCE bit to 1.

However, after setting the RTCE bit to 1, this processing is not required when shifting to STOP mode after the first INTRTC interrupt has occurred.

(1) Shifting to HALT/STOP mode when at least two input clocks (f_{RTC}) have elapsed after setting the RTCE bit to 1 (see **Figure 8 - 21**, **Example 1**).

(2) 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 8 - 21**, **Example 2**).

Figure 8 - 21 Procedure for Shifting to HALT/STOP Mode After Setting RTCE bit to 1



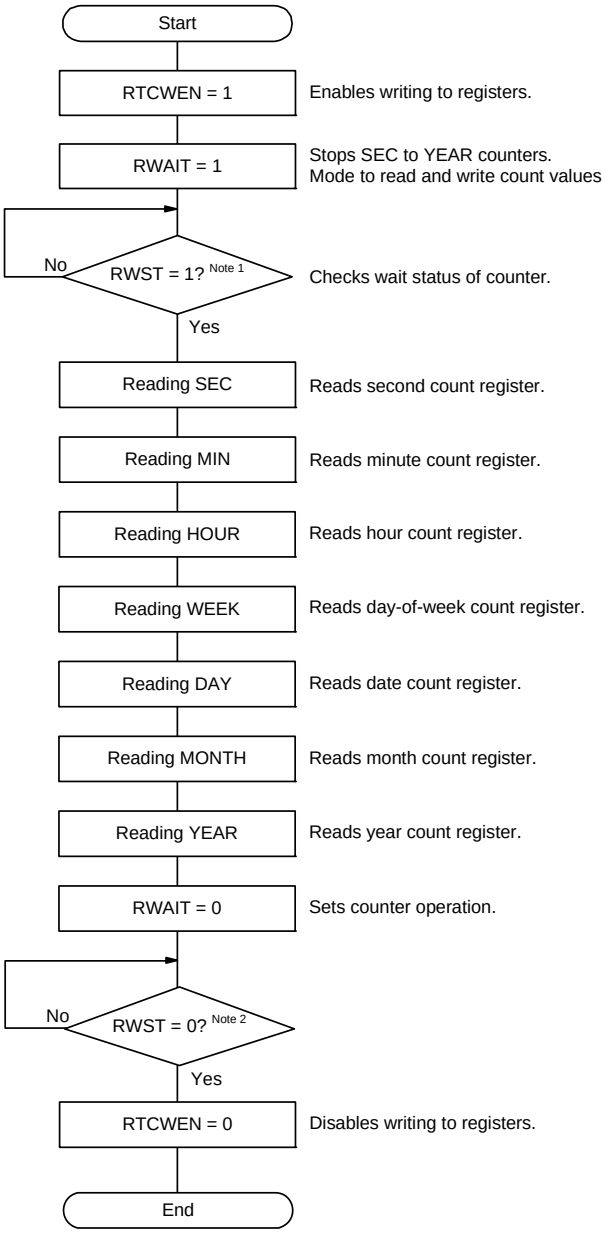
8.4.3 Reading real-time clock 2

Read the counter during counter operation (RTCE = 1) after setting RWAIT to 1 first.

Set RWAIT to 0 after completion of reading the counter.

When the alarm interrupt is in use, read from the counters according to the procedures shown in Figure 8 - 23.

Figure 8 - 22 Procedure for Reading Real-time Clock 2



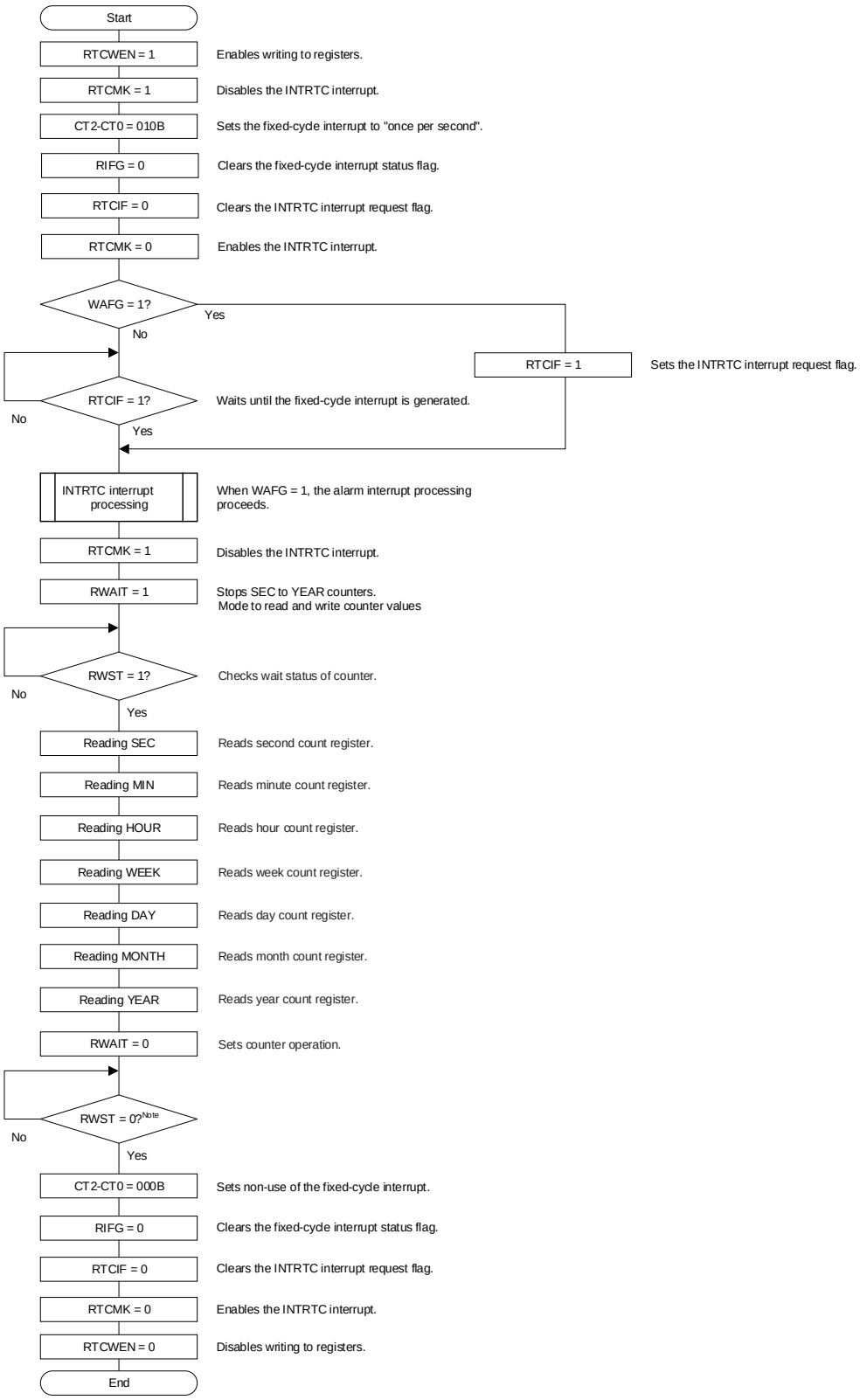
Note 1. When the counter is stopped (RTCE = 0), RWST is not set to 1.

Note 2. Be sure to confirm that RWST = 0 before setting STOP mode.

Caution **Complete the series of operations of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.**

Remark SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be read in any sequence. All the registers do not have to be set and only some registers may be read.

Figure 8 - 23 Procedure for Reading Real-time Clock 2 (When the Alarm Interrupt is in Use)



Note Be sure to confirm that RWST = 0 before setting STOP mode.

Caution **Complete the series of operations of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.**

Remark SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be read in any sequence. All the registers do not have to be set and only some registers may be read.

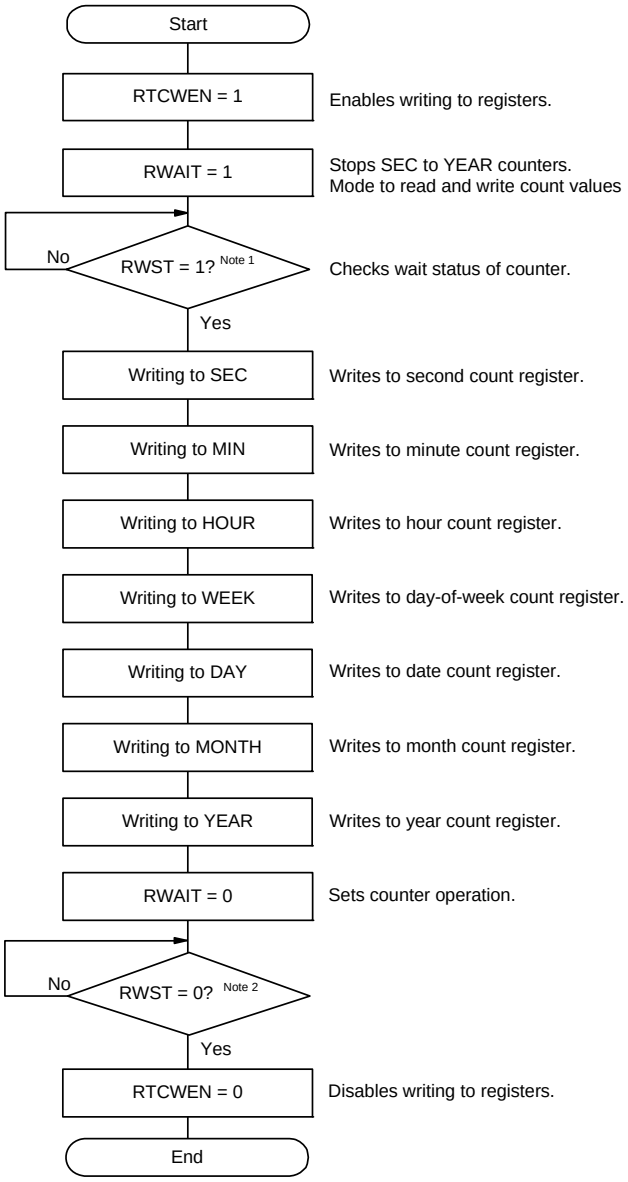
8.4.4 Writing to real-time clock 2 counter

Write the counter during counter operation (RTCE = 1) after setting RWAIT to 1 first.

Set RWAIT to 0 after completion of writing the counter.

When the alarm interrupt is in use, write to the counters according to the procedures shown in Figure 8 - 25.

Figure 8 - 24 Procedure for Writing Real-time Clock 2



Note 1. When the counter is stopped (RTCE = 0), RWST is not set to 1.

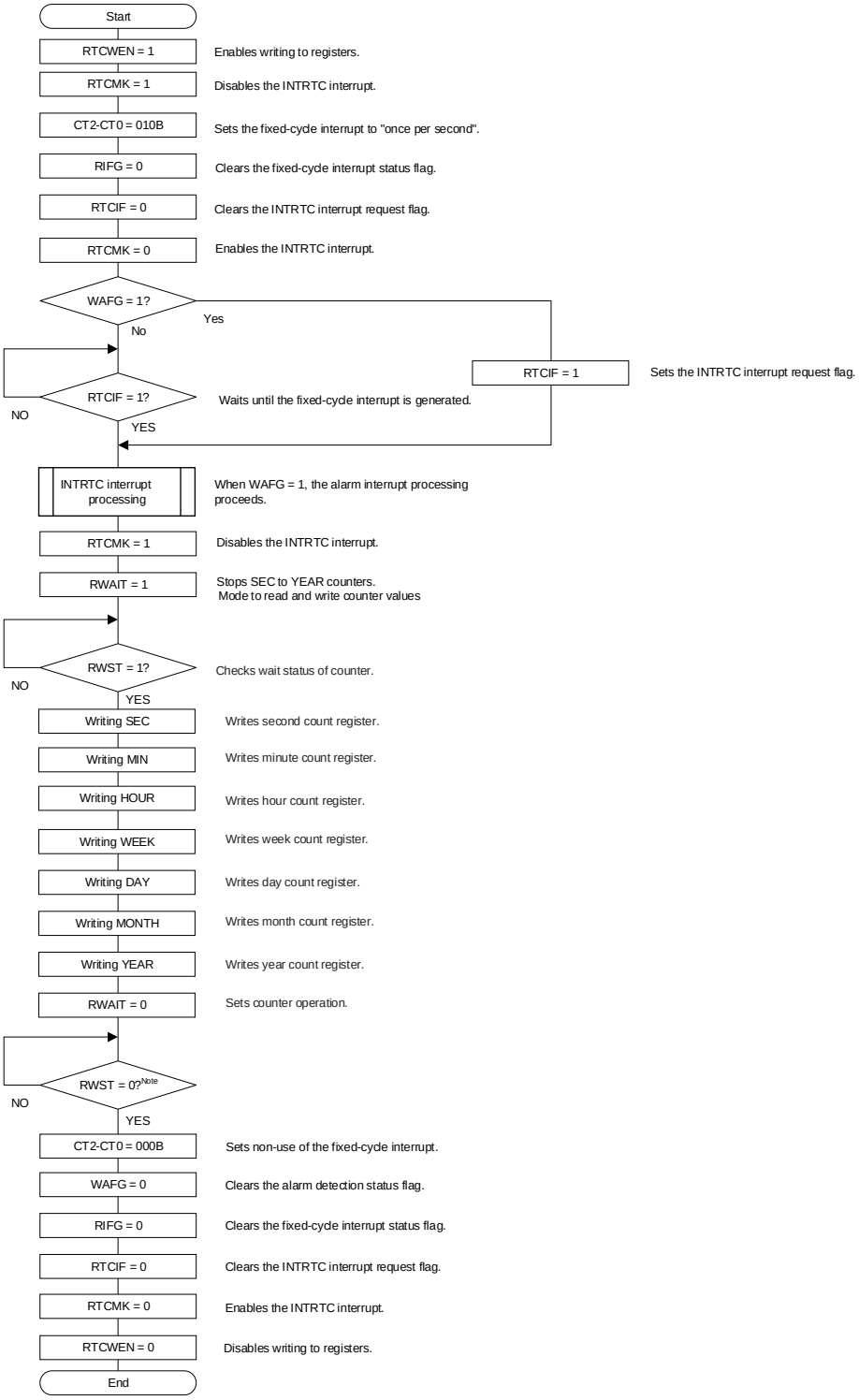
Note 2. Be sure to confirm that RWST = 0 before setting STOP mode.

Caution 1. Complete the series of operations of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.

Caution 2. When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR registers while counting is in progress (RTCE = 1), rewrite the registers after disabling interrupt processing of INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags after rewriting the registers.

Remark SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be written in any sequence. All the registers do not have to be set and only some registers may be written.

Figure 8 - 25 Procedure for Writing Real-time Clock 2 (When the Alarm Interrupt is in Use)



Note Be sure to confirm that RWST = 0 before setting STOP mode.

Caution 1. Complete the parts of the process from the start of INTRTC interrupt processing to clearing the RWAIT bit to 0 within 1 second.

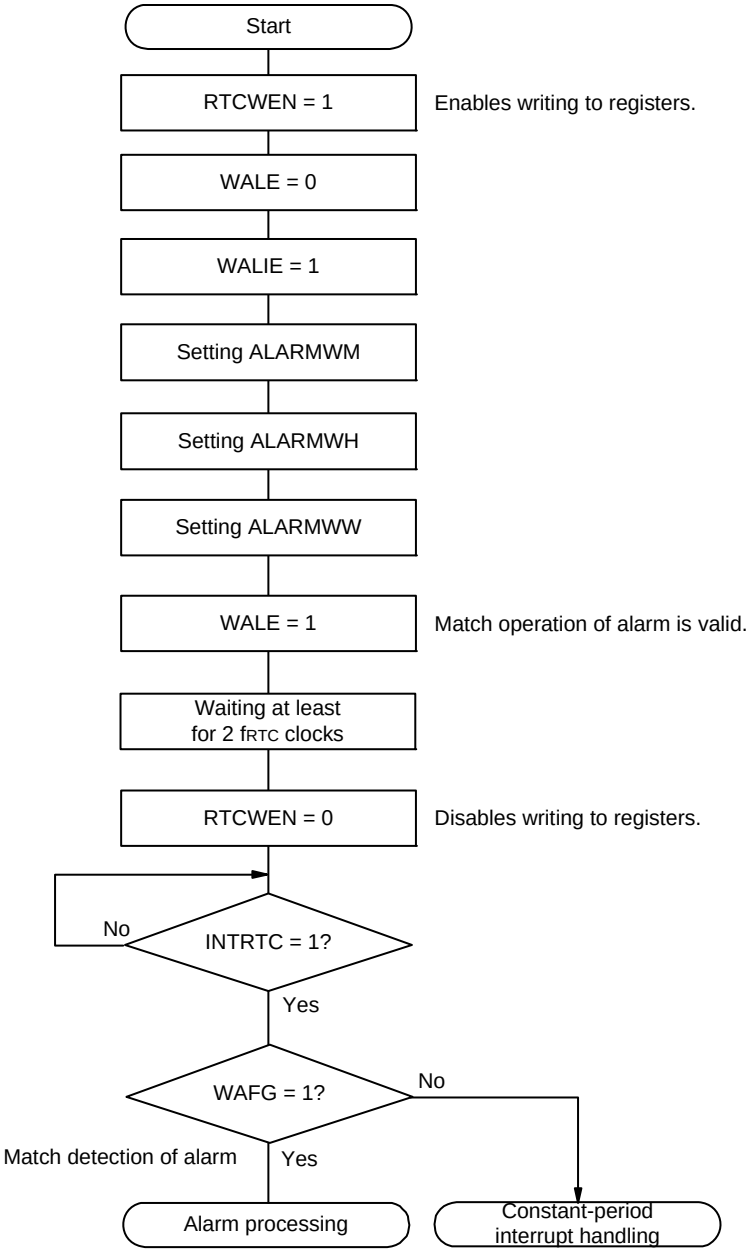
Caution 2. When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR registers while counting is in progress (RTCE = 1), rewrite the registers after disabling interrupt processing of INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags after rewriting the registers.

Remark SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be written in any sequence. All the registers do not have to be set and only some registers may be written.

8.4.5 Setting alarm of real-time clock 2

Set the alarm time after setting 0 to WALE (alarm operation invalid) first.

Figure 8 - 26 Alarm Setting Procedure

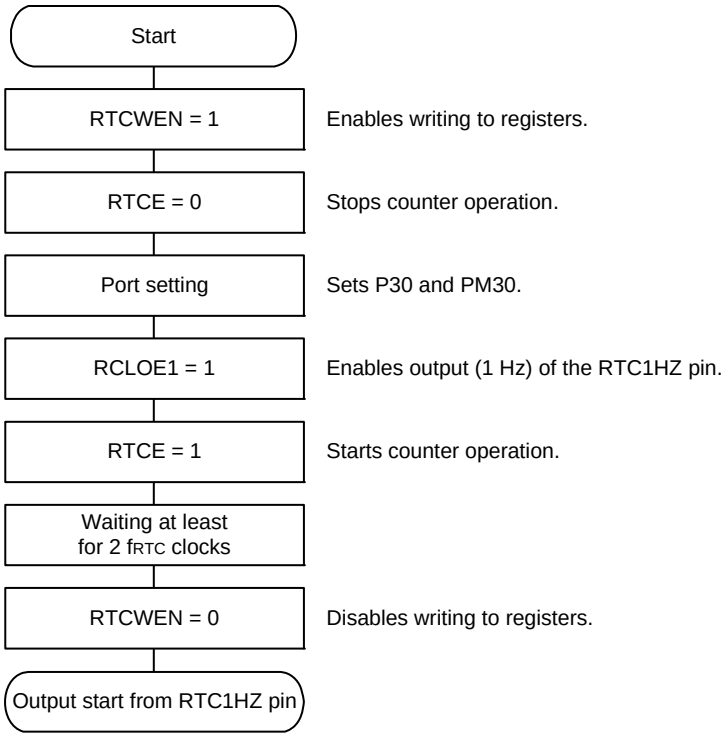


Remark 1. ALARMWM, ALARMWH, and ALARMWW may be written in any sequence.

Remark 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.

8.4.6 1 Hz output of real-time clock 2

Figure 8 - 27 1 Hz Output Setting Procedure

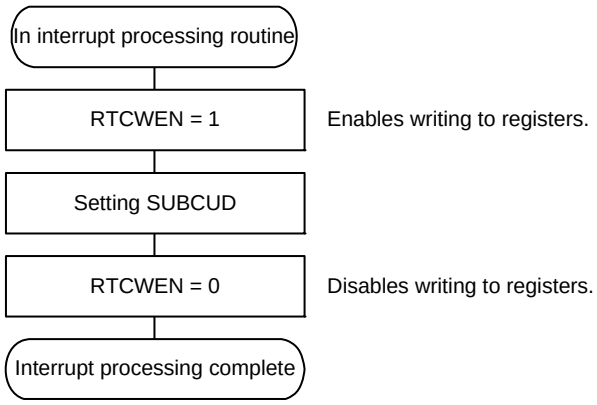


8.4.7 Clock error correction register setting procedure

To set the clock error correction register (SUBCUD), perform the following procedure in the interrupt handling routine of the correction timing signal interrupt (INTRTIT).

Caution The process from generation of a correction timing signal interrupt (INTRTIT) to the interrupt response and SUBCUD setting should be completed within 1 second (before the next timing of correction every second).

Set the clock error correction register after setting RTCWEN to 1 first. Then set RTCWEN to 0.



8.4.8 Example of watch error correction of real-time clock 2

The clock can be corrected every second with a minimum resolution and accuracy of 0.96 ppm when it is slow or fast, by setting a value to the clock error correction register.
The following shows how to calculate the target correction value, and how to calculate the F8 to F0 values of the clock error correction register from the target correction value.

Calculating the target correction value 1

(When using output frequency of the RTC1HZ pin)

[Measuring the oscillation frequency]

The oscillation frequency Note of each product is measured by outputting about 1 Hz from the RTC1HZ pin when the F15 of the watch error correction register (SUBCUD) is set to 1 (stops the watch error correction).

Note See **8.4.6 1 Hz output of real-time clock 2** for the setting procedure of outputting about 1 Hz from the RTC1HZ pin.

[Measuring the oscillation frequency]
(When the output frequency from the RTCCL pin is 0.9999817 Hz)

Oscillation frequency = 32768 × 0.9999817 ≈ 32767.4 Hz

Assume the target frequency to be 32768 Hz. Then the target correction value is calculated as follows.

Target correction value = Oscillation frequency ÷ Target frequency - 1
= 32767.4 ÷ 32768 - 1
≈ -18.3 ppm

Remark 1. The oscillation frequency is the input clock (f_{RTC}). It can be calculated from the output frequency of the RTC1HZ pin × 32768 when stops the watch error correction.
Remark 2. The target correction value is the oscillation frequency deviation (unit: [ppm]) of the crystal resonator.
Remark 3. The target frequency is the frequency resulting after watch error correction performed.

Calculating the F8 to F0 value of the watch error correction register

The F8 to F0 value of the SUBCUD register is calculated from the target correction value by using the following expression.

$$SUBCUD[8:0] = \left[\frac{\text{Target correction value [ppm]} \times 2^{20}}{10^6}\right]_{\text{Binary (9 digits)}} + 0\ 0010\ 0000\ B$$

Examples 1. When target correction value = -18.3 [ppm]

SUBCUD[8:0] = $(-18.3 \times 2^{20} / 10^6)$ Binary (9 digits) + 000100000B
= (-19.1889408) Binary (9 digits) + 000100000B
= (000010011B) 2's complement + 000100000B
= 111101101B + 000100000B
= 000001101B

Examples 2. When target correction value = 94.0 [ppm]

SUBCUD[8:0] = $(94.0 \times 2^{20} / 10^6)$ Binary (9 digits) + 000100000B
= (98.566144) Binary (9 digits) + 000100000B
= 001100011B + 000100000B
= 010000011B

CHAPTER 9 12-BIT INTERVAL TIMER

9.1 Functions of 12-bit Interval Timer

An interrupt (INTIT) is generated at any previously specified time interval. It can be utilized for wakeup from STOP mode and triggering an A/D converter's SNOOZE mode.

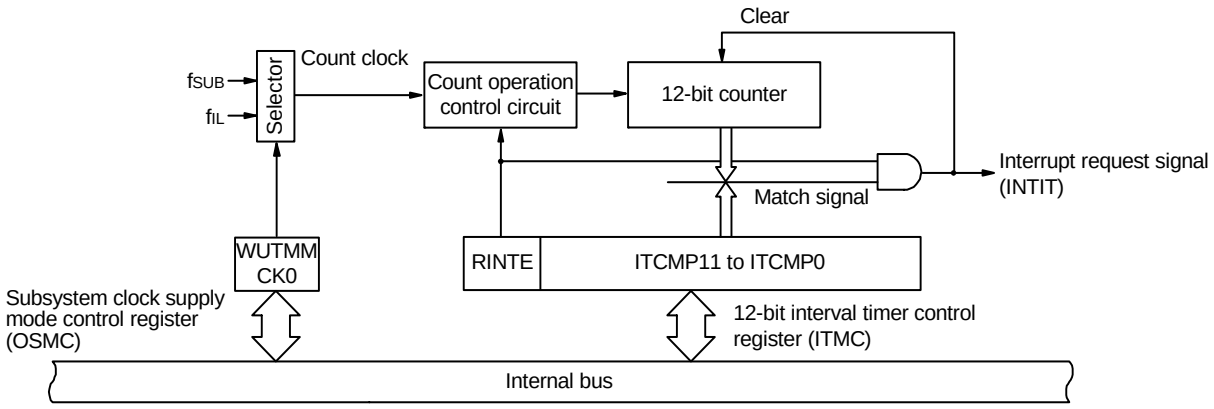
9.2 Configuration of 12-bit Interval Timer

The 12-bit interval timer includes the following hardware.

Table 9 - 1 Configuration of 12-bit Interval Timer

Item	Configuration
Counter	12-bit counter
Control registers	Peripheral enable register 1 (PER1)
	Subsystem clock supply mode control register (OSMC)
	12-bit interval timer control register (ITMC)

Figure 9 - 1 Block Diagram of 12-bit Interval Timer



9.3 Registers Controlling 12-bit Interval Timer

The 12-bit interval timer is controlled by the following registers.

- Peripheral enable register 1 (PER1)
- Subsystem clock supply mode control register (OSMC)
- 12-bit interval timer control register (ITMC)

9.3.1 Peripheral enable register 1 (PER1)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

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

Reset signal generation clears this register to 00H.

Figure 9 - 2 Format of Peripheral enable register 1 (PER1)

Address: F007AH After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	2	1	<0>
PER1	TMKAEN	0	CMPEN	TKB20EN	DTCEN	0	0	DACEN

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

Caution 1. When using the 12-bit interval timer, be sure to first set the TMKAEN bit to 1 and then set the interval timer control register (ITMC), while oscillation of the count clock (f_{RTC}) is stable. If TMKAEN = 0, writing to the registers 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)).

Caution 2. Clock supply to peripheral functions other than the real-time clock 2, 12-bit interval timer, and LCD controller/driver 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.

Caution 3. Be sure to clear bits 1, 2, and 6 to 0.

9.3.2 Subsystem clock supply mode control register (OSMC)

The WUTMMCK0 bit can be used to select the 12-bit interval timer 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 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 for real-time clock 2, 12-bit interval timer, and LCD driver/controller.	Selection of clock output from PCLBUZn pin of clock output/buzzer output
0	Subsystem clock (f_{SUB})	Selecting the subsystem clock (f_{SUB}) is enabled.
1	Low-speed on-chip oscillator clock (f_{IL})	Selecting the subsystem clock (f_{SUB}) is disabled.

Note Be sure to select the subsystem clock (WUTMMCK0 bit = 0) if the subsystem clock is oscillating.

Caution **The subsystem clock and low-speed on-chip oscillator clock can only be switched by using the WUTMMCK0 bit if the real-time clock 2, 12-bit interval timer, and LCD driver/controller are all stopped. These are stopped as follows:**

Real-time clock 2: Set the RTCE bit to 0.
12-bit interval timer: Set the RINTE bit to 0.
LCD driver/controller: Set the SCOC and VLCON bits to 0.

Remark RTCE: Bit 7 of real-time clock control register 0 (RTCC0)
RINTE: Bit 15 of the 12-bit interval timer control register (ITMC)
SCOC: Bit 6 of LCD mode register 1 (LCDM1)
VLCON: Bit 5 of LCD mode register 1 (LCDM1)

9.3.3 12-bit interval timer control register (ITMC)

This register is used to set up the starting and stopping of the 12-bit interval timer operation and to specify the timer compare value.

The ITMC register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0FFFH.

Figure 9 - 4 Format of 12-bit interval timer control register (ITMC)

Address: FFF90H After reset: 0FFFH R/W

Symbol	15	14	13	12	11 to 0
ITMC	RINTE	0	0	0	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 prohibited
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]	

Caution 1. Before changing the RINTE bit from 1 to 0, use the interrupt mask flag register to disable the INTIT interrupt servicing. When the operation starts (from 0 to 1) again, clear the TMKAIF flag, and then enable the interrupt servicing.

Caution 2. The value read from the RINTE bit is applied one count clock cycle after setting the RINTE bit.

Caution 3. When setting the RINTE bit after returned from standby mode and entering standby mode again, confirm that the written value of the RINTE bit is reflected, or wait that more than one clock of the count clock has elapsed after returned from standby mode. Then enter standby mode.

Caution 4. Only change the setting of the 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.

9.4 12-bit Interval Timer Operation

9.4.1 12-bit interval timer operation timing

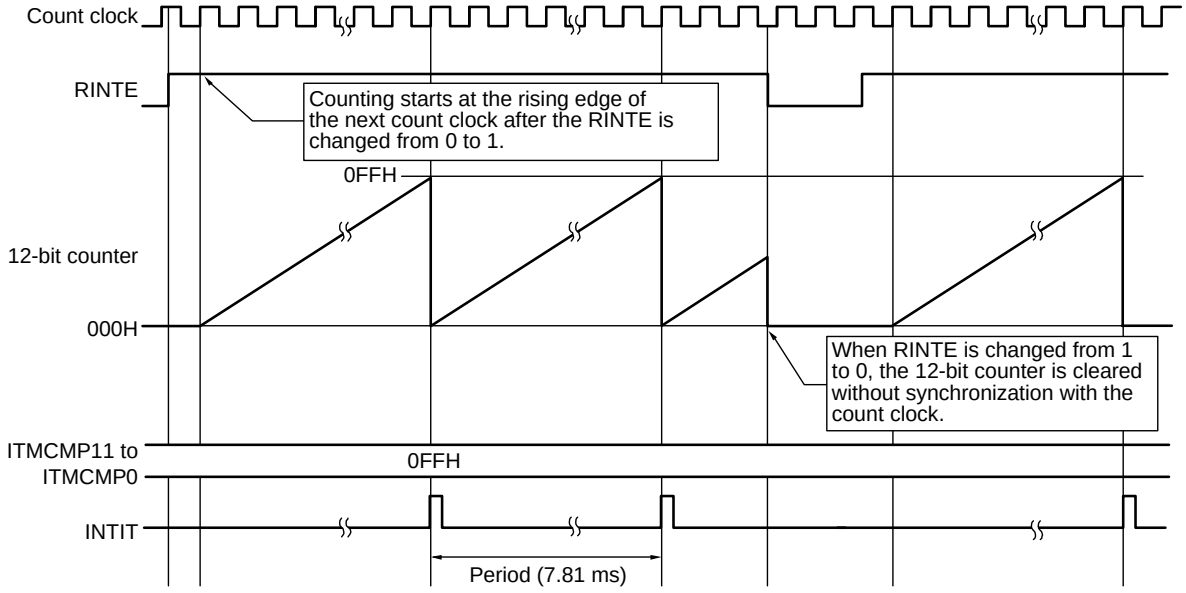
The count value specified for the ITMCMP11 to ITMCMP0 bits is used as an interval to operate an 12-bit interval timer that repeatedly generates interrupt requests (INTIT).

When the RINTE bit is set to 1, the 12-bit counter starts counting.

When the 12-bit counter value matches the value specified for the 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 9 - 5 12-bit Interval Timer Operation Timing (ITMCMP11 to ITMCMP0 = 0FFH, count clock: fSUB = 32.768 kHz)

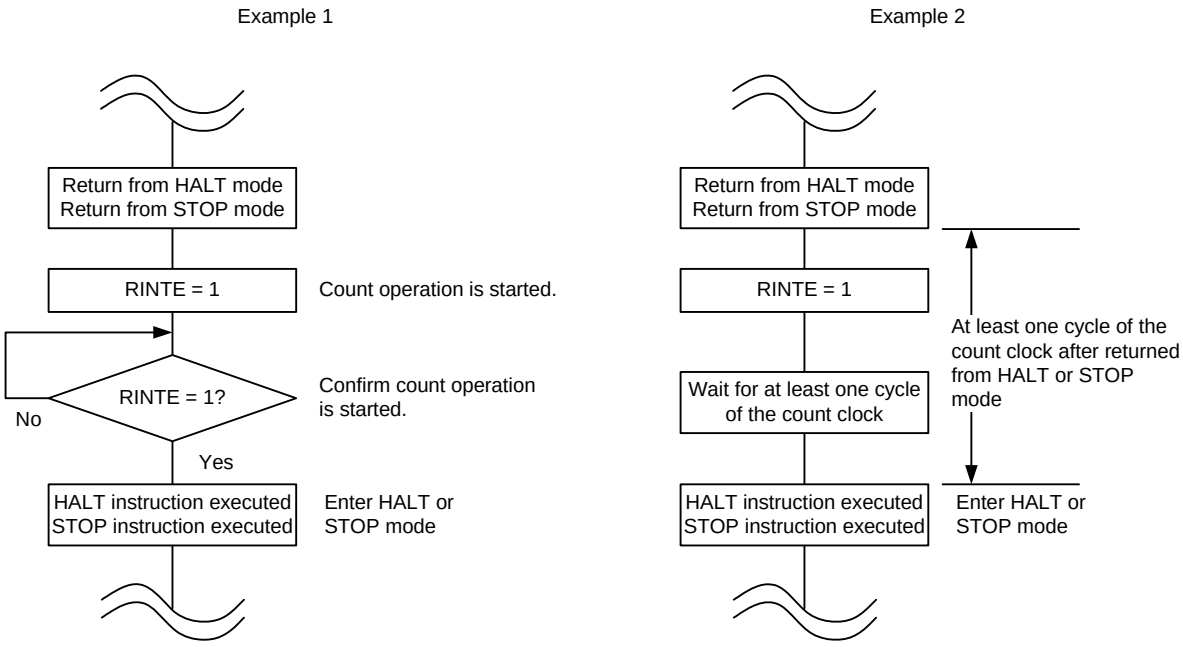


9.4.2 Start of count operation and re-enter to HALT/STOP mode after returned from HALT/STOP mode

When setting the RINTE bit after returned from HALT or STOP mode and entering HALT or STOP mode again, write 1 to the RINTE bit, and confirm the written value of the RINTE bit is reflected or wait for at least one cycle of the count clock.
Then, enter HALT or STOP mode.

- After setting RINTE to 1, confirm by polling that the RINTE bit has become 1, and then enter HALT or STOP mode (see **Example 1** in **Figure 9 - 6**).
- After setting RINTE to 1, wait for at least one cycle of the count clock and then enter HALT or STOP mode (see **Example 2** in **Figure 9 - 6**).

Figure 9 - 6 Procedure of entering to HALT or STOP mode after setting RINTE to 1



CHAPTER 10 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

10.1 Functions of Clock Output/Buzzer Output Controller

The clock output controller is intended for clock output for supply to peripheral ICs.

Buzzer output is a function to output a square wave of buzzer frequency.

One pin can be used to output a clock or buzzer sound.

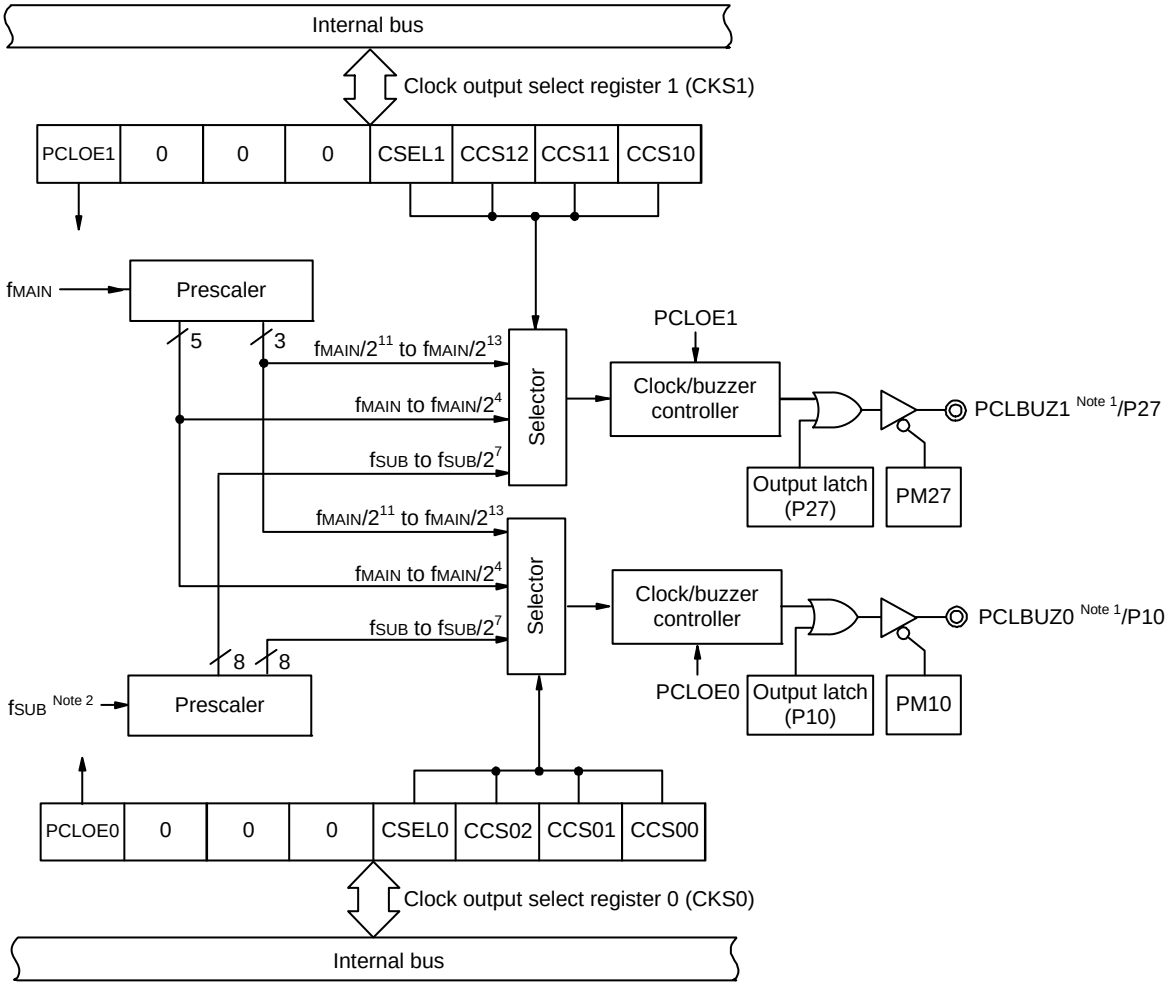
Two output pins, PCLBUZ0 and PCLBUZ1, are available.

The PCLBUZn pin outputs a clock selected by clock output select register n (CKSn).

Figure 10 - 1 shows the Block Diagram of Clock Output/Buzzer Output Controller.

Remark n = 0, 1

Figure 10 - 1 Block Diagram of Clock Output/Buzzer Output Controller



Note 1. For output frequencies available from PCLBUZ0 and PCLBUZ1, refer to **34.4** or **35.4 AC Characteristics**.

Note 2. Selecting f_{SUB} as the output clock of the clock output/buzzer output controller is prohibited when the WUTMMCK0 bit of the OSMC register is set to 1.

Remark The clock output/buzzer output pins in above diagram shows the information with PIOR2 = 0.

10.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 10 - 1 Configuration of Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	Clock output select registers n (CKSn) Port mode registers 0, 1, 2 (PM0, PM1, PM2) Port registers 0, 1, 2 (P0, P1, P2)

10.3 Registers Controlling Clock Output/Buzzer Output Controller

The following register is used to control the clock output/buzzer output controller.

- Clock output select registers n (CKSn)
- Port mode registers 0, 1, 2 (PM0, PM1, PM2)
- Port registers 0, 1, 2 (P0, P1, P2)

10.3.1 Clock output select registers n (CKSn)

These registers set output enable/disable for clock output or for the buzzer frequency output pin (PCLBUZn), and set the output clock.

Select the clock to be output from the PCLBUZn pin by using the CKSn register.

The CKSn register are set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 10 - 2 Format of Clock output select registers n (CKSn)

Address: FFFA5H (CKS0), FFFA6H (CKS1) After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CKSn	PCLOEn	0	0	0	CSELn	CCSn2	CCSn1	CCSn0

PCLOEn	PCLBUZn pin output enable/disable specification
0	Output disable (default)
1	Output enable

CSELn	CCSn2	CCSn1	CCSn0	PCLBUZn pin output clock selection				
					f_{MAIN} = 5 MHz	f_{MAIN} = 10 MHz	f_{MAIN} = 20 MHz	f_{MAIN} = 24 MHz
0	0	0	0	f_{MAIN}	5 MHz	Setting prohibited Note 1	Setting prohibited Note 1	Setting prohibited Note 1
0	0	0	1	$f_{MAIN}/2$	2.5 MHz	5 MHz	Setting prohibited Note 1	Setting prohibited Note 1
0	0	1	0	$f_{MAIN}/2^2$	1.25 MHz	2.5 MHz	5 MHz	6 MHz
0	0	1	1	$f_{MAIN}/2^3$	625 kHz	1.25 MHz	2.5 MHz	3 MHz
0	1	0	0	$f_{MAIN}/2^4$	312.5 kHz	625 kHz	1.25 MHz	1.5 MHz
0	1	0	1	$f_{MAIN}/2^{11}$	2.44 kHz	4.88 kHz	9.77 kHz	11.7 kHz
0	1	1	0	$f_{MAIN}/2^{12}$	1.22 kHz	2.44 kHz	4.88 kHz	5.86 kHz
0	1	1	1	$f_{MAIN}/2^{13}$	610 Hz	1.22 kHz	2.44 kHz	2.93 kHz
1	0	0	0	f_{SUB} Note 2	32.768 kHz			
1	0	0	1	$f_{SUB}/2$ Note 2	16.384 kHz			
1	0	1	0	$f_{SUB}/2^2$ Note 2	8.192 kHz			
1	0	1	1	$f_{SUB}/2^3$ Note 2	4.096 kHz			
1	1	0	0	$f_{SUB}/2^4$ Note 2	2.048 kHz			
1	1	0	1	$f_{SUB}/2^5$ Note 2	1.024 kHz			
1	1	1	0	$f_{SUB}/2^6$ Note 2	512 Hz			
1	1	1	1	$f_{SUB}/2^7$ Note 2	256 Hz			

Note 1. Use the output clock within a range of 8 MHz. See **34.4** or **35.4 AC Characteristics** for details.

Note 2. Selecting f_{SUB} as the output clock of the clock output/buzzer output controller is prohibited when the WUTMMCK0 bit of the OSMC register is set to 1.

Caution 1. Change the output clock after disabling clock output (PCLOEn = 0).

Caution 2. To shift to STOP mode when the main system clock is selected (CSELn = 0), set PCLOEn = 0 before executing the STOP instruction.

Remark 1. n = 0, 1

Remark 2. f_{MAIN}: Main system clock frequency
f_{SUB}: Subsystem clock frequency

10.3.2 Registers that control port functions of clock output/buzzer output pins

Using the clock output/buzzer output requires setting of the registers that control the port functions for the port pins with which the clock output/buzzer output pin functions for the target channel are multiplexed (port mode register (PMxx), port register (Pxx)). For details, see **4.3.1 Port mode registers (PMxx)** and **4.3.2 Port registers (Pxx)**.

Using a port pin which is multiplexed with a clock output/buzzer output pin function (e.g. P10/SCK20/SCL20/PCLBUZ0, P27/PCLBUZ1) for clock output/buzzer output requires setting the corresponding bits in the port mode register (PMxx) and port register (Pxx) to 0.

Example: When P10/SCK20/SCL20/PCLBUZ0 is to be used for clock output/buzzer output
Set the PM10 bit of port mode register 1 to 0.
Set the P10 bit of port register 1 to 0.

10.4 Operations of Clock Output/Buzzer Output Controller

One pin can be used to output a clock or buzzer sound.

The PCLBUZ0 pin outputs a clock/buzzer selected by clock output select register 0 (CKS0).

The PCLBUZ1 pin outputs a clock/buzzer selected by clock output select register 1 (CKS1).

10.4.1 Operation as output pin

The PCLBUZn pin is output as the following procedures.

<1> Set 0 in the bit of the port mode register (PMxx) and port register (Px) which correspond to the port which has a pin used as the PCLBUZ0 pin.

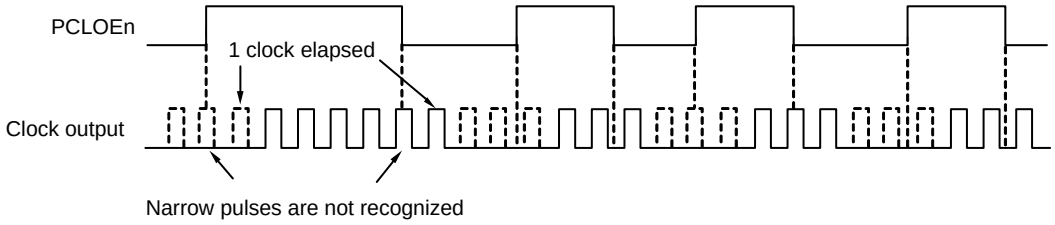
<2> Select the output frequency with bits 0 to 3 (CCSn0 to CCSn2, CSELn) of the clock output select register (CKSn) of the PCLBUZn pin (output in disabled status).

<3> Set bit 7 (PCLOEn) of the CKSn register to 1 to enable clock/buzzer output.

Remark 1. The controller used for outputting the clock starts or stops outputting the clock one clock after enabling or disabling clock output (PCLOEn bit) is switched. At this time, pulses with a narrow width are not output. Figure 10 - 3 shows enabling or stopping output using the PCLOEn bit and the timing of outputting the clock.

Remark 2. n = 0, 1

Figure 10 - 3 Timing of Outputting Clock from PCLBUZn Pin



10.5 Cautions of clock output/buzzer output controller

When the main system clock is selected for the PCLBUZn output (CSELn = 0), if STOP mode is entered within 1.5 clock cycles output from the PCLBUZn pin after the output is disabled (PCLOEn = 0), the PCLBUZn output width becomes shorter.

CHAPTER 11 WATCHDOG TIMER

11.1 Functions of Watchdog Timer

The counting operation of the watchdog timer is set by the option byte (000C0H).
The watchdog timer operates on the low-speed on-chip oscillator clock (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 24 RESET FUNCTION**.
When 75% + 1/2f_{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)

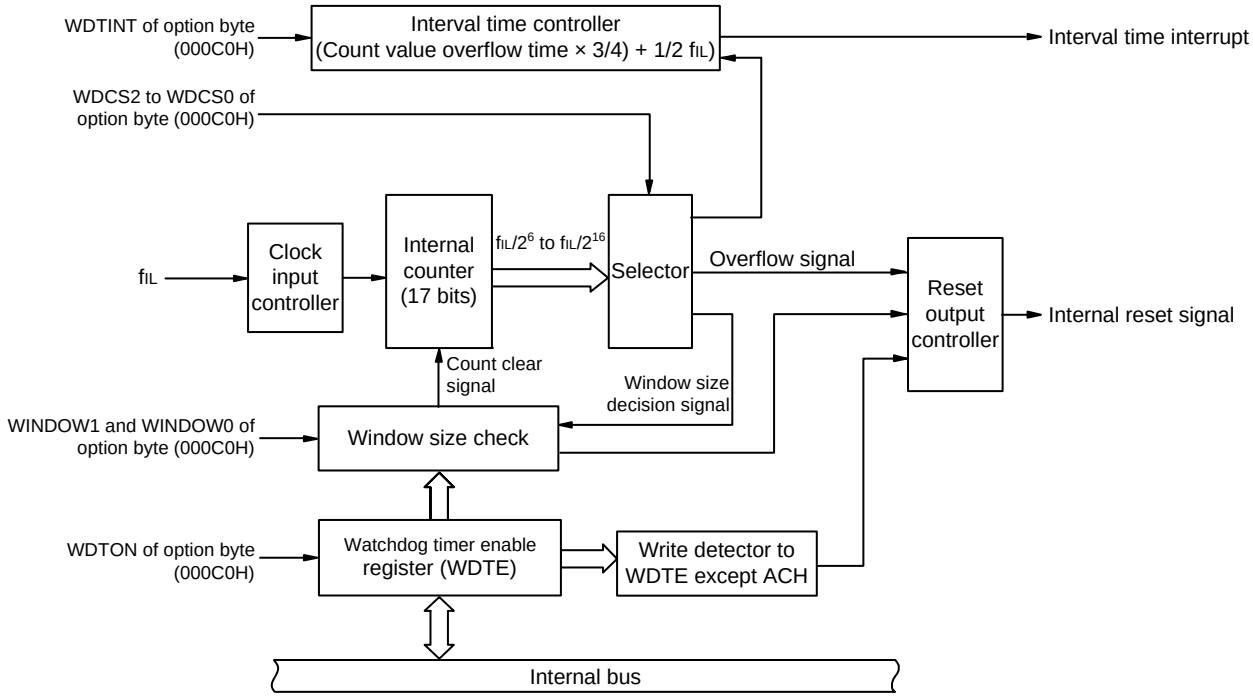
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

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 29 OPTION BYTE**.

Figure 11 - 1 Block Diagram of Watchdog Timer



Remark fIL: Low-speed on-chip oscillator clock

11.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

11.3.1 Watchdog timer enable register (WDTE)

Writing “ACH” to the WDTE register clears the watchdog timer counter and starts counting again.

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

Reset signal generation sets this register to 9AH or 1AH [Note].

Figure 11 - 2 Format of Watchdog timer enable register (WDTE)

Address: FFFABH After reset: 9AH/1AH [Note] R/W

Symbol	7	6	5	4	3	2	1	0
WDTE								

Note The WDTE register reset value differs depending on the WDTON bit setting value of the option byte (000C0H). To operate watchdog timer, set the WDTON bit to 1.

WDTON Bit Setting Value	WDTE Register Reset Value
0 (watchdog timer count operation disabled)	1AH
1 (watchdog timer count operation enabled)	9AH

Caution 1. If a value other than “ACH” is written to the WDTE register, an internal reset signal is generated.

Caution 2. If a 1-bit memory manipulation instruction is executed for the WDTE register, an internal reset signal is generated.

Caution 3. The value read from the WDTE register is 9AH/1AH (this differs from the written value (ACH)).

11.4 Operation of Watchdog Timer

11.4.1 Controlling operation of watchdog timer

1. When the watchdog timer is used, its operation is specified by the option byte (000C0H).

• Enable counting operation of the watchdog timer by setting bit 4 (WDTON) of the option byte (000C0H) to 1 (the counter starts operating after a reset release) (for details, see **CHAPTER 29 OPTION BYTE**).

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 Setting overflow time of watchdog timer** and **CHAPTER 29 OPTION BYTE**).
• 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 Setting window open period of watchdog timer** and **CHAPTER 29 OPTION BYTE**).

2. After a reset release, the watchdog timer starts counting.
3. By writing “ACH” to the watchdog timer enable register (WDTE) after the watchdog timer starts counting and before the overflow time set by the option byte, the watchdog timer is cleared and starts counting again.
4. After that, write the WDTE register the second time or later after a reset release during the window open period. If the WDTE register is written during a window close period, an internal reset signal is generated.
5. If the overflow time expires without “ACH” written to the WDTE register, an internal reset signal is generated.
   An internal reset signal is generated in the following cases.

• If a 1-bit manipulation instruction is executed on the WDTE register
• If data other than “ACH” is written to the WDTE register

Caution 1.When data is written to the watchdog timer enable register (WDTE) for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

Caution 2. After “ACH” is written to the WDTE register, an error of up to 2 clocks (f_{IL}) may occur before the watchdog timer is cleared.

Caution 3. The watchdog timer can be cleared immediately before the count value overflows.

Caution 4. The operation of the watchdog timer in the HALT and STOP 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)
1	0	1	$2^{13}/f_{IL}$ (474.89 ms) Note
1	1	0	$2^{14}/f_{IL}$ (949.79 ms) Note
1	1	1	$2^{16}/f_{IL}$ (3799.18 ms) Note

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

<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), and
- ACH is written to the WDTE register (FFFABH) when the watchdog timer counter has reached or exceeded 75% of the overflow time.

This interrupt can be masked by clearing the watchdog timer counter through steps 1 to 5 below.

1. Set the WDTIMK bit of the interrupt mask flag register 0 (MK0L) to 1 before clearing the watchdog timer counter.
2. Clear the watchdog timer counter.
3. Wait for at least 80 μs.
4. Clear the WDTIIF bit of the interrupt request flag register (IF0L) to 0.
5. Clear the WDTIMK bit of the interrupt mask flag register 0 (MK0L) to 0.

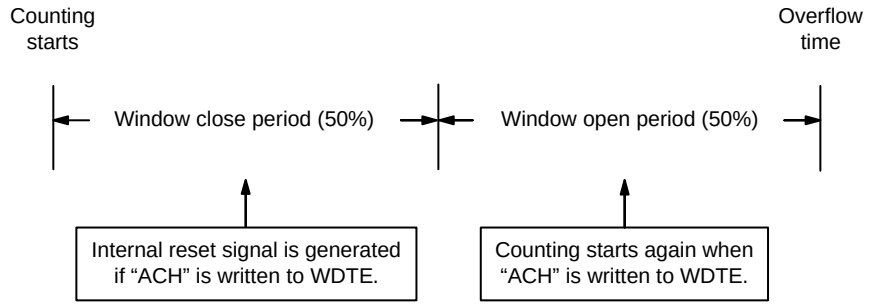
Remark f_{IL}: Low-speed on-chip oscillator clock frequency

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	$2^{13}/f_{IL}$ (474.90 ms)	237.44 ms to 321.26 ms
1	1	0	$2^{14}/f_{IL}$ (949.80 ms)	474.89 ms to 642.51 ms
1	1	1	$2^{16}/f_{IL}$ (3799.19 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.) × (1 - 0.5) = 0 to 2^9/12.75 kHz × 0.5 = 0 to 20.08 ms
- Window open time:
  $2^9/f_{IL}$ (MIN.) × (1 - 0.5) to $2^9/f_{IL}$ (MAX.) = 2^9/12.75 kHz × 0.5 to 2^9/17.25 kHz = 20.08 to 29.68 ms

11.4.4 Setting watchdog timer interval interrupt

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 not 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.

			80/85-pin (products with USB)	80/85-pin (products without USB)	100-pin (products with USB)	100-pin (products without USB)
Analog input channels		Total	9 ch	11 ch	13 ch	13 ch
	High accuracy channel	Pins based on input buffer power supply AV_{DD}	3 ch (ANI0 to ANI2)	5 ch (ANI0 to ANI2, ANI5, ANI6)	7 ch (ANI0 to ANI6)	
	Standard channel	Pins based on input buffer power supply V_{DD}	6 ch (ANI16 to ANI21)			

Remark Most of the following descriptions in this chapter use the 100-pin as an example.

12.1 Function of A/D Converter

The A/D converter converts analog input signals into digital values, and is configured to control analog inputs, including up to 13 channels of A/D converter analog inputs (ANI0 to ANI6 and ANI16 to ANI21). 12-bit resolution or 8-bit resolution can also be selected by using the ADTYP bit of A/D converter mode register 2 (ADM2).
The A/D converter has the following function.

• 12-bit or 8-bit resolution A/D conversion
12-bit or 8-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from ANI0 to ANI6 and ANI16 to ANI21. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated (when in the select mode).

Caution The valid resolution differs depending on the voltage conditions of AV_{DD} and AV_{REFP}. For details, see 34.6.1 or 35.6.1 A/D converter characteristics.

Remark When using the converter with a resolution of 10 bits, select the 12-bit resolution mode (ADTYP = 0). Use the higher 10 bits of the conversion result. Do not use the lower 2 bits.

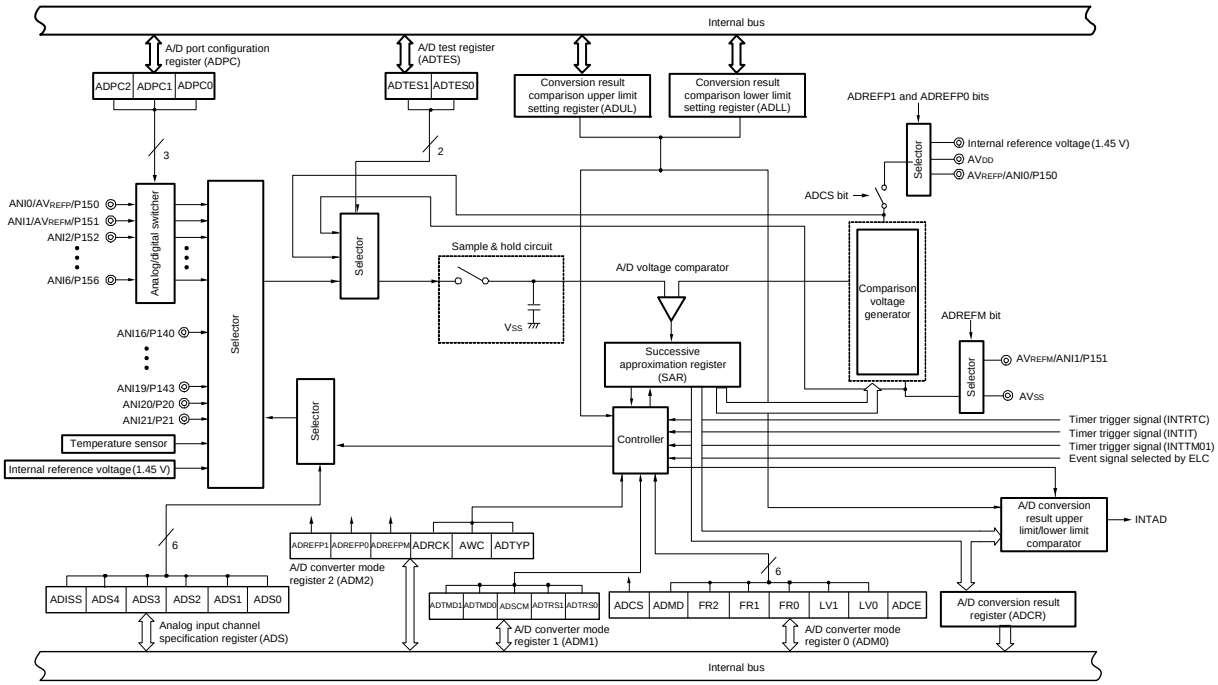
Various A/D conversion modes can be specified by using the mode combinations below.

Trigger mode	Software trigger	Conversion is started by software manipulation.
	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 A/D power supply 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 Mode Note	Number of Sampling Clock	
Normal 1	11 f_{AD}	Set a value to the number of sampling clocks, at which the sampling capacitor is fully charged, depending on the output impedance of the analog input source.
Normal 2	23 f_{AD}	
Low-voltage 1	33 f_{AD}	
Low-voltage 2	187 f_{AD}	

Note The operation modes selectable differ depending on the analog input channel, AV_{DD} voltage, trigger mode, and f_{CLK}. For details, **Tables 12 - 3** to **12 - 6 A/D Conversion Time Selection**.

Figure 12 - 1 Block Diagram of A/D Converter



Remark Analog input pin for Figure 12 - 1 when a 100-pin product is used.

12.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0 to ANI6 and ANI16 to ANI21 pins

These are the analog input pins of the 13 channels of the A/D converter. They input analog signals to be converted into digital signals. Pins other than the one selected as the analog input pin can be used as I/O port pins.

(2) Sample & hold circuit

The sample & hold circuit samples each of the analog input voltages sequentially sent from the input circuit, and sends them to the A/D voltage comparator. This circuit also holds the sampled analog input voltage during A/D conversion.

(3) A/D voltage comparator

This A/D voltage comparator compares output from the voltage tap of the comparison voltage generator with the sampled voltage value. 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 10 of the SAR register is automatically set, and the next comparison is made. The voltage tap of the comparison voltage generator is selected by the value of bit 11, to which the result has been already set.

Bit 11 = 0: (1/4 AV_{REF})
Bit 11 = 1: (3/4 AV_{REF})

The voltage tap of the comparison voltage generator and the analog input voltage are compared and bit 10 of the SAR register is manipulated according to the result of the comparison.

Analog input voltage ≥ Voltage tap of comparison voltage generator: Bit 10 = 1
Analog input voltage ≤ Voltage tap of comparison voltage generator: Bit 10 = 0

Comparison is continued like this to bit 0 of the SAR register.
When performing A/D conversion at a resolution of 8 bits, the comparison continues until bit 4 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 AV_{DD}.)

(4) Comparison voltage generator

The comparison voltage generator generates the comparison voltage input from an analog input pin.

(5) 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.

(6) 12-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 lower 12 bits (the higher 4 bits are fixed to 0).

(7) 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.

(8) 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 an interrupt request signal (INTAD) through the A/D conversion result upper limit/lower limit comparator.

(9) 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 to ANI6 and ANI16 to ANI21 are converted to digital signals based on the voltage applied between AV_{REFP} and the - side reference voltage (AV_{REFM}/AV_{SS}).

In addition to AV_{REFP}, it is possible to select AV_{DD} or the internal reference voltage (1.45 V) as the + side reference voltage of the A/D converter.

(10) 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 AV_{SS} as the - side reference voltage of the A/D converter.

12.3 Registers Controlling A/D Converter

The A/D converter uses 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)
- 12-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 2 and 14 (PMC2, PMC14)
- Port mode registers 2, 14, and 15 (PM2, PM14, PM15)

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	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

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

Caution 1. When setting the A/D converter, be sure to set the following registers first while the ADCEN bit is set to 1.
If ADCEN = 0, the values of the A/D converter control registers are cleared to their initial values and writing to them is ignored (except for port mode registers 2, 14, and 15 (PM2, PM14, PM15), port mode control registers 2 and 14 (PMC2, 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)**
- **12-bit A/D conversion result register (ADCR)**
- **8-bit A/D conversion result register (ADCRH)**
- **Analog input channel specification register (ADS)**
- **Conversion result comparison upper limit setting register (ADUL)**
- **Conversion result comparison lower limit setting register (ADLL)**
- **A/D test register (ADTES)**

Caution 2. Be sure to clear bits 6 and 1 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

Note 1. For details of the FR2 to FR0, LV1, LV0 bits, and A/D conversion, see **Tables 12 - 3** to **12 - 6 A/D Conversion Time Selection**.

Note 2. While in the software trigger mode or hardware trigger no-wait mode, the operation of the A/D voltage comparator is controlled by the ADCS and ADCE bits, and it takes stabilization wait status from the start of operation for the operation to stabilize. Therefore, when the ADCS bit is set to 1 after stabilization wait status 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. If the ADCS bit was set to 1 before the stabilization time elapsed, ignore the first conversion data.
[Stabilization wait status]
If a high-accuracy channel is selected as the analog input channel: 0.5 μs
If a test mode setting (ADTES1 bit of ADTES register = 1) is selected: 0.5 μs
If a standard channel is selected as the analog input channel: 2 μs
If a temperature sensor output voltage/internal reference voltage are selected as the analog input channel: (ADISS bit of ADS register = 1): 2 μs

Caution 1. Change the ADMD, FR2 to FR0, LV1, and LV0 bits while in the conversion stopped status (ADCS = 0, ADCE = 0).

Caution 2. Setting ADCS = 1, ADCE = 0 is prohibited.

Caution 3. Do not change the ADCE and ADCS bits from 0 to 1 at the same time by using an 8-bit manipulation instruction. Be sure to set these bits in the order described in 12.7 A/D Converter Setup Flowchart.

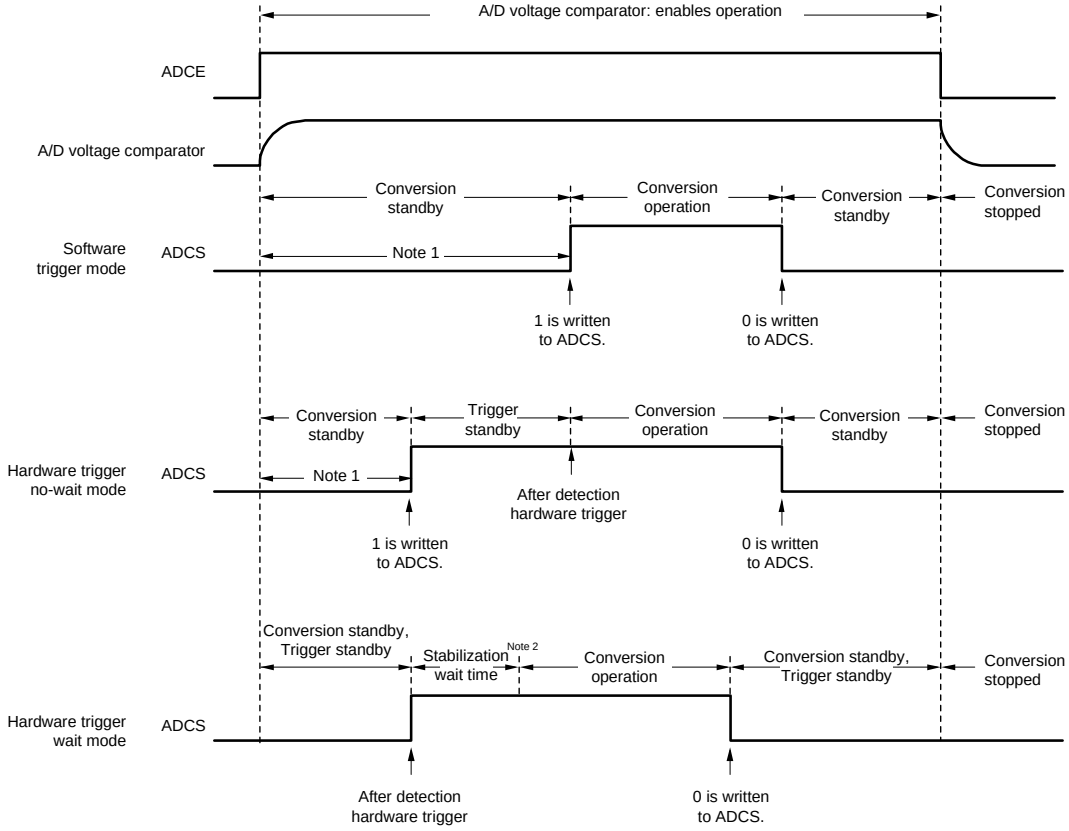
Table 12 - 1 Settings of ADCS and ADCE Bits

ADCS	ADCE	A/D Conversion Operation
0	0	Conversion stopped state
0	1	Conversion standby state
1	0	Setting prohibited
1	1	Conversion-in-progress state

Table 12 - 2 Setting and Clearing Conditions for ADCS Bit

A/D Conversion Mode			Set Conditions	Clear Conditions
Software trigger	Select mode	Sequential conversion mode	When 1 is written to ADCE and 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 1 is written to ADCE and a hardware trigger is input	When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when A/D conversion ends.
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when conversion ends on the specified four channels.

Figure 12 - 4 Timing Chart When A/D Voltage Comparator Is Used



Note 1. While in the software trigger mode or hardware trigger no-wait mode, the time from the rising of the ADCE bit to the falling of the ADCS bit must be following time or longer to stabilize the internal circuit.
[Stabilization wait status]
If a high-accuracy channel is selected as the analog input channel: 0.5 μs
If a test mode setting (ADTES1 bit of ADTES register = 1) is selected: 0.5 μs
If a standard channel is selected as the analog input channel: 2 μs
If a temperature sensor output voltage/internal reference voltage are selected as the analog input channel: (ADISS bit of ADS register = 1): 2 μs

Note 2. For the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the A/D power supply stabilization wait time do not occur after a hardware trigger is detected.

Caution 1. If using the hardware trigger wait mode, setting the ADCS bit to 1 is prohibited (but the bit is automatically switched to 1 when the hardware trigger signal is detected). However, it is possible to clear the ADCS bit to 0 to specify the A/D conversion standby status.

Caution 2. While in the one-shot conversion mode of the hardware trigger no-wait mode, the ADCS flag is not automatically cleared to 0 when A/D conversion ends. Instead, 1 is retained.

Caution 3. Only rewrite the value of the ADCE bit when ADCS = 0 (while in the conversion stopped/conversion standby status).

Caution 4. To complete A/D conversion, specify at least the following time as the hardware trigger interval:
Hardware trigger no wait mode: 2 f_{CLK} clock + conversion start time + A/D conversion time
Hardware trigger wait mode: 2 f_{CLK} clock + conversion start time + A/D power supply stabilization wait time + A/D conversion time

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 12 - 3 A/D Conversion Time Selection (1/4)
(1) 12-bit resolution mode (ADTYP = 0) When there is no stabilization wait time
(software trigger mode/hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of Conversion Clock (Number of Sampling Clock)	Conversion Clock Number (f_{CLK})	Conversion Time Selection				
									AV_{DD} = 1.6 to 3.6 V	AV_{DD} = 1.6 to 3.6 V	AV_{DD} = 1.8 to 3.6 V	AV_{DD} = 2.4 to 3.6 V	AV_{DD} = 2.7 to 3.6 V
FR2	FR1	FR0	LV1	LV0					f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 24 MHz
0	0	0	0	0	Normal 1	f_{CLK}/32	54 f_{AD} (number of sampling clock: 11 f_{AD})	1728/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	72 µs Note
0	0	1				f_{CLK}/16		864/f_{CLK}				54 µs Note	36 µs Note
0	1	0				f_{CLK}/8		432/f_{CLK}			54 µs Note	27 µs Note	18 µs Note
0	1	1				f_{CLK}/6		324/f_{CLK}			40.5 µs Note	20.25 µs Note	13.5 µs Note
1	0	0				f_{CLK}/5		270/f_{CLK}			33.75 µs Note	16.875 µs Note	11.25 µs Note
1	0	1				f_{CLK}/4		216/f_{CLK}		54 µs Note	27 µs Note	13.5 µs Note	9 µs Note
1	1	0				f_{CLK}/2		108/f_{CLK}		27 µs Note	13.5 µs Note	6.75 µs Note	4.5 µs Note
1	1	1				f_{CLK}/1		54/f_{CLK}	54 µs Note	13.5 µs Note	6.75 µs Note	3.375 µs Note	Setting prohibited
0	0	0	0	1	Normal 2	f_{CLK}/32	66 f_{AD} (number of sampling clock: 23 f_{AD})	2112/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	88 µs
0	0	1				f_{CLK}/16		1056/f_{CLK}				66 µs	44 µs
0	1	0				f_{CLK}/8		528/f_{CLK}			66 µs Note	33 µs	22 µs
0	1	1				f_{CLK}/6		396/f_{CLK}			49.5 µs Note	24.75 µs	16.5 µs
1	0	0				f_{CLK}/5		330/f_{CLK}			41.25 µs Note	20.625 µs	13.75 µs
1	0	1				f_{CLK}/4		264/f_{CLK}		66 µs Note	33 µs Note	16.5 µs	11 µs
1	1	0				f_{CLK}/2		132/f_{CLK}		33 µs Note	16.5 µs Note	8.25 µs	5.5 µs
1	1	1				f_{CLK}/1		66/f_{CLK}	66 µs Note	16.5 µs Note	8.25 µs Note	4.125 µs	Setting prohibited
0	0	0	1	0	Low voltage 1	f_{CLK}/32	76 f_{AD} (number of sampling clock: 33 f_{AD})	2432/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	101.33 µs
0	0	1				f_{CLK}/16		1216/f_{CLK}				76 µs	50.67 µs
0	1	0				f_{CLK}/8		608/f_{CLK}			76 µs	38 µs	25.33 µs
0	1	1				f_{CLK}/6		456/f_{CLK}			57 µs	28.5 µs	19 µs
1	0	0				f_{CLK}/5		380/f_{CLK}			47. 5 µs	23.75 µs	15.83 µs
1	0	1				f_{CLK}/4		304/f_{CLK}		76 µs Note	38 µs	19 µs	12.67 µs
1	1	0				f_{CLK}/2		152/f_{CLK}		38 µs Note	19 µs	9.5 µs	6.33 µs
1	1	1				f_{CLK}/1		76/f_{CLK}	76 µs Note	19 µs Note	9.5 µs	4.75 µs	Setting prohibited
0	0	0	1	1	Low voltage 2	f_{CLK}/32	230 f_{AD} (number of sampling clock: 187 f_{AD})	7360/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	306.67 µs
0	0	1				f_{CLK}/16		3680/f_{CLK}				230 µs	153.33 µs
0	1	0				f_{CLK}/8		1840/f_{CLK}			230 µs	115 µs	76.67 µs
0	1	1				f_{CLK}/6		1380/f_{CLK}			172.5 µs	86.25 µs	57.5 µs
1	0	0				f_{CLK}/5		1150/f_{CLK}			143.75 µs	71. 875 µs	47.92 µs
1	0	1				f_{CLK}/4		920/f_{CLK}		230 µs	115 µs	57.5 µs	38.33 µs
1	1	0				f_{CLK}/2		460/f_{CLK}		115 µs	57.5 µs	28.75 µs	19.17 µs
1	1	1				f_{CLK}/1		230/f_{CLK}	230 µs	57.5 µs	28.75 µs	14.375 µs	Setting prohibited

Note When using ANI16 to ANI21, setting this value is prohibited.

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 34.6.1 or 35.6.1 A/D converter characteristics.

Caution 2. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, while in the conversion stopped/conversion standby status (ADCS = 0, ADCE=0).

Caution 3. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

Caution 4. When software trigger mode/hardware trigger no-wait mode, specify the conversion time so that the following conditions are satisfied:

- **f_{AD} is used within a range of 1 to 16 MHz.**
- **When using ANI16 to ANI21, the A/D converter is used in the following range of AV_{DD}, in accordance with the settings of the LV1 and LV0 bits:**
  When LV1 = 0, LV0 = 0: Setting prohibited
  When LV1 = 0, LV0 = 1: 2.4 V ≤ AV_{DD} ≤ 3.6 V
  When LV1 = 1, LV0 = 0: 1.8 V ≤ AV_{DD} ≤ 3.6 V
  When LV1 = 1, LV0 = 1: 1.6 V ≤ AV_{DD} ≤ 3.6 V
- **If temperature sensor output voltage or internal reference voltage (ADISS bit of ADS register = 1) is specified for the analog input channel, the A/D converter is used in the following range of AV_{DD}:**
  When LV1 = 0, LV0 = 0: Setting prohibited
  When other than LV1 = 0, LV0 = 0: 2.4 V ≤ AV_{DD} ≤ 3.6 V

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 12 - 4 A/D Conversion Time Selection (2/4)

(2) 12-bit resolution mode (ADTYP = 0) When there is A/D power supply stabilization wait time (hardware trigger wait mode (except second and subsequent conversion in sequential conversion mode and conversion of channel specified by scan 1, 2, and 3 in scan mode Note 1))

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of A/D power supply stabilization wait time	Number of Conversion Clock (Number of Sampling Clock)	A/D Power Supply Stabilization Wait Time +Conversion Time	Conversion Time Selection				
										AV_{DD} = 1.6 to 3.6 V	AV_{DD} = 1.6 to 3.6 V	AV_{DD} = 1.8 to 3.6 V	AV_{DD} = 2.4 to 3.6 V	AV_{DD} = 2.7 to 3.6 V
FR2	FR1	FR0	LV1	LV0						f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 24 MHz
0	0	0	0	0	Normal 1	f_{CLK}/32	4 f_{CLK}	54 f_{AD} (number of sampling clock: 11 f_{AD})	1732/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	72.1667 µs Note 2
0	0	1				f_{CLK}/16			868/f_{CLK}				54.25 µs Note 2	36.1667 µs Note 2
0	1	0				f_{CLK}/8			436/f_{CLK}			54.5 µs Note 2	27.25 µs Note 2	18.1667 µs Note 2
0	1	1				f_{CLK}/6			328/f_{CLK}			41 µs Note 2	20.5 µs Note 2	13.6667 µs Note 2
1	0	0				f_{CLK}/5			274/f_{CLK}			34.25 µs Note 2	17.125 µs Note 2	11.4167 µs Note 2
1	0	1				f_{CLK}/4			220/f_{CLK}		55 µs Note 2	27.5 µs Note 2	13.75 µs Note 2	9.1667 µs Note 2
1	1	0				f_{CLK}/2			112/f_{CLK}		28 µs Note 2	14 µs Note 2	7 µs Note 2	4.6667 µs Note 2
1	1	1				f_{CLK}/1	2 f_{CLK}		56/f_{CLK}	56 µs Note 2	14 µs Note 2	7 µs Note 2	3.5 µs Note 2	Setting prohibited
0	0	0	0	1	Normal 2	f_{CLK}/32	58 f_{CLK}	66 f_{AD} (number of sampling clock: 23 f_{AD})	2170/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	90.4167 µs
0	0	1				f_{CLK}/16			1114/f_{CLK}				69.625 µs	46.4167 µs
0	1	0				f_{CLK}/8			586/f_{CLK}			73.25 µs Note 2	36.625 µs	24.4167 µs
0	1	1				f_{CLK}/6			454/f_{CLK}			56.75 µs Note 2	28.375 µs	18.9167 µs
1	0	0				f_{CLK}/5			388/f_{CLK}			48.5 µs Note 2	24.25 µs	16.1667 µs
1	0	1				f_{CLK}/4			322/f_{CLK}		80.ms Note 2	40.25 µs Note 2	20.125 µs	13.4167 µs
1	1	0				f_{CLK}/2			190/f_{CLK}		47.5 µs Note 2	23.75 µs Note 2	11.875 µs	7.9167 µs
1	1	1				f_{CLK}/1	29 f_{CLK}		95/f_{CLK}	95 µs Note 2	23.75 µs Note 2	11.875 µs Note 2	5.9375 µs	Setting prohibited
0	0	0	1	0	Low voltage 1	f_{CLK}/32	15 f_{CLK}	76 f_{AD} (number of sampling clock: 33 f_{AD})	2447/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	101.958 µs Note 2
0	0	1				f_{CLK}/16			1231/f_{CLK}				76.9375 µs Note 2	51.292 µs Note 2
0	1	0				f_{CLK}/8			623/f_{CLK}			77.875 µs	38.9375 µs Note 2	25.958 µs Note 2
0	1	1				f_{CLK}/6			471/f_{CLK}			58.875 µs	29.4375 µs Note 2	19.625 µs Note 2
1	0	0				f_{CLK}/5			395/f_{CLK}			49.375 µs	24.6875 µs Note 2	16.458 µs Note 2
1	0	1				f_{CLK}/4			319/f_{CLK}		79.75 µs Note 2	39.875 µs	19.9375 µs Note 2	13.292 µs Note 2
1	1	0				f_{CLK}/2			167/f_{CLK}		41.75 µs Note 2	20.875 µs	10.4375 µs Note 2	6.958 µs Note 2
1	1	1				f_{CLK}/1			91/f_{CLK}	91 µs Note 2	22.75 µs Note 2	11.375 µs	5.6875 µs Note 2	Setting prohibited
0	0	0	1	1	Low voltage 2	f_{CLK}/32	8 f_{CLK}	230 f_{AD} (number of sampling clock: 187 f_{AD})	7368/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	307 µs Note 2
0	0	1				f_{CLK}/16			3688/f_{CLK}				230.5 µs Note 2	153.67 µs Note 2
0	1	0				f_{CLK}/8			1848/f_{CLK}			231 µs Note 2	115.5 µs Note 2	77 µs Note 2
0	1	1				f_{CLK}/6			1388/f_{CLK}			173.5 µs Note 2	86.75 µs Note 2	57.83 µs Note 2
1	0	0				f_{CLK}/5			1158/f_{CLK}			144.75 µs Note 2	72.375 µs Note 2	48.25 µs Note 2
1	0	1				f_{CLK}/4			928/f_{CLK}		232 µs	116 µs Note 2	58 µs Note 2	38.67 µs Note 2
1	1	0				f_{CLK}/2			468/f_{CLK}		117 µs	58.5 µs Note 2	29.25 µs Note 2	19.5 µs Note 2
1	1	1				f_{CLK}/1			238/f_{CLK}	238 µs	59.5 µs	29.75 µs Note 2	14.875 µs Note 2	Setting prohibited

Note 1. For the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected (see **Table 12 - 3**).

Note 2. When using ANI16 to ANI21, setting this value is prohibited.

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 34.6.1 or 35.6.1 A/D converter characteristics.

Caution 2. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, while in the conversion stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

Caution 4. When hardware trigger wait mode, specify the conversion time so that the following conditions are satisfied:

- **f_{AD} is used within a range of 1 to 16 MHz.**
- **When using ANI16 to ANI21, the A/D converter is used in the following conditions:**
  When LV1 = 0, LV0 = 0: Setting prohibited
  When LV1 = 0, LV0 = 1: 2.4 V ≤ AV_{DD} ≤ 3.6 V
  When LV1 = 1, LV0 = 0: 1.8 V ≤ AV_{DD} ≤ 3.6 V, 1 MHz ≤ f_{CLK} ≤ 8 MHz
  When LV1 = 1, LV0 = 1: 1.6 V ≤ AV_{DD} ≤ 3.6 V, 1 MHz ≤ f_{CLK} ≤ 4 MHz
- **If temperature sensor output voltage or internal reference voltage (ADISS bit of ADS register = 1) is specified for the analog input channel, the A/D converter is used in the following conditions:**
  When LV1 = 0, LV0 = 0: Setting prohibited
  When LV1 = 0, LV0 = 1: 2.4 V ≤ AV_{DD} ≤ 3.6 V
  When LV1 = 1, LV0 = 0: 2.4 V ≤ AV_{DD} ≤ 3.6 V, 1 MHz ≤ f_{CLK} ≤ 8 MHz
  When LV1 = 1, LV0 = 1: 2.4 V ≤ AV_{DD} ≤ 3.6 V, 1 MHz ≤ f_{CLK} ≤ 4 MHz

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 12 - 5 A/D Conversion Time Selection (3/4)

(3) 8-bit resolution mode (ADTYP = 1) When there is no stabilization wait time (software trigger mode/hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of Conversion Clock (Number of Sampling Clock)	Conversion Clock Number (f_{CLK})	Conversion Time Selection				
									AV_{DD} = 1.6 to 3.6 V	AV_{DD} = 1.6 to 3.6 V	AV_{DD} = 1.8 to 3.6 V	AV_{DD} = 2.4 to 3.6 V	AV_{DD} = 2.7 to 3.6 V
FR2	FR1	FR0	LV1	LV0					f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 24 MHz
0	0	0	0	0	Normal 1	f_{CLK}/32	41 f_{AD} (number of sampling clock: 11 f_{AD})	1312/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	54.6667 μs Note
0	0	1				f_{CLK}/16		656/f_{CLK}				41 μs Note	27.3333 μs Note
0	1	0				f_{CLK}/8		328/f_{CLK}			41 μs Note	20.5 μs Note	13.6667 μs Note
0	1	1				f_{CLK}/6		246/f_{CLK}			30.75 μs Note	15.375 μs Note	10.25 μs Note
1	0	0				f_{CLK}/5		205/f_{CLK}			25.625 μs Note	12.8125 μs Note	8.5417 μs Note
1	0	1				f_{CLK}/4		164/f_{CLK}		41 μs Note	20.5 μs Note	10.25 μs Note	6.8333 μs Note
1	1	0				f_{CLK}/2		82/f_{CLK}		20.5 μs Note	10.25 μs Note	5.125 μs Note	3.4167 μs Note
1	1	1				f_{CLK}/1		41/f_{CLK}	41 μs Note	10.25 μs Note	5.125 μs Note	2.5625 μs Note	Setting prohibited
0	0	0	0	1	Normal 2	f_{CLK}/32	53 f_{AD} (number of sampling clock: 23 f_{AD})	1696/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	70.6667 μs
0	0	1				f_{CLK}/16		848/f_{CLK}				53 μs	35.3333 μs
0	1	0				f_{CLK}/8		424/f_{CLK}			53 μs Note	26.5 μs	17.6667 μs
0	1	1				f_{CLK}/6		318/f_{CLK}			39.75 μs Note	19.875 μs	13.25 μs
1	0	0				f_{CLK}/5		265/f_{CLK}			33.125 μs Note	16.5625 μs	11.0417 μs
1	0	1				f_{CLK}/4		212/f_{CLK}		53 μs Note	26.5 μs Note	13.25 μs	8.8333 μs
1	1	0				f_{CLK}/2		106/f_{CLK}		26.5 μs Note	13.25 μs Note	6.625 μs	4.4167 μs
1	1	1				f_{CLK}/1		53/f_{CLK}	53 μs Note	13.25 μs Note	6.625 μs Note	3.3125 μs	Setting prohibited
0	0	0	1	0	Low voltage 1	f_{CLK}/32	63 f_{AD} (number of sampling clock: 33 f_{AD})	2016/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	84.00 μs
0	0	1				f_{CLK}/16		1008/f_{CLK}				63 μs	42.00 μs
0	1	0				f_{CLK}/8		504/f_{CLK}			63 μs	31.5 μs	21.00 μs
0	1	1				f_{CLK}/6		378/f_{CLK}			47.25 μs	23.625 μs	15.75 μs
1	0	0				f_{CLK}/5		315/f_{CLK}			39.375 μs	19.6875 μs	13.13 μs
1	0	1				f_{CLK}/4		252/f_{CLK}		63 μs Note	31.5 μs	15.75 μs	10.50 μs
1	1	0				f_{CLK}/2		126/f_{CLK}		31.5 μs Note	15.75 μs	7.875 μs	5.25 μs
1	1	1				f_{CLK}/1		63/f_{CLK}	63 μs Note	15.75 μs Note	7.875 μs	3.9375 μs	Setting prohibited
0	0	0	1	1	Low voltage 2	f_{CLK}/32	217 f_{AD} (number of sampling clock: 187 f_{AD})	6944/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	289.33 μs
0	0	1				f_{CLK}/16		3472/f_{CLK}				217 μs	144.67 μs
0	1	0				f_{CLK}/8		1736/f_{CLK}			217 μs	108.5 μs	72.33 μs
0	1	1				f_{CLK}/6		1302/f_{CLK}			162.75 μs	81.375 μs	54.25 μs
1	0	0				f_{CLK}/5		1085/f_{CLK}			135.625 μs	67.8125 μs	45.21 μs
1	0	1				f_{CLK}/4		868/f_{CLK}		217 μs	108.5 μs	54.25 μs	36.17 μs
1	1	0				f_{CLK}/2		434/f_{CLK}		108.5 μs	54.25 μs	27.125 μs	18.08 μs
1	1	1				f_{CLK}/1		217/f_{CLK}	217 μs	54.25 μs	27.125 μs	13.5625 μs	Setting prohibited

Note When using ANI16 to ANI21, setting this value is prohibited.

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 34.6.1 or 35.6.1 A/D converter characteristics.

Caution 2. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, while in the conversion stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

Caution 4. When software trigger mode/hardware trigger no-wait mode, specify the conversion time so that the following conditions are satisfied:

- **f_{AD} is used within a range of 1 to 16 MHz.**
- **When using ANI16 to ANI21, the A/D converter is used in the following range of AV_{DD}, in accordance with the settings of the LV1 and LV0 bits:**
  When LV1 = 0, LV0 = 0: Setting prohibited
  When LV1 = 0, LV0 = 1: 2.4 V ≤ AV_{DD} ≤ 3.6 V
  When LV1 = 1, LV0 = 0: 1.8 V ≤ AV_{DD} ≤ 3.6 V
  When LV1 = 1, LV0 = 1: 1.6 V ≤ AV_{DD} ≤ 3.6 V
- **If temperature sensor output voltage or internal reference voltage (ADISS bit of ADS register = 1) is specified for the analog input channel, the A/D converter is used in the following range of AV_{DD}:**
  When LV1 = 0, LV0 = 0: Setting prohibited
  When other than LV1 = 0, LV0 = 0: 2.4 V ≤ AV_{DD} ≤ 3.6 V

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 12 - 6 A/D Conversion Time Selection (4/4)

(4) 8-bit resolution mode (ADTYP = 1) When there is A/D power supply stabilization wait time

(hardware trigger wait mode (except second and subsequent conversion in sequential conversion mode and conversion of channel specified by scan 1, 2, and 3 in scan mode Note 1))

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of A/D power supply stabilization wait time	Number of Conversion Clock (Number of Sampling Clock)	A/D Power Supply Stabilization Wait Time +Conversion Time	Conversion Time Selection				
										AV_{DD} = 1.6 to 3.6 V	AV_{DD} = 1.6 to 3.6 V	AV_{DD} = 1.8 to 3.6 V	AV_{DD} = 2.4 to 3.6 V	AV_{DD} = 2.7 to 3.6 V
FR2	FR1	FR0	LV1	LV0						f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 24 MHz
0	0	0	0	0	Normal 1	f_{CLK}/32	4 f_{CLK}	41 f_{AD} (number of sampling clock: 11 f_{AD})	1316/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	54.8333 µs Note 2
0	0	1				f_{CLK}/16			660/f_{CLK}				41.25 µs Note 2	27.5000 µs Note 2
0	1	0				f_{CLK}/8			332/f_{CLK}			41.5 µs Note 2	20.75 µs Note 2	13.8333 µs Note 2
0	1	1				f_{CLK}/6			250/f_{CLK}			31.25 µs Note 2	15.625 µs Note 2	10.4167 µs Note 2
1	0	0				f_{CLK}/5			209/f_{CLK}			26.125 µs Note 2	13.0625 µs Note 2	8.7083 µs Note 2
1	0	1				f_{CLK}/4			168/f_{CLK}		42 µs Note 2	21 µs Note 2	10.5 µs Note 2	7.0000 µs Note 2
1	1	0				f_{CLK}/2			86/f_{CLK}		21.5 µs Note 2	10.75 µs Note 2	5.375 µs Note 2	3.5833 µs Note 2
1	1	1				f_{CLK}/1	2 f_{CLK}		43/f_{CLK}	43 µs Note 2	10.75 µs Note 2	5.375 µs Note 2	2.6875 µs Note 2	Setting prohibited
0	0	0	0	1	Normal 2	f_{CLK}/32	58 f_{CLK}	53 f_{AD} (number of sampling clock: 23 f_{AD})	1754/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	73.0833 µs Note 2
0	0	1				f_{CLK}/16			906/f_{CLK}				56.625 µs	37.7500 µs Note 2
0	1	0				f_{CLK}/8			482/f_{CLK}			60.25 µs Note 2	30.125 µs	20.0833 µs Note 2
0	1	1				f_{CLK}/6			376/f_{CLK}			47 µs Note 2	23.5 µs	15.6667 µs Note 2
1	0	0				f_{CLK}/5			323/f_{CLK}			40.375 µs Note 2	20.1875 µs	13.4583 µs Note 2
1	0	1				f_{CLK}/4			270/f_{CLK}		67.5 µs Note 2	33.75 µs Note 2	16.875 µs	11.2500 µs Note 2
1	1	0				f_{CLK}/2			164/f_{CLK}		41 µs Note 2	20.5 µs Note 2	10.25 µs	6.8333 µs Note 2
1	1	1				f_{CLK}/1	29 f_{CLK}		82/f_{CLK}	82 µs Note 2	20.5 µs Note 2	10.25 µs Note 2	5.125 µs	Setting prohibited
0	0	0	1	0	Low voltage 1	f_{CLK}/32	15 f_{CLK}	63 f_{AD} (number of sampling clock: 33 f_{AD})	2031/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	84.625 µs Note 2
0	0	1				f_{CLK}/16			1023/f_{CLK}				63.9375 µs Note 2	42.625 µs Note 2
0	1	0				f_{CLK}/8			519/f_{CLK}			64.875 µs	32.4375 µs Note 2	21.625 µs Note 2
0	1	1				f_{CLK}/6			393/f_{CLK}			49.125 µs	24.5625 µs Note 2	16.375 µs Note 2
1	0	0				f_{CLK}/5			330/f_{CLK}			41.25 µs	20.625 µs Note 2	13.75 µs Note 2
1	0	1				f_{CLK}/4			267/f_{CLK}		66.75 µs Note 2	33.375 µs	16.6875 µs Note 2	11.125 µs Note 2
1	1	0				f_{CLK}/2			141/f_{CLK}		35.25 µs Note 2	17.625 µs	8.8125 µs Note 2	5.875 µs Note 2
1	1	1				f_{CLK}/1			78/f_{CLK}	78 µs Note 2	19.5 µs Note 2	9.75 µs	4.875 µs Note 2	Setting prohibited
0	0	0	1	1	Low voltage 2	f_{CLK}/32	8 f_{CLK}	217 f_{AD} (number of sampling clock: 187 f_{AD})	6952/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	289.67 µs Note 2
0	0	1				f_{CLK}/16			3480/f_{CLK}				217.5 µs Note 2	145 µs Note 2
0	1	0				f_{CLK}/8			1744/f_{CLK}			218 µs Note 2	109 µs Note 2	72.67 µs Note 2
0	1	1				f_{CLK}/6			1310/f_{CLK}			163.75 µs Note 2	81.875 µs Note 2	54.58 µs Note 2
1	0	0				f_{CLK}/5			1093/f_{CLK}			136.625 µs Note 2	68.3125 µs Note 2	45.54 µs Note 2
1	0	1				f_{CLK}/4			876/f_{CLK}		219 µs	109.5 µs Note 2	54.75 µs Note 2	36.5 µs Note 2
1	1	0				f_{CLK}/2			442/f_{CLK}		110.5 µs	55.25 µs Note 2	27.625 µs Note 2	18.42 µs Note 2
1	1	1				f_{CLK}/1			225/f_{CLK}	225 µs	56.25 µs	28.125 µs Note 2	14.0625 µs Note 2	Setting prohibited

Note 1. For the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected (see **Table 12 - 5**).

Note 2. When using ANI16 to ANI21, setting this value is prohibited.

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 34.6.1 or 35.6.1 A/D converter characteristics.

Caution 2. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, while in the conversion stopped (ADCS = 0, ADCE = 0).

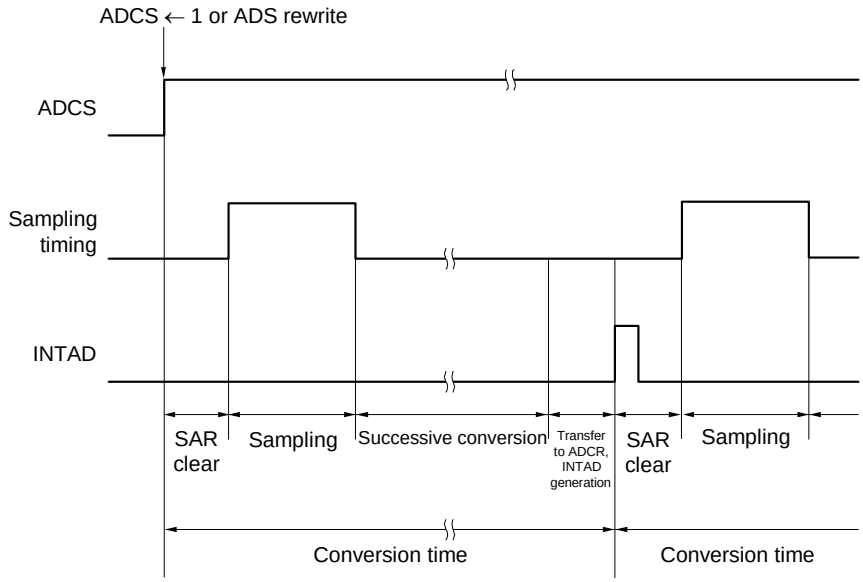
Caution 3. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

Caution 4. When hardware trigger wait mode, specify the conversion time so that the following conditions are satisfied:

- **f_{AD} is used within a range of 1 to 16 MHz.**
- **When using ANI16 to ANI21, the A/D converter is used in the following conditions:**
  When LV1 = 0, LV0 = 0: Setting prohibited
  When LV1 = 0, LV0 = 1: 2.4 V ≤ AV_{DD} ≤ 3.6 V
  When LV1 = 1, LV0 = 0: 1.8 V ≤ AV_{DD} ≤ 3.6 V, 1 MHz ≤ f_{CLK} ≤ 8 MHz
  When LV1 = 1, LV0 = 1: 1.6 V ≤ AV_{DD} ≤ 3.6 V, 1 MHz ≤ f_{CLK} ≤ 4 MHz
- **If temperature sensor output voltage or internal reference voltage (ADISS bit of ADS register = 1) is specified for the analog input channel, the A/D converter is used in the following conditions:**
  When LV1 = 0, LV0 = 0: Setting prohibited
  When LV1 = 0, LV0 = 1: 2.4 V ≤ AV_{DD} ≤ 3.6 V
  When LV1 = 1, LV0 = 0: 2.4 V ≤ AV_{DD} ≤ 3.6 V, 1 MHz ≤ f_{CLK} ≤ 8 MHz
  When LV1 = 1, LV0 = 1: 2.4 V ≤ AV_{DD} ≤ 3.6 V, 1 MHz ≤ f_{CLK} ≤ 4 MHz

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	Event signal selected by ELC
1	0	Real-time clock interrupt signal (INTRTC)
1	1	12-bit interval timer interrupt signal (INTIT)

Caution 1. Only rewrite the value of the ADM1 register while conversion operation is stopped (which is indicated by the ADCS and ADCE bits of A/D converter mode register 0 (ADM0) being 0).

Caution 2. To complete A/D conversion, specify at least the following time as the hardware trigger interval:
Hardware trigger no wait mode: 2 f_{CLK} clock + conversion start time + A/D conversion time
Hardware trigger wait mode: 2 f_{CLK} clock + conversion start time + A/D power supply stabilization wait time + A/D conversion time

Caution 3. In modes other than SNOOZE function, 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.

Remark 1. ×: don't care

Remark 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 A/D converter reference voltage, check the upper limit and lower limit A/D conversion result values, select the resolution, and specify whether to use the SNOOZE mode.

The ADM2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 12 - 7 Format of A/D converter mode register 2 (ADM2) (1/2)

Address: F0010H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

ADREFP1	ADREFP0	Selection of the + side reference voltage source of the A/D converter
0	0	Supplied from AV_{DD}
0	1	Supplied from P150/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) Stabilization wait time (A)
(4) Set ADCE = 1
(5) Stabilization wait time (B)

The stabilization wait time indicated by (3) is required when the value of the ADREFP1 and ADREFP0 bits is changed.
When ADREFP1 and ADREFP0 are changed to 1 and 0: A = 10 μs
When ADREFP1 and ADREFP0 are changed to 0 and 0 or 0 and 1: A = 1 μs
The stabilization wait time indicated by (5) is required when the value of the ADCE bit is changed to 1.
If a high-accuracy channel is selected as the analog input channel: 0.5 μs
If a test mode setting (ADTES1 bit of ADTES register = 1) is selected: 0.5 μs
If a standard channel is selected as the analog input channel: 2 μs
If a temperature sensor output voltage/internal reference voltage are selected as the analog input channel: (ADISS bit of ADS register = 1): 2 μs
After (5) stabilization time, start the A/D conversion.

• 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.
Be sure to perform A/D conversion while ADISS = 0.

Note This setting can be used only in HS (high-speed main) mode. For detail, see **Figure 29 - 4 Format of User Option Byte (000C2H/010C2H)**.

Caution 1. Rewrite the value of the ADM2 register while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 2. Do not set the ADREFP1 bit to 1 when shifting to STOP mode, or to HALT mode while the CPU is operating on the subsystem clock. Also, if the ADREFP1 bit is set to 1, the A/D converter reference voltage current (I_{ADREF}) indicated in 34.3.2 or 35.3.2 Supply current characteristics will be added to the current consumption when shifting to HALT mode while the CPU is operating on the main system clock.

Caution 3. When using AV_{REFP} and AV_{REFM}, specify ANI0 and ANI1 as the analog input channels and specify input mode by using the port mode register.

Figure 12 - 8 Format of A/D converter mode register 2 (ADM2) (2/2)

Address: F0010H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

ADREFM	Selection of the - side reference voltage source of the A/D converter
0	Supplied from AVSS
1	Supplied from P151/AVREFM/ANI1

ADRCK	Checking the upper limit and lower limit conversion result values
0	The A/D conversion end interrupt request signal (INTAD) is output when the ADLL register ≤ the ADCR register ≤ the ADUL register (AREA1).
1	The A/D conversion end interrupt request signal (INTAD) is output when the ADCR register < the ADLL register (AREA2) or the ADUL register < the ADCR register (AREA3).
Figure 12 - 9 shows the generation range of the A/D conversion end interrupt request signal (INTAD) for AREA1 to AREA3.	

AWC	Specification of the SNOOZE mode
0	Do not use the SNOOZE mode function.
1	Use the SNOOZE mode function.
When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and A/D conversion is performed without operating the CPU (the SNOOZE mode). • The SNOOZE mode function can only be specified when the high-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (fCLK). If any other clock is selected, specifying this mode is prohibited. • Using the SNOOZE mode function in the software trigger mode or hardware trigger no-wait mode is prohibited. • Using the SNOOZE mode function in the sequential conversion mode is prohibited. • When using the SNOOZE mode function, specify a hardware trigger interval of at least “shift time to SNOOZE mode Note 1 + A/D power supply stabilization wait time + A/D conversion time +2 fCLK 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 or normal operation.	

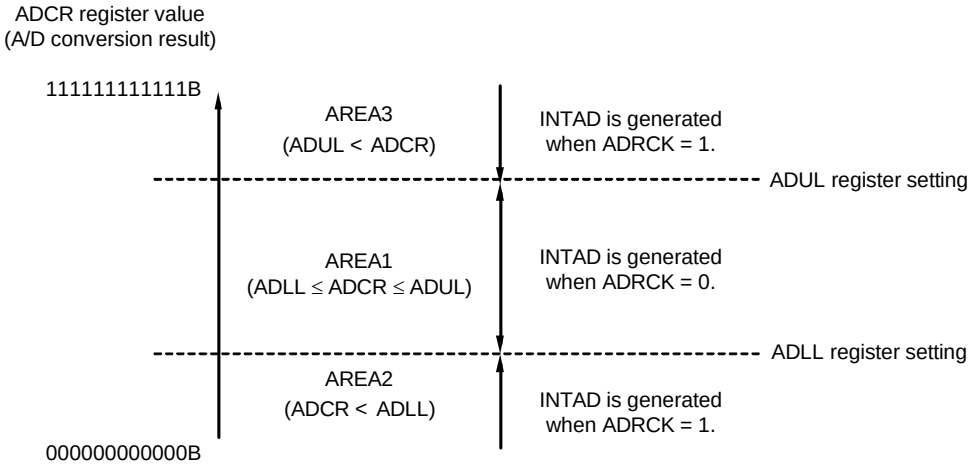
ADTYP	Selection of the A/D conversion resolution
0	12-bit resolution Note 2
1	8-bit resolution

Note 1. Refer to “Transition time from STOP mode to SNOOZE mode:” in **23.3.3 SNOOZE mode**.

Note 2. The valid resolution differs depending on the voltage conditions of AVDD and AVREFP. For details, see **34.6.1** or **35.6.1 A/D converter characteristics**.

Caution Rewrite the value of the ADM2 register while conversion operation is stopped (ADCS = 0, ADCE = 0).

Figure 12 - 9 ADRCK Bit Interrupt Signal Generation Range



Remark If INTAD does not occur, the A/D conversion result is not stored in the ADCR or ADCRH register.

12.3.5 12-bit A/D conversion result register (ADCR)

This register is a 16-bit register that stores the A/D conversion result. The higher 4 bits are fixed to 0. Each time A/D conversion ends, each time A/D conversion ends, the value of ADSAR[11:0] is stored in the A/D conversion result register (note that whether to store this value is determined by the setting of the ADRCK bit of the ADM2 register and by the settings of the ADUL and ADLL registers). The higher 4 bits of the conversion result are stored in FFF1FH and the lower 8 bits are stored in the lower 4 bits of FFF1EH Note.

The ADCR register can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (the value specified by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see **Figure 12 - 9**), the result is not stored.

Caution The valid resolution differs depending on the voltage conditions of AV_{DD} and AV_{REFP}. For details, see 34.6.1 or 35.6.1 A/D converter characteristics.

Remark 1. When using the converter with a resolution of 10 bits, select the 12-bit resolution mode (ADTYP = 0). Use the higher 10 bits of the conversion result. Do not use the lower 2 bits.

Remark 2. When using the converter with a resolution of 8 bits, select the 8-bit resolution mode (ADTYP = 1). Do not use the lower 4 bits of the ADCR register.
The higher 8 bits of the conversion result can be read by using the ADCRH register.

Figure 12 - 10 Format of 12-bit A/D conversion result register (ADCR)

Address: FFF1EH, FFF1FH After reset: 0000H R

Symbol	FFF1FH								FFF1EH							
ADCR	0	0	0	0												

Caution 1. 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 ADCR 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.

Caution 2. If INTAD does not occur, the A/D conversion result is not stored in the ADCR register.

12.3.6 8-bit A/D conversion result register (ADCRH)

This register is an 8-bit register that indicates bits [11:4] of the ADCR register. The higher 8 bits of 12-bit resolution are stored Note.

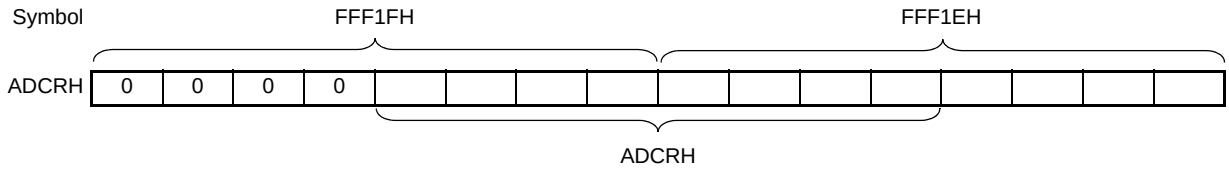
The ADCRH register can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (the value specified by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see **Figure 12 - 9**), the result is not stored.

Figure 12 - 11 Format of 8-bit A/D conversion result register (ADCRH)

Address: FFF1FH Note After reset: 00H R



Note The ADCRH data (the lower 4 bits of FFF1FH + the higher 4 bits of FFF1EH) is to be read as a FFF1FH address.

Caution 1. 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.

Caution 2. If INTAD does not occur, the A/D conversion result is not stored in the ADCRH register.

12.3.7 Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.

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

Reset signal generation clears this register to 00H.

Figure 12 - 12 Format of Analog input channel specification register (ADS) (1/2)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

• Select mode (ADMD = 0)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P150/ANI0/AVREFP pin
0	0	0	0	0	1	ANI1	P151/ANI1/AVREFM pin
0	0	0	0	1	0	ANI2	P152/ANI2 pin
0	0	0	0	1	1	ANI3	P153/ANI3 pin
0	0	0	1	0	0	ANI4	P154/ANI4 pin
0	0	0	1	0	1	ANI5	P155/ANI5 pin
0	0	0	1	1	0	ANI6	P156/ANI6 pin
0	1	0	0	0	0	ANI16	P140/ANI16 pin
0	1	0	0	0	1	ANI17	P141/ANI17 pin
0	1	0	0	1	0	ANI18	P142/ANI18 pin
0	1	0	0	1	1	ANI19	P143/ANI19 pin
0	1	0	1	0	0	ANI20	P20/ANI20 pin
0	1	0	1	0	1	ANI21	P21/ANI21 pin
1	0	0	0	0	0	—	Temperature sensor output Note
1	0	0	0	0	1	—	Internal reference voltage output (1.45 V) Note
Other than the above						Setting prohibited	

Note This setting can be used only in HS (high-speed main) mode. For detail, see **Figure 29 - 4 Format of User Option Byte (000C2H/010C2H)**.

• Scan mode (ADMD = 1)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel			
						Scan 0	Scan 1	Scan 2	Scan 3
0	0	0	0	0	0	ANI0	ANI1	ANI2	ANI3
0	0	0	0	0	1	ANI1	ANI2	ANI3	ANI4
0	0	0	0	1	0	ANI2	ANI3	ANI4	ANI5
0	0	0	0	1	1	ANI3	ANI4	ANI5	ANI6
0	1	0	0	0	0	ANI16	ANI17	ANI18	ANI19
0	1	0	0	0	1	ANI17	ANI18	ANI19	ANI20
0	1	0	0	1	0	ANI18	ANI19	ANI20	ANI21
Other than the above						Setting prohibited			

(**Cautions** are listed on the next page.)

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

Caution 2. Set a channel to be set the analog input by ADPC and PMCx registers in the input mode by using Port mode registers 2, 14, and 15 (PM2, PM14, PM15).

Caution 3. Do not set the pin that is set by the A/D port configuration register (ADPC) as digital I/O by the ADS register.

Caution 4. Do not set the pin that is set by Port mode control registers 2 and 14 (PMC2, PMC14) as digital I/O by the ADS register.

Caution 5. Rewrite the value of the ADISS bit while conversion operation is stopped (ADCS = 0, ADCE = 0).

Caution 6. If using AV_{REFP} as the + side reference voltage source of the A/D converter, do not select ANI0 as an A/D conversion channel.

Caution 7. If using AV_{REFM} as the - side reference voltage source of the A/D converter, do not select ANI1 as an A/D conversion channel.

Caution 8. If ADISS is set to 1, the internal reference voltage (1.45 V) cannot be used for the + side reference voltage source. Also, after setting the ADISS to 1, the result of the first conversion cannot be used. For details about the setting flow, see 12.7.4 Setup when temperature sensor output voltage/internal reference voltage is selected (example for software trigger mode and one-shot conversion mode).

Caution 9. Do not set the ADISS bit to 1 when shifting to STOP mode, or to HALT mode while the CPU is operating on the subsystem clock. Also, if the ADISS bit is set to 1, the temperature sensor operating current indicated in 34.3.2 Supply current characteristics (I_{TMPS}) will be added to the current consumption when shifting to HALT mode while the CPU is operating on the main system clock.

Caution 10. Ignore the conversion result if the corresponding ANI pin does not exist in the product used.

12.3.8 Conversion result comparison upper limit setting register (ADUL)

This register is used to specify the setting for checking the upper limit of the A/D conversion results.

The A/D conversion results and ADUL register value are compared, and interrupt signal (INTAD) generation is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in **Figure 12 - 9**).

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

Reset signal generation sets this register to FFH.

Caution 1. When 12-bit resolution A/D conversion is selected, the higher 8 bits of the 12-bit A/D conversion result register (ADCR) are compared with the ADUL register.

Caution 2. Only rewrite the value of the ADUL register and ADLL register while conversion operation is stopped (which is indicated by the ADCE bit of A/D converter mode register 0 (ADM0) being 0).

Caution 3. Make sure that ADUL > ADLL when setting these registers.

Figure 12 - 13 Format of Conversion result comparison upper limit setting register (ADUL)

Address: F0011H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
ADUL	ADUL7	ADUL6	ADUL5	ADUL4	ADUL3	ADUL2	ADUL1	ADUL0

12.3.9 Conversion result comparison lower limit setting register (ADLL)

This register is used to specify the setting for checking the lower limit of the A/D conversion results.

The A/D conversion results and ADLL register value are compared, and interrupt signal (INTAD) generation is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in **Figure 12 - 9**).

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

Reset signal generation clears this register to 00H.

Figure 12 - 14 Format of Conversion result comparison lower limit setting register (ADLL)

Address: F0012H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADLL	ADLL7	ADLL6	ADLL5	ADLL4	ADLL3	ADLL2	ADLL1	ADLL0

Caution 1. When 12-bit resolution A/D conversion is selected, the higher 8 bits of the 12-bit A/D conversion result register (ADCR) are compared with the ADLL register.

Caution 2. Only rewrite the value of the ADUL register and ADLL register while conversion operation is stopped (which is indicated by the ADCE bit of A/D converter mode register 0 (ADM0) being 0).

Caution 3. Make sure that ADUL > ADLL when setting these registers.

12.3.10 A/D test register (ADTES)

This register is used to select the + side reference voltage or - side reference voltage for the A/D converter, an analog input channel (ANIxx), the temperature sensor output voltage, or the internal reference voltage (1.45 V) as the target for A/D conversion. When using as the A/D test function, 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 - 15 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	- side reference voltage (setting at ADREFM bit of ADM2 register)
1	1	+ side reference voltage (setting at ADREFP1, ADREFP0 bits of the ADM2 register)
Other than the above		Setting prohibited

Note Temperature sensor output/internal reference voltage (1.45 V) can be used only in HS (high-speed main) mode. For detail, see **Figure 29 - 4 Format of User Option Byte (000C2H/010C2H)**.

Caution For details on the A/D test function, refer to CHAPTER 27 SAFETY FUNCTIONS.

12.3.11 Registers that control port functions of analog input pins

Set the registers that control the port functions for the port pins with which the analog input pin functions of the A/D converter are multiplexed (port mode registers (PMxx), port mode control registers (PMCxx), and A/D port configuration register (ADPC)).

For details, see as follows.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.6 Port mode control registers (PMCxx)**
- **4.3.7 A/D port configuration register (ADPC)**

When using the ANI0 to ANI6 pins for analog input of the A/D converter, set the corresponding bit in the port mode register (PMxx) to 1 and select analog input through the A/D port configuration register (ADPC).

When using the ANI16 to ANI21 pins for analog input of the A/D converter, set the corresponding bits in the port mode register (PMxx) and port mode control register (PMCxx) 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 11 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 10 of the SAR register is automatically set to 1, and the operation proceeds to the next comparison.

The series resistor string voltage tap is selected according to the preset value of bit 11, as described below.

- Bit 11 = 1: (3/4) AV_{REF}
- Bit 11 = 0: (1/4) AV_{REF}

The voltage tap and sampled voltage are compared and bit 10 of the SAR register is manipulated as follows.

- Sampled voltage ≥ Voltage tap: Bit 10 = 1
- Sampled voltage < Voltage tap: Bit 10 = 0

<6> Comparison is continued in this way up to bit 0 of the SAR register.

<7> Upon completion of the comparison of 12 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.

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.

Note 1. If the A/D conversion result is outside the A/D conversion result range specified by the ADRCK bit and the ADUL and ADLL registers (see **Figure 12 - 9**), the A/D conversion result interrupt request signal is not generated and no A/D conversion results are stored in the ADCR and ADCRH registers.

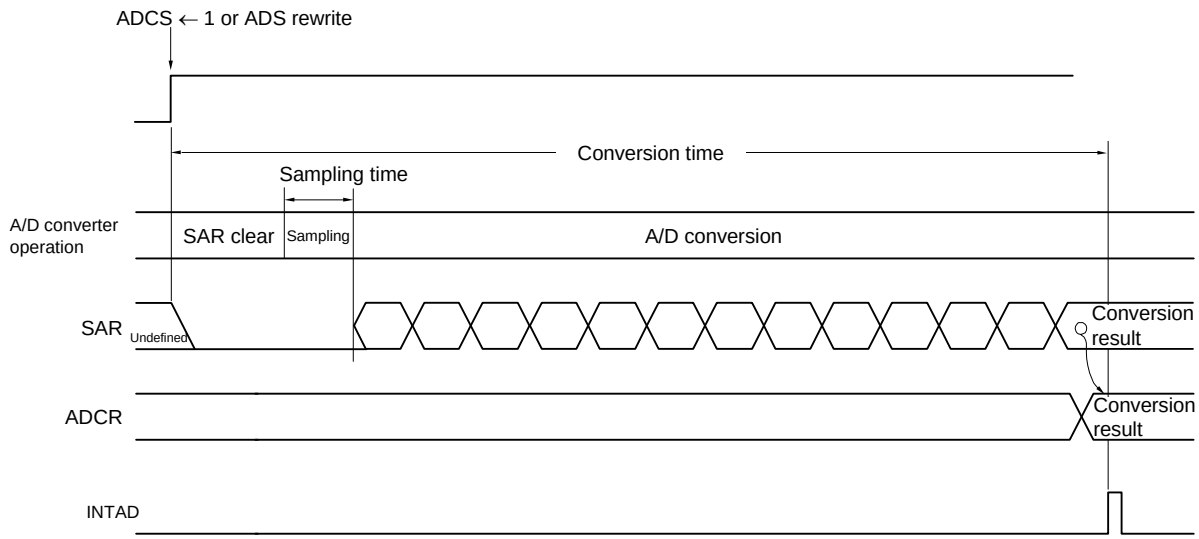
Note 2. While in the sequential conversion mode, the ADCS flag is not automatically cleared to 0. This flag is not automatically cleared to 0 while in the one-shot conversion mode of the hardware trigger no-wait mode, either. Instead, 1 is retained.

Remark 1. Two types of the A/D conversion result registers are available.

- ADCR register (16 bits): Store 12-bit A/D conversion value
- ADCRH register (8 bits): Store 8-bit A/D conversion value

Remark 2. AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and AV_{DD}.

Figure 12 - 16 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.

Rewriting and overwriting 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 ANI6 and ANI16 to ANI21) and the theoretical A/D conversion result (stored in the 12-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$ADCR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 4096 + 0.5\right)$$

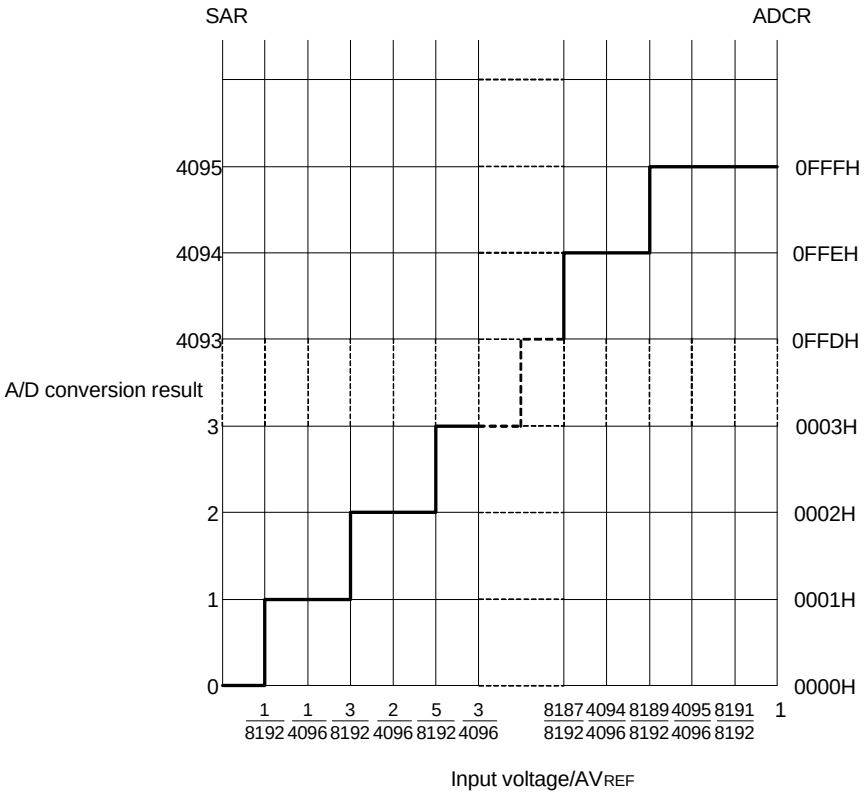
or

$$(ADCR - 0.5) \times \frac{AV_{REF}}{4096} \leq V_{AIN} < (ADCR + 0.5) \times \frac{AV_{REF}}{4096}$$

where, INT( ): Function which returns integer part of value in parentheses
V_{AIN}: Analog input voltage
AV_{REF}: AV_{REF} pin voltage
ADCR: A/D conversion result register (ADCR) value

Figure 12 - 17 shows the Relationship Between Analog Input Voltage and A/D Conversion Result.

Figure 12 - 17 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 AV_{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 Note, 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.

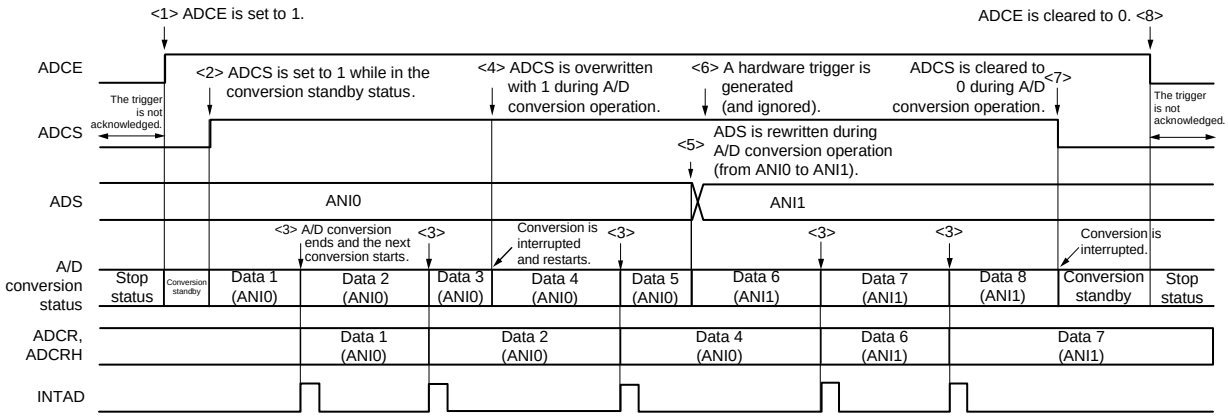
<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<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.

Note If a high-accuracy channel is selected as the analog input channel: Stabilization wait time = 0.5 μs
If a standard channel is selected as the analog input channel: Stabilization wait time = 2 μs

Figure 12 - 18 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 [Note], 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.

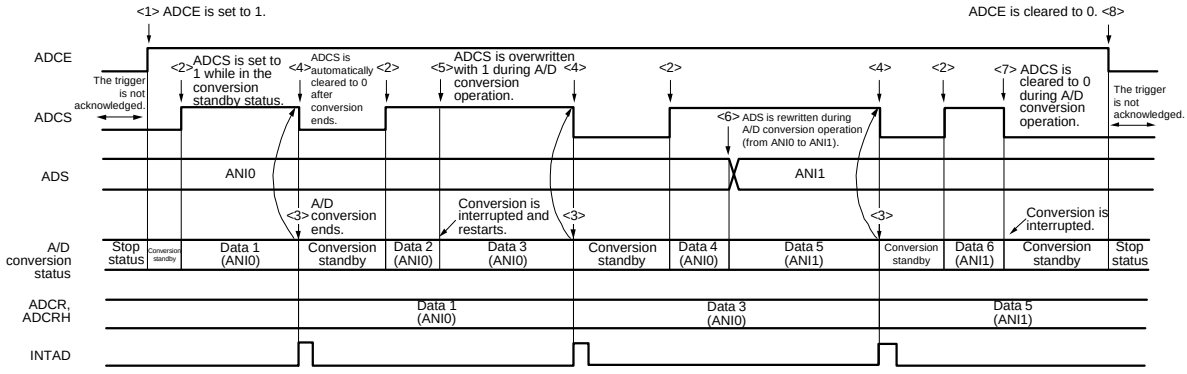
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<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.

Note If a high-accuracy channel is selected as the analog input channel: Stabilization wait time = 0.5 μs
If a standard channel is selected as the analog input channel: Stabilization wait time = 2 μs

Figure 12 - 19 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 Note, 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.

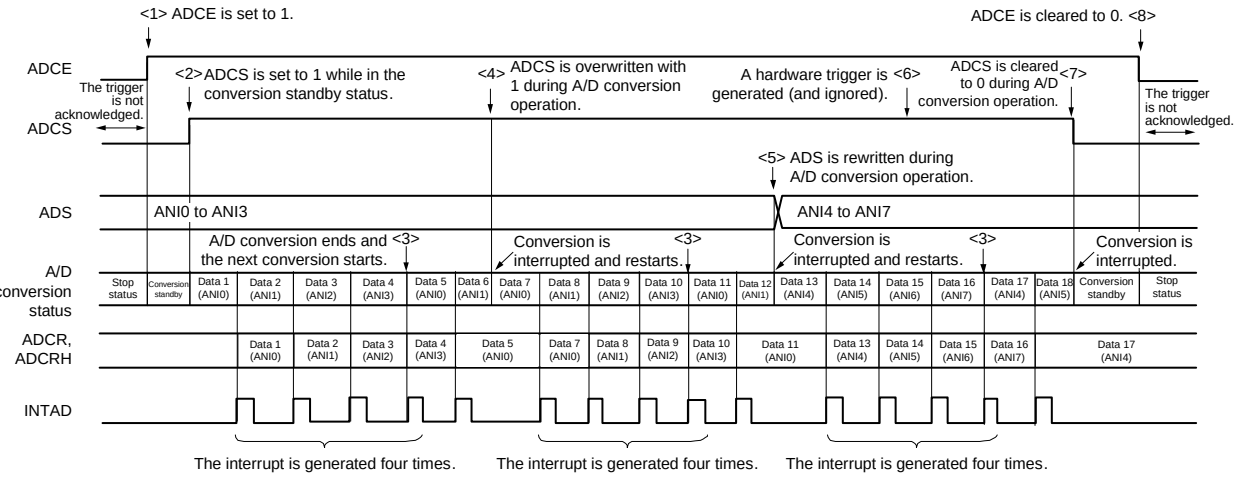
<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<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.

Note If a high-accuracy channel is selected as the analog input channel: Stabilization wait time = 0.5 μs
If a standard channel is selected as the analog input channel: Stabilization wait time = 2 μs

Figure 12 - 20 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 Note, 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.

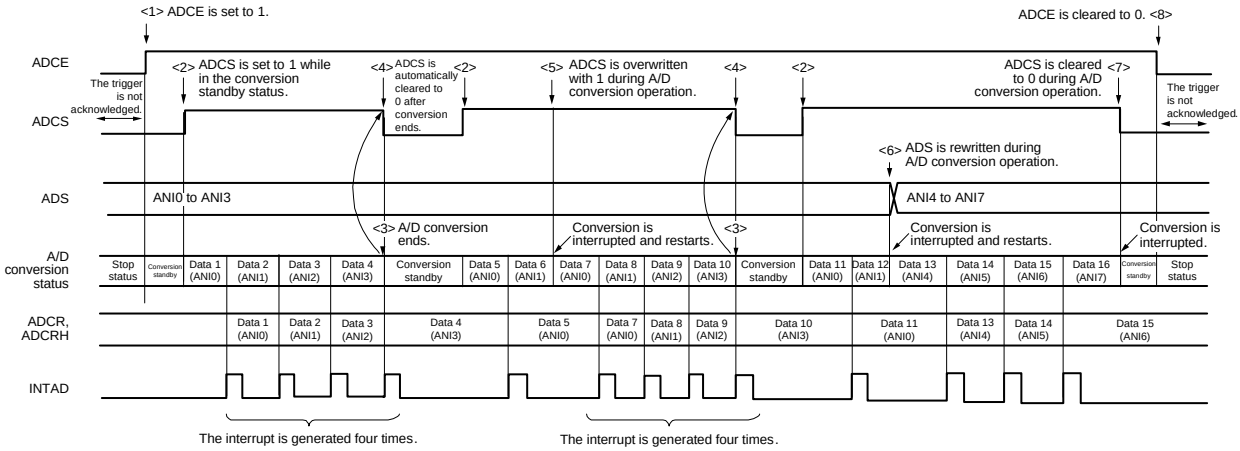
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<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.

Note If a high-accuracy channel is selected as the analog input channel: Stabilization wait time = 0.5 μs
If a standard channel is selected as the analog input channel: Stabilization wait time = 2 μs

Figure 12 - 21 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 Note, 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.

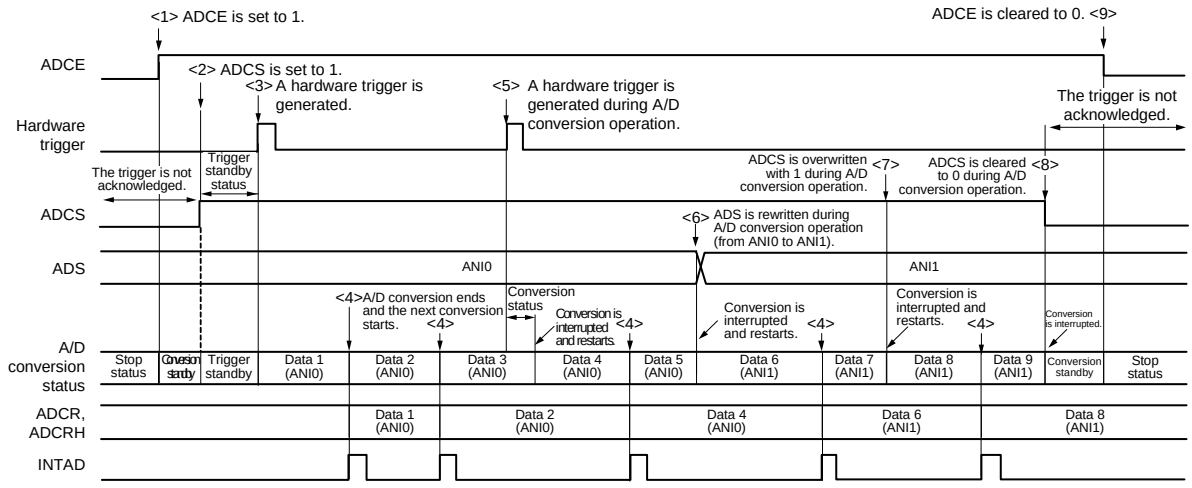
<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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.

Note If a high-accuracy channel is selected as the analog input channel: Stabilization wait time = 0.5 μs
If a standard channel is selected as the analog input channel: Stabilization wait time = 2 μs

Figure 12 - 22 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 Note, 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.

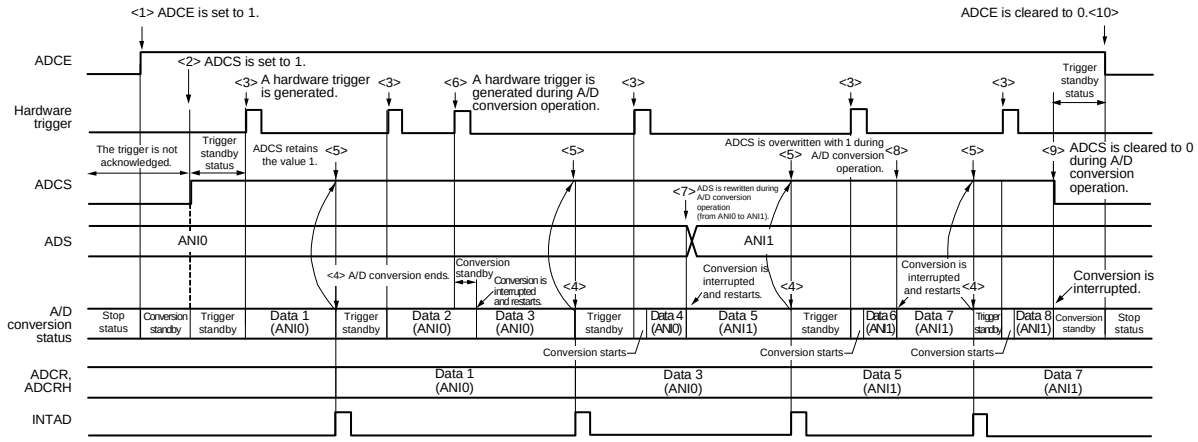
<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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.

Note If a high-accuracy channel is selected as the analog input channel: Stabilization wait time = 0.5 μs
If a standard channel is selected as the analog input channel: Stabilization wait time = 2 μs

Figure 12 - 23 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 Note, 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.

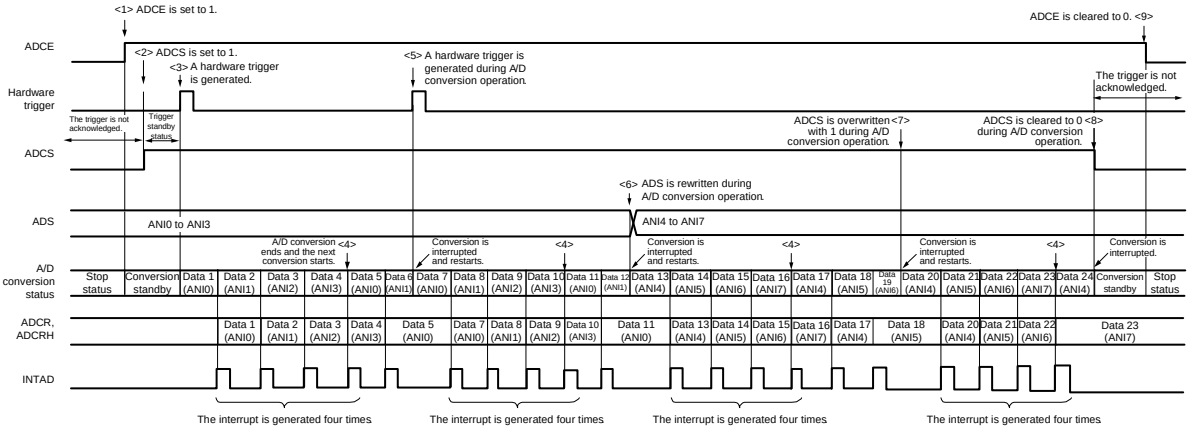
<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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.

Note If a high-accuracy channel is selected as the analog input channel: Stabilization wait time = 0.5 μs
If a standard channel is selected as the analog input channel: Stabilization wait time = 2 μs

Figure 12 - 24 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 Note, 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.

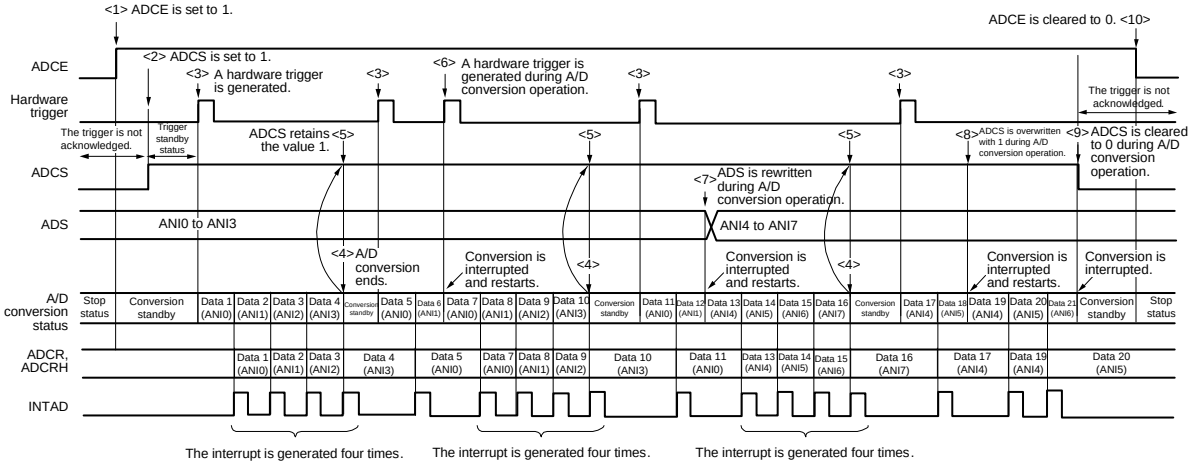
<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<10> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Note If a high-accuracy channel is selected as the analog input channel: Stabilization wait time = 0.5 μs
If a standard channel is selected as the analog input channel: Stabilization wait time = 2 μs

Figure 12 - 25 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.)

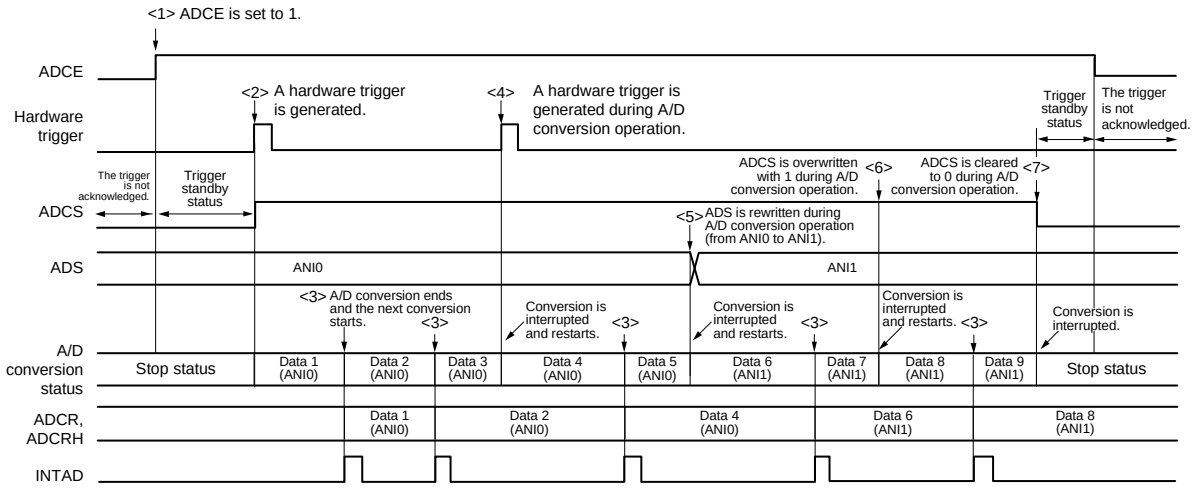
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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 - 26 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.

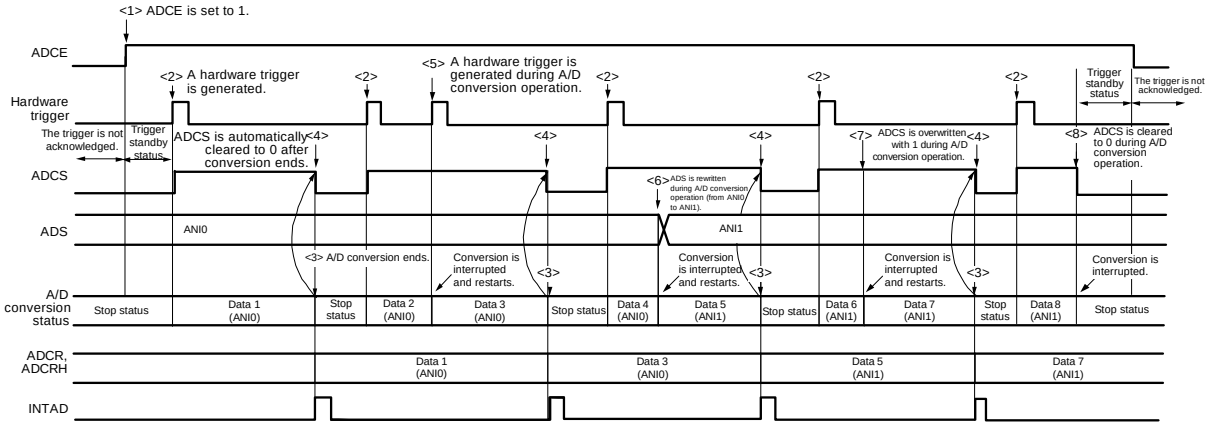
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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 - 27 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.

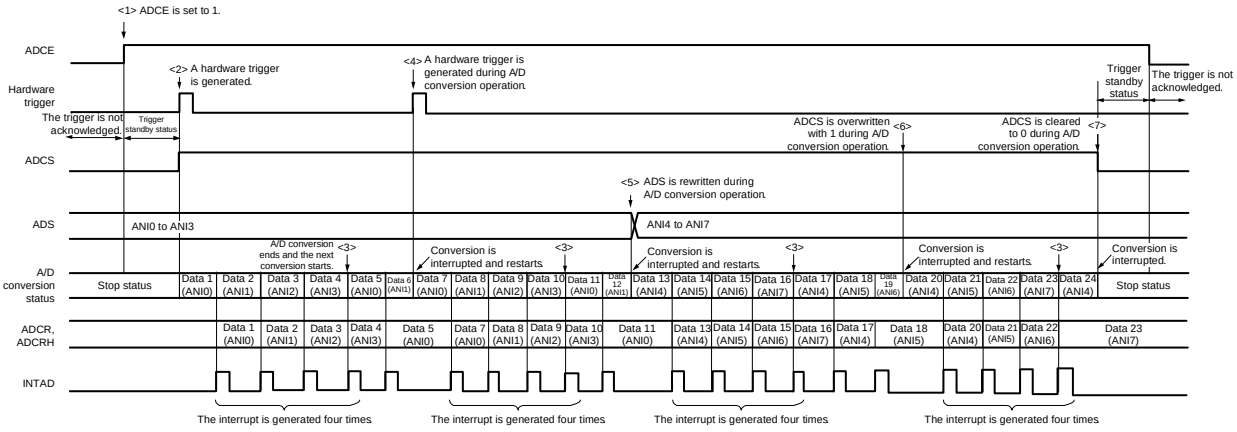
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<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 - 28 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.

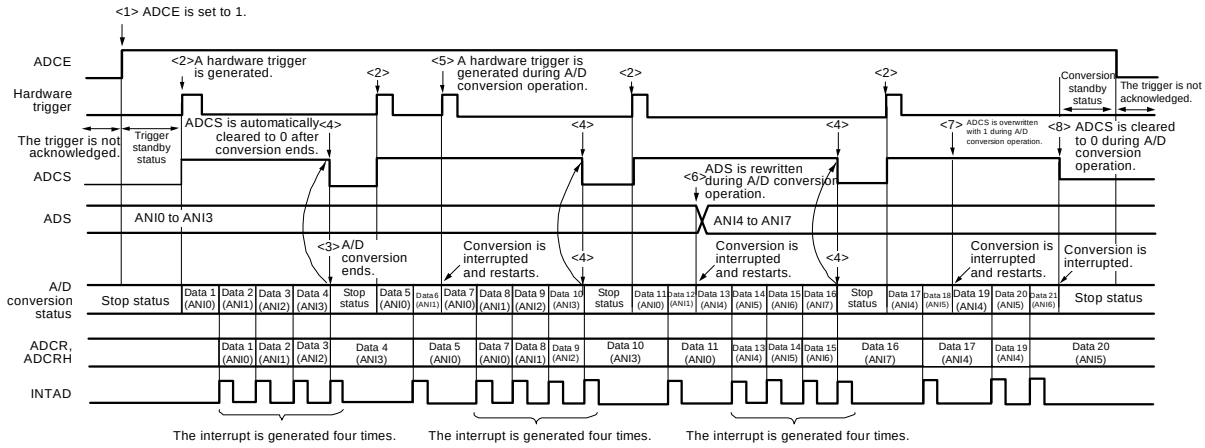
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<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 - 29 Example of Hardware Trigger Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing

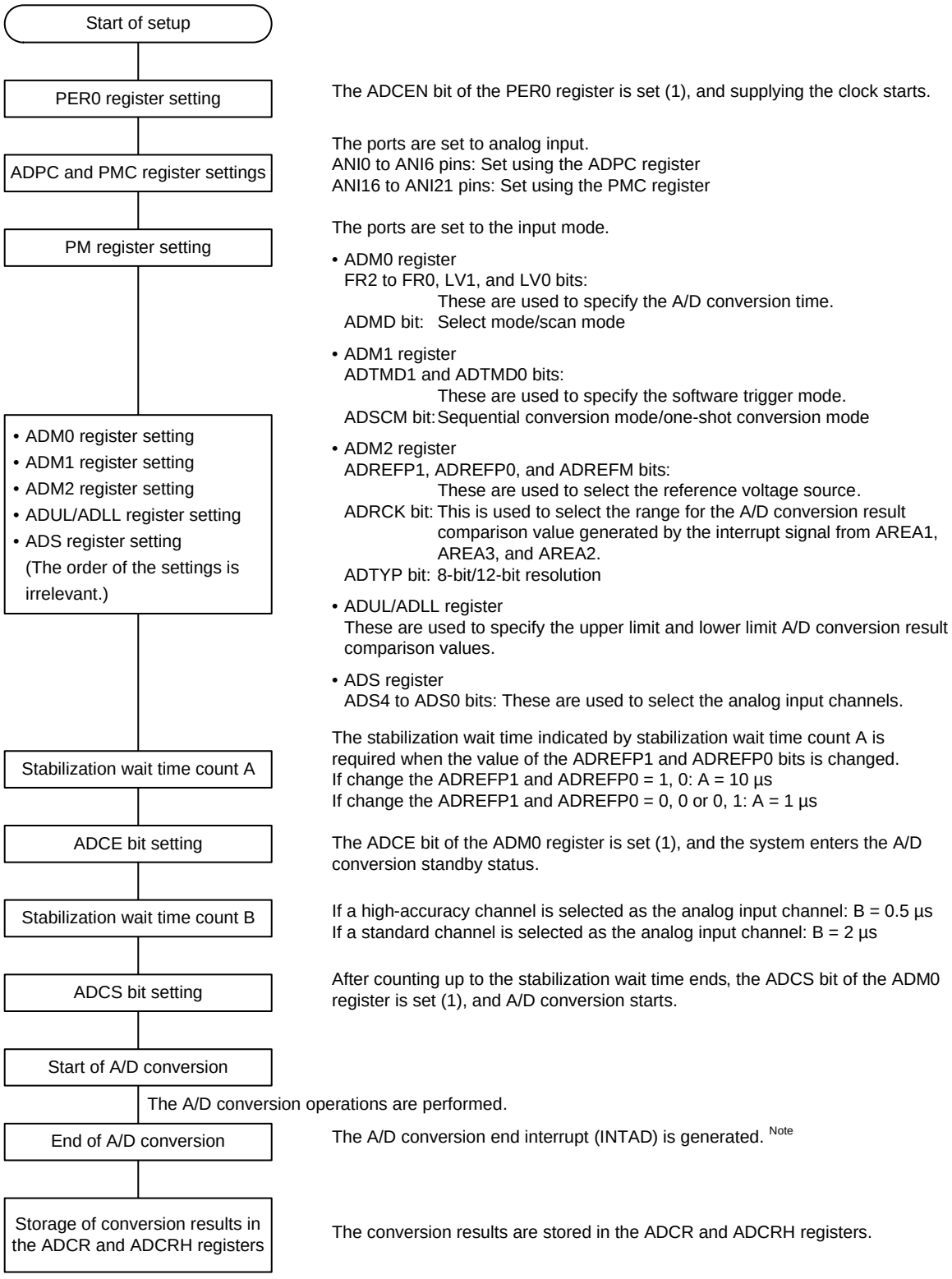


12.7 A/D Converter Setup Flowchart

The A/D converter setup flowchart in each operation mode is described below.

12.7.1 Setting up software trigger mode

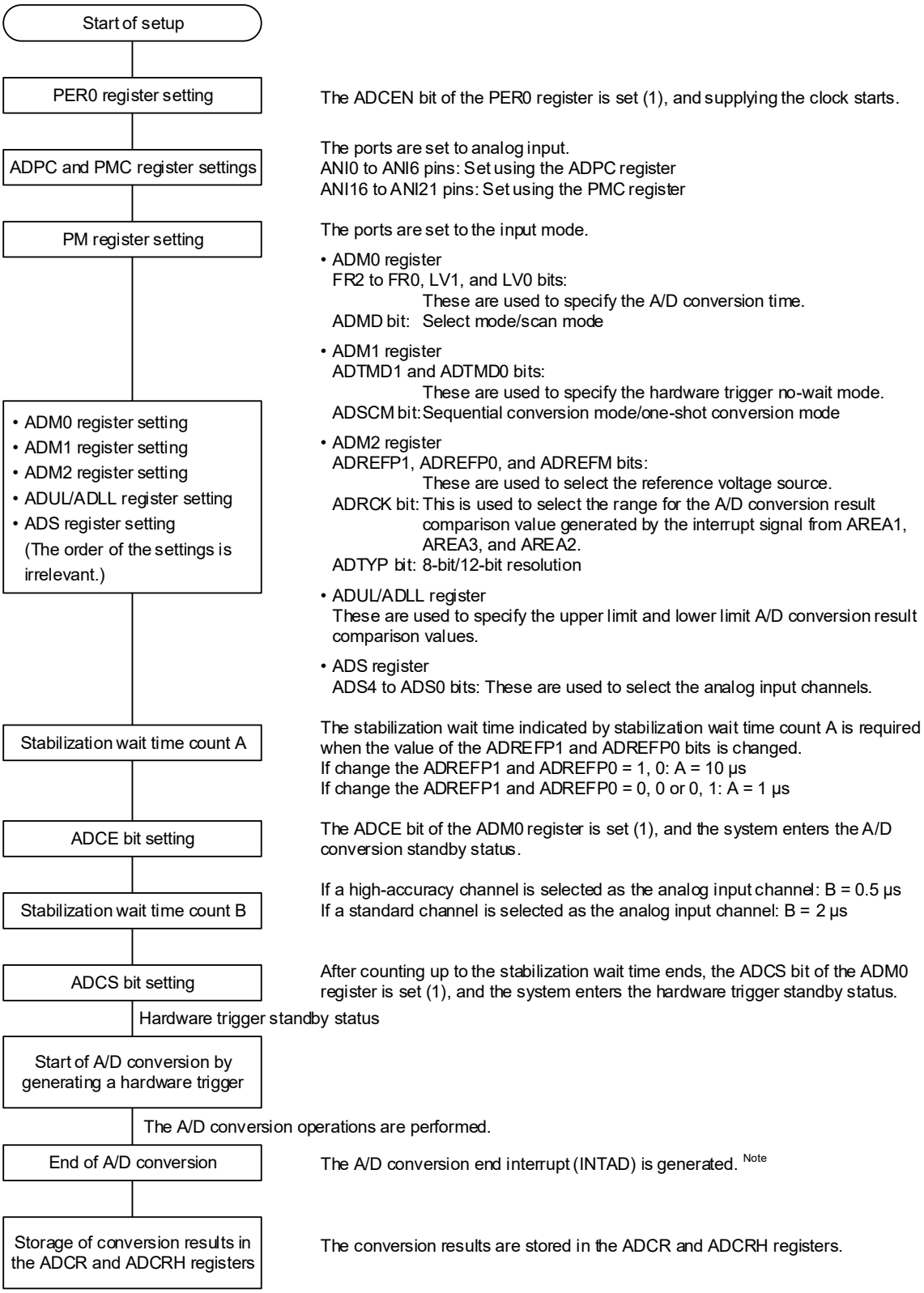
Figure 12 - 30 Setting up Software Trigger Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no A/D conversion end interrupt request signal (INTAD) being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

12.7.2 Setting up hardware trigger no-wait mode

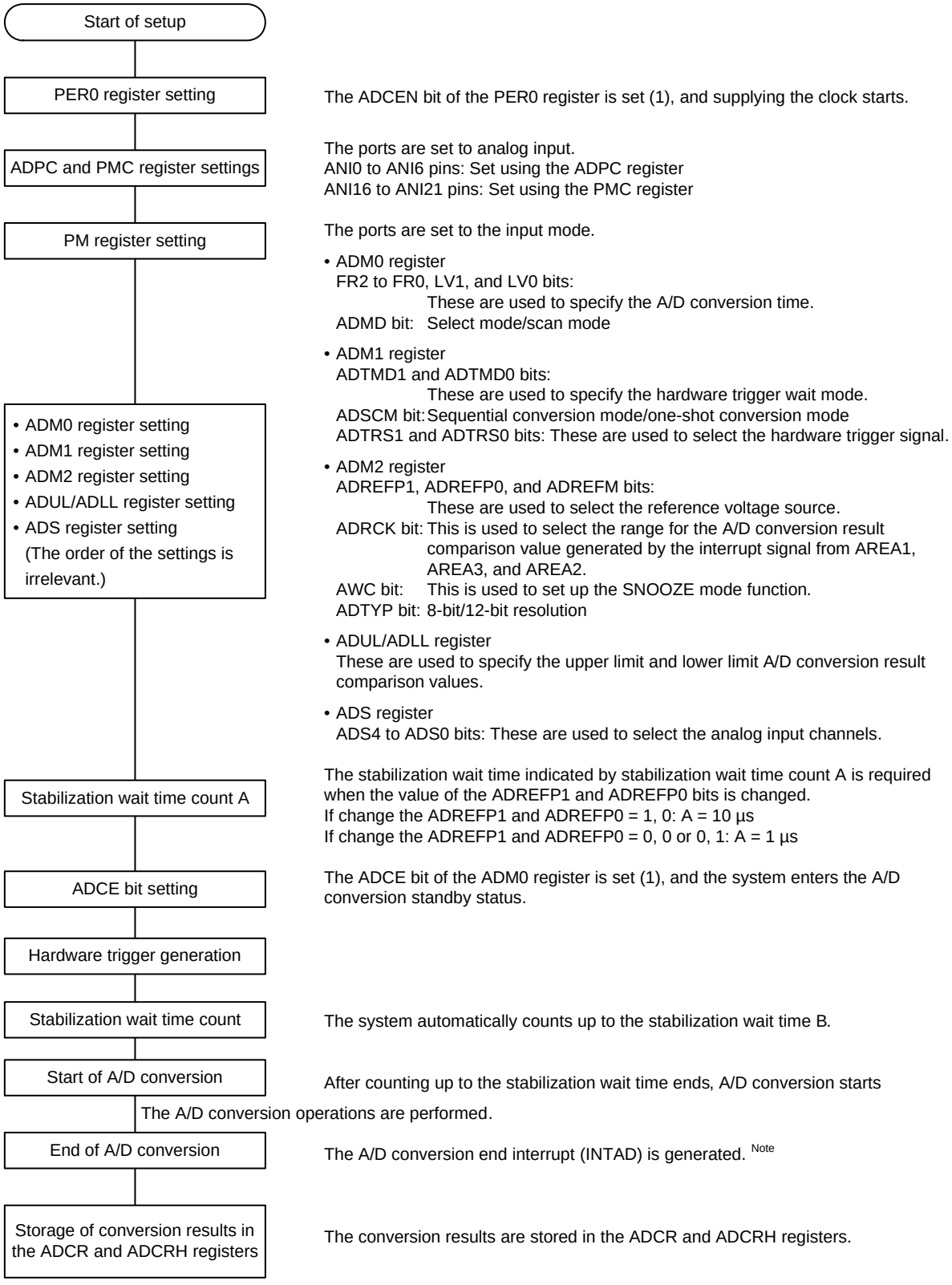
Figure 12 - 31 Setting up Hardware Trigger No-Wait Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no A/D conversion end interrupt request signal (INTAD) being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

12.7.3 Setting up hardware trigger wait mode

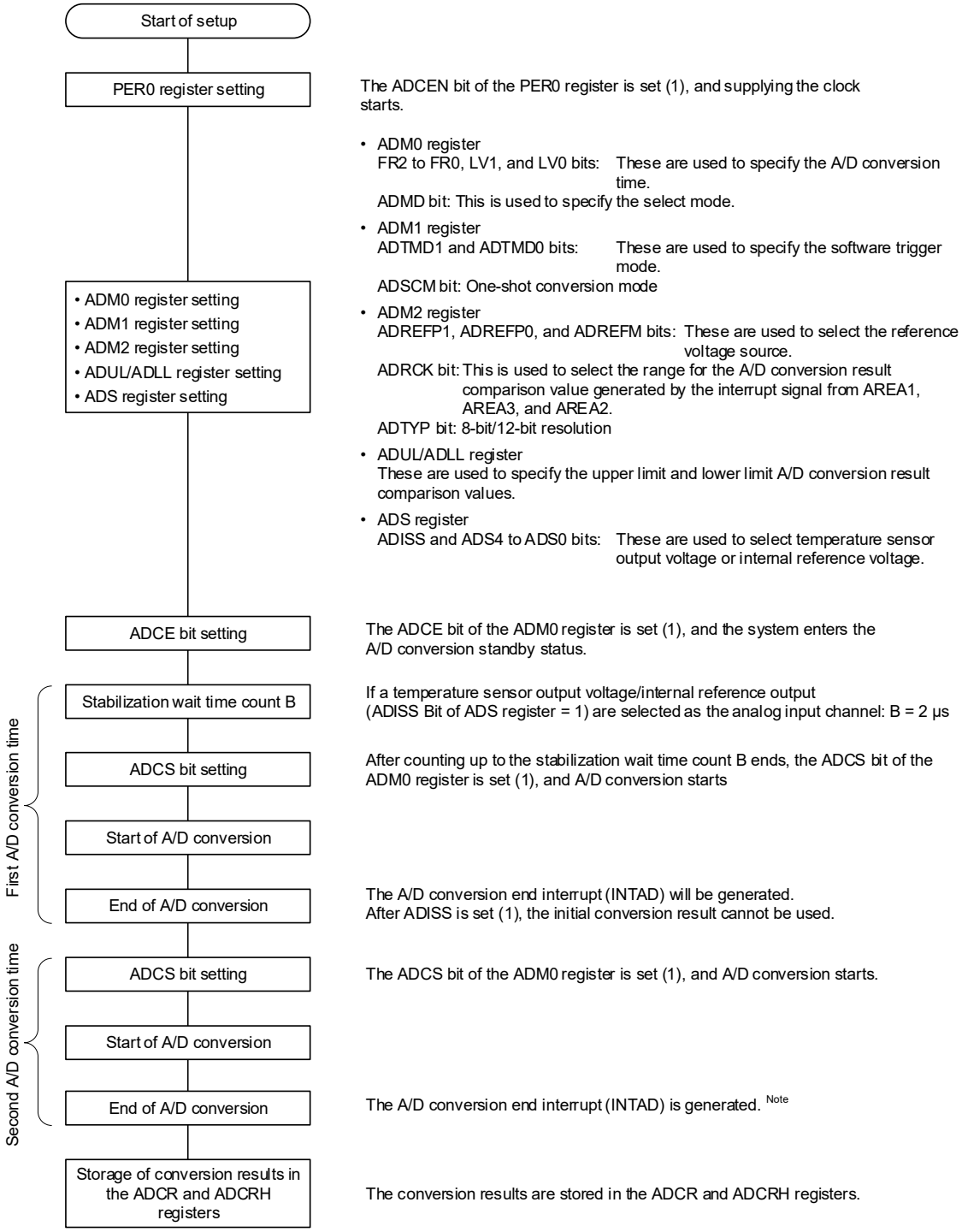
Figure 12 - 32 Setting up Hardware Trigger Wait Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no A/D conversion end interrupt request signal (INTAD) being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

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 - 33 Setup when Temperature Sensor Output Voltage/Internal Reference Voltage is Selected

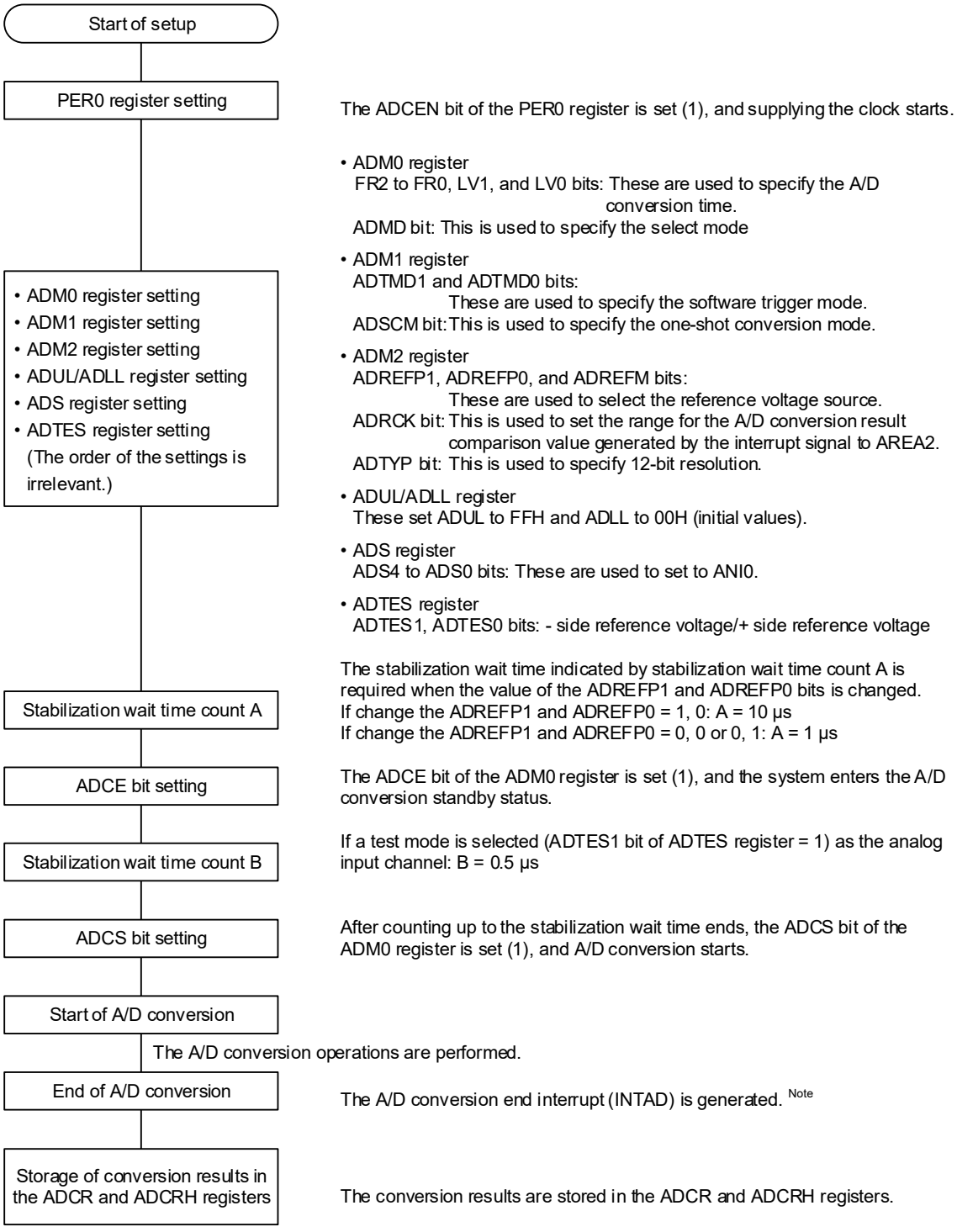


Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no A/D conversion end interrupt request signal (INTAD) being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

Caution **This setting can be used only in HS (high-speed main) mode. For detail, see Figure 29 - 4 Format of User Option Byte (000C2H/010C2H).**

12.7.5 Setting up test mode

Figure 12 - 34 Setting up Test Trigger Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no A/D conversion end interrupt request signal (INTAD) being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

Caution For the procedure for testing the A/D converter, see 27.3.8 A/D test function.

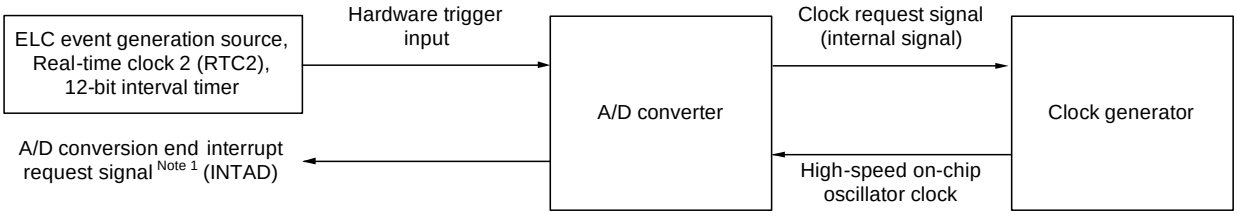
12.8 SNOOZE Mode Function

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, A/D conversion can be performed without operating the CPU by inputting a hardware trigger. This is effective for reducing the operation current.
If the A/D conversion result range is specified using the ADUL and ADLL registers, A/D conversion results can be judged at a certain interval of time in SNOOZE mode. Using this function enables power supply voltage monitoring and input key judgment based on A/D inputs.

In the SNOOZE mode, only the following two conversion modes can be used:
• Hardware trigger wait mode (select mode, one-shot conversion mode)
• Hardware trigger wait mode (scan mode, one-shot conversion mode)

Caution The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for f_{CLK}.

Figure 12 - 35 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 switching to the STOP mode, set bit 2 (AWC) of A/D converter mode register 2 (ADM2) 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 stabilization wait time, and then A/D conversion starts.
The SNOOZE mode operation after A/D conversion ends differs depending on whether an interrupt signal is generated Note 1.

Note 1. Depending on the setting of the A/D conversion result comparison function (ADRCK bit, ADUL/ADLL register), there is a possibility of no interrupt signal being generated.
Note 2. Be sure to set the ADM1 register to E1H, E2H or E3H.

Remark 1. The hardware trigger is the event selected by the ELC, or INTRTC or INTIT.
INTRTC can be used as an ELC event generation source or directly used as a trigger.
Remark 2. Specify the hardware trigger by using 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.

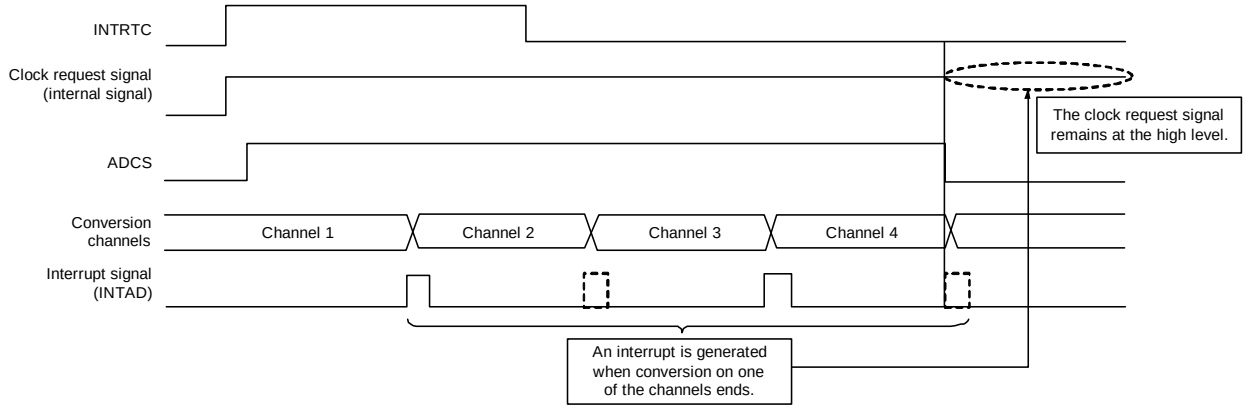
- While in the select mode

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 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 - 36 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 - 37 Operation Example When No Interrupt Is Generated After A/D Conversion Ends (While in Scan Mode)

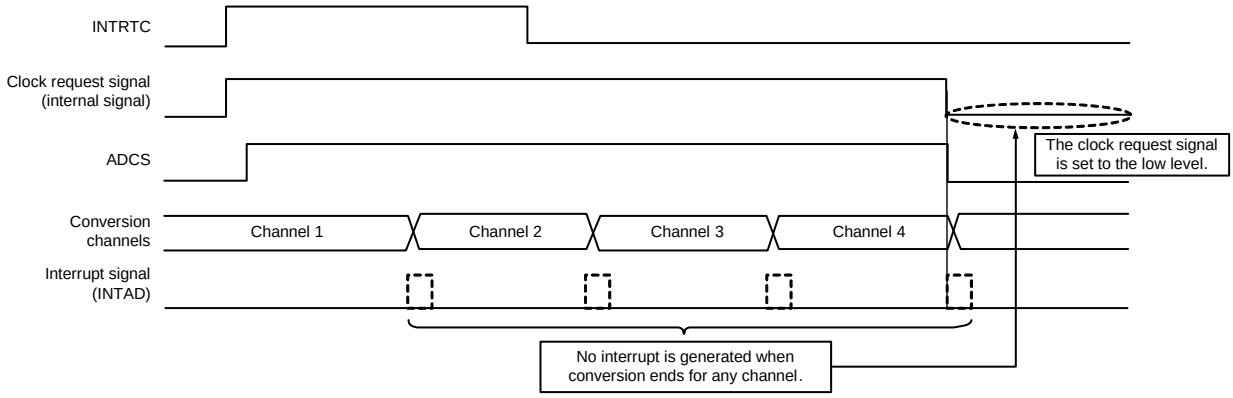
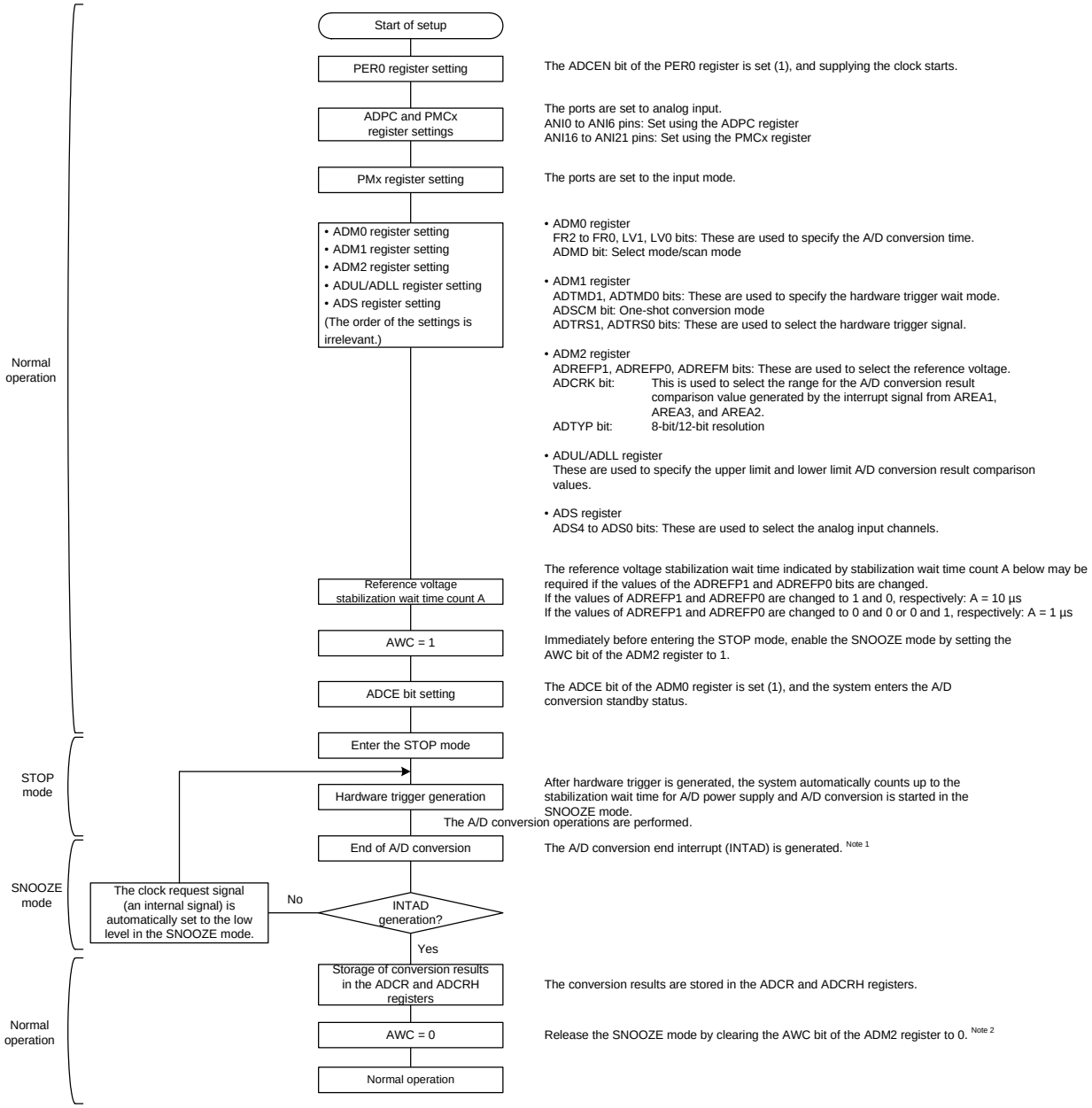


Figure 12 - 38 Flowchart for Setting up SNOOZE Mode



Note 1. If the A/D conversion end interrupt request signal (INTAD) is not generated by setting ADRCK bit and ADUL/ADLL register, the result is not stored in the ADCR and ADCRH registers.
The system enters the STOP mode again. If a hardware trigger is input later, A/D conversion operation is again performed in the SNOOZE mode.

Note 2. If the AWC bit is left set to 1, A/D conversion will not start normally in spite of the subsequent SNOOZE or normal operation mode. Be sure to clear the AWC bit to 0.

12.9 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 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.

$1\ LSB = 1/2^{12} = 1/4096$
$= 0.00091\%FSR$

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 - 39 Overall Error

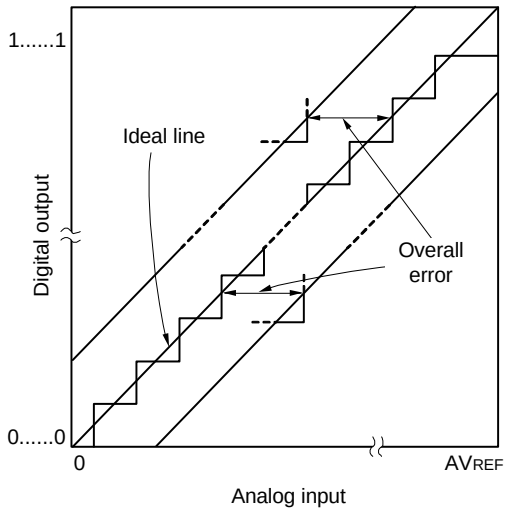
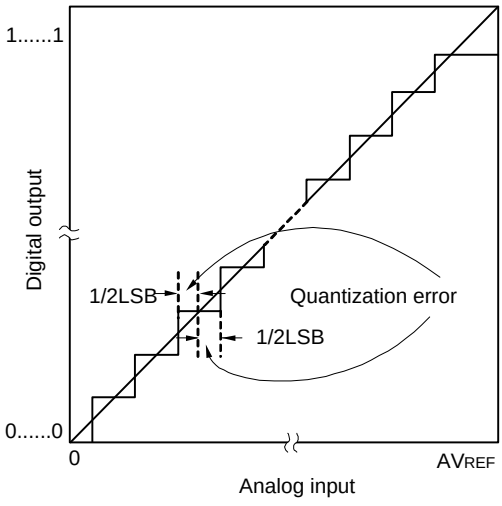


Figure 12 - 40 Quantization Error



(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 - 41 Zero-Scale Error

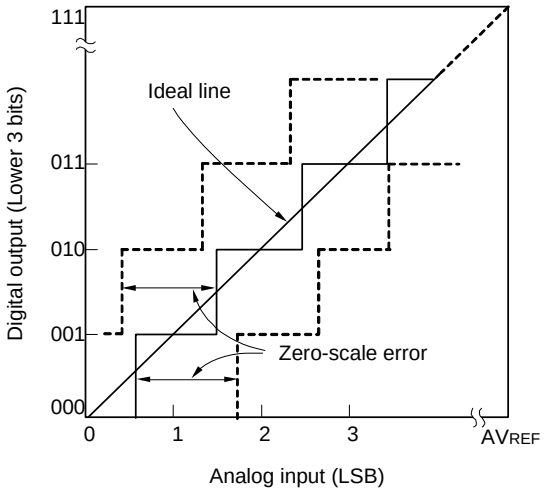


Figure 12 - 42 Full-Scale Error

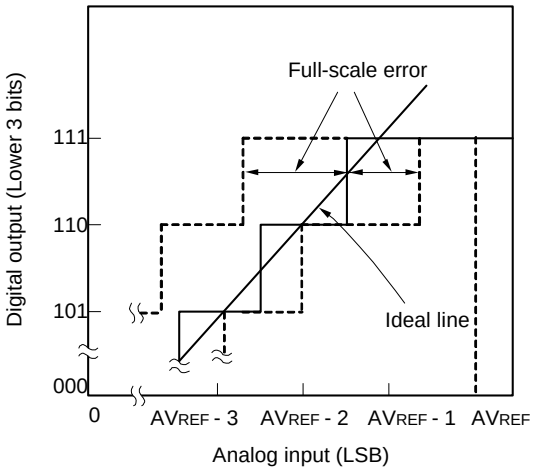


Figure 12 - 43 Integral Linearity Error

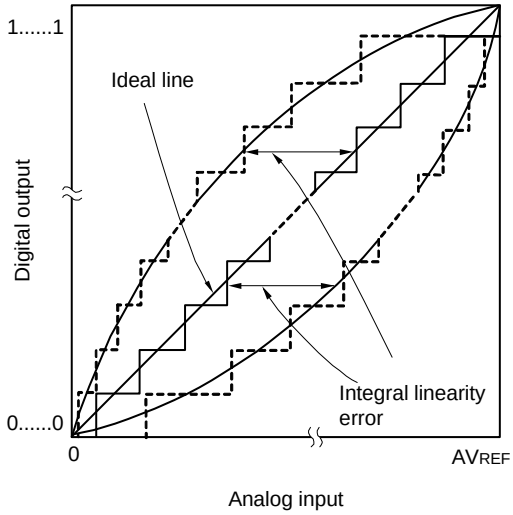
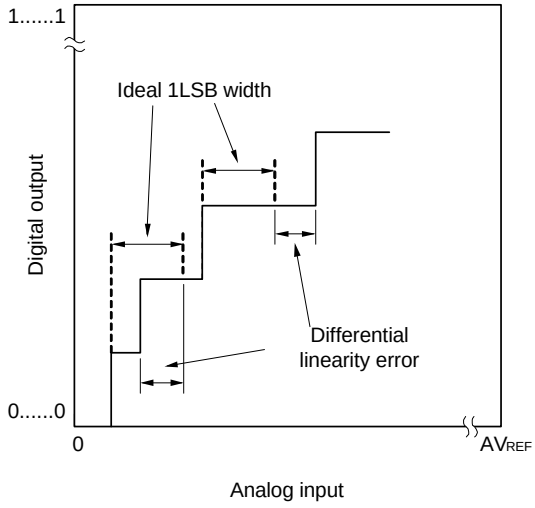


Figure 12 - 44 Differential Linearity Error



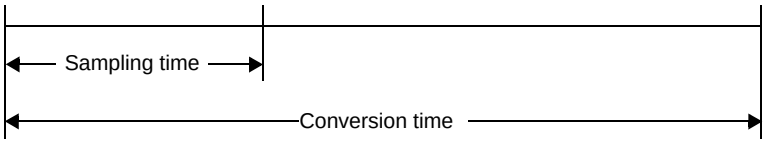
(8) Conversion time

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 ANI6 and ANI16 to ANI21 pins

Observe the rated range of the ANI0 to ANI6 and ANI16 to ANI21 pin input voltage. If a voltage of AV_{DD} and AV_{REFP} or higher or a voltage lower than AV_{SS} and AV_{REFM} (even in the range of absolute maximum ratings) is input to an analog input channel, the converted value of that channel becomes undefined. In addition, the converted values of the other channels may also be affected.

When internal reference voltage (1.45 V) is selected reference voltage source for the + side of the A/D converter, do not input internal reference voltage (1.45 V) or higher voltage to a pin selected by the ADS register. However, it is no problem that a pin not selected by the ADS register is input voltage greater than the internal reference voltage (1.45 V).

Caution Internal reference voltage (1.45 V) can be used only in HS (high-speed main) mode. For detail, see Figure 29 - 4 Format of User Option Byte (000C2H/010C2H).

(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 register 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 12-bit resolution, attention must be paid to noise input to the AV_{REFP}, AV_{DD} ANI0 to ANI6, and ANI16 to ANI21 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.

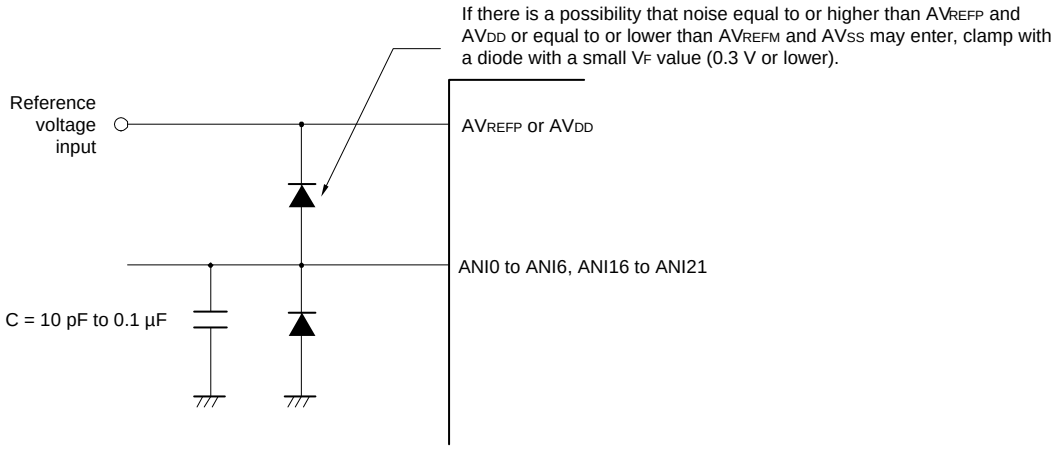
<2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting external C as shown in Figure 12 - 45 is recommended.

<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.

<5> Separate digital and analog signals so that they do not cross or approach each other.

Figure 12 - 45 Analog Input Pin Connection



(5) Analog input (ANIn) pins

<1> ANI0 to ANI6 pins (high-accuracy channel) are also used as P150 to P156 pins.
When A/D conversion is performed with any of the high-accuracy channel (ANI0 to ANI6) pins selected, do not change the output value P150 to P156 while conversion is in progress; otherwise the conversion accuracy may be degraded.

<2> If a pin adjacent to the pin whose value is being A/D converted is used as a digital I/O port pin, the A/D conversion value might differ from the expected value due to coupling noise. To prevent coupling noise, make sure that pulses whose voltage suddenly change, such as digital pulses, are not input or output to a pin adjacent to the pin whose value is being A/D converted.

(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.

However, in order to perform sampling accurately, the output impedance of the analog input source should be 1 kΩ or lower. If it is not possible to keep the output impedance below this level, it is recommended to either extend the sampling time or connect a capacitor of about 0.1 µF to the ANI0 to ANI12 and ANI16 to ANI30 pins. (See **Figure 12 - 45** for details.)

Also, if the ADCS bit is set to 0 or a reconversion is started during A/D conversion, the sampling capacitor will be insufficiently charged. This means that charging will start with an undefined conversion voltage from the next conversion in the case of setting the ADCS bit to 0, or from the current conversion in the case of starting a reconversion. To ensure that the capacitor is fully charged, therefore, either reduce the output impedance of the analog input source or specify a sufficiently long sampling time, irrespective of the analog signal voltage variation.

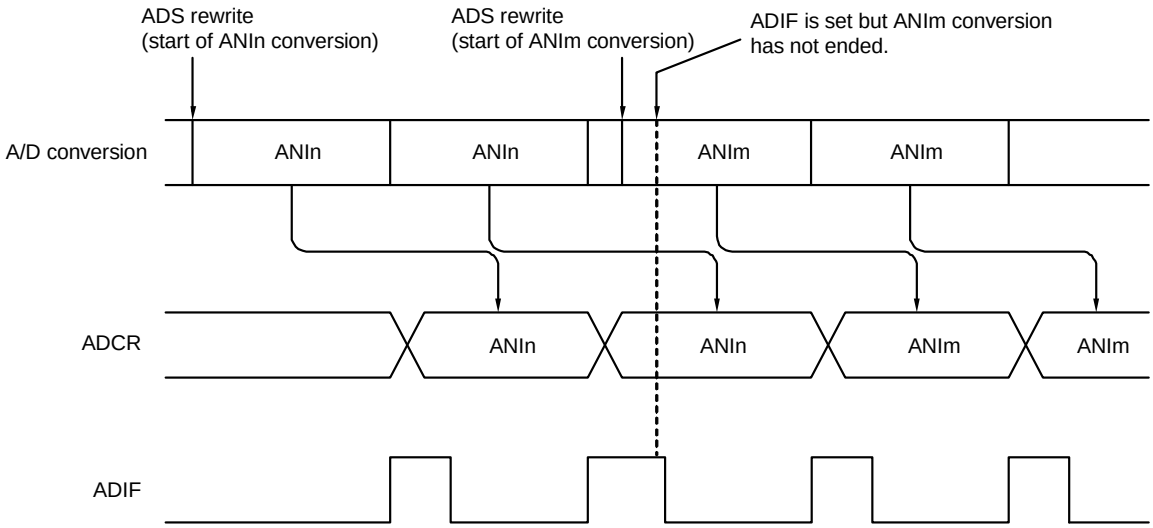
(7) Interrupt request flag (ADIF)

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 - 46 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 the stabilization wait time 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.

[Stabilization wait status]

If a high-accuracy channel is selected as the analog input channel: 0.5 μs
If a test mode setting (ADTES1 bit of ADTES register = 1) is selected: 0.5 μs
If a standard channel is selected as the analog input channel: 2 μs
If a temperature sensor output/internal reference voltage output are selected as the analog input channel:
(ADISS bit of ADS register = 1): 2 μs

(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 (PMCx), 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 PMCx 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 - 47 Internal Equivalent Circuit of ANIn Pin

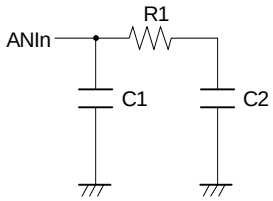


Table 12 - 7 Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{DD}, AV_{REFP}	ANIn pin	R1[kΩ]	C1[pF]	C2[pF]
2.4 V ≤ AV_{DD} ≤ 3.6 V	ANI0 to ANI6	7.4	8	6.3
	ANI16 to ANI21	12.3	8	7.4
1.8 V ≤ AV_{DD} ≤ 3.6 V	ANI0 to ANI6	11	8	6.3
	ANI16 to ANI21	41	8	7.4
1.6 V ≤ AV_{DD} ≤ 3.6 V	ANI0 to ANI6	510	8	6.3
	ANI16 to ANI21	650	8	7.4

Remark The resistance and capacitance values shown in Table 12 - 7 are not guaranteed values.

(11) Starting the A/D converter

Start the A/D converter after the AV_{REFP} and AV_{DD} voltages stabilize.

(12) Always set the AV_{DD} and V_{DD} to the same potential when the A/D voltage comparator operates.

CHAPTER 13 D/A CONVERTER

13.1 Functions of D/A Converter

The D/A converter is an 8-bit resolution converter that converts digital inputs into analog signals. It is used to control analog outputs for two independent channels (ANO0, ANO1).

The D/A converter has the following features.

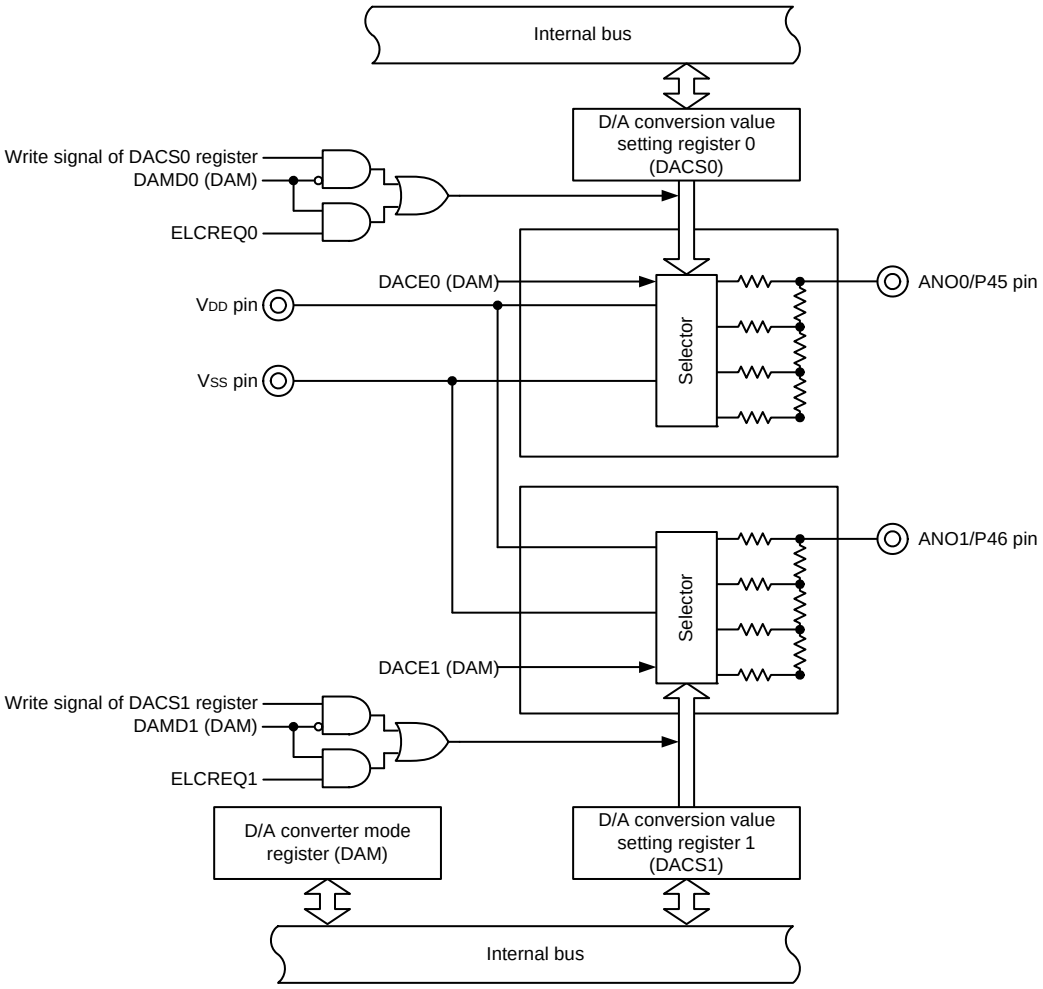
- 8-bit resolution × 2 channels
- R-2R ladder method
- Output analog voltage
  - 8-bit resolution: V_{DD} × m8/256 (m8: Value set to DACSi register)
- Operation mode
  - Normal mode
  - Real-time output mode

Remark i = 0, 1

13.2 Configuration of D/A Converter

Figure 13 - 1 shows the Block Diagram of D/A Converter.

Figure 13 - 1 Block Diagram of D/A Converter



Remark ELCREQ0 and ELCREQ1 are trigger signals (request signals from the ELC) that are used in the real-time output mode.

13.3 Registers Controlling D/A Converter

The D/A converter is controlled by the following registers.

- Port mode control register 4 (PMC4)
- Peripheral enable register 1 (PER1)
- D/A converter mode register (DAM)
- D/A conversion value setting registers 0, 1 (DACS0, DACS1)
- Port mode register 4 (PM4)

13.3.1 Port mode control register 4 (PMC4)

This register set the ANO0/P45 to ANO1/P46 digital I/O/analog I/O in 1-bit units.
This register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to FFH.

Figure 13 - 2 Format of Port mode control register 4 (PMC4)

Address: F0064H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PMC4	0	PMC46	PMC45	PMC44	PMC43	PMC42	PMC41	0

PMC4n	P4n pin digital I/O and analog I/O selection (n = 1 to 6)
0	Digital I/O (multiplexed function other than analog I/O)
1	Analog I/O

13.3.2 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.
When the D/A converter is used, be sure to set bit 0 (DACEN) of this register to 1.
The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 13 - 3 Format of Peripheral enable register 1 (PER1)

Address: F007AH After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	2	1	<0>
PER1	TMKAEN	0	CMPEN	TKB20EN	DTCEN	0	0	DACEN

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

Caution 1. When setting the D/A converter, be sure to set DACEN to 1 first.
If DACEN = 0, writing to a control register of the D/A converter is ignored, and all read values are default values (except for port mode register 4 (PM4) and port register 4 (P4)).

Caution 2. Be sure to clear bits 6, 2, and 1 to "0".

13.3.3 D/A converter mode register (DAM)

This register controls the operation of the D/A converter.
The DAM register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 13 - 4 Format of D/A converter mode register (DAM)

Address: FFF36H After reset: 00H R/W

Symbol	7	6	<5>	<4>	3	2	1	0
DAM	0	0	DACE1	DACE0	0	0	DAMD1	DAMD0

DACEi	D/A conversion operation control
0	Stops D/A conversion operation
1	Enables D/A conversion operation

DAMDi	D/A converter operation mode selection
0	Normal mode
1	Real-time output mode

Remark i = 0, 1

13.3.4 D/A conversion value setting register i (DACSi) (i = 0, 1)

This register is used to set the analog voltage value to be output to the ANO0 and ANO1 pins when the D/A converter is used.
The DACSi register can be read by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 13 - 5 Format of D/A conversion value setting register i (DACSi) (i = 0, 1)

Address: FFF74H (DACS0), FFF75H (DACS1) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
DACSi	DACSi7	DACSi6	DACSi5	DACSi4	DACSi3	DACSi2	DACSi1	DACSi0

Remark The relation between the resolution and analog output voltage (VANOi) of the D/A converter are as follows.
VANOi = Reference voltage for D/A converter × (DACSi)/256

When the D/A converter is not used, set the DACEi bit to 0 (output disable) and set the DACSi register to 00H to prevent current from flowing into the R-2R resistor ladder to reduce unnecessary current consumption.

13.3.5 Port mode register 4 (PM4)

When using the ANO0/P45 and ANO1/P46 pins as analog output ports, set bits PM45 and PM46 to 1.
If bits PM45 and PM46 are set to 0, these pins cannot be used as analog I/O ports.
The PM4 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to FFH.

Figure 13 - 6 Format of Port mode register 4 (PM4)

Address: FFF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	0	PM46	PM45	PM44	PM43	PM42	PM41	PM40

PM4n	P4n pin I/O mode selection (n = 0 to 6)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

The function of the ANO0/P45 and ANO1/P46 pins can be selected by using the port mode control register 4 (PMC4), the D/A converter mode register (DAM), and the PM4 register.

Table 13 - 1 Setting Functions of ANO0/P45 and ANO1/P46 Pins

PMC4	PM4	DAM	Functions of ANO0/P45 and ANO1/P46 Pins
Digital I/O selection	Input mode	—	Digital input
	Output mode	—	Digital output
Analog I/O selection	Input mode	Enables D/A conversion operation	Analog output
		Stops D/A conversion operation	Analog output (high-impedance output)
	Output mode	Enables D/A conversion operation	Setting prohibited
		Stops D/A conversion operation	

13.4 Operations of D/A Converter

13.4.1 Operation in Normal Mode

D/A conversion is performed using write operation to the DACSi register as the trigger.
The setting method is described below.

<1> Set the DACEN bit of the PER1 register (peripheral enable register 1) to 1 to start the supply of the input clock to the D/A converter.
<2> Set the PMC4 register (port mode control register 4) and the PM4 register (port mode register 4) to specify analog output.
<3> Set the DAMDi bit of the DAM register (D/A converter mode register) to 0 (normal mode).
<4> Set the analog voltage value to be output to the ANOi pin to the DACSi register (D/A conversion value setting register i).

Steps <1> and <4> above constitute the initial settings.

<5> Set the DACEi bit of the DAM register to 1 (D/A conversion enable).
D/A conversion starts, and then, after the settling time elapses, the analog voltage set in step <4> is output to the ANOi pin.
<6> To perform subsequent D/A conversions, write to the DACSi register.

The previous D/A conversion result is held until the next D/A conversion is performed.
When the DACEi bit of the DAM register is set to 0 (D/A conversion operation stop), D/A conversion stops.

Caution 1. Even if 1, 0, and then 1 is set to the DACEi bit, the analog voltage set by the DACSi register is output to the ANOi pin when a settling time has elapsed after 1 is set for the last time.

Caution 2. If the DACSi register is rewritten during the settling time, D/A conversion is aborted and reconversion by using the rewritten values starts.

Remark i = 0, 1

13.4.2 Operation in Real-Time Output Mode

D/A conversion is performed on each channel using the event signals from the ELC as triggers.
The setting method is described below.

<1> Set the DACEN bit of the PER1 register (peripheral enable register 1) to 1 to start the supply of the input clock to the D/A converter.
<2> Set the PMC4 register (port mode control register 4) and the PM4 register (port mode register 4) to specify analog output.
<3> Set the DAMDi bit of the DAM register (D/A converter mode register) to 0 (normal mode).
<4> Set the analog voltage value to be output to the ANOi pin to the DACSi register (D/A conversion value setting register i).
<5> Set the DACEi bit of the DAM register to 1 (D/A conversion enable).
D/A conversion starts, and then, after the settling time elapses, the analog voltage set in step <4> is output to the ANOi pin.
<6> Use the event output destination select register (ELSELRn; n = 00 to 25) to set the trigger signal used for real-time output mode.
<7> Set the DAMDi bit of the DAM register to 1 (real-time output mode).
<8> Start the operation of the event source.

Steps <1> to <8> above constitute the initial settings.

<9> Upon generation of the trigger signals used for real-time output mode, D/A conversion starts and the analog voltage set in step <4> will be output to the ANOi pin after a settling time has elapsed.
Set the analog voltage value to be output to the ANOi pin, to the DACSi register before performing the next D/A conversion (trigger signal used for real-time output mode is generated).

Set the analog voltage value to be output to the ANOi pin, to the DACSi register before performing the next D/A conversion (trigger signal used for real-time output mode is generated).
When the DACEi bit of the DAM register is set to 0 (D/A conversion operation stop), D/A conversion stops.

Caution 1. Even if 1, 0, and then 1 is set to the DACEi bit, the analog voltage set by the DACSi register is output to the ANOi pin when a settling time has elapsed after 1 is set for the last time.

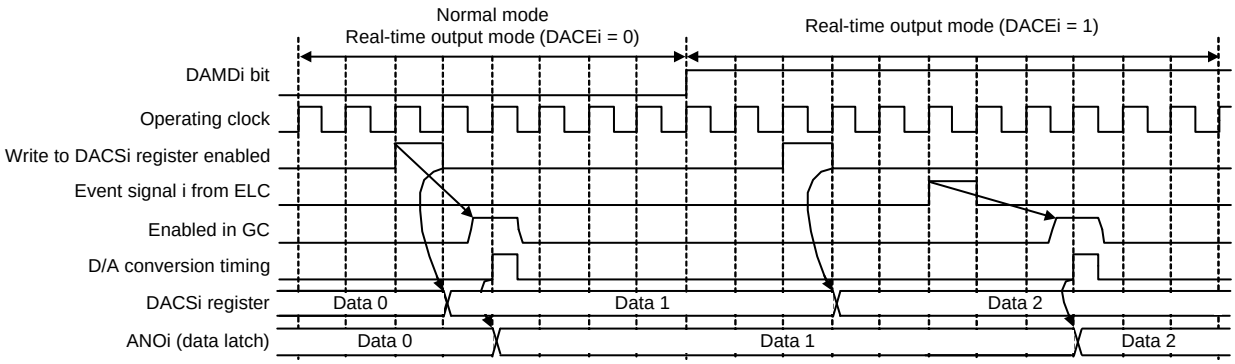
Caution 2. Set the interval between each generation of the trigger signal used for real-time output mode of the same channel to longer than the settling time. If a trigger signal used for real-time output mode is generated during the settling time, D/A conversion is aborted and reconversion starts.

Caution 3. Set the interval between each generation of the trigger signal used for real-time output mode of the same channel to longer than the three clocks of f_{CLK}. When a trigger is generated consecutively at intervals of three or fewer f_{CLK} clock cycles, D/A conversion is performed using only the first trigger.

13.4.3 Timing for Outputting D/A Conversion Value

Figure 13 - 7 shows the Timing for Outputting D/A Conversion Value.

Figure 13 - 7 Timing for Outputting D/A Conversion Value



Remark i = 0, 1

- Normal mode and real-time output mode (when conversion operation is disabled)
  The value is written to the data latch after one cycle of the operating clock when the DACSi register is written.
- Real-time output mode (when conversion operation is enabled)
  The value is written to the data latch (output from the ANOi pin) after three cycles of the operating clock when the event signal from the ELC is accepted.

13.5 Cautions for D/A Converter

Observe the following cautions when using the D/A converter.

(1) The operation of the D/A converter continues in the HALT and STOP modes. To lower the power consumption, therefore, clear the DACEi bit to 0, and execute the HALT or STOP instruction after stopping the operation of the D/A converter.

Remark i = 0, 1

(2) To stop the real-time output mode (including when changing to normal mode), one of the following procedures must be used:

- Wait for at least three clocks after stopping the trigger output source and then set bits DACEi and DAMDi to 0.
- After setting bits DACEi and DAMDi, set the DACEN bit of the PER1 register to 0 (DAC stop).
- When the DACEN bit is set to 0, all the registers in the DAC are cleared, so the settings of the SFRs are required to start the operation again.

(3) In real-time output mode, set the value of the DACSi register before a trigger signal used for real-time output mode is generated. Do not change the set value of the DACSi register while the trigger signal is output.

(4) Since the output impedance of the D/A converter is high, no current can be taken out from the ANO0 or ANO1 pin. If the input impedance of the load is low, insert a follower amplifier between the load and the ANO0 and ANO1 pins before use. In addition, the wiring length between the follower amplifier and the load must be as short as possible due to the high output impedance. If the wiring length is long, take measures such as placing a ground pattern around the wiring area.

(5) When entering STOP mode while real-time output mode is enabled, disable linking of ELC events before entering STOP mode.

CHAPTER 14 COMPARATOR

Table 14 - 1 lists the Comparator Pin Configuration.

Table 14 - 1 Comparator Pin Configuration

	100-pin products	80/85-pin products
VCOUT0, IVCMP0, IVREF0	√	√
VCOUT1, IVCMP1, IVREF1	√	—

14.1 Functions of Comparator

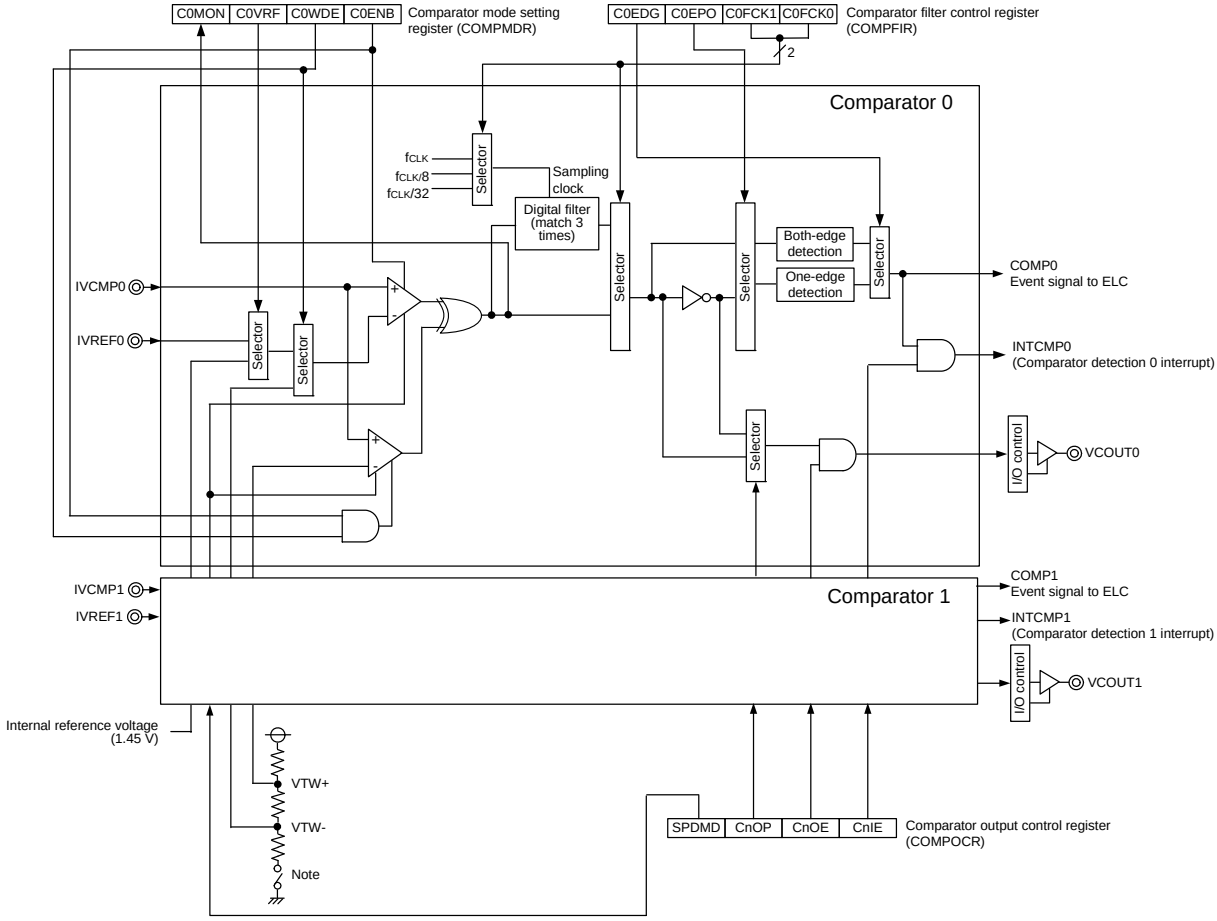
The comparator has the following functions.

- Comparator high-speed mode, comparator low-speed mode, or comparator window mode can be selected.
- The external reference voltage input or internal reference voltage can be selected as the reference voltage.
- The canceling width of the noise canceling digital filter can be selected.
- An interrupt signal can be generated by detecting an active edge of the comparator output.
- An event link controller (ELC) event signal can be output by detecting an active edge of the comparator output.

14.2 Configuration of Comparator

Figure 14 - 1 shows the Comparator Block Diagram.

Figure 14 - 1 Comparator Block Diagram



Note When setting either the C0WDE bit or C1WDE bit, or both bits to 1, this switch is turned ON, and the division resistor to generate the comparison voltage becomes enabled.

Remark n = 0, 1

14.3 Registers

Table 14 - 2 lists the Comparator Register Configuration.

Table 14 - 2 Comparator Register Configuration

Register Name	Symbol
Peripheral enable register 1	PER1
Comparator mode setting register	COMPMDR
Comparator filter control register	COMPFIR
Comparator output control register	COMPOCR
Port mode control register 4	PMC4
Port mode register 1	PM1
Port mode register 4	PM4
Port register 1	P1
Port register 4	P4

14.3.1 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable use of each peripheral hardware macro. 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 5 (CMPEN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Figure 14 - 2 Format of Peripheral enable register 1 (PER1)

Address: F007AH After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	2	1	<0>
PER1	TMKAEN	0	CMPEN	TKB20EN	DTCEN	0	0	DACEN

CMPEN	Control of comparator input clock
0	Stops input clock supply. • SFR used by the comparator cannot be written. • The Comparator is in the reset status.
1	Supplies input clock. • SFR used by the comparator can be read/written.

Caution 1. When setting the comparator, be sure to set the CMPEN bit to 1 first.
If CMPEN = 0, writing to a control register of the comparator is ignored, and all read values are default values (except for port mode register 1 (PM1), port register 1 (P1), and port mode control register 4 (PMC4)).

Caution 2. Be sure to clear bits 6, 2, and 1 to 0.

14.3.2 Comparator mode setting register (COMPMDR)

Figure 14 - 3 Format of Comparator mode setting register (COMPMDR)

Address: F0340H After reset: 00H R/W Note 1

Symbol	<7>	6	5	<4>	<3>	2	1	<0>
COMPMDR	C1MON	C1VRF	C1WDE	C1ENB	C0MON	C0VRF	C0WDE	C0ENB

C1MON	Comparator 1 monitor flag Notes 1, 4
0	In standard mode: IVCMP1 < comparator 1 reference voltage or comparator 1 stopped In window mode: IVCMP1 < low-voltage reference or IVCMP1 > high-voltage reference
1	In standard mode: IVCMP1 > comparator 1 reference voltage In window mode: Low-voltage reference < IVCMP1 < high-voltage reference

C1VRF	Comparator 1 reference voltage selection Notes 2, 5, 6, 7
0	Comparator 1 reference voltage is IVREF1 input
1	Comparator 1 reference voltage is internal reference voltage (1.45 V)

C1WDE	Comparator 1 window mode selection Note 3
0	Comparator 1 standard mode
1	Comparator 1 window mode

C1ENB	Comparator 1 operation enable
0	Comparator 1 operation disabled
1	Comparator 1 operation enabled

Note 1. The value written to this bit is ignored.

Note 2. Valid only when standard mode is selected. In window mode, the reference voltage in the comparator is selected regardless of the setting of this bit.

Note 3. Window mode cannot be set when low-speed mode is selected (the SPDMD bit in the COMPOCR register is 0).

Note 4. The initial value is 0 immediately after a reset is released. However, the value is undefined when C0ENB is set to 0 and C1ENB is set to 0 after operation of the comparator is enabled once.

Note 5. The internal reference voltage (1.45 V) can be selected in HS (high-speed main) mode. When the internal reference voltage (1.45 V) is selected in HS (high-speed main) mode, the temperature sensor output voltage cannot be A/D converted by the A/D converter.

Note 6. Do not select the internal reference voltage in STOP mode.

Note 7. Do not select the internal reference voltage when the subsystem clock (f_{SUB}) is selected as the CPU clock and the high-speed system clock (f_{MX}) and the high-speed on-chip oscillator clock (f_{HOCO}) are both stopped.

C0MON	Comparator 0 monitor flag Notes 1, 4
0	In standard mode: IVCMP0 < comparator 0 reference voltage or comparator 0 stopped In window mode: IVCMP0 < low-voltage reference or IVCMP0 > high-voltage reference
1	In standard mode: IVCMP0 > comparator 0 reference voltage In window mode: Low-voltage reference < IVCMP0 < high-voltage reference

C0VRF	Comparator 0 reference voltage selection Notes 2, 5, 6, 7
0	Comparator 0 reference voltage is IVREF0 input
1	Comparator 0 reference voltage is internal reference voltage (1.45 V)

C0WDE	Comparator 0 window mode selection Note 3
0	Comparator 0 standard mode
1	Comparator 0 window mode

C0ENB	Comparator 0 operation enable
0	Comparator 0 operation disabled
1	Comparator 0 operation enabled

Note 1. The value written to this bit is ignored.

Note 2. Valid only when standard mode is selected. In window mode, the reference voltage in the comparator is selected regardless of the setting of this bit.

Note 3. Window mode cannot be set when low-speed mode is selected (the SPDMD bit in the COMPOCR register is 0).

Note 4. The initial value is 0 immediately after a reset is released. However, the value is undefined when C0ENB is set to 0 and C1ENB is set to 0 after operation of the comparator is enabled once.

Note 5. The internal reference voltage (1.45 V) can be selected in HS (high-speed main) mode. When the internal reference voltage (1.45 V) is selected in HS (high-speed main) mode, the temperature sensor output voltage cannot be A/D converted by the A/D converter.

Note 6. Do not select the internal reference voltage in STOP mode.

Note 7. Do not select the internal reference voltage when the subsystem clock (f_{SUB}) is selected as the CPU clock and the high-speed system clock (f_{MX}) and the high-speed on-chip oscillator clock (f_{HOCO}) are both stopped.

14.3.3 Comparator filter control register (COMPFIR)

Figure 14 - 4 Format of Comparator filter control register (COMPFIR)

Address: F0341H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
COMPFIR	C1EDG	C1EPO	C1FCK1	C1FCK0	C0EDG	C0EPO	C0FCK1	C0FCK0

C1EDG	Comparator 1 edge detection selection Note 1
0	Interrupt request by comparator 1 one-edge detection
1	Interrupt request by comparator 1 both-edge detection

C1EPO	Comparator 1 edge polarity switching Note 1
0	Interrupt request at comparator 1 rising edge
1	Interrupt request at comparator 1 falling edge

C1FCK1	C1FCK0	Comparator 1 filter selection Note 1
0	0	No comparator 1 filter
0	1	Comparator 1 filter enabled, sampling at f_{CLK}
1	0	Comparator 1 filter enabled, sampling at $f_{CLK}/8$
1	1	Comparator 1 filter enabled, sampling at $f_{CLK}/32$

C0EDG	Comparator 0 edge detection selection Note 2
0	Interrupt request by comparator 0 one-edge detection
1	Interrupt request by comparator 0 both-edge detection

C0EPO	Comparator 0 edge polarity switching Note 2
0	Interrupt request at comparator 0 rising edge
1	Interrupt request at comparator 0 falling edge

C0FCK1	C0FCK0	Comparator 0 filter selection Note 2
0	0	No comparator 0 filter
0	1	Comparator 0 filter enabled, sampling at f_{CLK}
1	0	Comparator 0 filter enabled, sampling at $f_{CLK}/8$
1	1	Comparator 0 filter enabled, sampling at $f_{CLK}/32$

Note 1. If bits C1FCK1 to C1FCK0, C1EPO, and C1EDG are changed, a comparator 1 interrupt request and an event signal to the ELC may be generated. Change these bits only after setting the ELSELR19 register for the ELC to 0 (not linked to comparator 1 output), and bits 15 to 12 and 6 to 4 in forced output stop function control registers np (TKBPACTLnp, n = 0, 1, 2, p =0, 1) to 0. In addition, clear bit 7 (CPMIF1) in interrupt request flag register 2L (IF2L) to 0.

If bits C1FCK1 to C1FCK0 are changed from 00B (no comparator 1 filter) to a value other than 00B (comparator 1 filter enabled), allow the time for sampling four times to elapse until the filter output is updated, and then use the comparator 1 interrupt request or the event signal to the ELC.

Note 2. If bits C0FCK1 to C0FCK0, C0EPO, and C0EDG are changed, a comparator 0 interrupt request and an event signal to the ELC may be generated. Change these bits only after setting the ELSELR18 register for the ELC to 0 (not linked to comparator 1 output), and bits 15 to 12 and 6 to 4 in forced output stop function control registers np (TKBPACTLnp, n = 0, 1, 2, p =0, 1) to 0. In addition, clear bit 6 (CPMIF0) in interrupt request flag register 2L (IF2L) to 0.

If bits C0FCK1 to C0FCK0 are changed from 00B (no comparator 0 filter) to a value other than 00B (comparator 0 filter enabled), allow the time for sampling four times to elapse until the filter output is updated, and then use the comparator 0 interrupt request or the event signal to the ELC.

14.3.4 Comparator output control register (COMPOCR)

Figure 14 - 5 Format of Comparator output control register (COMPOCR)

Address: F0342H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
COMPOCR	SPDMD	C1OP	C1OE	C1IE	0	C0OP	C0OE	C0IE

SPDMD	Comparator speed selection Note 1
0	Comparator low-speed mode
1	Comparator high-speed mode

C1OP	VCOUT1 output polarity selection
0	Comparator 1 output is output to VCOUT1
1	Inverted comparator 1 output is output to VCOUT1

C1OE	VCOUT1 pin output enable
0	Comparator 1 VCOUT1 pin output disabled
1	Comparator 1 VCOUT1 pin output enabled

C1IE	Comparator 1 interrupt request enable Note 2
0	Comparator 1 interrupt request disabled
1	Comparator 1 interrupt request enabled

C0OP	VCOUT0 output polarity selection
0	Comparator 0 output is output to VCOUT0
1	Inverted comparator 0 output is output to VCOUT0

C0OE	VCOUT0 pin output enable
0	Comparator 0 VCOUT0 pin output disabled
1	Comparator 0 VCOUT0 pin output enabled

C0IE	Comparator 0 interrupt request enable Note 3
0	Comparator 0 interrupt request disabled
1	Comparator 0 interrupt request enabled

Note 1. When rewriting the SPDMD bit, be sure to set the CiENB bit (i = 0 or 1) in the COMPMDR register to 0 in advance.

Note 2. If C1IE is changed from 0 (interrupt request disabled) to 1 (interrupt request enabled), since bit 7 (CPMIF1) in interrupt request flag register 2L (IF2L) may set to 1 (interrupt requested), clear bit 7 (CPMIF1) in interrupt request flag register 2L (IF2L) to 0 before using an interrupt.

Note 3. If C0IE is changed from 0 (interrupt request disabled) to 1 (interrupt request enabled), since bit 6 (CPMIF0) in interrupt request flag register 2L (IF2L) may set to 1 (interrupt requested), clear bit 6 (CPMIF0) in interrupt request flag register 2L (IF2L) to 0 before using an interrupt.

14.3.5 Port mode control register 4 (PMC4)

This register specifies digital I/O or analog I/O mode for P41 to P46 in 1-bit units.

The PMC1 register can be set by a 1-bit or 8-bit manipulation instruction.

Reset signal generation resets this register to 00H.

Figure 14 - 6 Format of Port mode control register 4 (PMC4)

Address: F0064H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PMC4	0	PMC46	PMC45	PMC44	PMC43	PMC42	PMC41	0

PMC4n	P4n pin digital I/O and analog I/O selection (n = 1 to 6)
0	Digital I/O (multiplexed function other than analog I/O)
1	Analog I/O

14.3.6 Port mode register 1 (PM1)

This register specifies I/O mode for port 1 in 1-bit units.

When using the VCOUT0/P11 and VCOUT1/P12 pins for the comparator output function, set the PM11 and PM12 bits and the output latches of P11 and P12 to 0.

The PM1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 14 - 7 Format of Port mode register 1 (PM1)

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

14.3.7 Port mode register 4 (PM4)

When using the IVREF1/P41, IVCMP1/P42, IVCMP0/P43, and IVREF0/P44 pins as analog input ports, set bits PM41 to PM44 to 1. At this time, the output latches of P41 to P44 may be 0 or 1.

If bits PM41 to PM44 are set to 0, these pins cannot be used as analog input ports.

The PM4 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Caution If a pin is set as an analog input port, not the pin level but 0 is always read.

Figure 14 - 8 Format of Port mode register 4 (PM4)

Address: FFF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	0	PM46	PM45	PM44	PM43	PM42	PM41	0

PM4n	P4n pin I/O mode selection (n = 1 to 6)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

14.4 Operation

Comparator 0 and comparator 1 operate independently. Their setting methods and operations are the same. Table 14 - 3 lists the Procedure for Setting Comparator Associated Registers.

Table 14 - 3 Procedure for Setting Comparator Associated Registers

<table>
<tr><th>Step</th><th>Register</th><th>Bit</th><th colspan="3">Setting Value</th></tr>
<tr><td>1</td><td>PER1</td><td>CMPEN</td><td colspan="3">1 (input clock supply)</td></tr>
<tr><td rowspan="2">2</td><td>PMC4</td><td>PMC4n</td><td colspan="3" rowspan="2">Select the function of pins IVCMPi and IVREFi.
Set the PMC4n bit to 1 (analog input).
Set the PM4n bit to 1 (input mode).
Refer to 14.3.5 Port mode control register 4 (PMC4) and 14.3.7 Port mode register 4 (PM4).</td></tr>
<tr><td>PM4</td><td>PM4n</td></tr>
<tr><td>3</td><td>COMPOCR</td><td>SPDMD</td><td colspan="3">Select the comparator response speed (0: Low-speed mode/1: High-speed mode). Note 1</td></tr>
<tr><td rowspan="3">4</td><td rowspan="3">COMPMDR</td><td>CiWDE</td><td colspan="2">0 (standard mode)</td><td>1 (window mode) Note 2</td></tr>
<tr><td>CiVRF</td><td>0
(Reference = IVREFi input)</td><td>1
(Reference = internal reference voltage (1.45 V)) Note 4</td><td>Window comparator operation (reference = internal VREF)</td></tr>
<tr><td>CiENB</td><td colspan="3">1 (operation enabled)</td></tr>
<tr><td>5</td><td colspan="5">Wait for comparator stabilization time (max. 100 μs).</td></tr>
<tr><td rowspan="2">6</td><td rowspan="2">COMPFIR</td><td>CiFCK1 - CiFCK0</td><td colspan="3">Select whether the digital filter is used or not and the sampling clock.</td></tr>
<tr><td>CiEOP, CiEDG</td><td colspan="3">Select the edge detection condition for an interrupt request (rising edge/falling edge/both edges).</td></tr>
<tr><td rowspan="2">7</td><td rowspan="2">COMPOCR</td><td>CiOP, CiOE</td><td colspan="3">Set the VCOUTi output (select the polarity and set output enabled or disabled).
Refer to 14.4.4 Comparator i Output (i = 0 or 1).</td></tr>
<tr><td>CiIE</td><td colspan="3">Set the interrupt request output enabled or disabled.
Refer to 14.4.4 Comparator i Output (i = 0 or 1).</td></tr>
<tr><td>8</td><td>PR2L Note 5</td><td>CMPPR0i,
CMPPR1i</td><td colspan="3">When using an interrupt: Select the interrupt priority level.</td></tr>
<tr><td>9</td><td>MK2L Note 5</td><td>CMPMKi</td><td colspan="3">When using an interrupt: Select the interrupt masking.</td></tr>
<tr><td>10</td><td>IF2L Note 5</td><td>CMPIFi</td><td colspan="3">When using an interrupt: 0 (no interrupt requested: initialization) Note 3</td></tr>
</table>

Note 1. Comparator 0 and comparator 1 cannot be set independently.

Note 2. Can be set in high-speed mode (SPDMD = 1).

Note 3. After the setting of the comparator, an unnecessary interrupt may occur until operation becomes stable, so initialize the interrupt flag.

Note 4. Can be set in HS (high-speed main) mode.

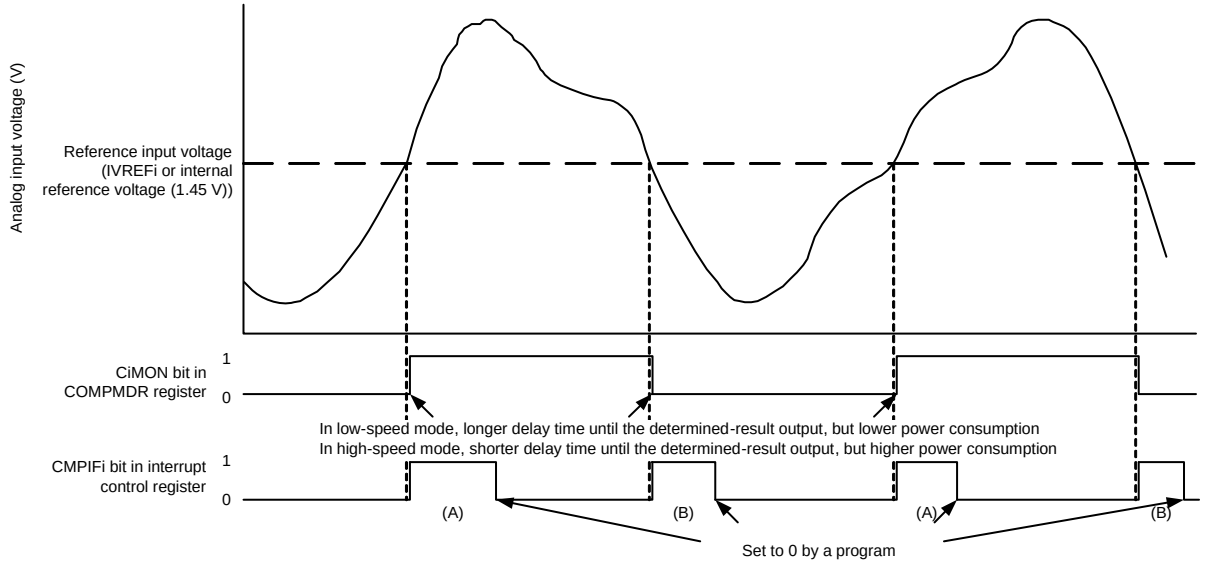
Note 5. PR2L, MK2L, and IF2L are the interrupt control registers for comparator i.

Remark i = 0, 1, n = 2 to 5

Figures 14 - 9 and 14 - 10 show comparator i (i = 0 or 1) operation examples. In standard mode, the CiMON bit in the COMPMDR register is set to 1 when the analog input voltage is higher than the reference input voltage, and the CiMON bit is set to 0 when the analog input voltage is lower than the reference input voltage.
In window mode, the CiMON bit in the COMPMDR register is set to 1 when the analog input voltage meets the following condition, and the CiMON bit is set to 0 when the analog input voltage does not meet the following condition:
“Low-voltage reference voltage < analog input voltage < high-voltage reference voltage”
When using the comparator i interrupt, set CiIE in the COMPOCR register to 1 (interrupt request output enabled). If the comparison result changes at this time, a comparator i interrupt request is generated. For details on interrupt requests, refer to **14.4.2 Comparator i (i = 0 or 1) Interrupts**.

Figure 14 - 9 Comparator i (i = 0 or 1) Operation Example in Standard Mode

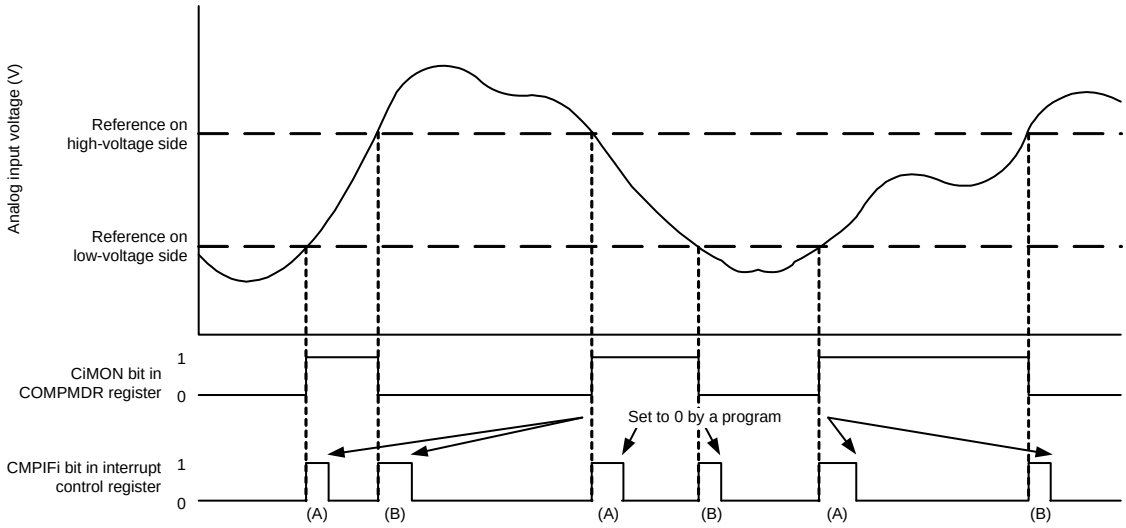
• Operation example in standard mode



Caution The above diagram applies when CiFCK1 - CiFCK0 in the COMPFIR register = 00B (no filter) and CiEDG = 1 (both edges). When CiEDG = 0 and CiEOP = 0 (rising edge), CMPIFi changes as shown by (A) only. When CiEDG = 0 and CiEOP = 1 (falling edge), CMPIFi changes as shown by (B) only.

Figure 14 - 10 Comparator i (i = 0 or 1) Operation Example in Window Mode

• Operation example in window mode



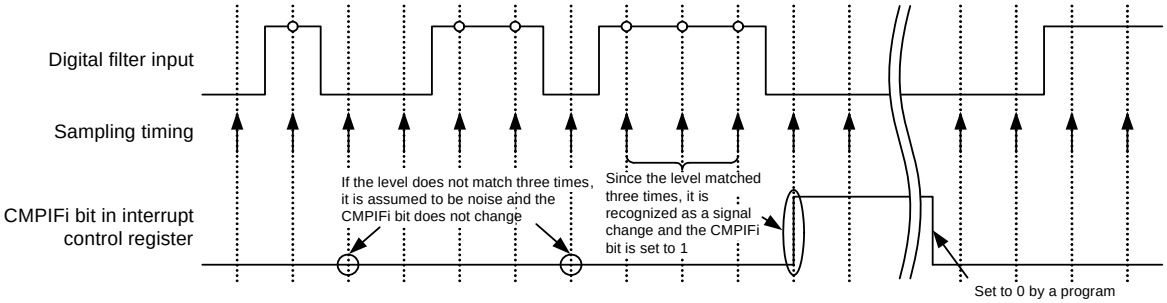
Caution The above diagram applies when CiFCK1 to CiFCK0 in the COMPFIR register = 00B (no filter) and CiEDG = 1 (both edges). When CiEDG = 0 and CiEOP = 0 (rising edge), CMPIFi changes as shown by (A) only. When CiEDG = 0 and CiEOP = 1 (falling edge), CMPIFi changes as shown by (B) only.

14.4.1 Comparator i Digital Filter (i = 0 or 1)

Comparator i contains a digital filter. The sampling clock can be selected by bits CiFCK1 to CiFCK0 in the COMPFIR register. The comparator i output signal is sampled every sampling clock, and when the level matches three times, that value is determined as the digital filter output at the next sampling clock.

Figure 14 - 11 shows the Comparator i (i = 0 or 1) Digital Filter and Interrupt Operation Example.

Figure 14 - 11 Comparator i (i = 0 or 1) Digital Filter and Interrupt Operation Example



Caution The above operation example applies when bits CiFCK1 to CiFCK0 in the COMPFIR register is 01B, 10B, or 11B (digital filter enabled).

14.4.2 Comparator i (i = 0 or 1) Interrupts

The comparator generates interrupt requests from two sources, comparator 0 and comparator 1. The comparator i interrupt each uses a priority level specification flag, an interrupt mask flag, an interrupt request flag, and a single vector.

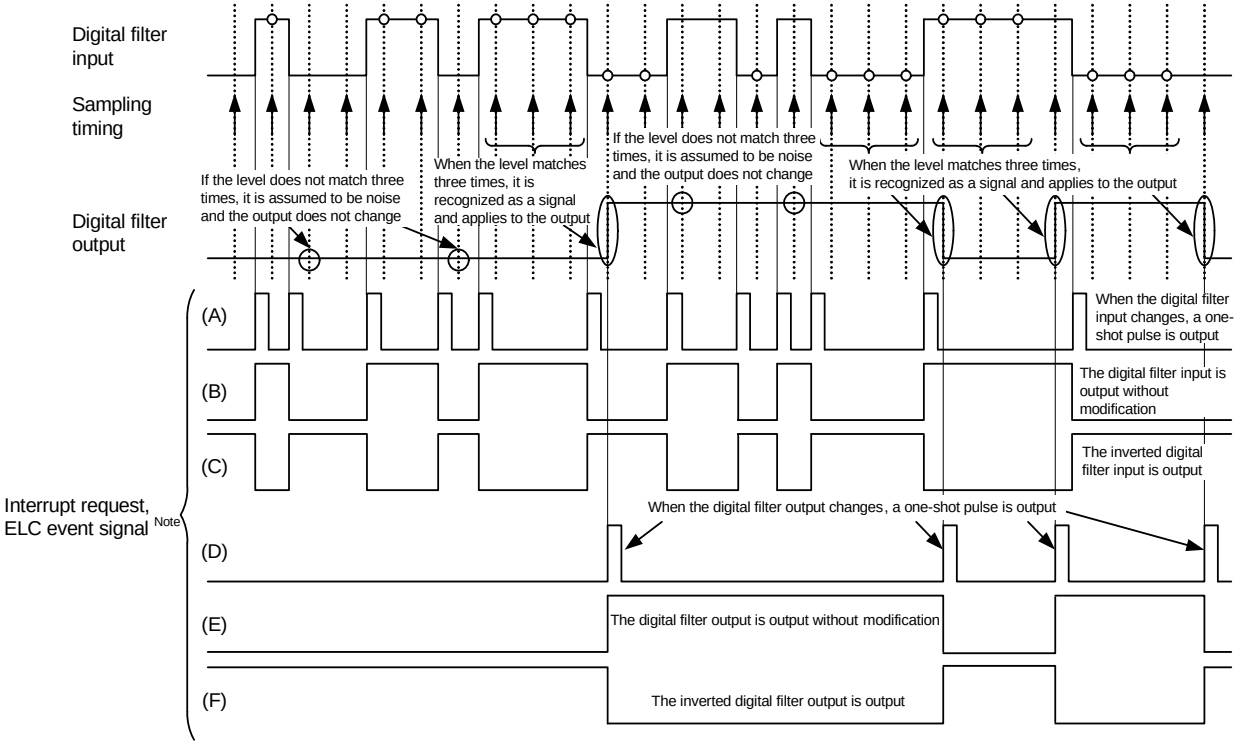
When using the comparator i interrupt, set the CiIE bit in the COMPOCR register to 1 (interrupt request output enabled). The condition for interrupt request generation can be set by the COMPFIR register. The comparator outputs can also be passed through the digital filter. Three different sampling clocks can be selected for the digital filter.

For details on the register setting and interrupt request generation, refer to **14.3.3 Comparator filter control register (COMPFIR)** and **14.3.4 Comparator output control register (COMPOCR)**.

14.4.3 Event signal output to event link controller (ELC)

An event signal to the ELC is generated by detecting the edge for the digital filter output set by the COMPFIR register, which is the same as the condition for interrupt request generation. However, unlike interrupt requests, ELC events are always output regardless of the CiIE bit in the COMPOCR register. Set registers ELSELR18 and ELSELR19 for the ELC to select the event output destination and to stop linking events.

Figure 14 - 12 Digital Filter and Interrupt Request/Event Signal Output to the ELC Operation



Note When the CiIE bit (i = 0, 1) is 1, the same waveform is generated for an interrupt request and an ELC event.
When the CiIE bit (i = 0, 1) is 0, the value is fixed at 0 for an interrupt request only.

The waveforms of (A), (B), and (C) are shown for an operation example when the CiFCK bits (i = 0, 1) in the COMPFIR register are 00B (no digital filter). The waveforms (D), (E), and (F) are shown for an operation example when the CiFCK bits (i = 0, 1) in the COMPFIR register are 01B, 10B, or 11B (digital filter enabled).
(A) and (D) apply when the CiEDG bit is set to 1 (both edges), (B) and (E) when the CiEDG bit is 0 and the CiEPO bit is 0 (rising edge), and (C) and (F) when the CiEDG bit is 0 and the CiEPO bit is 1 (falling edge).

14.4.4 Comparator i Output (i = 0 or 1)

The comparison result from the comparator can be output to external pins. Bits CiOP and CiOE in the COMPOCR register can be used to set the output polarity (non-inverted output or inverted output) and output enabled or disabled. For the correspondence between the register setting and the comparator output, refer to **14.3.4 Comparator output control register (COMPOCR)**.

To output the comparator comparison result to the VCOUTi output pin, use the following procedure to set the ports. Note that the ports are set to input after reset.

<1> Set the mode for the comparator (Steps 1 to 4 as listed in Table 14 - 3 Procedure for Setting Comparator Associated Registers).
<2> Set the VCOUTi output for the comparator (set the COMPOCR register to select the polarity and enable the output).
<3> Set the corresponding port register bit for the VCOUTi output pin to 0.
<4> Set the corresponding port mode register for the VCOUTi output pin to output (start outputting from the pin).

14.4.5 Stopping or Supplying Comparator Clock

To stop the comparator clock by setting peripheral enable register 1 (PER1), use the following procedure:

<1> Set the CiENB bit in the COMPMDR register to 0 (stop the comparator).
<2> Set the CMPIFi bit in registers IF2L and IF2H to 0 (clear any unnecessary interrupt before stopping the comparator).
<3> Set the CMPEN bit in the PER1 register to 0.

When the clock is stopped by setting PER1, all the internal registers in the comparator are initialized. To use the comparator again, follow the procedure in Table 14 - 3 to set the registers.

Caution 1. The temperature sensor output voltage cannot be A/D converted by the A/D converter while the comparator n reference voltage select bit (CnVRF) in the comparator mode setting register (COMPMDR) is 1 (comparator n reference voltage is internal reference voltage (1.45 V)).

Caution 2. When DTC activation is enabled under either of the following conditions, a DTC transfer is started and an interrupt is generated after completion of the transfer. Therefore, enable DTC activation after confirming the comparator monitor flag (CnMON) as necessary. (n = 0, 1)

- **The comparator is set to an interrupt request on one-edge detection (CnEDG = 0), an interrupt request at the rising edge for the comparator, and IVCMP > IVREF (or internal reference voltage: 1.45 V)**
- **The comparator is set to an interrupt request on one-edge detection (CnEDG = 0), an interrupt request at the falling edge for the comparator, and IVCMP < IVREF (or internal reference voltage: 1.45 V)**

CHAPTER 15 SERIAL ARRAY UNIT

Serial array unit has up to four serial channels. Each channel can achieve Simplified SPI (CSI[Note]), UART, and simplified I^2C communication.
Function assignment of each channel supported by the RL78/L1C 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.

• 80-pin, 85-pin, and 100-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0 (supporting LIN-bus)	IIC00
	1	—		—
	2	CSI10	UART1	IIC10
	3	—		—
1	0	CSI20	UART2	IIC20
	1	—		—
	2	CSI30	UART3	IIC30
	3	—		—

When “UART0” is used for channels 0 and 1 of the unit 0, CSI00 and IIC00 cannot be used, but CSI10, UART1, or IIC10 can be used for channels 2 and 3.

Caution Most of the following descriptions in this chapter use the units and channels of the 100-pin products as an example.

15.1 Functions of Serial Array Unit

Each serial interface supported by the RL78/L1C has the following features.

15.1.1 Simplified SPI (CSI00, CSI10, CSI20, CSI30)

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, CSI10, CSI20, CSI30) Communication**.

[Data transmission/reception]

- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable

[Clock control]

- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel
- Maximum transfer rate Note

  During master communication: Max. f_{CLK}/2 (CSI00 only)

  Max. f_{CLK}/4

  During slave communication: Max. f_{MCK}/6

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

In addition, CSI00 and CSI20 support 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 CSI00 and CSI20 can be specified for asynchronous reception.

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **CHAPTER 34** or **CHAPTER 35 ELECTRICAL SPECIFICATIONS**.

15.1.2 UART (UART0 to UART3)

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 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 to UART3) 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 and UART2 reception support the SNOOZE mode. When RxD input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only UART0, UART2 can be specified for asynchronous reception.

The LIN-bus is accepted in UART0 (0 and 1 channels of unit 0).

[LIN-bus functions]

- Wakeup signal detection
- Break field (BF) detection
- Sync field measurement, baud rate calculation

} Using the external interrupt (INTP0) and timer array unit

Note Only UART0, UART2 can be specified for the 9-bit data length.

15.1.3 Simplified I²C (IIC00, IIC10, IIC20, IIC30)

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I²C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master.
Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.
For details about the settings, see **15.8 Operation of Simplified I²C (IIC00, IIC10, IIC20, IIC30) Communication**.

[Data transmission/reception]
- Master transmission, master reception (only master function with a single master)
- ACK output function Note and ACK detection function
- Data length of 8 bits (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Manual generation of start condition and stop condition

[Interrupt function]
- Transfer end interrupt

[Error detection flag]
- ACK error, or overrun error

* [Functions not supported by simplified I²C]
- Slave transmission, slave reception
- Arbitration loss detection function
- Clock stretch detection functions

Note When receiving the last data, ACK will not be output if 0 is written to the SOEmn bit (serial output enable register m (SOEm)) and serial communication data output is stopped. See the processing flow in **15.8.3 (2)** for details.

Remark 1. To use an I²C bus of full function, see **CHAPTER 16 SERIAL INTERFACE IICA**.
Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12

15.2 Configuration of Serial Array Unit

The serial array unit includes the following hardware.

Table 15 - 1 Configuration of Serial Array Unit

Item	Configuration
Shift register	8 bits or 9 bits Note 1
Buffer register	Lower 8 bits or 9 bits of serial data register mn (SDRmn) Notes 1, 2
Serial clock I/O	SCK00, SCK10, SCK20, SCK30 pins (for Simplified SPI), SCL00, SCL10, SCL20, SCL30 pins (for simplified I^2C)
Serial data input	SI00, SI10, SI20, SI30 pins (for Simplified SPI), RxD1 to RxD3 pins (for UART), RXD0 pin (for UART supporting LIN-bus)
Serial data output	SO00, SO10, SO20, SO30 pins (for Simplified SPI), TxD1 to TxD3 pins (for UART), TXD0 pin (for UART supporting LIN-bus)
Serial data I/O	SDA00, SDA10, SDA20, SDA30 pins (for simplified I^2C)
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Serial clock select register m (SPSm) • Serial channel enable status register m (SEm) • Serial channel start register m (SSm) • Serial channel stop register m (STm) • Serial output enable register m (SOEm) • Serial output register m (SOm) • Serial output level register m (SOLm) • Serial standby control register m (SSCm) • Input switch control register (ISC) • Noise filter enable register 0 (NFEN0)
	<Registers of each channel> • Serial data register mn (SDRmn) • Serial mode register mn (SMRmn) • Serial communication operation setting register mn (SCRmn) • Serial status register mn (SSRmn) • Serial flag clear trigger register mn (SIRmn)
	• Port input mode registers 0 to 4 (PIM0 to PIM4) • Port output mode registers 0 to 4 (POM0 to POM4) • Port mode registers 0 to 4 (PM0 to PM4) • Port registers 0 to 4 (P0 to P4)

Note 1. The number of bits used as the shift register and buffer register differs depending on the unit and channel.
- mn = 00, 01, 10, 11: lower 9 bits
- Other than above: lower 8 bits

Note 2. The lower 8 bits of serial data register mn (SDRmn) can be read or written as the following SFR, depending on the communication mode.
- CSIp communication ..... SIOp (CSIp data register)
- UARTq reception ..... RXDq (UARTq receive data register)
- UARTq transmission ..... TXDq (UARTq transmit data register)
- IICr communication ..... SIOr (IICr data register)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 10, 20, 30), q: UART number (q = 0 to 3), r: IIC number (r = 00, 10, 20, 30)

Figure 15 - 1 shows the Block Diagram of Serial Array Unit 0.

Figure 15 - 1 Block Diagram of Serial Array Unit 0

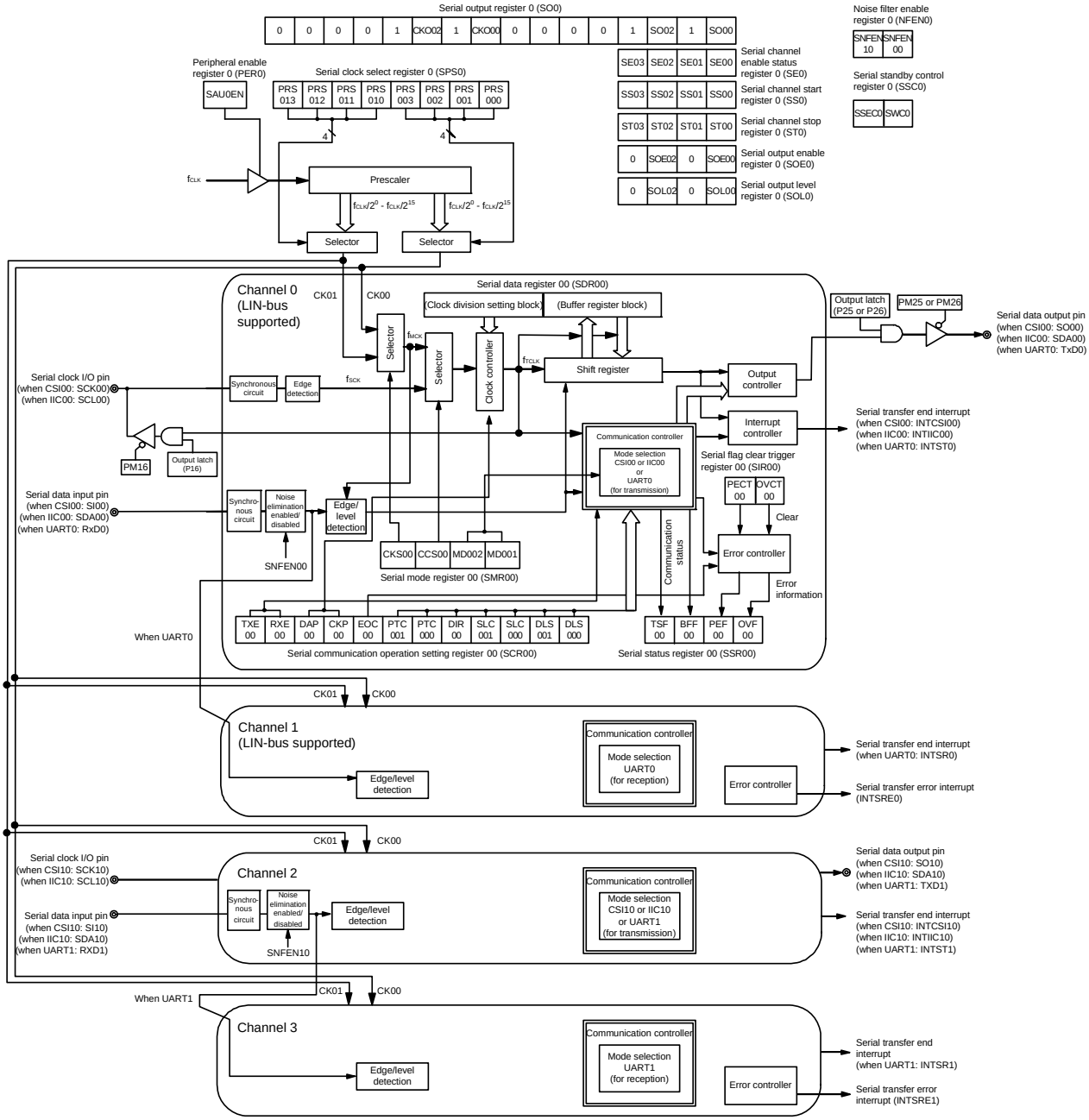
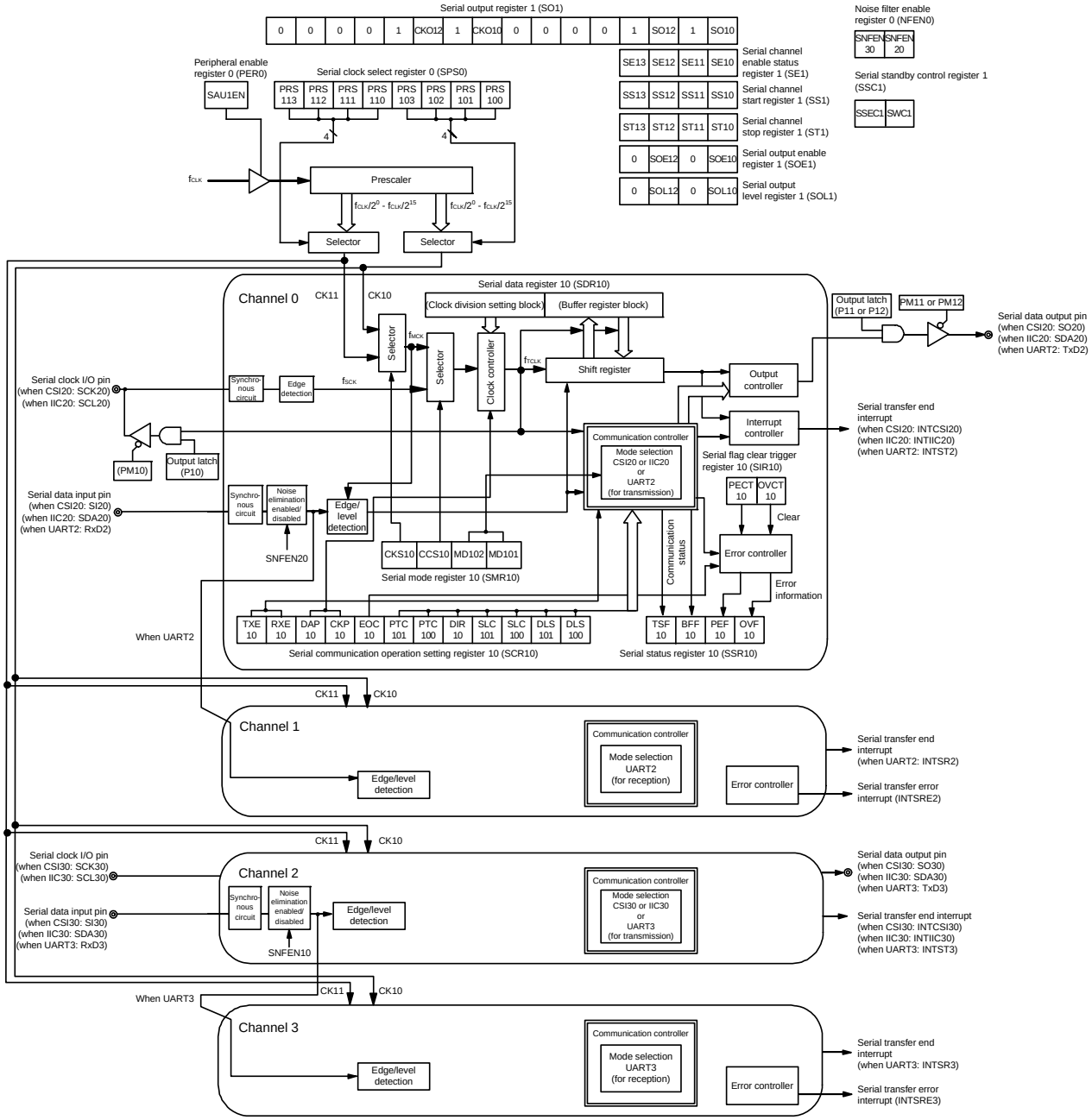


Figure 15 - 2 shows the Block Diagram of Serial Array Unit 1 (100-pin products).

Figure 15 - 2 Block Diagram of Serial Array Unit 1 (100-pin products)



15.2.1 Shift register

This is a 9-bit register that converts parallel data into serial data or vice versa.

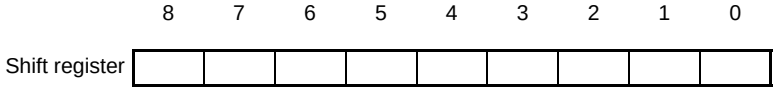
In case of the UART communication of nine bits of data, nine bits (bits 0 to 8) are used Note 1.

During reception, it converts data input to the serial pin into parallel data.

When data is transmitted, the value set to this register is output as serial data from the serial output pin.

The shift register cannot be directly manipulated by program.

To read or write the shift register, use the lower 8/9 bits of serial data register mn (SDRmn).



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 data to be transferred to the shift register to the lower 8/9 bits.

The data stored in the lower 8/9 bits of this register is as follows, depending on the setting of bits 0 and 1 (DLSmn0, DLSmn1) of serial communication operation setting register mn (SCRmn), regardless of the output sequence of the data.

- 7-bit data length (stored in bits 0 to 6 of SDRmn register)
- 8-bit data length (stored in bits 0 to 7 of SDRmn register)
- 9-bit data length (stored in bits 0 to 8 of SDRmn register) Note 1

The SDRmn register can be read or written in 16-bit units.

The lower 8/9 bits of the SDRmn register can be read or written Note 2 as the following SFR, depending on the communication mode.

- CSIp communication..... SIOp (CSIp data register)
- UARTq reception ..... RXDq (UARTq receive data register)
- UARTq transmission..... TXDq (UARTq transmit data register)
- IICr communication..... SIOr (IICr data register)

Reset signal generation clears the SDRmn register to 0000H.

Note 1. Only UART0, UART2 can be specified for the 9-bit data length.

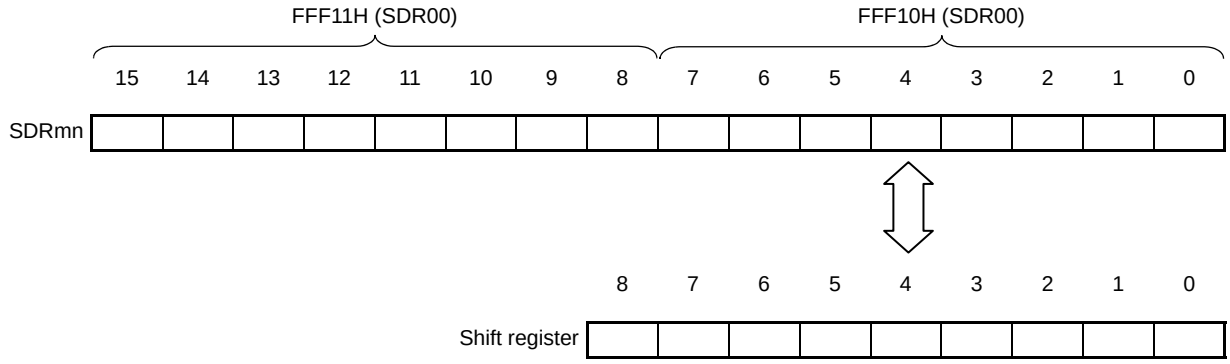
Note 2. When operation is stopped (SEmn = 0), do not rewrite SDRmn[7:0] by an 8-bit memory manipulation instruction (SDRmn[15:9] are all cleared to 0).

Remark 1. After data is received, “0” is stored in bits 0 to 8 in bit portions that exceed the data length.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 10, 20, 30), q: UART number (q = 0 to 3), r: IIC number (r = 00, 10, 20, 30)

Figure 15 - 3 Format of Serial data register mn (SDRmn) (mn = 00, 01, 10, 11)

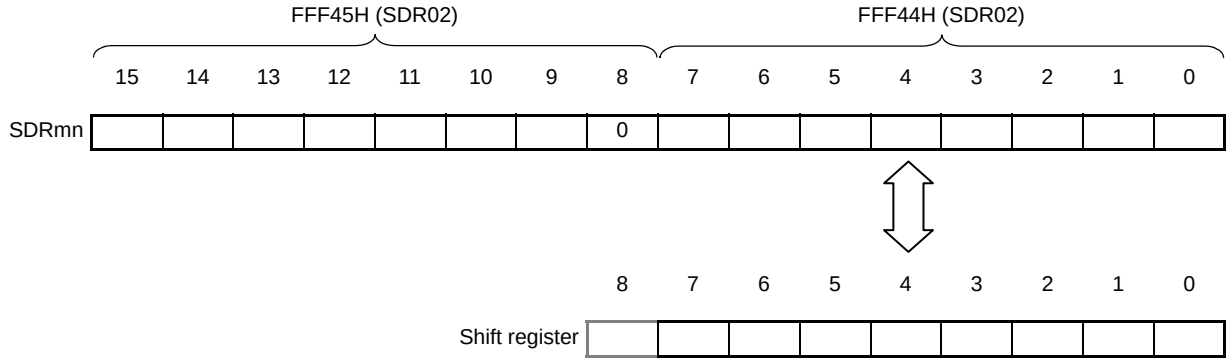
Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01) After reset: 0000H R/W
FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11)



Remark For the function of the higher 7 bits of the SDRmn register, see **15.3 Registers Controlling Serial Array Unit**.

Figure 15 - 4 Format of Serial data register mn (SDRmn) (mn = 02, 03, 12, 13)

Address: FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03), After reset: 0000H R/W
FFF14H, FFF15H (SDR12), FFF16H, FFF17H (SDR13)



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**.

15.3 Registers Controlling Serial Array Unit

Serial array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Serial clock select register m (SPSm)
- Serial mode register mn (SMRmn)
- Serial communication operation setting register mn (SCRmn)
- Serial data register mn (SDRmn)
- Serial flag clear trigger register mn (SIRmn)
- Serial status register mn (SSRmn)
- Serial channel start register m (SSm)
- Serial channel stop register m (STm)
- Serial channel enable status register m (SEm)
- Serial output enable register m (SOEm)
- Serial output level register m (SOLm)
- Serial output register m (SOm)
- Serial standby control register m (SSCm)
- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port input mode registers 0 to 4 (PIM0 to PIM4)
- Port output mode registers 0 to 4 (POM0 to POM4)
- Port mode registers 0 to 4 (PM0 to PM4)
- Port registers 0 to 4 (P0 to P4)

Remark m: Unit number (m = 0, 1), 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.

When serial array unit 1 is used, be sure to set bit 3 (SAU1EN) 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 - 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	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

SAUmEN	Control of serial array unit m input clock supply
0	Stops supply of input clock. • SFR used by serial array unit m cannot be written. • Serial array unit m is in the reset status.
1	Enables input clock supply. • SFR used by serial array unit m can be read/written.

Caution 1. When setting serial array unit m, be sure to first set the following registers with the SAUmEN bit set to 1. If SAUmEN = 0, control registers of serial array unit m 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 to 4 (PIM0 to PIM4), port output mode registers 0 to 4 (POM0 to POM4), port mode registers 0 to 4 (PM0 to PM4), and port registers 0 to 4 (P0 to P4)).

- **Serial clock select register m (SPSm)**
- **Serial mode register mn (SMRmn)**
- **Serial communication operation setting register mn (SCRmn)**
- **Serial data register mn (SDRmn)**
- **Serial flag clear trigger register mn (SIRmn)**
- **Serial status register mn (SSRmn)**
- **Serial channel start register m (SSm)**
- **Serial channel stop register m (STm)**
- **Serial channel enable status register m (SEm)**
- **Serial output enable register m (SOEm)**
- **Serial output level register m (SOLm)**
- **Serial output register m (SOm)**
- **Serial standby control register m (SSCm)**

Caution 2. Be sure to clear bits 1 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 - 6 Format of Serial clock select register m (SPSm)

Address: F0126H, F0127H (SPS0), F0166H, F0167H (SPS1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPSm	0	0	0	0	0	0	0	0	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Section of operation clock (CKmk) Note					
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz	24 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	12 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	6 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	3 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz	1.5 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz	750 kHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz	375 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	187.5 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	93.8 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	46.9 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	23.4 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	11.7 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	5.86 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz	2.93 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.46 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	153 Hz	305 Hz	610 Hz	732 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Caution Be sure to clear bits 15 to 8 to "0".

Remark 1. f_{CLK}: CPU/peripheral hardware clock frequency
Remark 2. m: Unit number (m = 0, 1)
Remark 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), UART, or simplified I^2C), and an interrupt source. This register is also used to invert the level of the receive data only in the UART mode.

Rewriting the SMRmn register is prohibited when the 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 - 7 Format of Serial mode register mn (SMRmn) (1/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03), F0150H, F0151H (SMR10) to F0156H, F0157H (SMR13) — After reset: 0020H — R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSmn Note	0	SISmn0 Note	1	0	0	MDmn2	MDmn1	MDmn0

CKSmn	Selection of operation clock (f_{MCK}) of channel n
0	Operation clock CKm0 set by the SPSm register
1	Operation clock CKm1 set by the SPSm register
Operation clock (f_{MCK}) is used by the edge detector. In addition, depending on the setting of the CCSmn bit and the higher 7 bits of the SDRmn register, a transfer clock (f_{TCLK}) is generated.	

CCSmn	Selection of transfer clock (f_{TCLK}) of channel n
0	Divided operation clock f_{MCK} specified by the CKSmn bit
1	Clock input f_{SCK} from the SCKp pin (slave transfer in Simplified SPI (CSI) mode)
Transfer clock f_{TCLK} is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCSmn = 0, the division ratio of operation clock (f_{MCK}) is set by the higher 7 bits of the SDRmn register.	

STSmn Note	Selection of start trigger source
0	Only software trigger is valid (selected for Simplified SPI (CSI), UART transmission, and simplified I^2C).
1	Valid edge of the RxDq pin (selected for UART reception)
Transfer is started when the above source is satisfied after 1 is set to the SSm register.	

Note The SMR01, SMR03, SMR11, and SMR13 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, SMR02, SMR10, or SMR12 register) to "0". Be sure to set bit 5 to "1".**

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 10, 20, 30), q: UART number (q = 0 to 3), r: IIC number (r = 00, 10, 20, 30)

Figure 15 - 8 Format of Serial mode register mn (SMRmn) (2/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03), F0150H, F0151H (SMR10) to F0156H, F0157H (SMR13) After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn Note	0	SIS mn0 Note	1	0	0	MD mn2	MD mn1	MD mn0

SIS mn0 Note	Controls inversion of level of receive data of channel n in UART mode
0	Falling edge is detected as the start bit. The input communication data is captured as is.
1	Rising edge is detected as the start bit. The input communication data is inverted and captured.

MD mn2	MD mn1	Setting of operation mode of channel n
0	0	Simplified SPI (CSI) mode
0	1	UART mode
1	0	Simplified I^2C mode
1	1	Setting prohibited

MD mn0	Selection of interrupt source of channel n
0	Transfer end interrupt
1	Buffer empty interrupt (Occurs when data is transferred from the SDRmn register to the shift register.)
For successive transmission, the next transmit data is written by setting the MDmn0 bit to 1 when SDRmn data has run out.	

Note The SMR01, SMR03, SMR11, and SMR13 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, SMR02, SMR10, or SMR12 register) to “0”. Be sure to set bit 5 to “1”.**

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 10, 20, 30), q: UART number (q = 0 to 3), r: IIC number (r = 00, 10, 20, 30)

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.

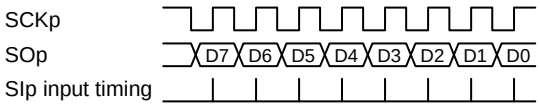
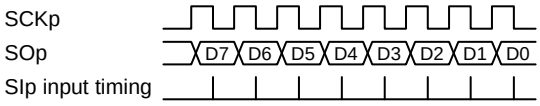
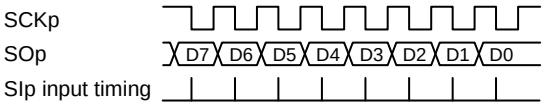
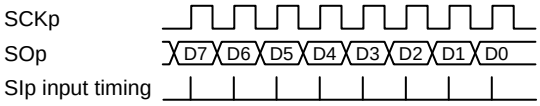
Reset signal generation sets the SCRmn register to 0087H.

Figure 15 - 9 Format of Serial communication operation setting register mn (SCRmn) (1/2)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03), F0158H, F0159H (SCR10) to F015EH, F015FH (SCR13) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLCm n1 Note 1	SLC mn0	0	1	DLSm n1 Note 2	DLS mn0

TXE mn	RXE mn	Setting of operation mode of channel n
0	0	Disable communication.
0	1	Reception only
1	0	Transmission only
1	1	Transmission/reception

DAP mn	CKP mn	Selection of data and clock phase in Simplified SPI (CSI) mode	Type
0	0		1
0	1		2
1	0		3
1	1		4
Be sure to set DAPmn, CKPmn = 0, 0 in the UART mode and simplified I²C mode.			

EOC mn	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, simplified I²C mode, and during UART transmission Note 3.	

Note 1. The SCR00, SCR02, SCR10, and SCR12 registers only.

Note 2. The SCR00, SCR01, SCR10 and SCR11 registers only. Others are fixed to 1.

Note 3. When using CSImn not with EOCmn = 0, error interrupt INTSREn may be generated.

Caution Be sure to clear bits 3, 6, and 11 to "0" (Also clear bit 5 of the SCR01, SCR03, SCR11, or SCR13 register to 0). Be sure to set bit 2 to "1".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 10, 20, 30)

Figure 15 - 10 Format of Serial communication operation setting register mn (SCRmn) (2/2)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03), F0158H, F0159H (SCR10) to F015EH, F015FH (SCR13) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLCmn1 Note 1	SLC mn0	0	1	DLSmn1 Note 2	DLS mn0

PTC mn1	PTC mn0	Setting of parity bit in UART mode	
		Transmission	Reception
0	0	Does not output the parity bit.	Receives without parity
0	1	Outputs 0 parity Note 3.	No parity judgment
1	0	Outputs even parity.	Judged as even parity.
1	1	Outputs odd parity.	Judges as odd parity.
Be sure to set PTCmn1, PTCmn0 = 0, 0 in the Simplified SPI (CSI) mode and simplified I^2C mode.			

DIR mn	Selection of data transfer sequence in Simplified SPI (CSI) and UART modes
0	Inputs/outputs data with MSB first.
1	Inputs/outputs data with LSB first.
Be sure to clear DIRmn = 0 in the simplified I^2C mode.	

SLCmn1 Note 1	SLC mn0	Setting of stop bit in UART mode
0	0	No stop bit
0	1	Stop bit length = 1 bit
1	0	Stop bit length = 2 bits (mn = 00, 02, 10, 12 only)
1	1	Setting prohibited
When the transfer end interrupt is selected, the interrupt is generated when all stop bits have been completely transferred. Set 1 bit (SLCmn1, SLCmn0 = 0, 1) during UART reception and in the simplified I^2C mode. Set no stop bit (SLCmn1, SLCmn0 = 0, 0) in the Simplified SPI (CSI) mode. Set 1 bit (SLCmn1, SLCmn0 = 0, 1) or 2 bits (SLCmn1, SLCmn0 = 1, 0) during UART transmission.		

DLSmn1 Note 2	DLS mn0	Setting of data length in Simplified SPI (CSI) and UART modes
0	1	9-bit data length (stored in bits 0 to 8 of the SDRmn register) (settable in UART mode only)
1	0	7-bit data length (stored in bits 0 to 6 of the SDRmn register)
1	1	8-bit data length (stored in bits 0 to 7 of the SDRmn register)
Other than above		Setting prohibited
Be sure to set DLSmn1, DLSmn0 = 1, 1 in the simplified I^2C mode.		

Note 1. The SCR00, SCR02, and SCR10 registers only.

Note 2. The SCR00, SCR01, SCR10 and SCR11 registers only. Others are fixed to 1.

Note 3. 0 is always added regardless of the data contents.

Caution **Be sure to clear bits 3, 6, and 11 to "0" (Also clear bit 5 of the SCR01, SCR03, SCR11, or SCR13 register to 0). Be sure to set bit 2 to "1".**

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 10, 20, 30)

15.3.5 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, SDR01, SDR10, SDR11 or bits 7 to 0 (lower 8 bits) of SDR02, SDR03, SDR12 and SDR13 function as a transmit/receive buffer register, and bits 15 to 9 (higher 7 bits) are used as a register that sets the division ratio of the operation clock (f_{MCK}).

If the CCSmn bit of serial mode register mn (SMRmn) is cleared to 0, the clock set by dividing the operation clock by bits 15 to 9 (higher 7 bits) of the SDRmn register is used as the transfer clock.

If the CCSmn bit of serial mode register mn (SMRmn) is set to 1, set bits 15 to 9 (upper 7 bits) of SDR00, SDR01, SDR10, and SDR11 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 only be written or read when the operation is stopped (SEmn = 0). During operation (SEmn = 1), a value is written only to the lower 8/9 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 - 11 Format of Serial data register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01) After reset: 0000H R/W
FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11)

	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
FFF14H, FFF15H (SDR12), FFF16H, FFF17H (SDR13)

	FFF45H (SDR02)								FFF44H (SDR02)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn								0								

SDRmn[15:9]							Transfer clock set by dividing the operating clock
0	0	0	0	0	0	0	f_{MCK}/2
0	0	0	0	0	0	1	f_{MCK}/4
0	0	0	0	0	1	0	f_{MCK}/6
0	0	0	0	0	1	1	f_{MCK}/8
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	1	1	1	0	f_{MCK}/254
1	1	1	1	1	1	1	f_{MCK}/256

Caution 1. Be sure to clear bit 8 of the SDR02, SDR03, SDR12, and SDR13 registers to "0".

Caution 2. Setting SDRmn[15:9] = (0000000B, 0000001B) is prohibited when UART is used.

Caution 3. Setting SDRmn[15:9] = 0000000B is prohibited when simplified I^2C is used. Set SDRmn[15:9] to 0000001B or greater.

Caution 4. When operation is stopped (SEmn = 0), do not rewrite SDRmn[7:0] by an 8-bit memory manipulation instruction (SDRmn[15:9] are all cleared to 0).

Remark 1. For the function of the lower 8/9 bits of the SDRmn register, see **15.2 Configuration of Serial Array Unit**.

Remark 2. m: Unit number (m = 0, 1), 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 - 12 Format of Serial flag clear trigger register mn (SIRmn)

Address: F0108H, F0109H (SIR00) to F010EH, F010FH (SIR03), F0148H, F0149H (SIR10) to F014EH, F014FH (SIR13) — 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 Note	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, SIR03, SIR11, and SIR13 registers only.

Caution Be sure to clear bits 15 to 3 (or bits 15 to 2 for the SIR00, SIR02, SIR10, or SIR12 register) to "0".

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Remark 2. When the SIRmn register is read, 0000H is always read.

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 - 13 Format of Serial status register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03), F0140H, F0141H (SSR10) to F0146H, F0147H (SSR13) 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 Note 1	BFFmn Note 1	0	0	FEFmn Note 2	PEFmn	OVFmn

TSFmn Note 1	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 Note 1	Buffer register status indication flag of channel n
0	Valid data is not stored in the SDRmn register.
1	Valid data is stored in the SDRmn register.
<Clear conditions> • Transferring transmit data from the SDRmn register to the shift register ends during transmission. • Reading receive data from the SDRmn register ends during reception. • The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is enabled). <Set conditions> • Transmit data is written to the SDRmn register while the TXEmn bit of the SCRmn register is set to 1 (transmission or transmission and reception mode in each communication mode). • Receive data is stored in the SDRmn register while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode). • A reception error occurs.	

Note 1. The SSR00, SSR02, SSR10, and SSR12 registers only.

Note 2. The SSR01, SSR03, SSR11, and SSR13 registers only.

Caution When the simplified SPI (CSI) is handling reception in the SNOOZE mode (SWCm = 1), the BFFmn flag will not change.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 15 - 14 Format of Serial status register mn (SSRmn) (2/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03), F0140H, F0141H (SSR10) to F0146H, F0147H (SSR13)

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 Note 1	BFFmn Note 1	0	0	FEFmn Note 2	PEFmn	OVFmn

FEFmn Note 2	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/ACK error detection flag of channel n
0	No error occurs.
1	Parity error occurs (during UART reception) or ACK is not detected (during I^2C transmission).
<Clear condition> • 1 is written to the PECTmn bit of the SIRmn register. <Set condition> • The parity of the transmit data and the parity bit do not match when UART reception ends (parity error). • No ACK signal is returned from the slave channel at the ACK reception timing during I^2C transmission (ACK is not detected).	

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 1. The SSR00, SSR02, SSR10, and SSR12 registers only.

Note 2. The SSR01, SSR03, SSR11, and SSR13 registers only.

Caution 1. If data is written to the SDRmn register when BFFmn = 1, the transmit/receive data stored in the register is discarded and an overrun error (OVEmn = 1) is detected.

Caution 2. When the simplified SPI (CSI) is handling reception in the SNOOZE mode (SWCm = 1), the OVFmn flag will not change.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

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 - 15 Format of Serial channel start register m (SSm)

Address: F0122H, F0123H (SS0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03	SS02	SS01	SS00

Address: F0162H, F0163H (SS1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS1	0	0	0	0	0	0	0	0	0	0	0	0	SS13	SS12	SS11	SS10

SSmn	Operation start trigger of channel n
0	No trigger operation
1	Sets the SEmn bit to 1 and enters the communication wait status Note.

Note If set the SSmn = 1 to during a communication operation, will wait status to stop the communication. At this time, holding status value of control register and shift register, SCKmn and SOmn pins, and FEFmn, PEFmn, OVFmn flags.

Caution 1. Be sure to clear bits 15 to 4 of the SS0 register and bits 15 to 4 of the SS1 register to "0".

Caution 2. For the UART reception, set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after 4 or more f_{MCK} clocks have elapsed.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Remark 2. When the SSm register is read, 0000H is always read.

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 - 16 Format of Serial channel stop register m (STm)

Address: F0124H, F0125H (ST0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST0	0	0	0	0	0	0	0	0	0	0	0	0	ST03	ST02	ST01	ST00

Address: F0164H, F0165H (ST1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST1	0	0	0	0	0	0	0	0	0	0	0	0	ST13	ST12	ST11	ST10

STmn	Operation stop trigger of channel n
0	No trigger operation
1	Clears the SEmn bit to 0 and stops the communication operation Note.

Note Holding status value of the control register and shift register, the SCKmn and SOmn pins, and FEFmn, PEFmn, OVFmn flags.

Caution Be sure to clear bits 15 to 4 of the ST0 register and bits 15 to 4 of the ST1 register to "0".

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Remark 2. When the STm register is read, 0000H is always read.

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 - 17 Format of Serial channel enable status register m (SEm)

Address: F0120H, F0121H (SE0) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	0	0	0	0	0	0	SE03	SE02	SE01	SE00

Address: F0160H, F0161H (SE1) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE1	0	0	0	0	0	0	0	0	0	0	0	0	SE13	SE12	SE11	SE10

SEmn	Indication of operation enable/stop status of channel n
0	Operation stops
1	Operation is enabled.

Remark m: Unit number (m = 0, 1), 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 - 18 Format of Serial output enable register m (SOEm)

Address: F012AH, F012BH (SOE0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02	0	SOE00

Address: F016AH, F016BH (SOE1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE12	0	SOE10

SOE mn	Serial output enable/stop of channel n
0	Stops output by serial communication operation.
1	Enables output by serial communication operation.

Caution Be sure to clear bits 15 to 3 and 1 of the SOEm register to “0”.

Remark m: Unit number (m = 0, 1), 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 - 19 Format of Serial output register m (SOm)

Address: F0128H, F0129H (SO0) After reset: 0F0FH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO 02	1	CKO 00	0	0	0	0	1	SO 02	1	SO 00

Address: F0168H, F0169H (SO1) After reset: 0F0FH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	CKO 12	1	CKO 10	0	0	0	0	1	SO 12	1	SO 10

CKO mn	Serial clock output of channel n
0	Serial clock output value is “0”.
1	Serial clock output value is “1”.

SO mn	Serial data output of channel n
0	Serial data output value is “0”.
1	Serial data output value is “1”.

Caution **Be sure to clear bits 15 to 12 and 7 to 4 of the SO0 register to “0”. And be sure to set bits 11, 9, 3 and 1 to “1”.**
Be sure to clear bits 15 to 12 and 7 to 4 of the SO1 register to “0”. And be sure to set bits 11, 9, 3 and 1 to “1”.

Remark m: Unit number (m = 0, 1), 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 and simplifies I^2C mode.

Inverting channel n by using this register is reflected on pin output only when serial output is enabled (SOEmn = 1). When serial output is disabled (SOEmn = 0), the value of the SOmn bit is output as is.

Rewriting the SOLm register is prohibited when the register is in operation (when SEmn = 1).

The SOLm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SOLm register can be set with an 8-bit memory manipulation instruction with SOLmL.

Reset signal generation clears the SOLm register to 0000H.

Figure 15 - 20 Format of Serial output level register m (SOLm)

Address: F0134H, F0135H (SOL0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL02	0	SOL00

Address: F0174H, F0175H (SOL1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL12	0	SOL10

SOLmn	Selects inversion of the level of the transmit data of channel n in UART mode
0	Communication data is output as is.
1	Communication data is inverted and output.

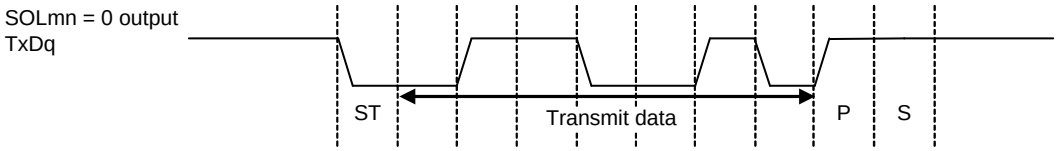
Caution Be sure to clear bits 15 to 3, and 1 of the SOLm register to "0".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

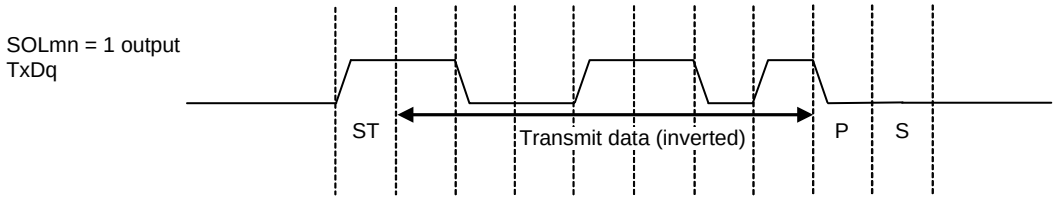
Figure 15 - 21 shows examples in which the level of transmit data is reversed during UART transmission.

Figure 15 - 21 Examples of Reverse Transmit Data

(a) Non-reverse Output (SOLmn = 0)



(b) Reverse Output (SOLmn = 1)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

15.3.14 Serial standby control register m (SSCm)

The SSC0 register is used to control the startup of reception (the SNOOZE mode) while in the STOP mode when receiving CSI00 or UART0 serial data.

The SSC1 register is used to control the startup of reception (the SNOOZE mode) while in the STOP mode when receiving CSI20 and UART2 serial data.

The SSCm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSCm register can be set with an 8-bit memory manipulation instruction with SSCmL.

Reset signal generation clears the SSCm register to 0000H.

Caution The maximum transfer rate in the SNOOZE mode is as follows.

- **When using CSI00, CSI20: Up to 1 Mbps**
- **When using UART0, UART2: 4800 bps only**

Figure 15 - 22 Format of Serial standby control register m (SSCm)

Address: F0138H (SSC0), F0178H (SSC1) After reset: 0000H R/W

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

SSECm	Selection of whether to enable or disable the generation of communication error interrupts in the SNOOZE mode
0	Enable the generation of error interrupts (INTSRE0/INTSRE2).
1	Stop the generation of error interrupts (INTSRE0/INTSRE2).
• The SSECm bit can be set to 1 or 0 only when both the SWCm and EOCmn bits are set to 1 during UART reception in the SNOOZE mode. In other cases, clear the SSECm bit to 0. • Setting SSECm, SWCm = 1, 0 is prohibited.	

SWCm	Setting of the SNOOZE mode
0	Do not use the SNOOZE mode function.
1	Use the SNOOZE mode function.
• When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and simplified SPI (CSI) or UART reception is performed without operating the CPU (the SNOOZE mode). • The SNOOZE mode function can only be specified when the high-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (f_{CLK}). If any other clock is selected, specifying this mode is prohibited. • Even when using SNOOZE mode, be sure to set the SWCm bit to 0 in normal operation mode and change it to 1 just before shifting to STOP mode. Also, be sure to change the SWCm bit to 0 after returning from STOP mode to normal operation mode.	

Caution Setting SSECm, SWCm = 1, 0 is prohibited.

Figure 15 - 23 Interrupt in UART Reception Operation in SNOOZE Mode

EOCmn Bit	SSECm Bit	Reception Ended Successfully	Reception Ended in an Error
0	0	INTSRx is generated.	INTSRx is generated.
0	1	INTSRx is generated.	INTSRx is generated.
1	0	INTSRx is generated.	INTSREx is generated.
1	1	INTSRx is generated.	No interrupt is generated.

15.3.15 Input switch control register (ISC)

The ISC1 and ISC0 bits of the ISC register are used to realize a LIN-bus 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.

When bit 1 is set to 1, the input signal of the serial data input (RXD0) pin is selected as a timer input, so that wake up signal can be detected, the low width of the break field, and the pulse width of the sync field can be measured by the timer.

The 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 - 24 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	Uses the input signal of the TI07 pin as a timer input (normal operation).
1	Input signal of the RxD0 pin is used as timer input (detects the wakeup signal and measures the low width of the break field and the pulse width of the sync field).

ISC0	Switching external interrupt (INTP0) input
0	Uses the input signal of the INTP0 pin as an external interrupt (normal operation).
1	Uses the input signal of the RxD0 pin as an external interrupt (wakeup signal detection).

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) or simplified I^2C communication, by clearing the corresponding bit of this register to 0.

Enable the noise filter of the pin used for UART communication, by setting the corresponding bit of this register to 1.

When the noise filter is enabled, after synchronization is performed with the operation clock (f_{MCK}) of the target channel, 2-clock match detection is performed. When the noise filter is disabled, only synchronization is performed with the operation clock (f_{MCK}) of the target channel.

The NFEN0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the NFEN0 register to 00H.

Figure 15 - 25 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	SNFEN30	0	SNFEN20	0	SNFEN10	0	SNFEN00

SNFEN30	Use of noise filter of RxD3 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN30 to 1 to use the RxD3 pin. Clear SNFEN30 to 0 to use the other than RxD3 pin.	

SNFEN20	Use of noise filter of RxD2 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN20 to 1 to use the RxD2 pin. Clear SNFEN20 to 0 to use the other than RxD2 pin.	

SNFEN10	Use of noise filter of RxD1 pin
0	Noise filter OFF
1	Noise filter ON
Set the SNFEN10 bit to 1 to use the RxD1 pin. Clear the SNFEN10 bit to 0 to use the other than RxD1 pin.	

SNFEN00	Use of noise filter of RxD0 pin
0	Noise filter OFF
1	Noise filter ON
Set the SNFEN00 bit to 1 to use the RxD0 pin. Clear the SNFEN00 bit to 0 to use the other than RxD0 pin.	

Caution Be sure to clear bits 7, 5, 3, and 1 to “0”.

15.3.17 Registers that control port functions of serial input/output pins

Using the serial array unit requires setting of the registers that control the port functions for the port pins with which the serial array unit pin functions for the target channel are multiplexed (port mode register (PMxx), port register (Pxx), port input mode register (PIMxx), port output mode register (POMxx)).

For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.2 Port registers (Pxx)**, **4.3.4 Port input mode registers (PIMxx)**, and **4.3.5 Port output mode registers (POMxx)**.

Using a port pin which is multiplexed with a serial data output or serial clock output pin function (e.g. P02/SO10/TxD1/(PCLBUZ0/SEG50) for serial data output or serial clock output requires setting the corresponding bits in the LCD port function register (PFSEGx) and port mode register (PMxx) to 0, and the corresponding bit in the port register (Pxx) to 1.

Using a port pin in N-ch open-drain output (V_{DD} tolerance) mode requires setting the corresponding bit in the port output mode register (POMxx) to 1. When connecting an external device operating at a different voltage (1.8 V or 2.5 V), see **4.4.4 Handling different potential (1.8 V, 2.5 V) by using I/O buffers**.

Example When P02/SO10/TxD1/(PCLBUZ0)/SEG50 is to be used for serial data output
Set the PFSEG50 bit of LCD port function register 6 to 0.
Set the PM02 bit of port mode register 0 to 0.
Set the P02 bit of port register 0 to 1.

Using a port pin which is multiplexed with a serial data input or serial clock input pin function (e.g. P01/SI10/RxD1/SDA10/SEG49) for serial data input or serial clock input requires setting the corresponding bits in the LCD port function register (PFSEGx) and port mode register (PMxx) to 1. At this time, the value of the corresponding bit in the port register (Pxx) may be 0 or 1.

Using a TTL input buffer requires setting the corresponding bit in the port input mode register (PIMxx) to 1. When connecting an external device operating at a different voltage (1.8 V or 2.5 V), see **4.4.4 Handling different potential (1.8 V, 2.5 V) by using I/O buffers**.

Example When P01/SI10/RxD1/SDA10/SEG49 is to be used for serial data input
Set the PFSEG49 bit of LCD port function register 6 to 0.
Set the PM03 bit of port mode register 0 to 1.
Set the P03 bit of port register 0 to 0 or 1.

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 by units

The stopping of the operation by units is set by using peripheral enable register 0 (PER0).

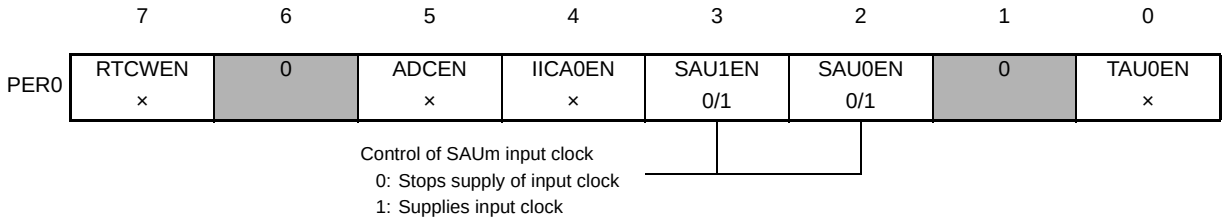
The PER0 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

To stop the operation of serial array unit 0, set bit 2 (SAU0EN) to 0.

To stop the operation of serial array unit 1, set bit 3 (SAU1EN) to 0.

Figure 15 - 26 Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units

(a) Peripheral enable register 0 (PER0)... Set only the bit of SAUm to be stopped to 0.



Caution 1. If SAUmEN = 0, writing to a control register of serial array unit m is ignored, and, even if the register is read, only the default value is read 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 to 4 (PIM0 to PIM4)**
- **Port output mode registers 0 to 4 (POM0 to POM4)**
- **Port mode registers 0 to 4 (PM0 to PM4)**
- **Port registers 0 to 4 (P0 to P4)**

Caution 2. Be sure to clear bits 1, 6 to 0.

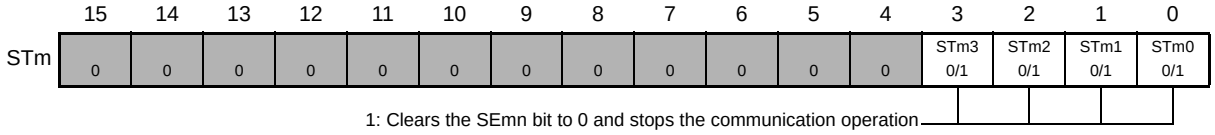
Remark ×: Bits not used with serial array units (depending on the settings of other peripheral functions)
0/1: Set to 0 or 1 depending on the usage of the user

15.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

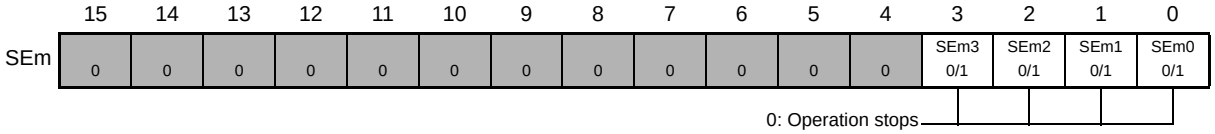
Figure 15 - 27 Each Register Setting When Stopping the Operation by Channels

(a) Serial channel stop register m (STm)... This register is a trigger register that is used to enable stopping communication/count by each channel.



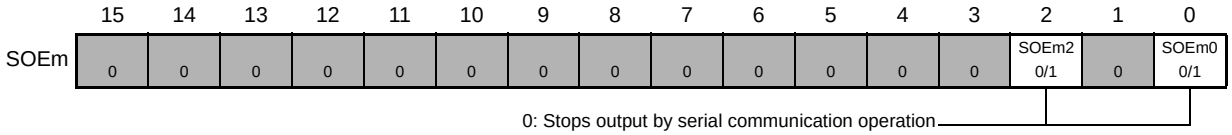
* Because the STmn bit is a trigger bit, it is cleared immediately when SEmn = 0.

(b) Serial Channel Enable Status Register m (SEm)... This register indicates whether data transmission/ reception operation of each channel is enabled or stopped.



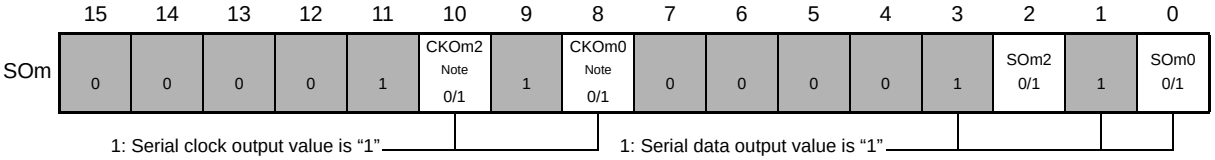
* The SEm register is a read-only status register, whose operation is stopped by using the STm register.
With a channel whose operation is stopped, the value of the CKOmn bit of the SOm register can be set by software.

(c) Serial output enable register m (SOEm)... This register is a register that is used to enable or stop output of the serial communication operation of each channel.



* 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.



* When using pins corresponding to each channel as port function pins, set the corresponding CKOmn, SOmn bits to “1”.

Note Serial array unit 0 only.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Remark 2. ■: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

15.5 Operation of Simplified SPI (CSI00, CSI10, CSI20, CSI30) 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 (CSI00 only)
  Max. f_{CLK}/4
  During slave communication: Max. f_{MCK}/6

[Interrupt function]
- Transfer end interrupt/buffer empty interrupt

[Error detection flag]
- Overrun error

In addition, CSI00 and CSI20 support 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.

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **CHAPTER 34** or **CHAPTER 35 ELECTRICAL SPECIFICATIONS**.

The channels supporting Simplified SPI (CSI00, CSI10, CSI20, CSI30) are channels 0 and 2 of SAU0, and channels 0 and 2 of SAU1

• 80-pin, 85-pin, and 100-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0 (supporting LIN-bus)	IIC00
	1	—		—
	2	CSI10	UART1	IIC10
	3	—		—
1	0	CSI20	UART2	IIC20
	1	—		—
	2	CSI30	UART3	IIC30
	3	—		—

Simplified SPI (CSI00, CSI10, CSI20, CSI30) 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	CSI10	CSI20	CSI30
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1	Channel 2 of SAU1
Pins used	SCK00, SO00	SCK10, SO10	SCK20, SO20	SCK30, SO30
Interrupt	INTCSI00	INTCSI10	INTCSI20	INTCSI30
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	None			
Transfer data length	7 or 8 bits			
Transfer rate Note	Max. $f_{CLK}/2$ [Hz] (CSI00 only), $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK}: System clock frequency			
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts from the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock 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			

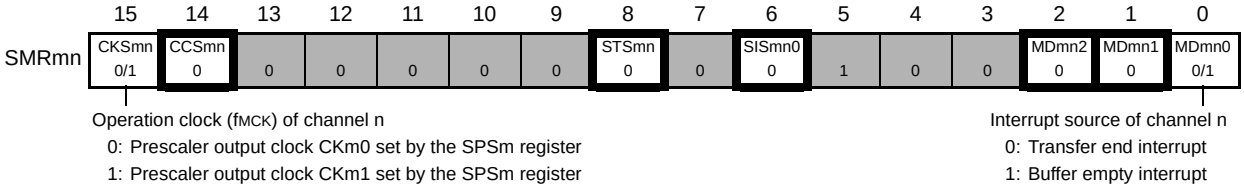
Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 34** or **CHAPTER 35 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12

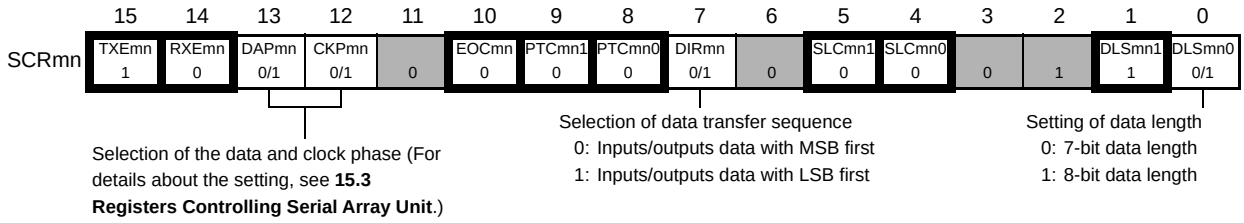
(1) Register setting

Figure 15 - 28 Example of Contents of Registers for Master Transmission of Simplified SPI (CSI00, CSI10, CSI20, CSI30)

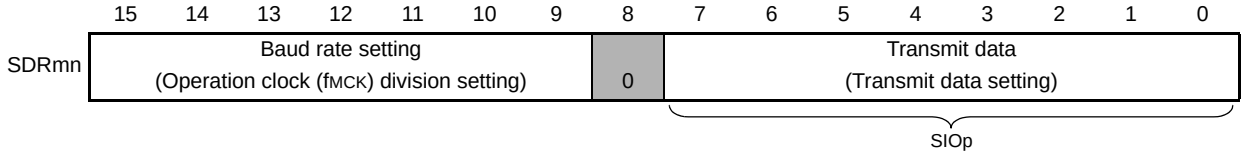
(a) Serial mode register mn (SMRmn)



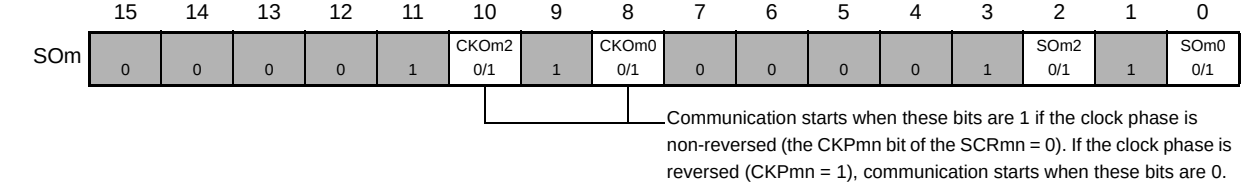
(b) Serial communication operation setting register mn (SCRmn)



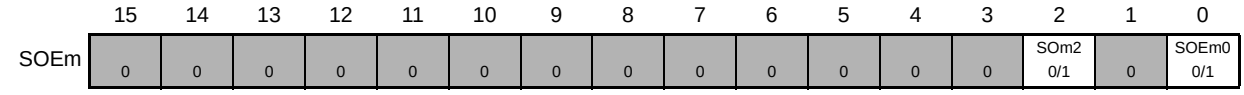
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



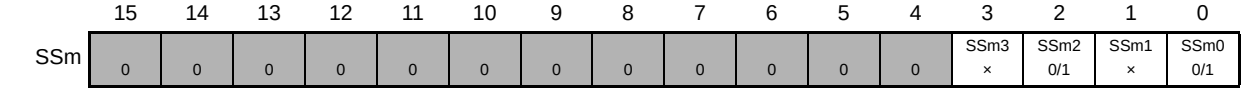
(d) Serial output register m (SOm)... Sets only the bits of the target channel.



(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.



Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20, 30), mn = 00, 02, 10, 12

Remark 2. ■: Setting is fixed in the simplified SPI (CSI) master transmission mode,
▒: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 15 - 29 Initial Setting Procedure for Master Transmission

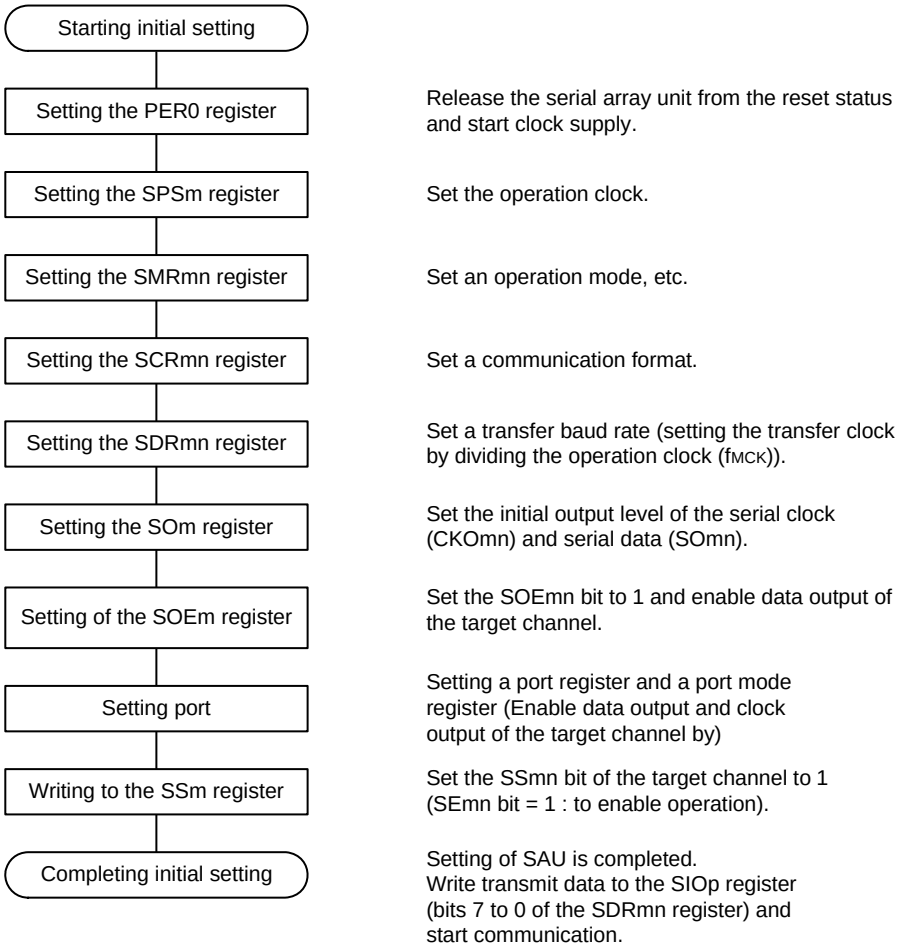


Figure 15 - 30 Procedure for Stopping Master Transmission

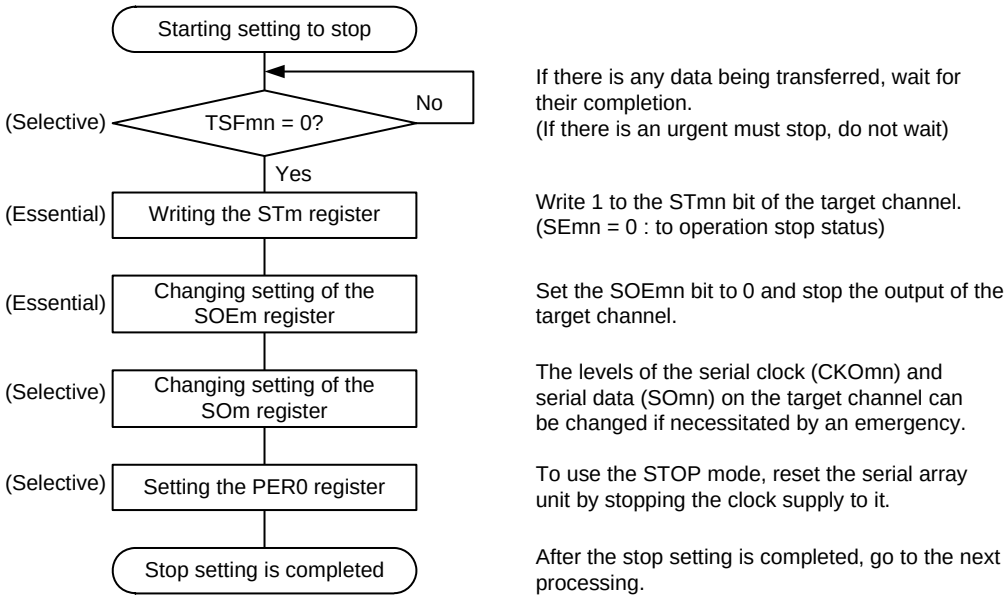
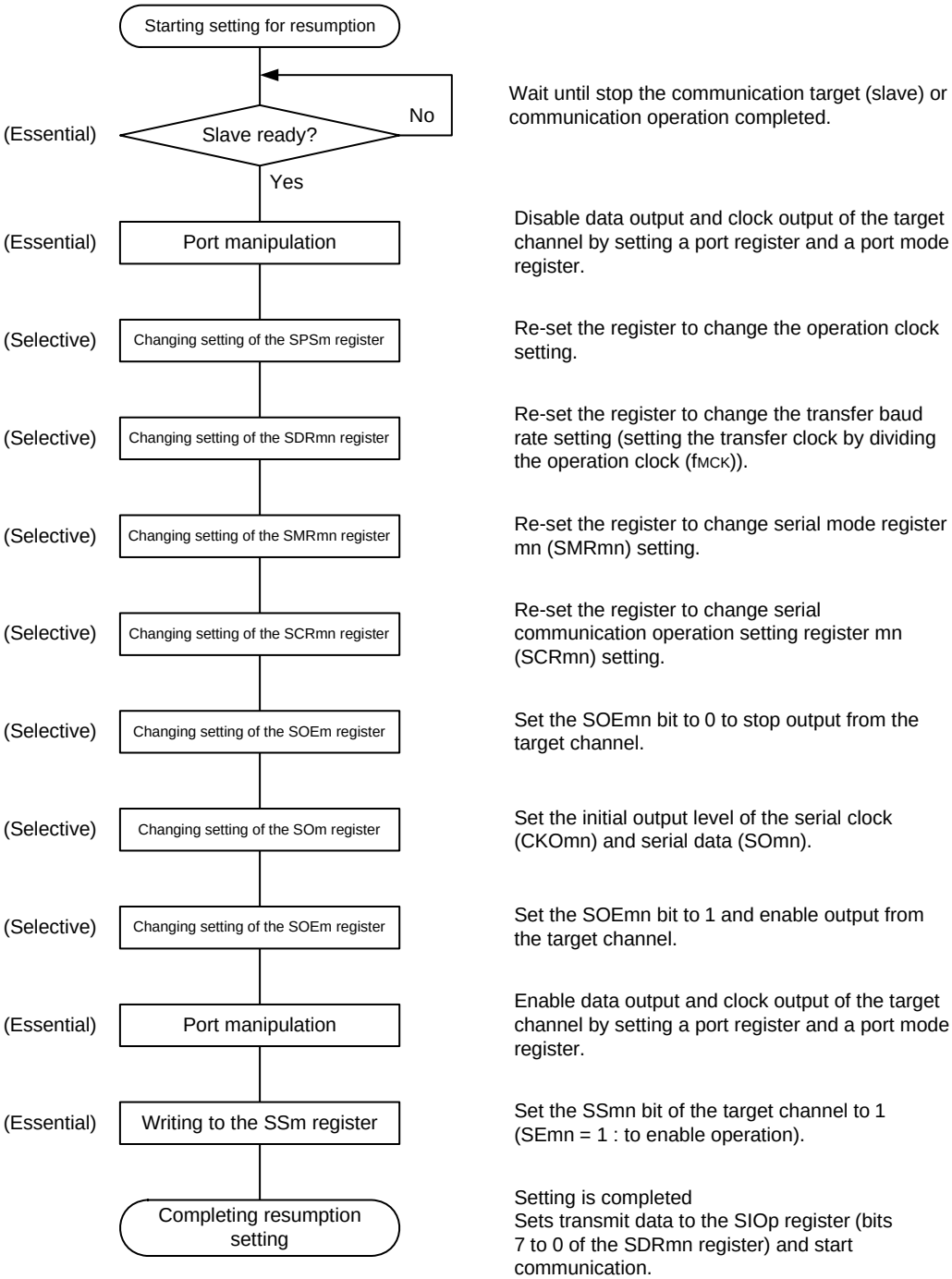


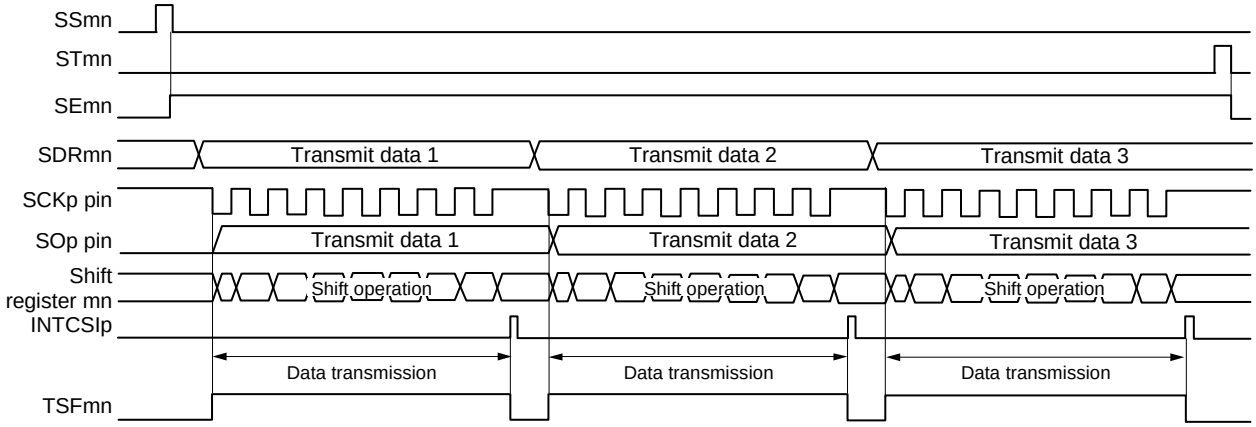
Figure 15 - 31 Procedure for Resuming Master Transmission



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.

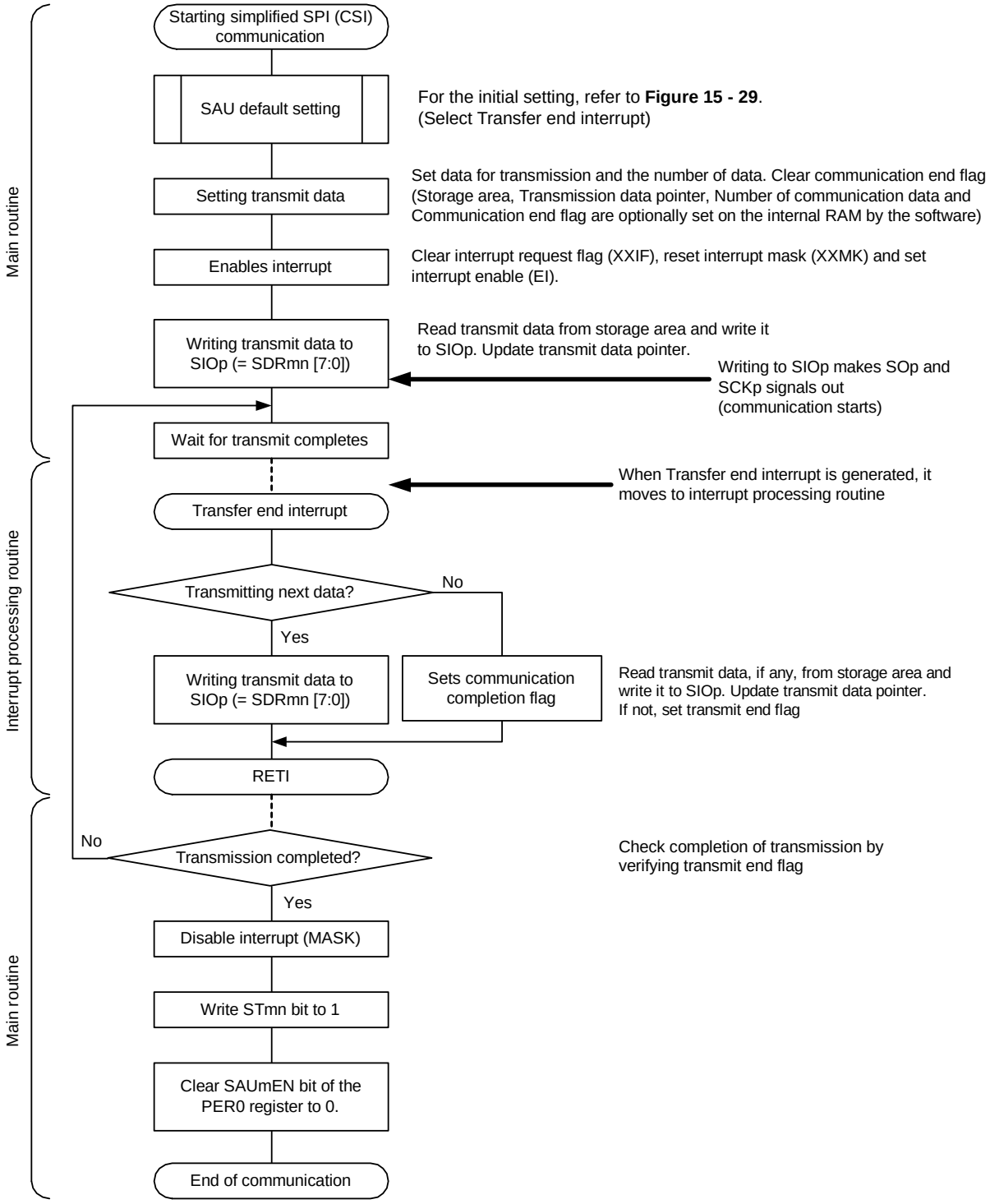
(3) Processing flow (in single-transmission mode)

Figure 15 - 32 Timing Chart of Master Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



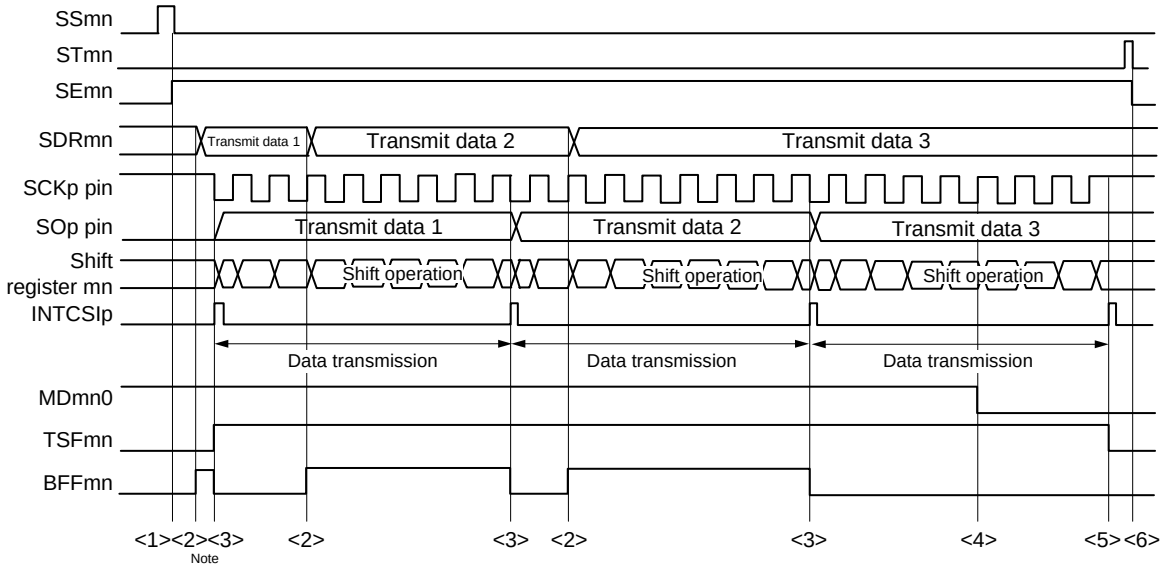
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20, 30), mn = 00, 02, 10, 12

Figure 15 - 33 Flowchart of Master Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 15 - 34 Timing Chart of Master Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

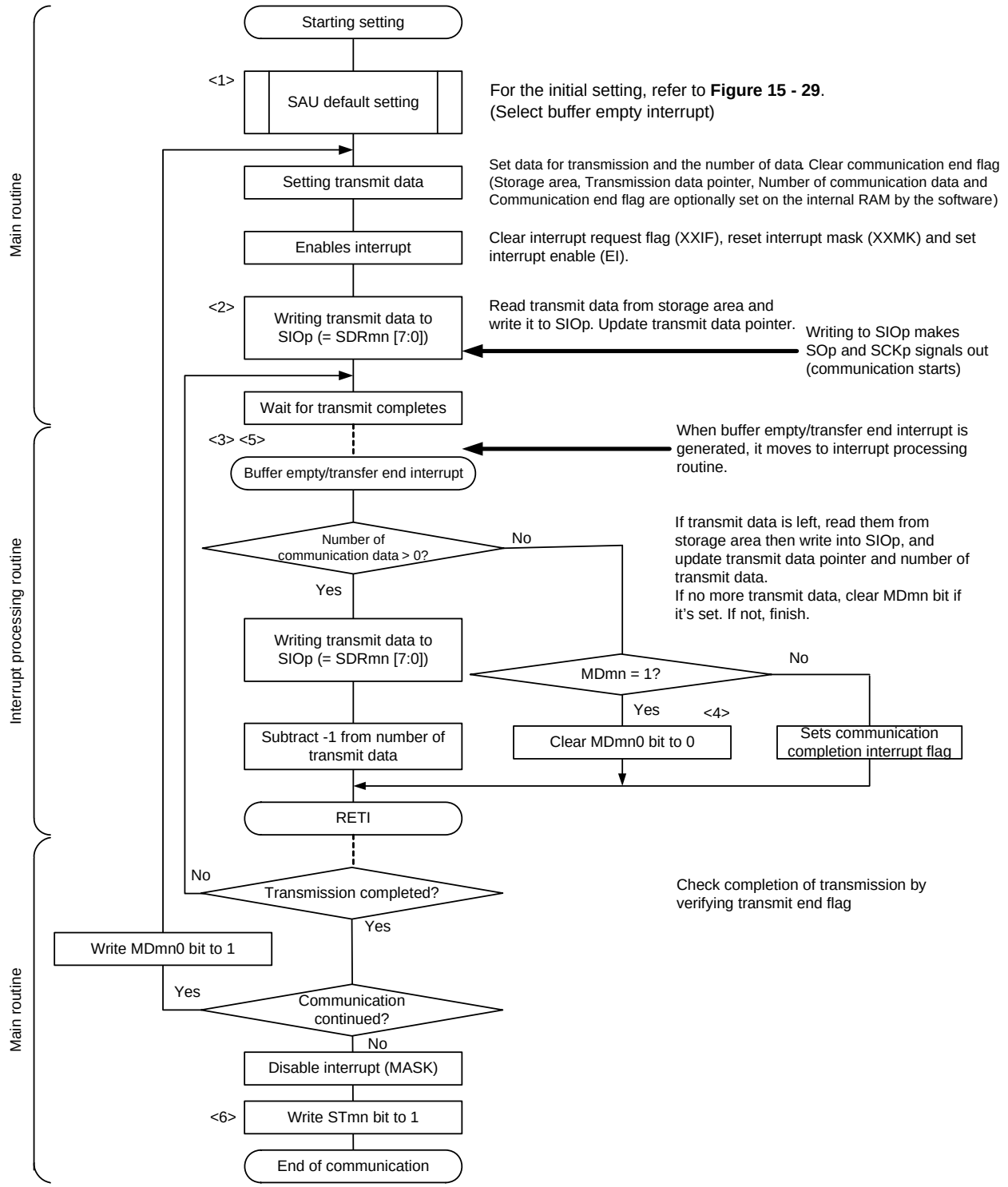


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

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, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20, 30), mn = 00, 02, 10, 12

Figure 15 - 35 Flowchart of Master Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 15 - 34 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	CSI10	CSI20	CSI30
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1	Channel 2 of SAU1
Pins used	SCK00, SI00	SCK10, SI10	SCK20, SI20	SCK30, SI30
Interrupt	INTCSI00	INTCSI10	INTCSI20	INTCSI30
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	Overrun error detection flag (OVFmn) only			
Transfer data length	7 or 8 bits			
Transfer rate Note	Max. $f_{CLK}/2$ [Hz] (CSI00 only), $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK}: System clock frequency			
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data input starts from the start of the operation of the serial clock. • DAPmn = 1: Data input starts half a clock 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 34** or **CHAPTER 35 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12

(1) Register setting

Figure 15 - 36 Example of Contents of Registers for Master Reception of Simplified SPI (CSI00, CSI10, CSI20, CSI30)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **15.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting (Operation clock (fMCK) division setting)							0	Receive data (Write FFH as dummy data.)							

SIOp

(d) Serial output register m (SOm)... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 0/1	1	CKOm0 0/1	0	0	0	0	1	SOm2 ×	1	SOm0 ×

Communication starts when these bits are 1 if the clock phase is non-reversed (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when these bits are 0.

(e) Serial output enable register m (SOEm)... The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	0	SOEm0 ×

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20, 30), mn = 00, 02, 10, 12

Remark 2. ■: Setting is fixed in the simplified SPI (CSI) master reception mode,
■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 15 - 37 Initial Setting Procedure for Master Reception

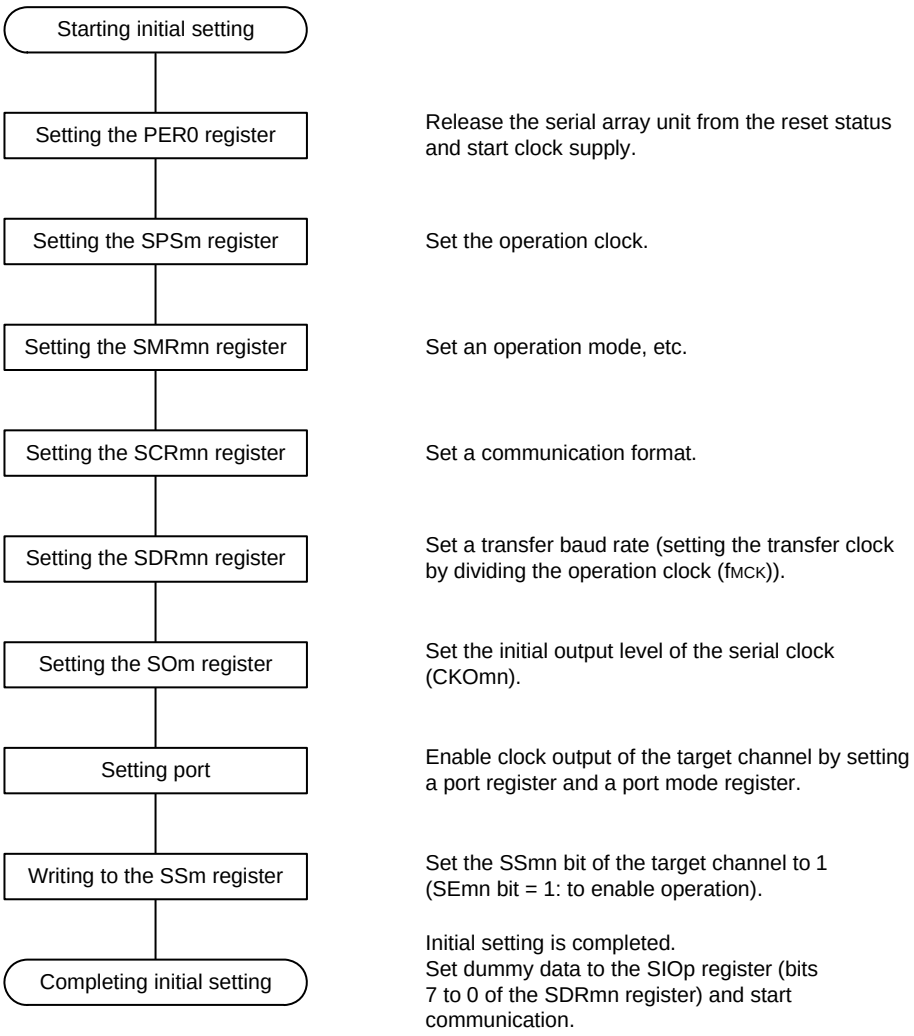


Figure 15 - 38 Procedure for Stopping Master Reception

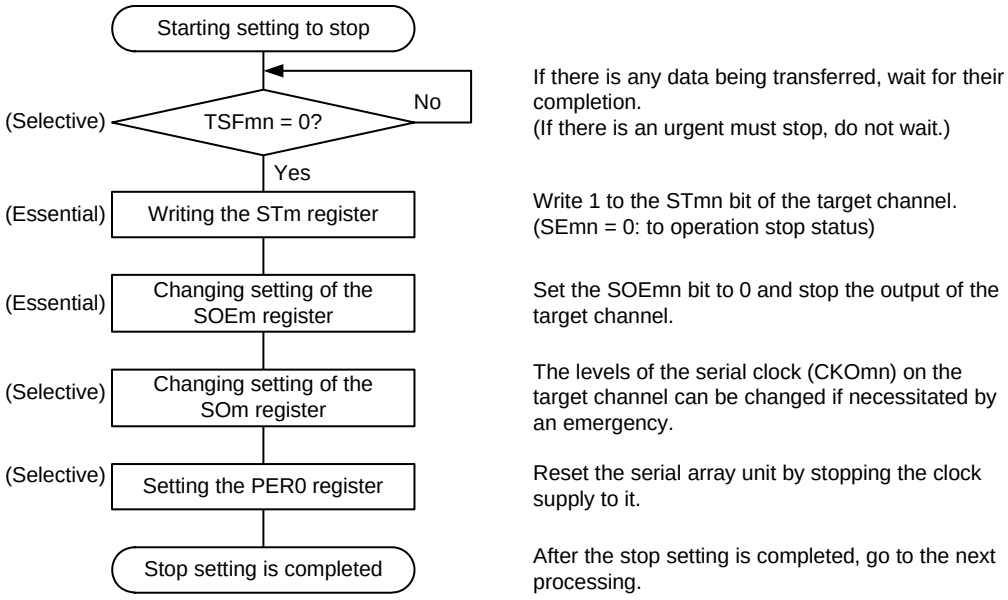
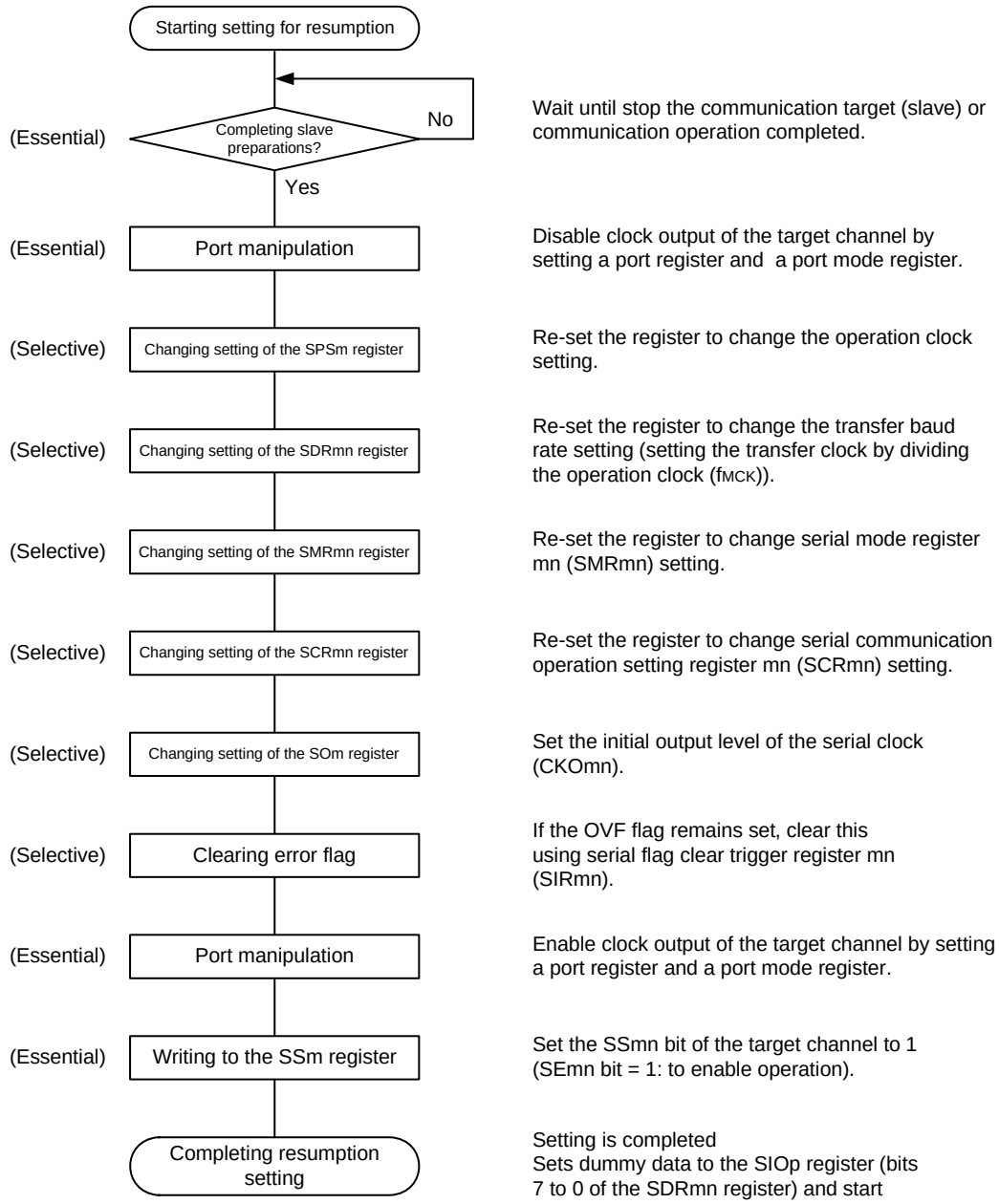


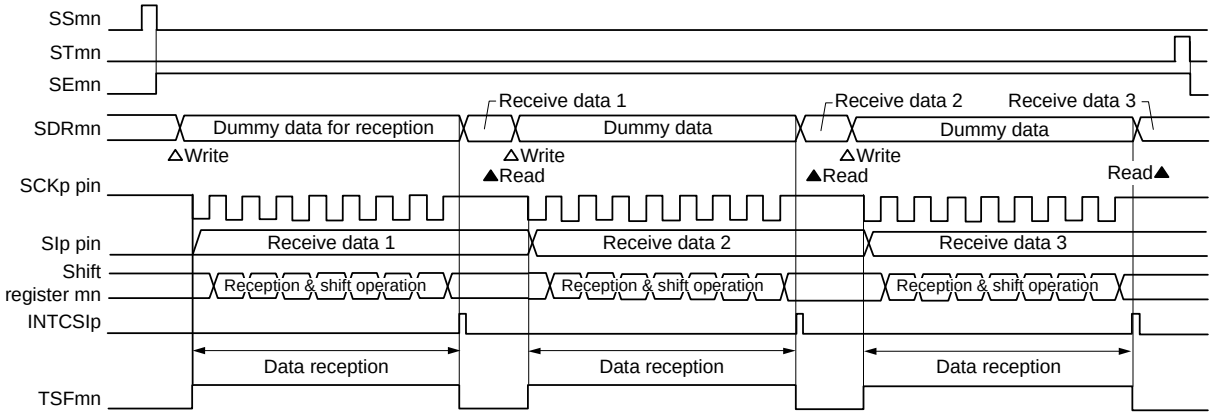
Figure 15 - 39 Procedure for Resuming Master Reception



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

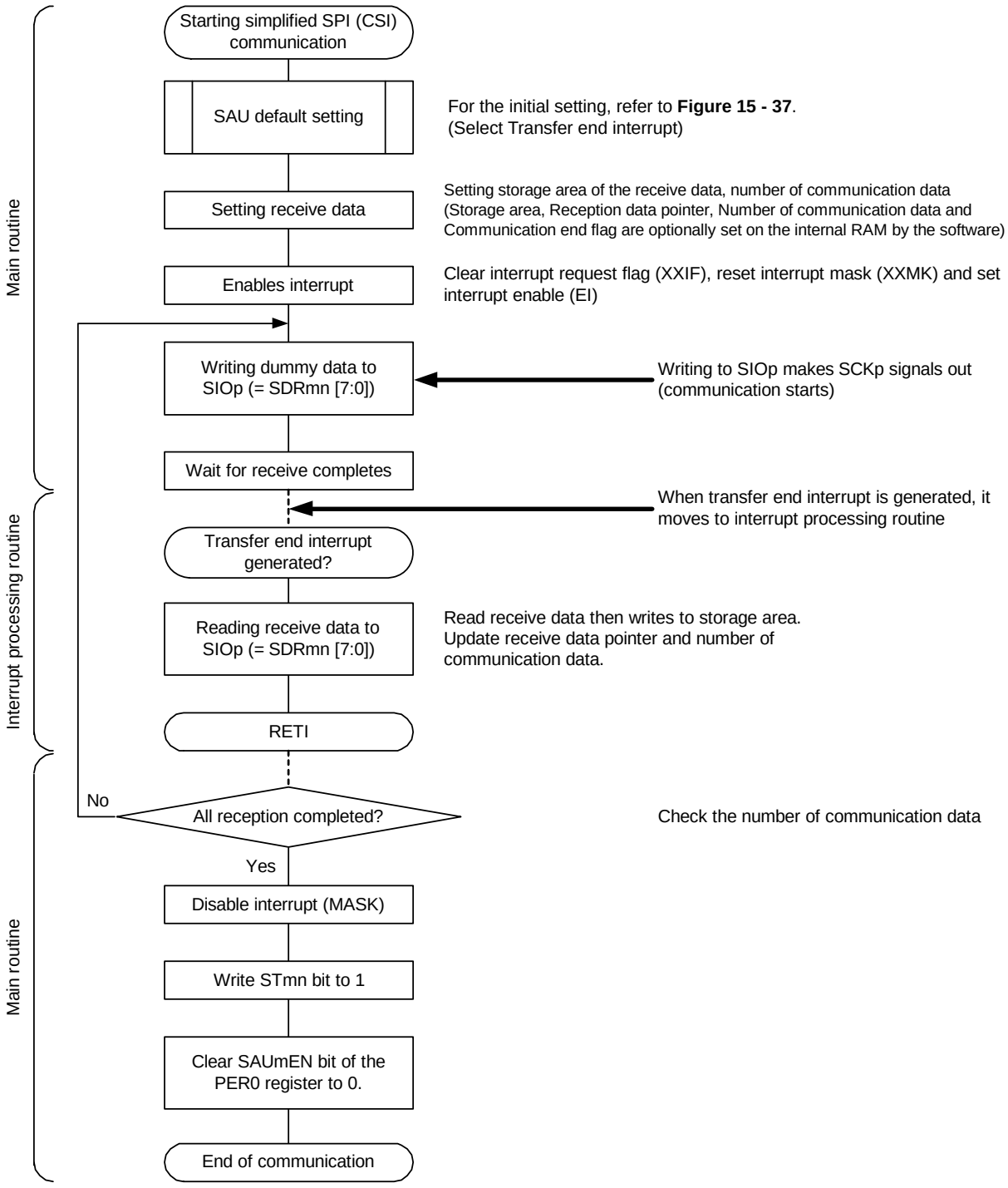
(3) Processing flow (in single-reception mode)

Figure 15 - 40 Timing Chart of Master Reception (in Single-Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



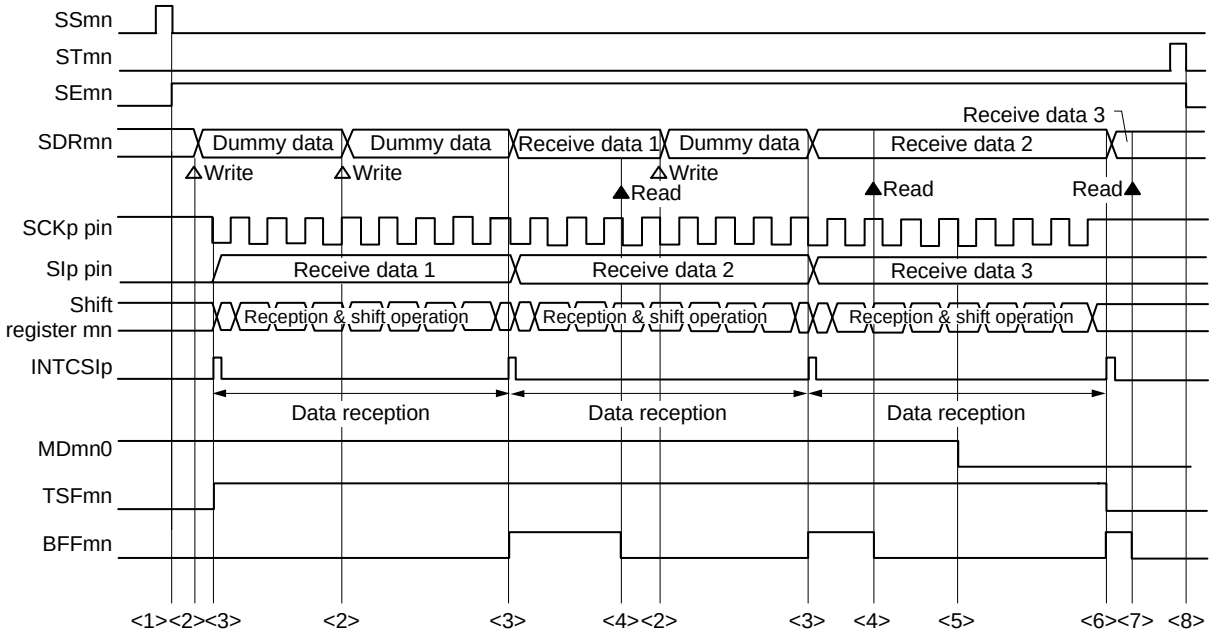
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20, 30), mn = 00, 02, 10, 12

Figure 15 - 41 Flowchart of Master Reception (in Single-Reception Mode)



(4) Processing flow (in continuous reception mode)

Figure 15 - 42 Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)

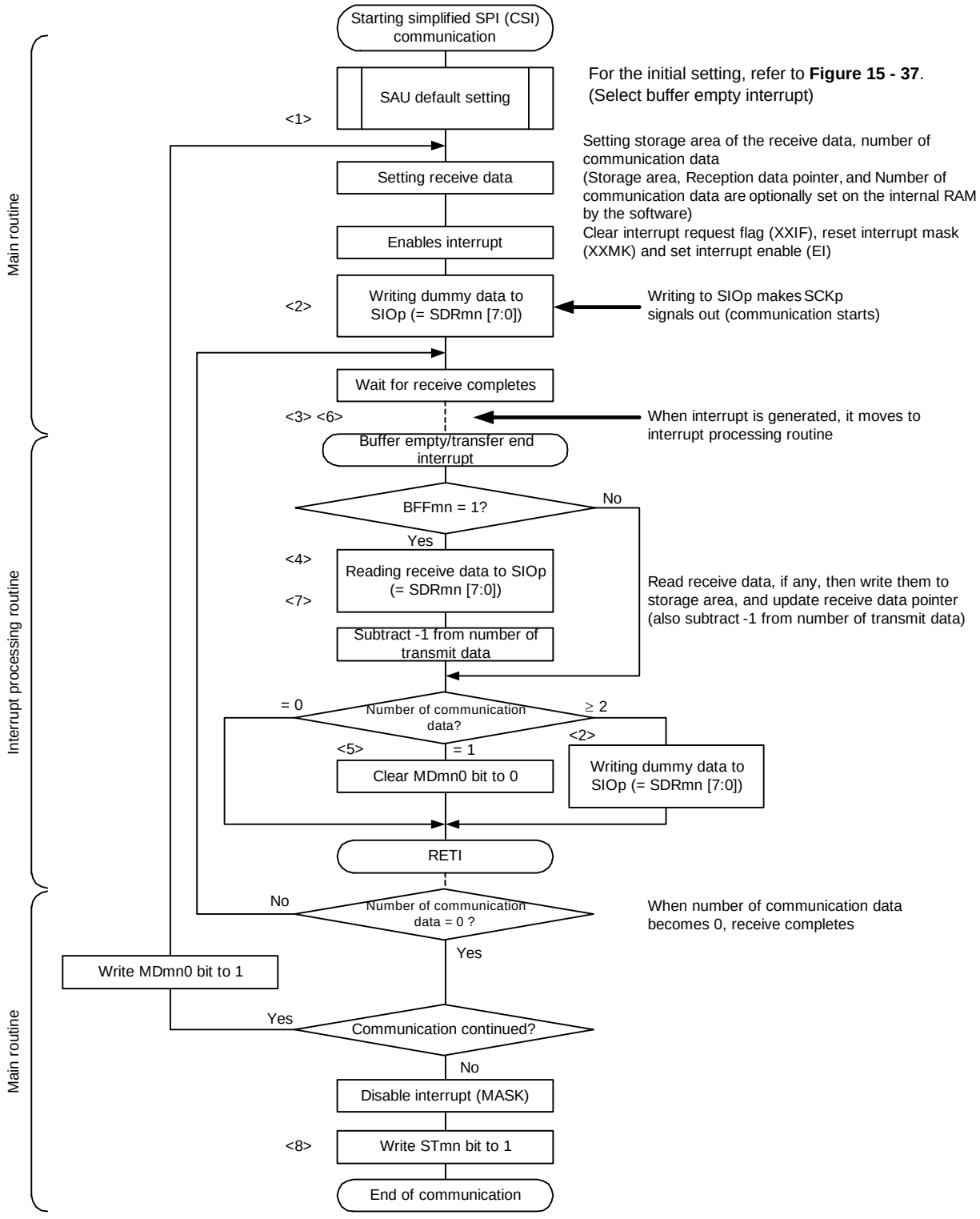


Caution The MDmn0 bit can be rewritten even during operation.
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in Figure 15 - 43 Flowchart of Master Reception (in Continuous Reception Mode).

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20, 30), mn = 00, 02, 10, 12

Figure 15 - 43 Flowchart of Master Reception (in Continuous Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in Figure 15 - 42 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	CSI10	CSI20	CSI30
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1	Channel 2 of SAU1
Pins used	SCK00, SI00, SO00	SCK10, SI10, SO10	SCK20, SI20, SO20	SCK30, SI30, SO30
Interrupt	INTCSI00	INTCSI10	INTCSI20	INTCSI30
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	Overrun error detection flag (OVFmn) only			
Transfer data length	7 or 8 bits			
Transfer rate Note	Max. $f_{CLK}/2$ [Hz] (CSI00 only), $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK}: System clock frequency			
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data I/O starts at the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock 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			

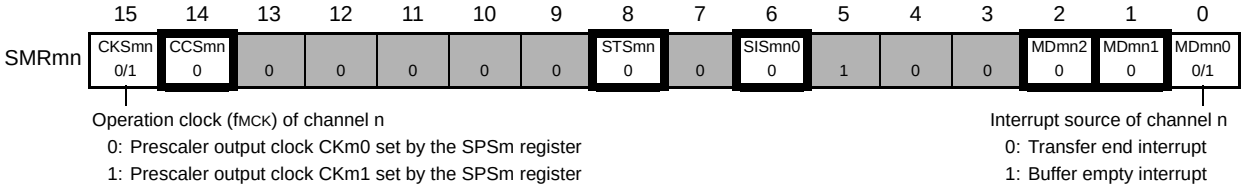
Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 34** or **CHAPTER 35 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSIl number (p = 00, 10, 20, 30), mn = 00, 02, 10, 12

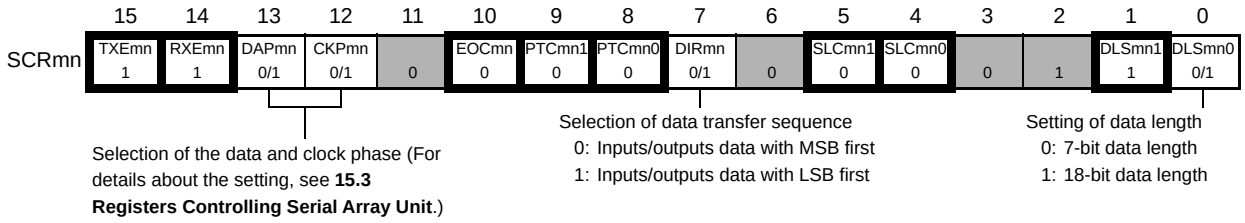
(1) Register setting

Figure 15 - 44 Example of Contents of Registers for Master Transmission/Reception of Simplified SPI (CSI00, CSI10, CSI20, CSI30)

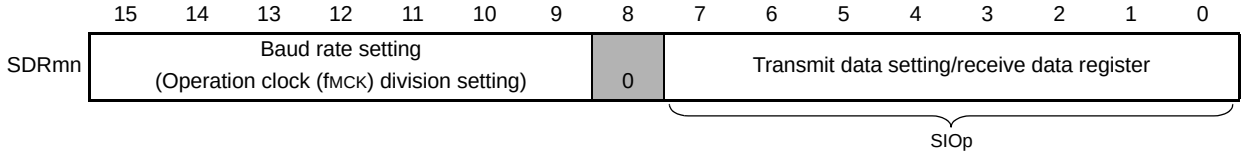
(a) Serial mode register mn (SMRmn)



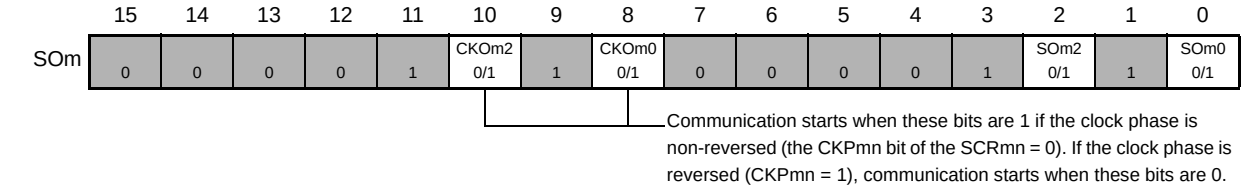
(b) Serial communication operation setting register mn (SCRmn)



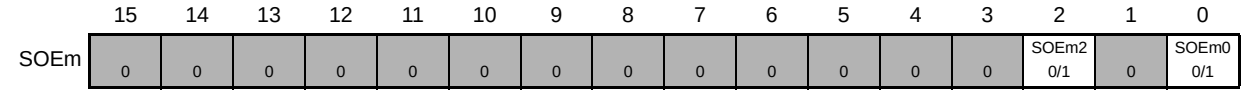
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



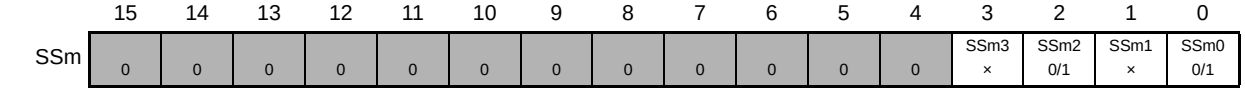
(d) Serial output register m (SOm)... Sets only the bits of the target channel.



(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.



Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20, 30), mn = 00, 02, 10, 12

Remark 2. ■: Setting is fixed in the simplified SPI (CSI) master transmission/reception mode,
■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 15 - 45 Initial Setting Procedure for Master Transmission/Reception

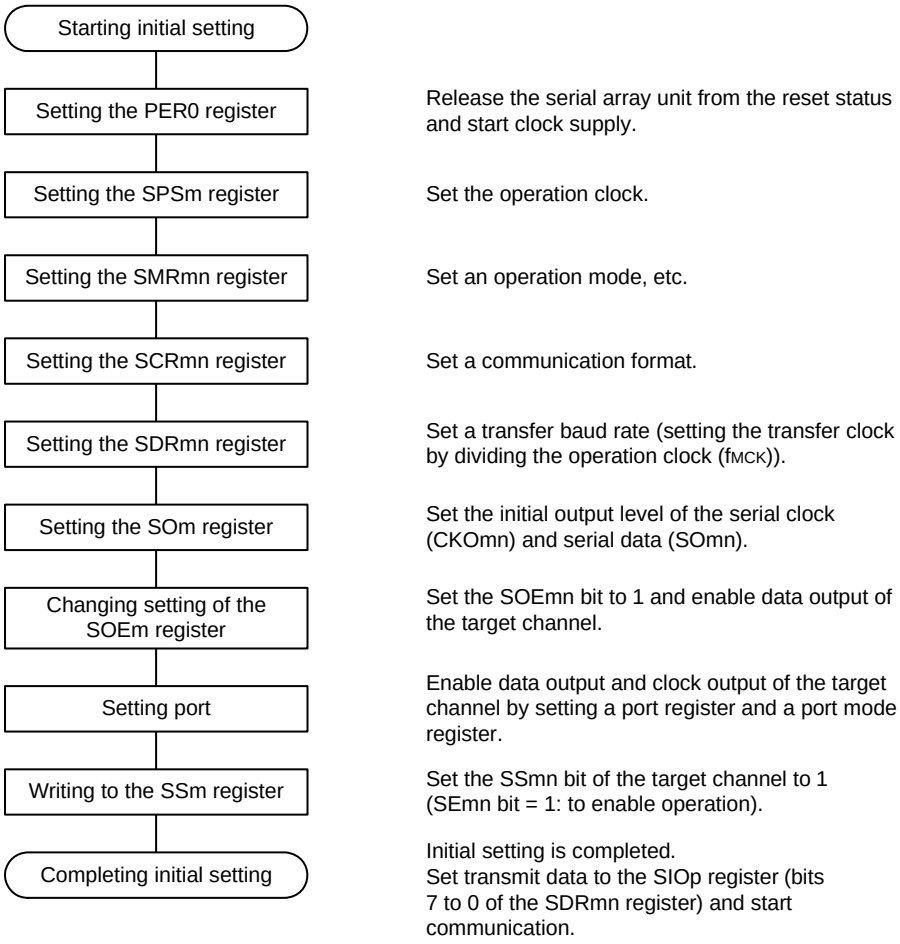


Figure 15 - 46 Procedure for Stopping Master Transmission/Reception

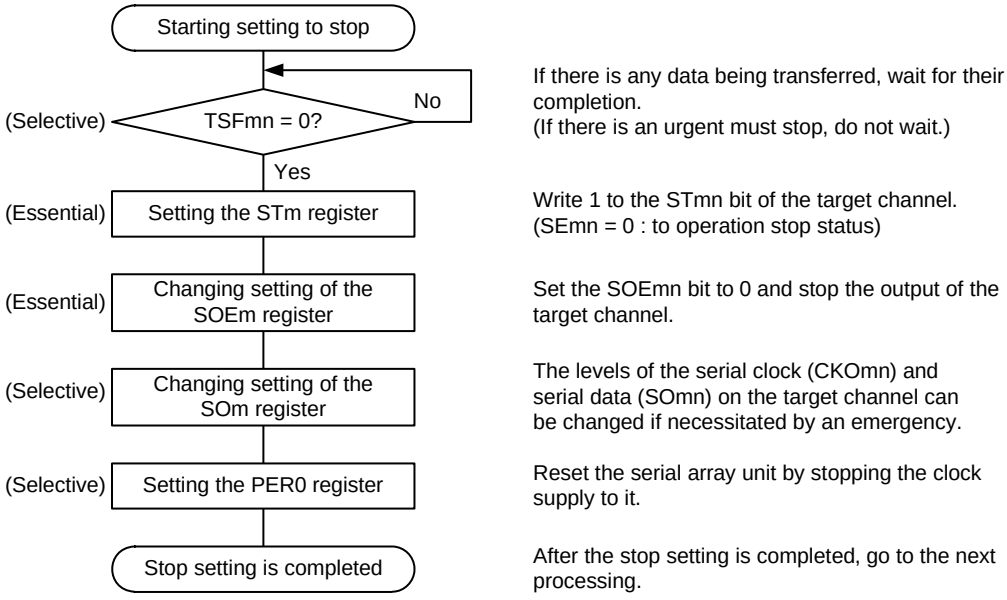
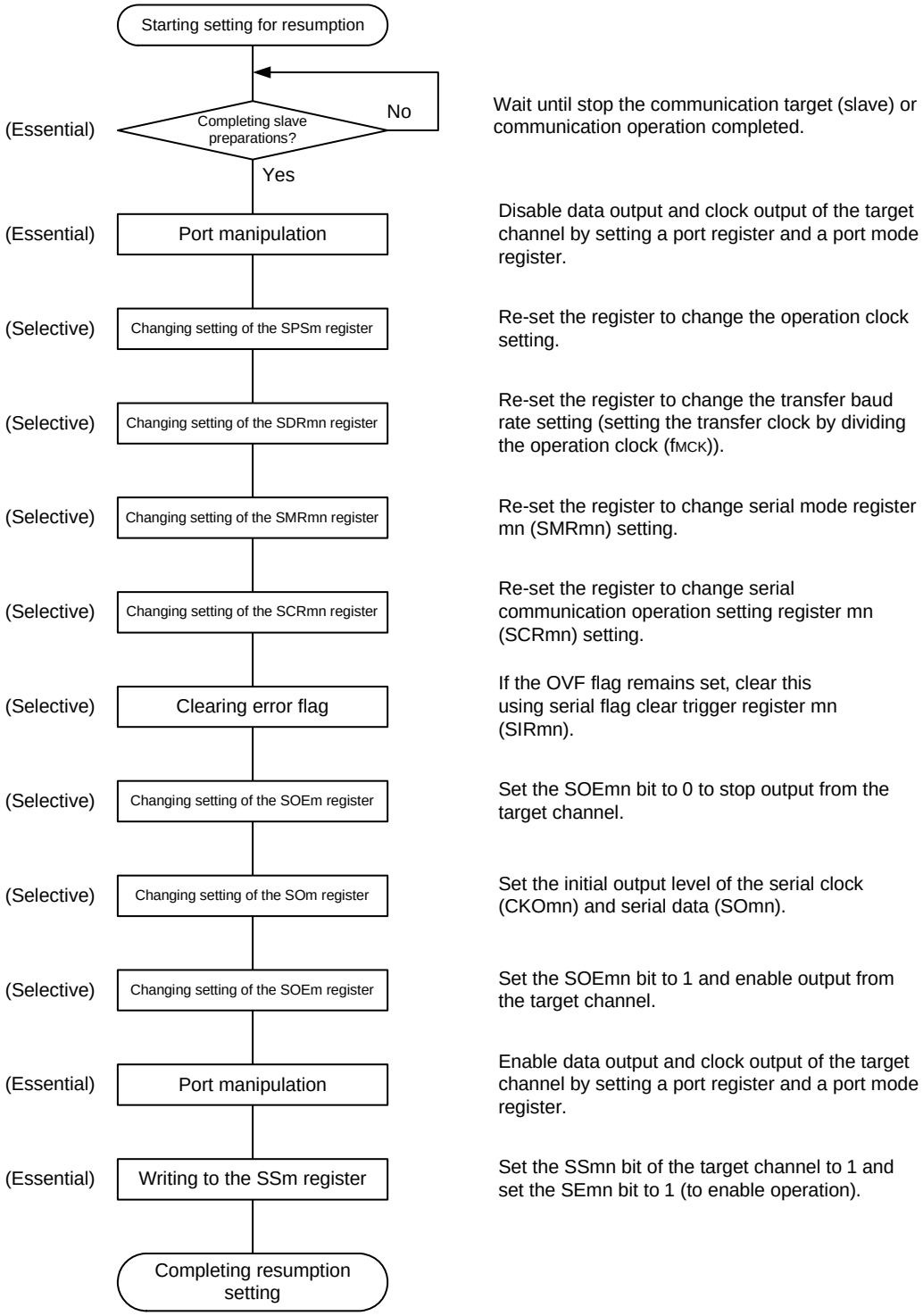
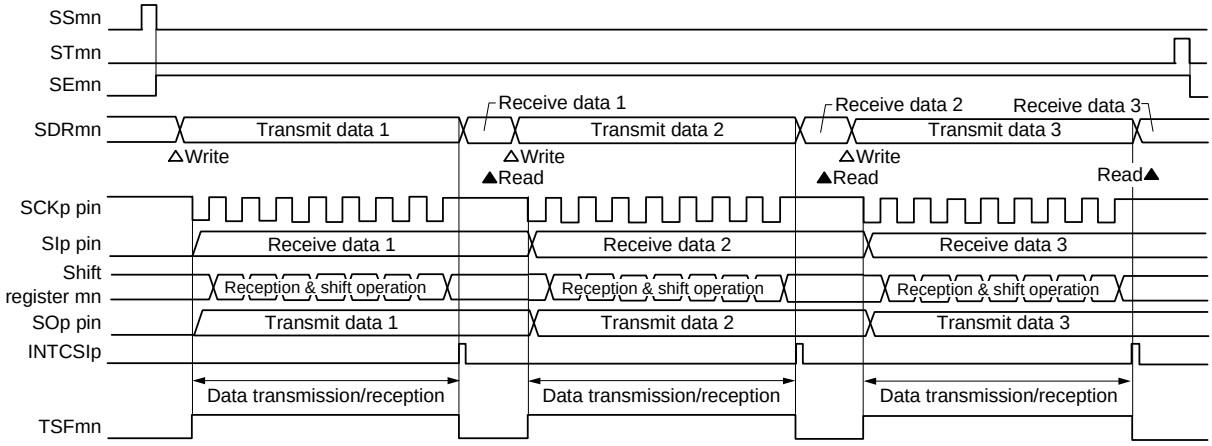


Figure 15 - 47 Procedure for Resuming Master Transmission/Reception



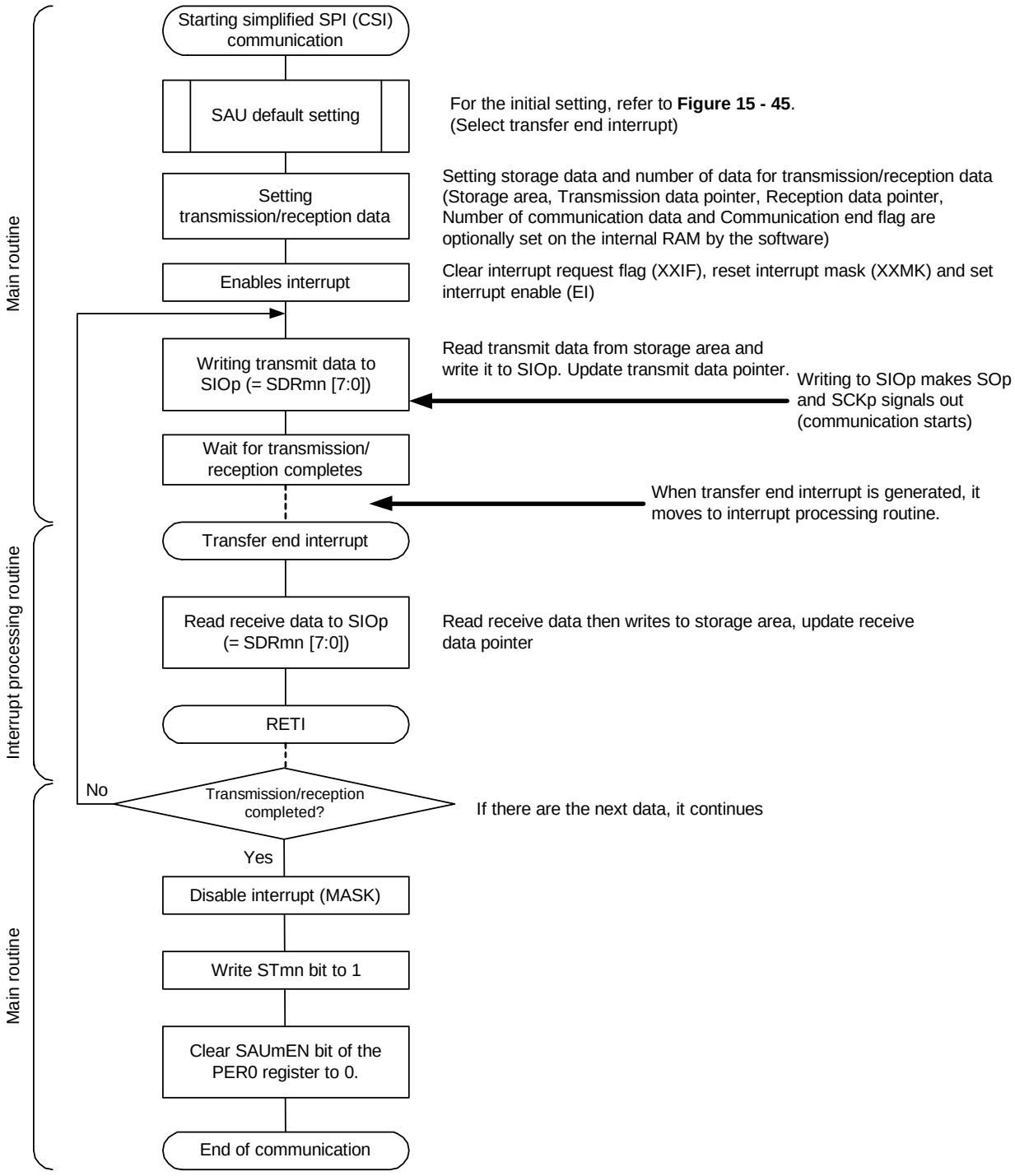
(3) Processing flow (in single-transmission/reception mode)

Figure 15 - 48 Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



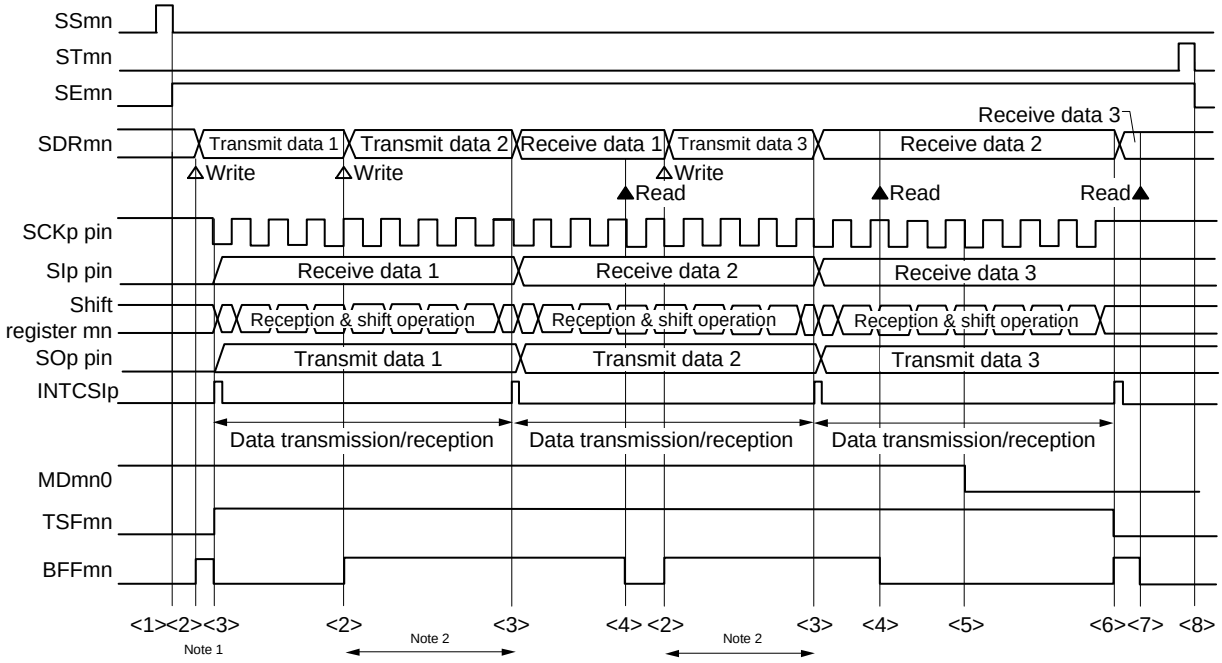
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20, 30), mn = 00, 02, 10, 12

Figure 15 - 49 Flowchart of Master Transmission/Reception (in Single- Transmission/Reception Mode)



(4) Processing flow (in continuous transmission/reception mode)

Figure 15 - 50 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Note 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

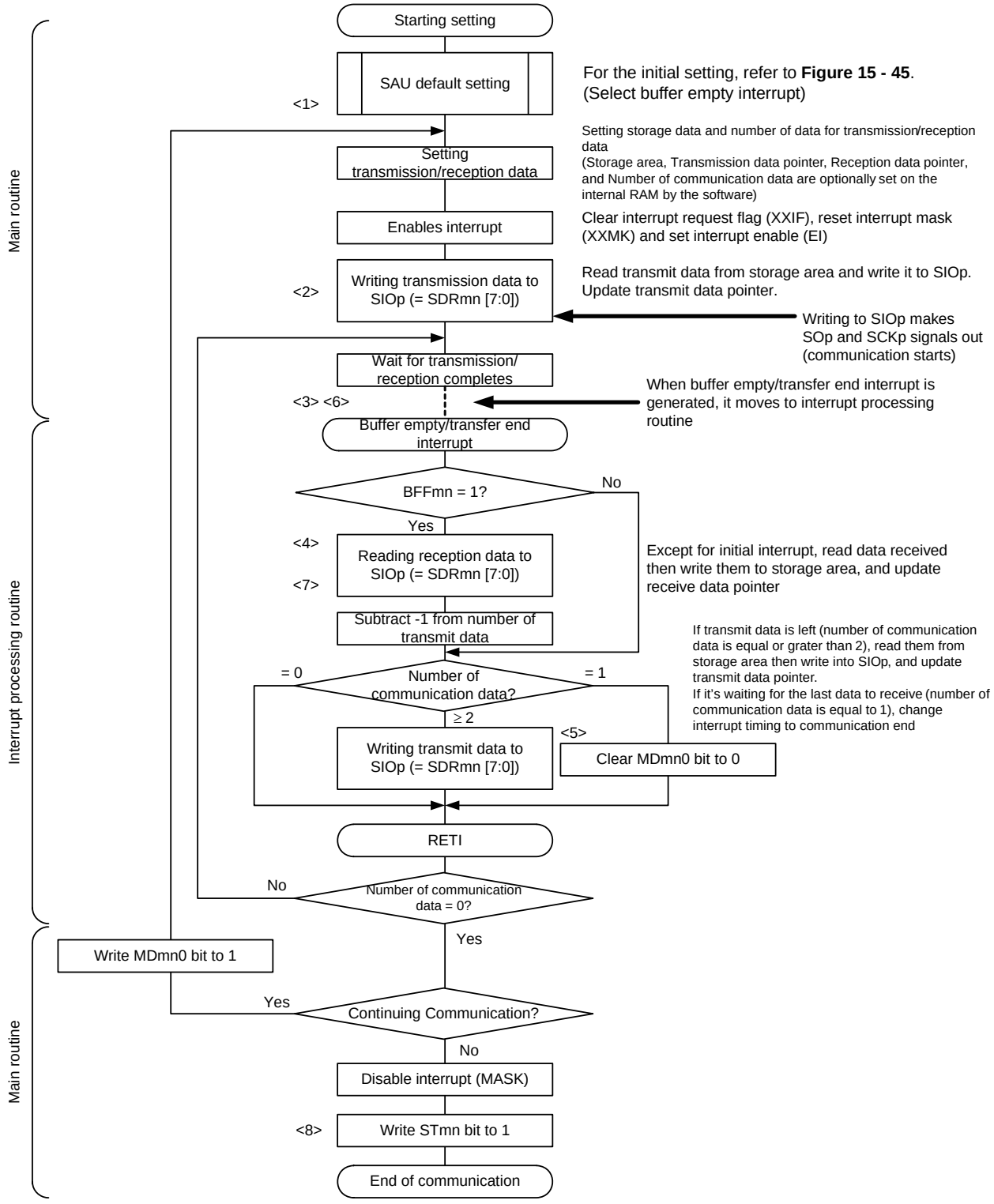
Note 2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in Figure 15 - 51 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode).

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20, 30), mn = 00, 02, 10, 12

Figure 15 - 51 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 - 50 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

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	CSI10	CSI20	CSI30
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1	Channel 2 of SAU1
Pins used	SCK00, SO00	SCK10, SO10	SCK20, SO20	SCK30, SO30
Interrupt	INTCSI00	INTCSI10	INTCSI20	INTCSI30
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	Overrun error detection flag (OVFmn) only			
Transfer data length	7 or 8 bits			
Transfer rate	Max. f_{MCK}/6 [Hz] Notes 1, 2.			
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts from the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.			
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse			
Data direction	MSB or LSB first			

Note 1. Because the external serial clock input to the SCK00, SCK10, SCK20, and SCK30 pins is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 34** or **CHAPTER 35 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel
f_{SCK}: Serial clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12

(1) Register setting

Figure 15 - 52 Example of Contents of Registers for Slave Transmission of Simplified SPI (CSI00, CSI10, CSI20, CSI30)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **15.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	Transmit data setting							

SIOp

(d) Serial output register m (SOm)... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 ×	1	CKOm0 ×	0	0	0	0	1	SOm2 0/1	1	SOm0 0/1

(e) Serial output enable register m (SOEm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	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 0/1	SSm1 ×	SSm0 0/1

Remark 1. mm: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20, 30), mn = 00, 02, 10, 12

Remark 2. ■: Setting is fixed in the simplified SPI (CSI) slave transmission mode,
■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 15 - 53 Initial Setting Procedure for Slave Transmission

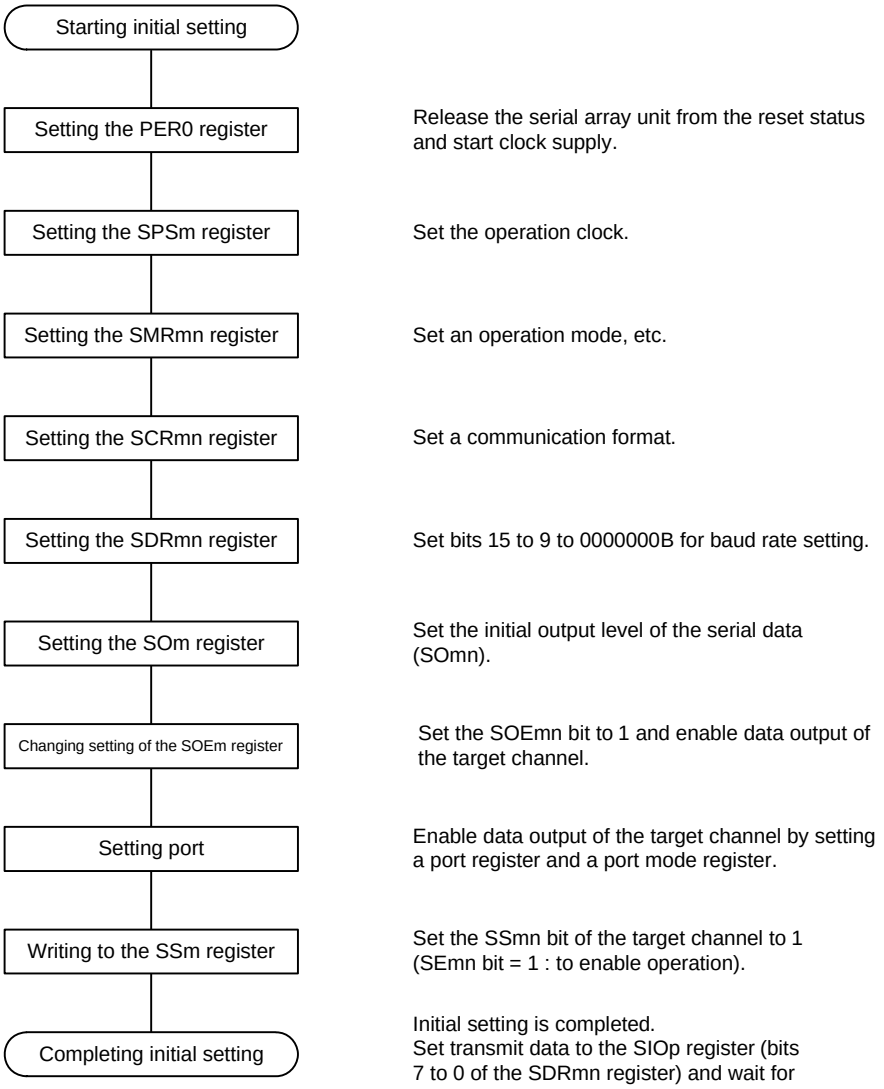


Figure 15 - 54 Procedure for Stopping Slave Transmission

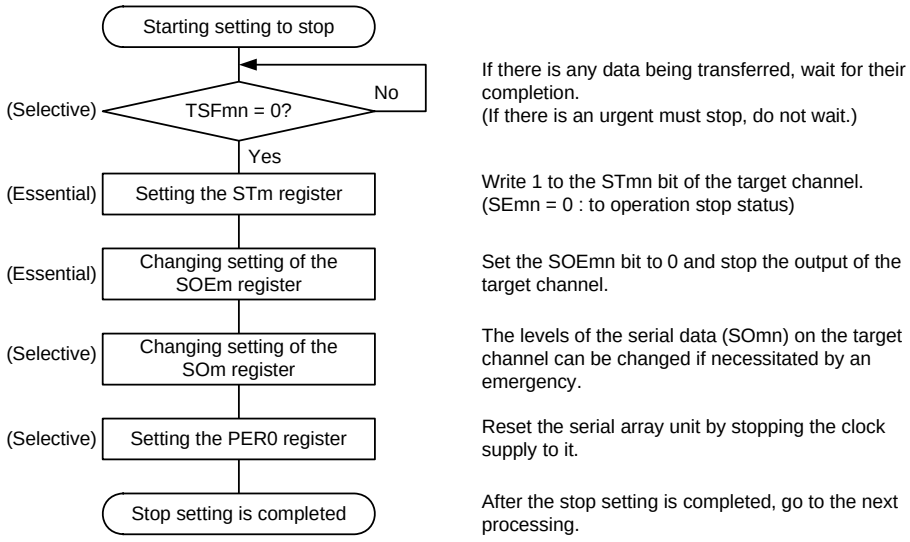
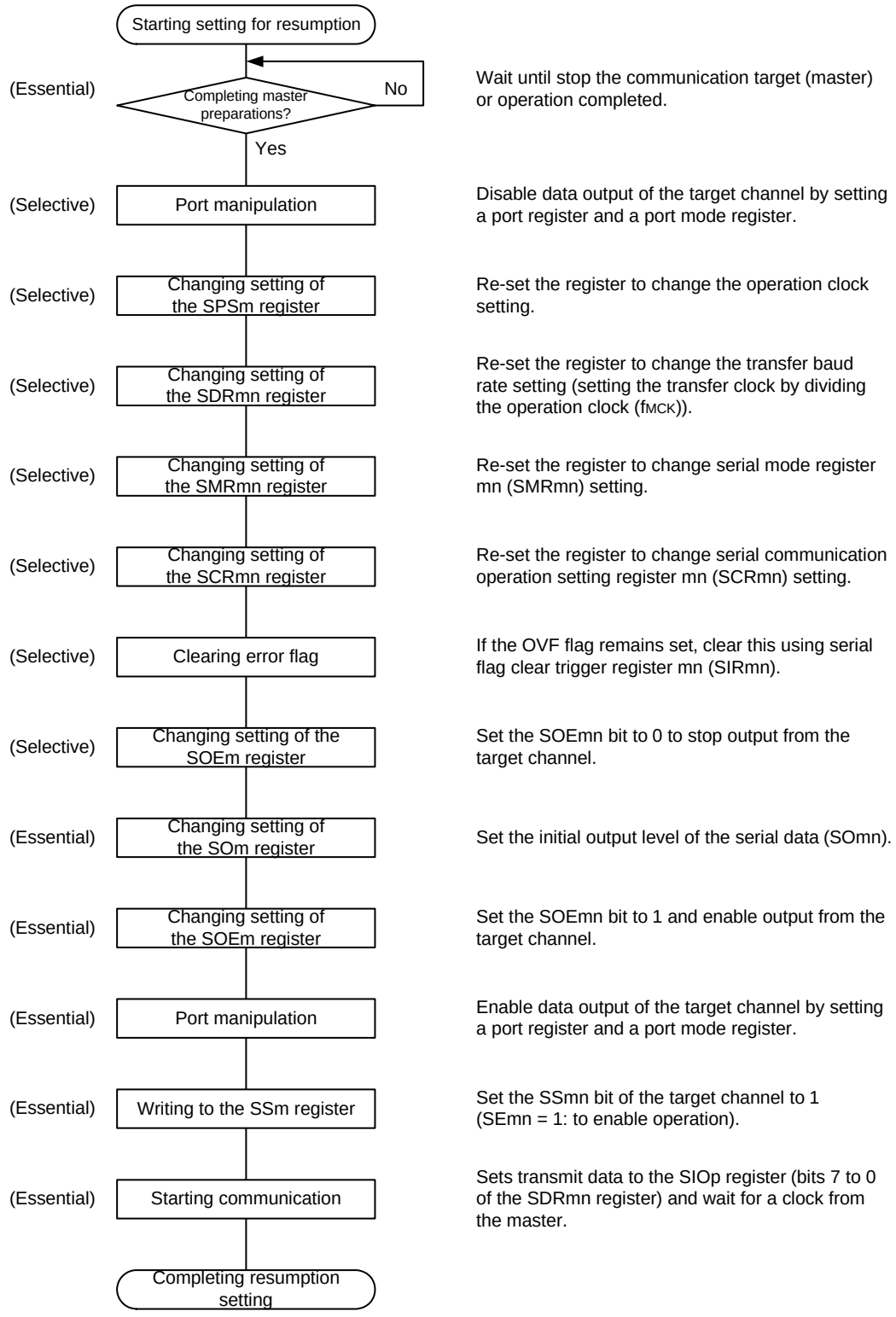


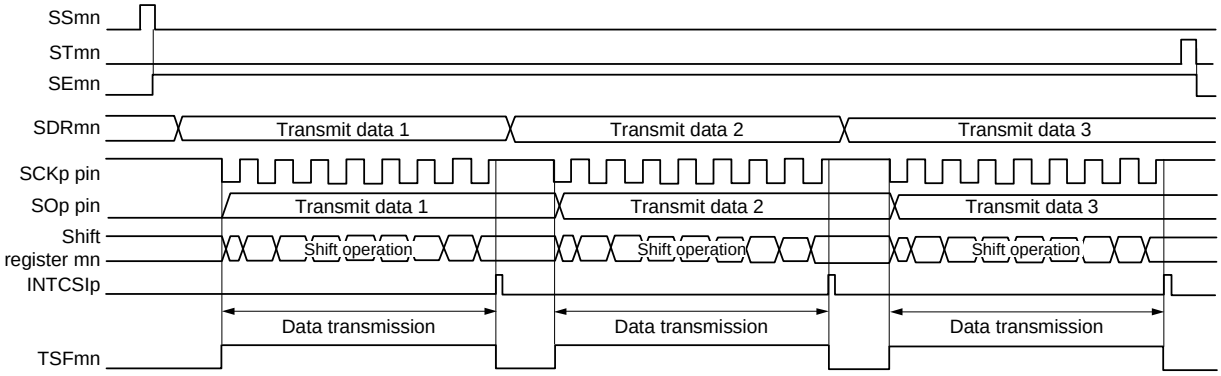
Figure 15 - 55 Procedure for Resuming Slave Transmission



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.

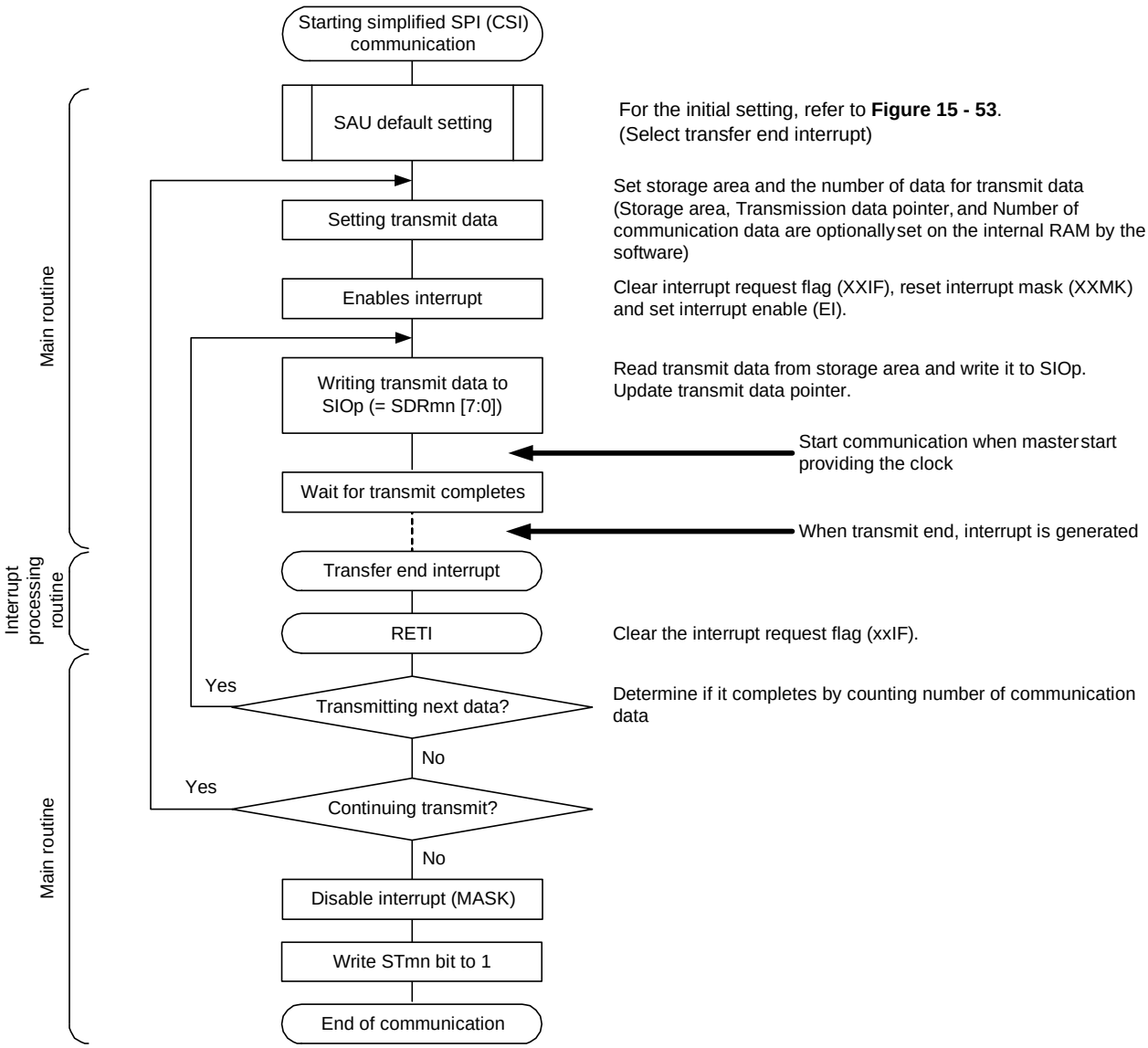
(3) Processing flow (in single-transmission mode)

Figure 15 - 56 Timing Chart of Slave Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



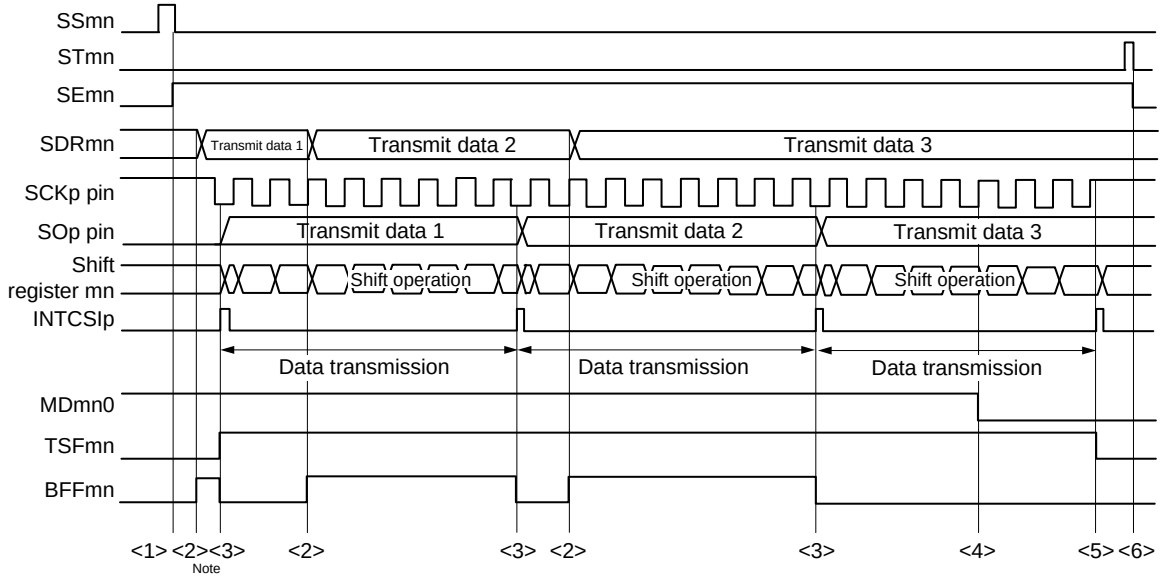
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20, 30), mn = 00, 02, 10, 12

Figure 15 - 57 Flowchart of Slave Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 15 - 58 Timing Chart of Slave Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

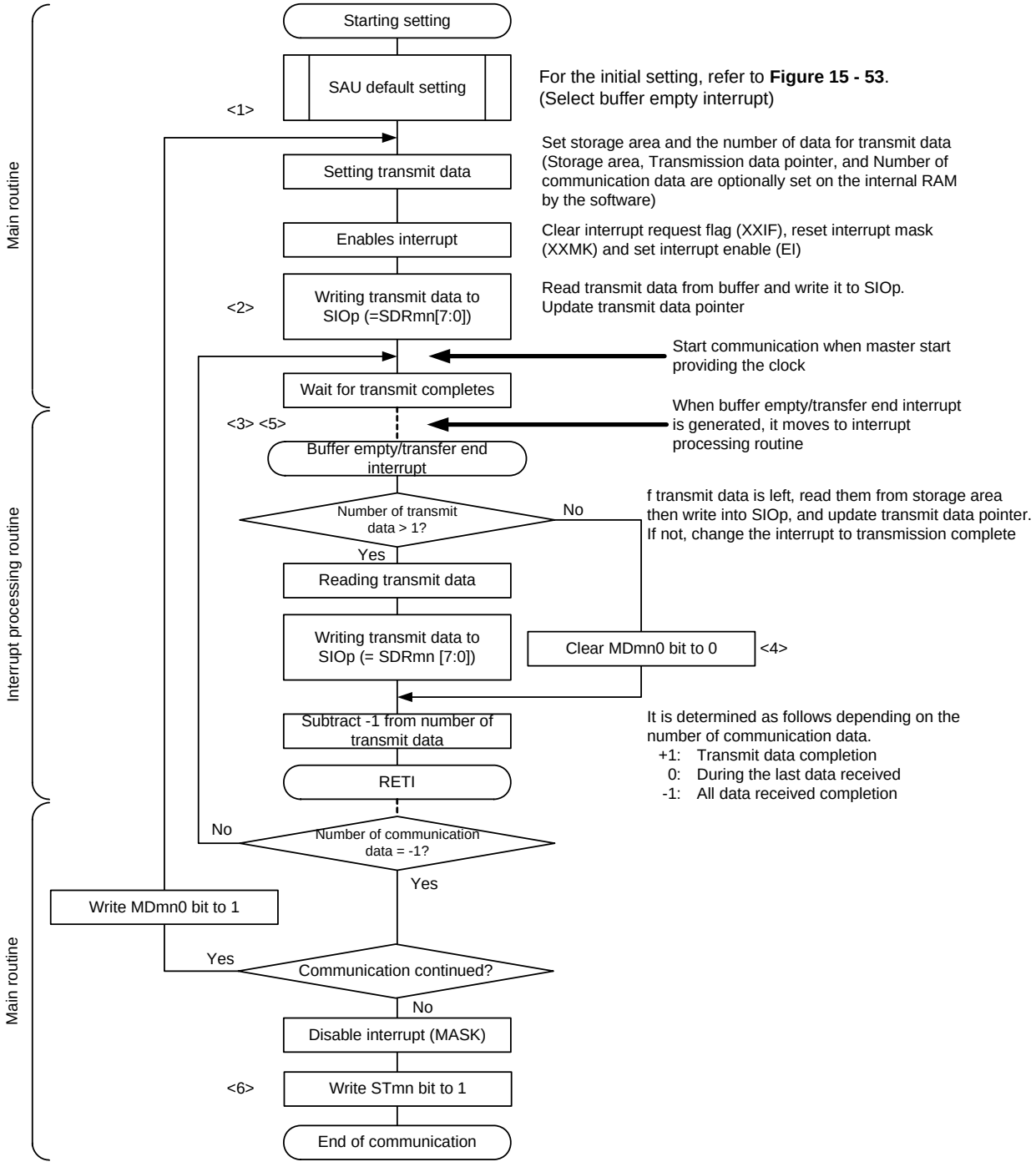


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20, 30), mn = 00, 02, 10, 12

Figure 15 - 59 Flowchart of Slave Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 15 - 58 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	CSI10	CSI20	CSI30
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1	Channel 2 of SAU1
Pins used	SCK00, SI00	SCK10, SI10	SCK20, SI20	SCK30, SI30
Interrupt	INTCSI00	INTCSI10	INTCSI20	INTCSI30
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error detection flag	Overrun error detection flag (OVFmn) only			
Transfer data length	7 or 8 bits			
Transfer rate	Max. f_{MCK}/6 [Hz] Notes 1, 2			
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data input starts from the start of the operation of the serial clock. • DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.			
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse			
Data direction	MSB or LSB first			

Note 1. Because the external serial clock input to the SCK00 and SCK10 pins is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 34** or **CHAPTER 35 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel
f_{SCK}: Serial clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12

(1) Register setting

Figure 15 - 60 Example of Contents of Registers for Slave Reception of Simplified SPI (CSI00, CSI10, CSI20, CSI30)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0

Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **15.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	Receive data							

SIOp

(d) Serial output register m (SOm)... The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 ×	1	CKOm0 ×	0	0	0	0	1	SOm2 ×	1	SOm0 ×

(e) Serial output enable register m (SOEm)... The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	0	SOEm0 ×

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20, 30), mn = 00, 02, 10, 12

Remark 2. ■: Setting is fixed in the simplified SPI (CSI) slave reception mode,
■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 15 - 61 Initial Setting Procedure for Slave Reception

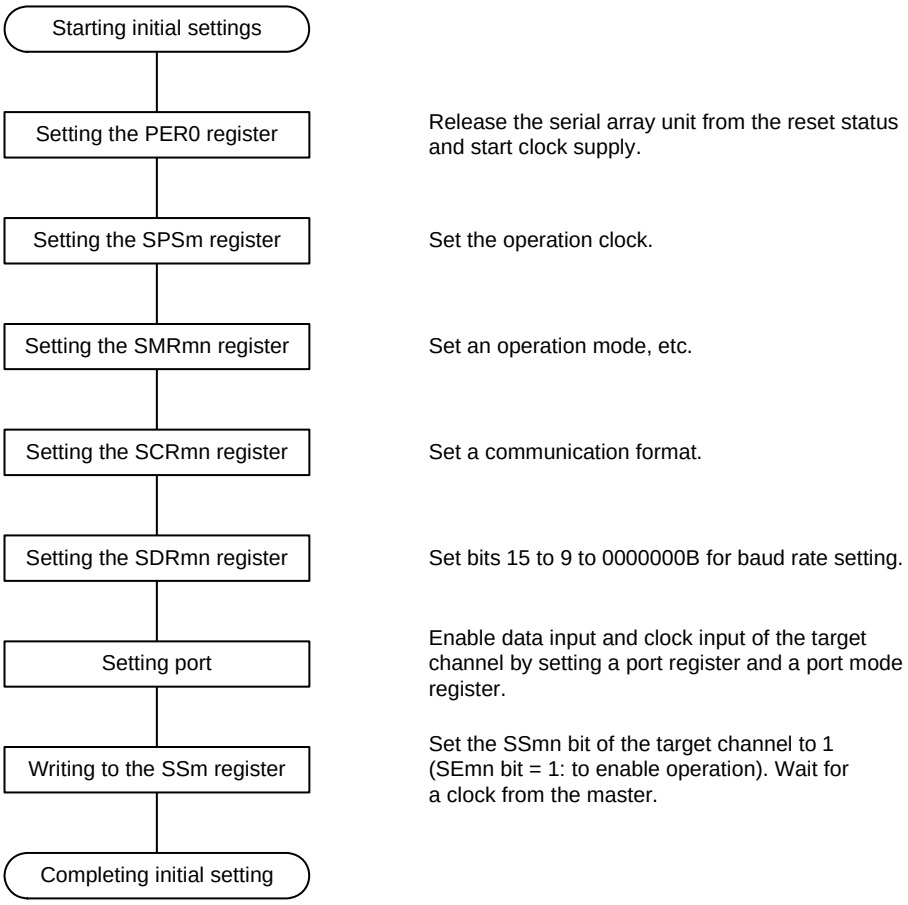


Figure 15 - 62 Procedure for Stopping Slave Reception

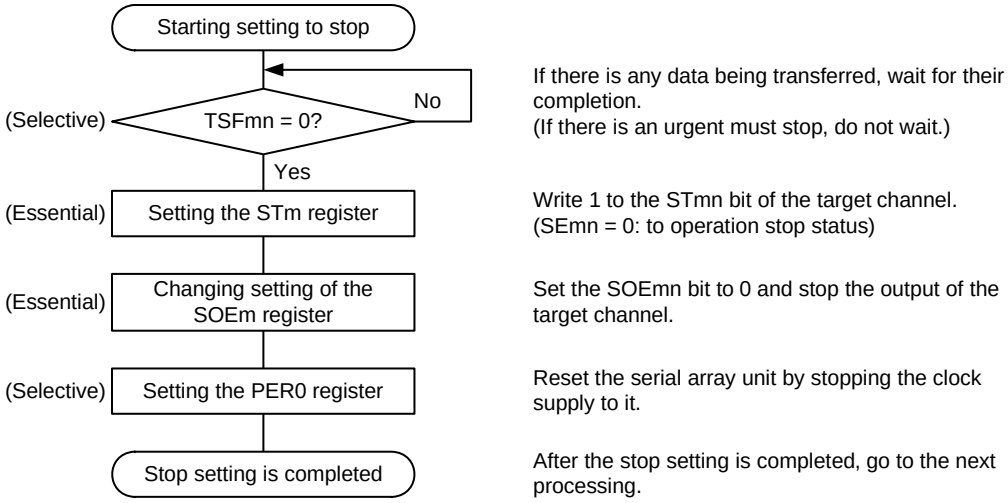
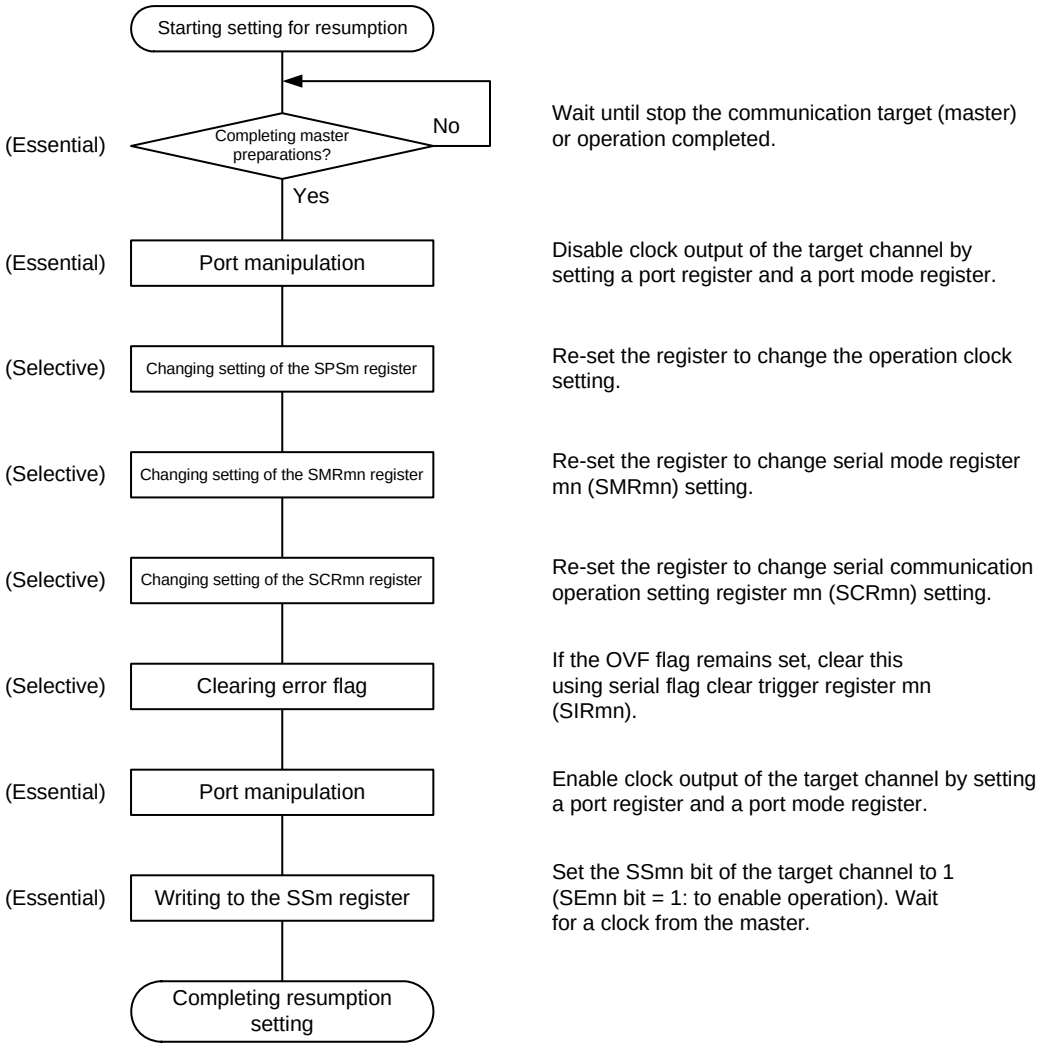


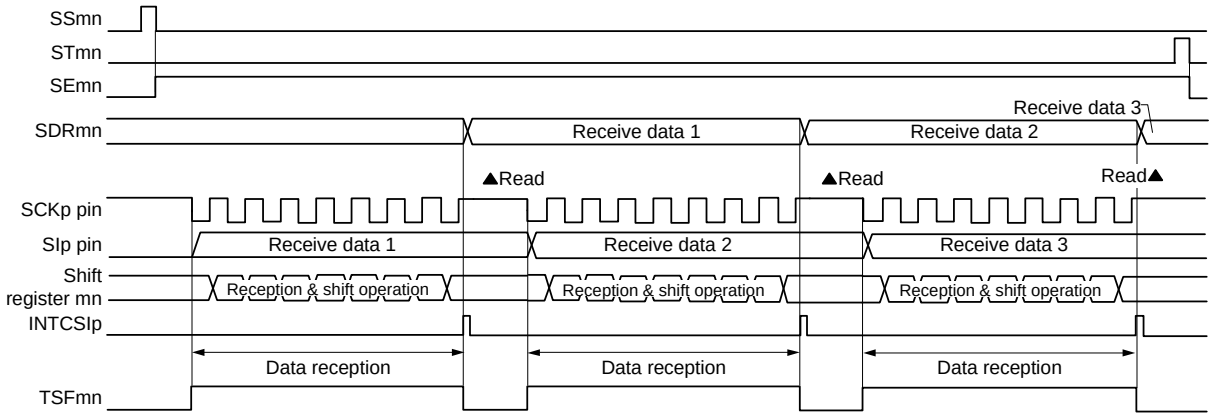
Figure 15 - 63 Procedure for Resuming Slave Reception



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.

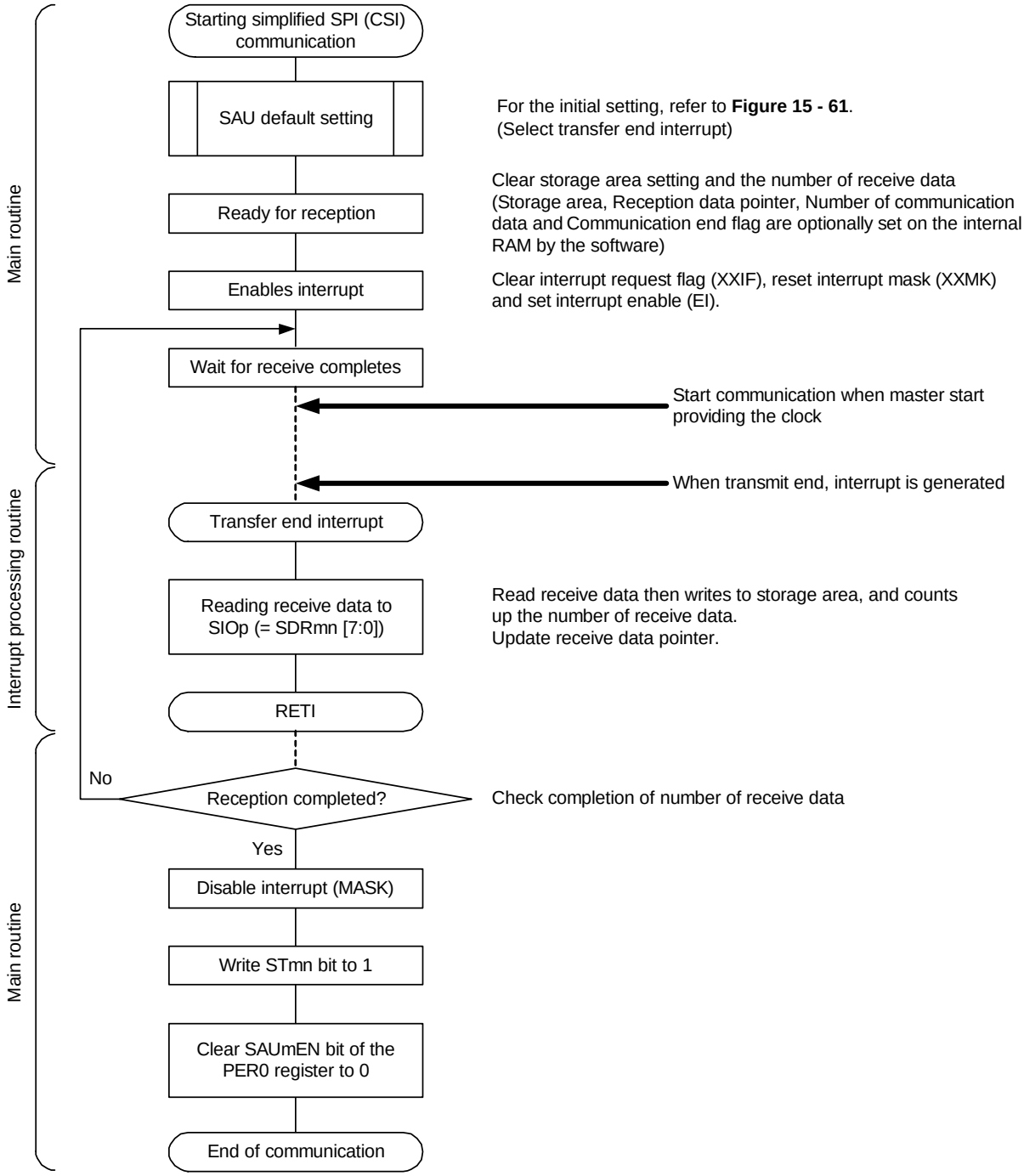
(3) Processing flow (in single-reception mode)

Figure 15 - 64 Timing Chart of Slave Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20, 30), mn = 00, 02, 10, 12

Figure 15 - 65 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	CSI10	CSI20	CSI30
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1	Channel 2 of SAU1
Pins used	SCK00, SI00, SO00	SCK10, SI10, SO10	SCK20, SI20, SO20	SCK30, SI30, SO30
Interrupt	INTCSI00	INTCSI10	INTCSI20	INTCSI30
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	Overrun error detection flag (OVFmn) only			
Transfer data length	7 or 8 bits			
Transfer rate	Max. f_{MCK}/6 [Hz] Notes 1, 2.			
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data I/O starts from the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.			
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse			
Data direction	MSB or LSB first			

Note 1. Because the external serial clock input to the SCK00, SCK10, SCK20, and SCK30 pins is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 34** or **CHAPTER 35 ELECTRICAL SPECIFICATIONS**).

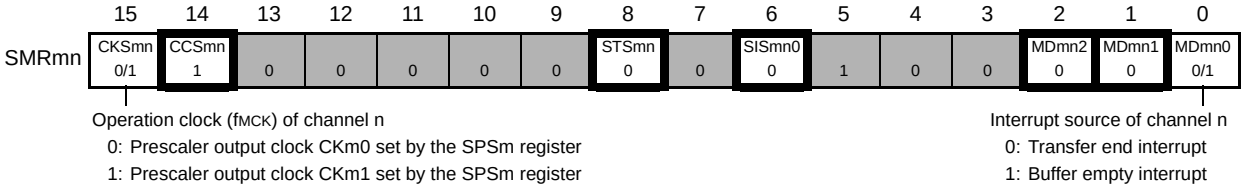
Remark 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: Serial clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12

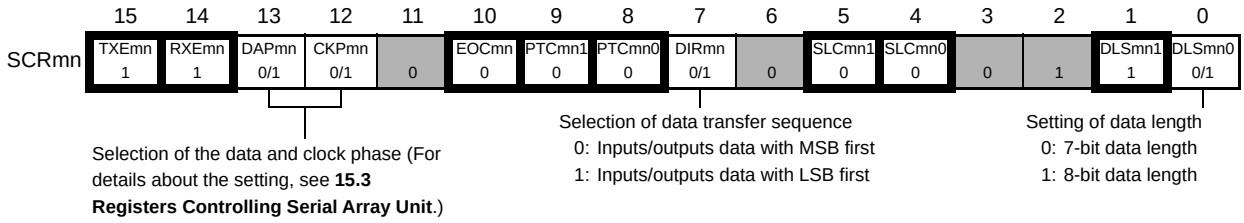
(1) Register setting

Figure 15 - 66 Example of Contents of Registers for Slave Transmission/Reception of Simplified SPI (CSI00, CSI10, CSI20, CSI30)

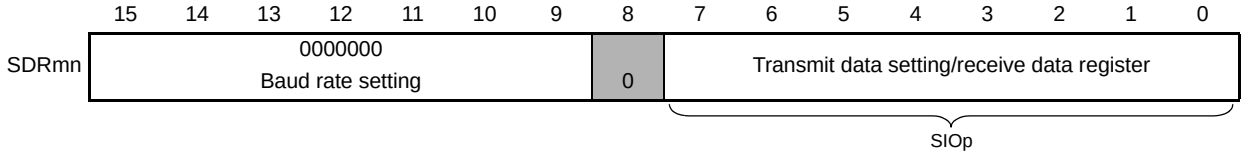
(a) Serial mode register mn (SMRmn)



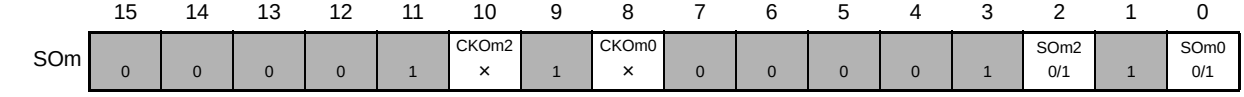
(b) Serial communication operation setting register mn (SCRmn)



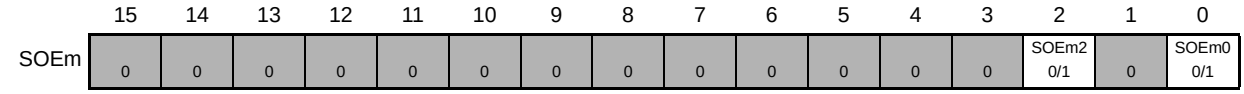
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



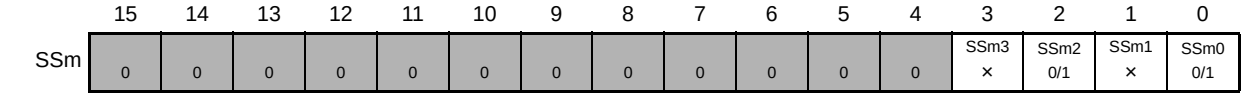
(d) Serial output register m (SOm)... Sets only the bits of the target channel.



(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.



Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20, 30), mn = 00, 02, 10, 12

Remark 2. ■: Setting is fixed in the simplified SPI (CSI) master transmission/reception mode

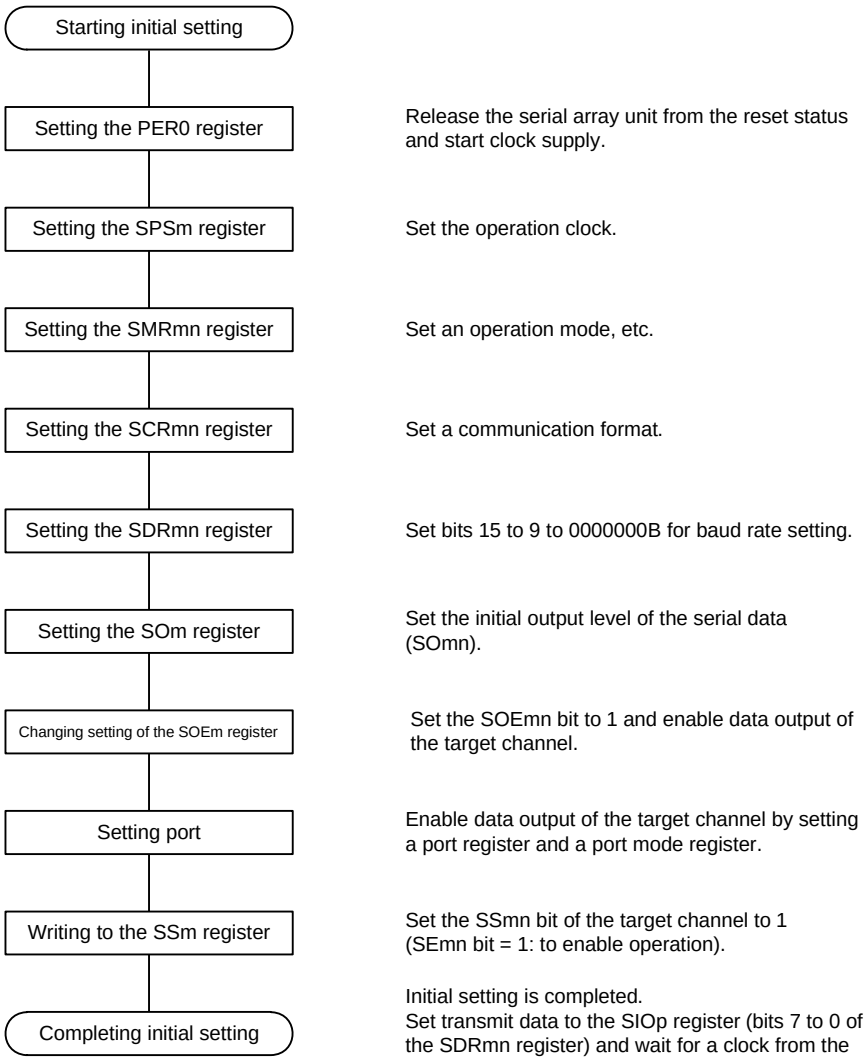
■: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 15 - 67 Initial Setting Procedure for Slave Transmission/Reception



Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Figure 15 - 68 Procedure for Stopping Slave Transmission/Reception

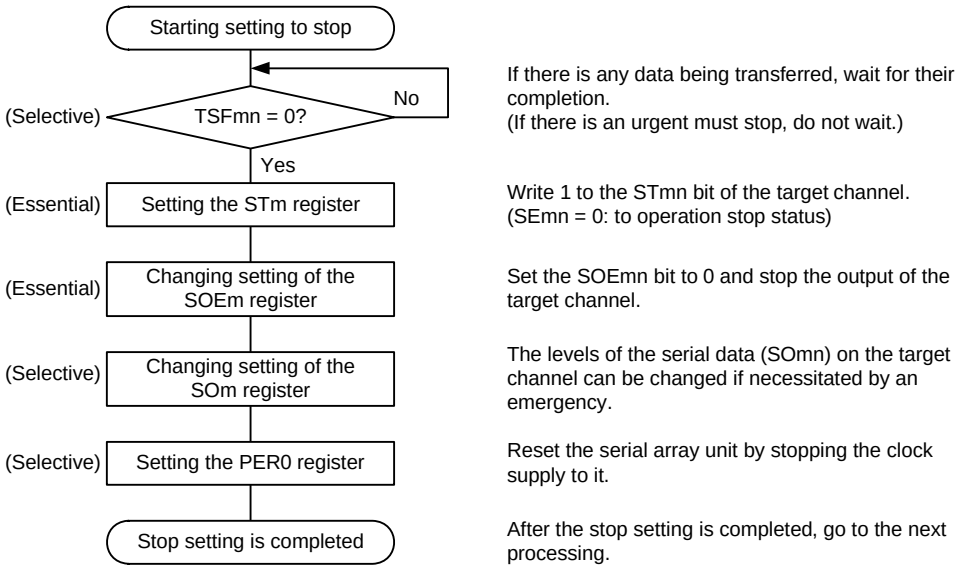
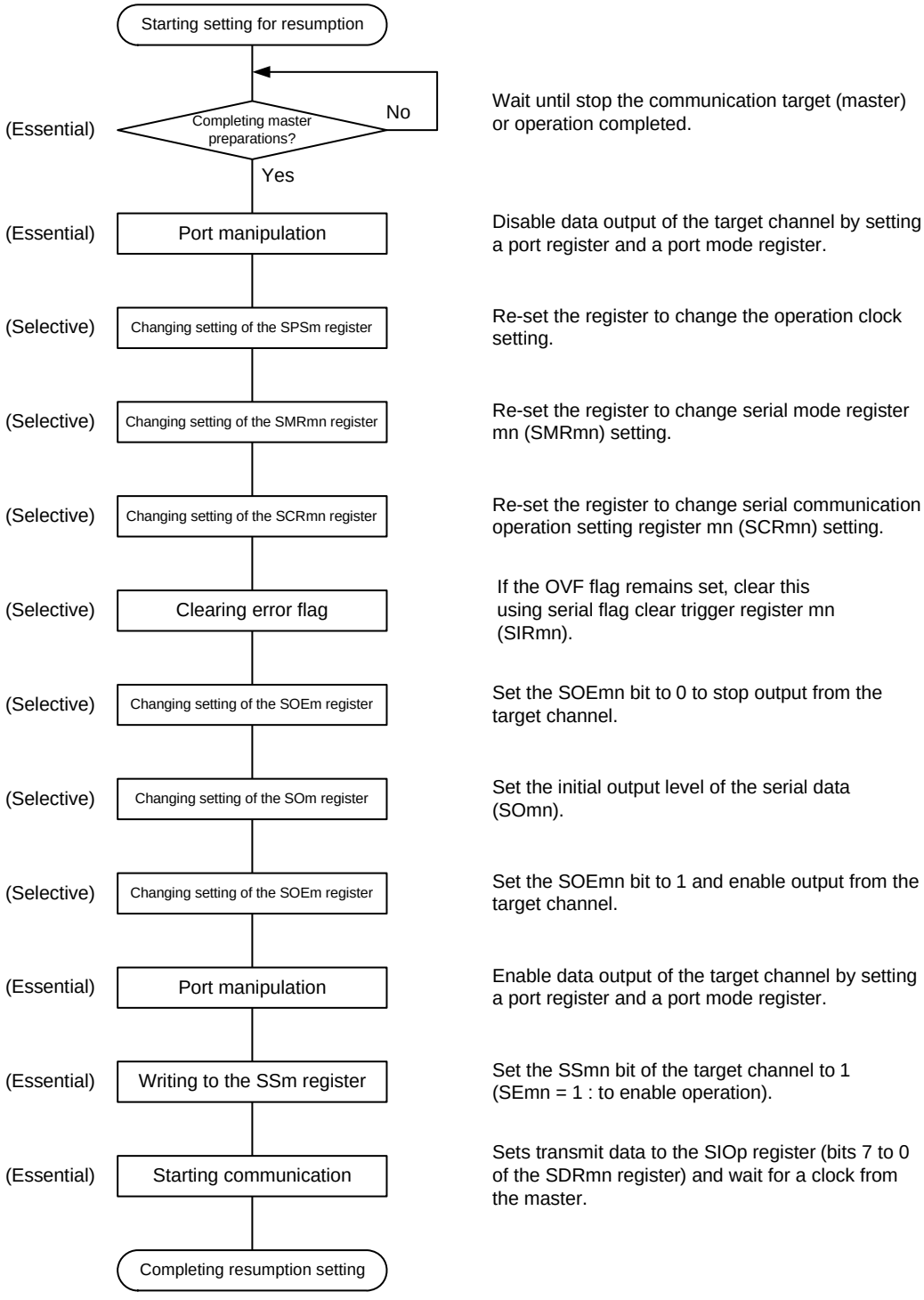


Figure 15 - 69 Procedure for Resuming Slave Transmission/Reception

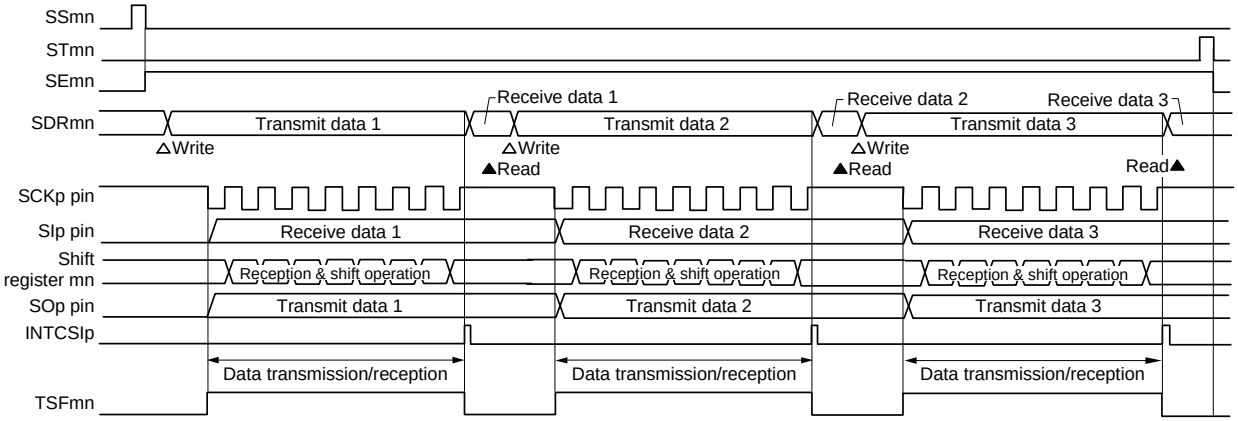


Caution 1. Be sure to set transmit data to the SIOp register before the clock from the master is started.

Caution 2. If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

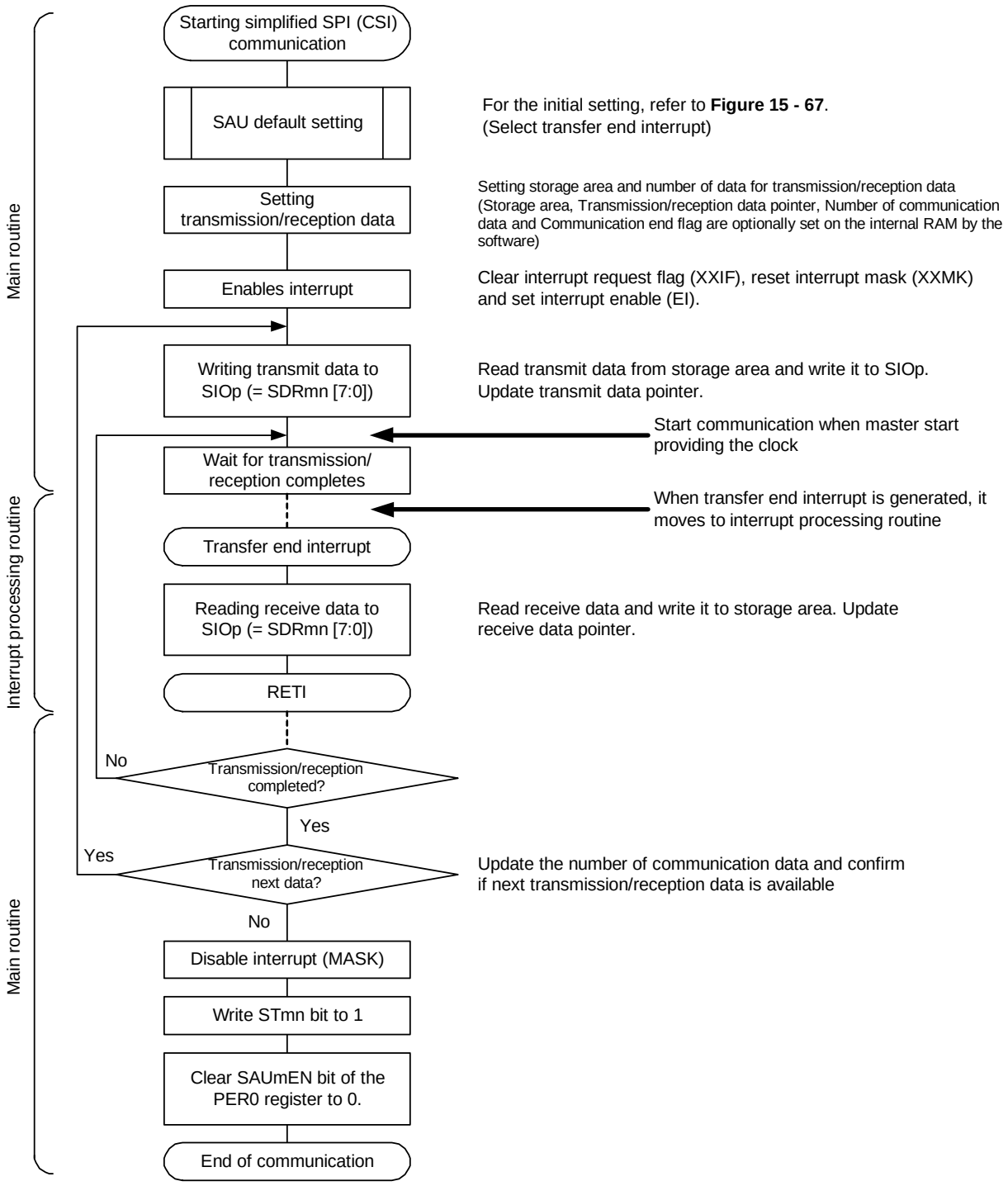
(3) Processing flow (in single-transmission/reception mode)

Figure 15 - 70 Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20, 30), mn = 00, 02, 10, 12

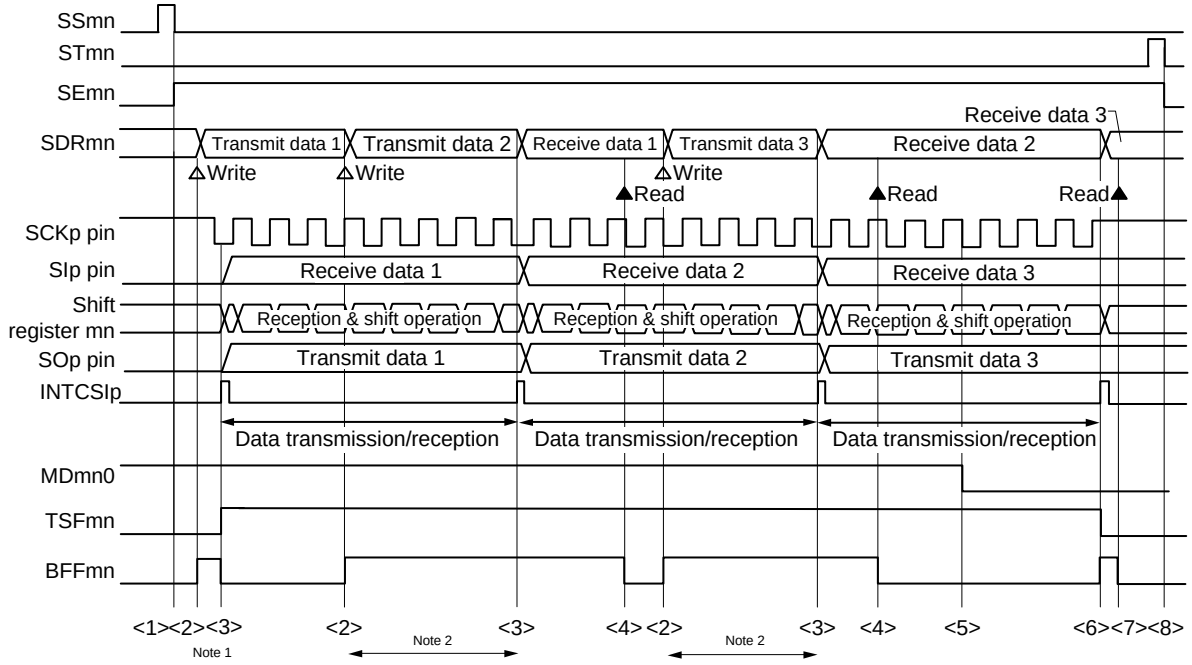
Figure 15 - 71 Flowchart of Slave Transmission/Reception (in Single- Transmission/Reception Mode)



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 - 72 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Note 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

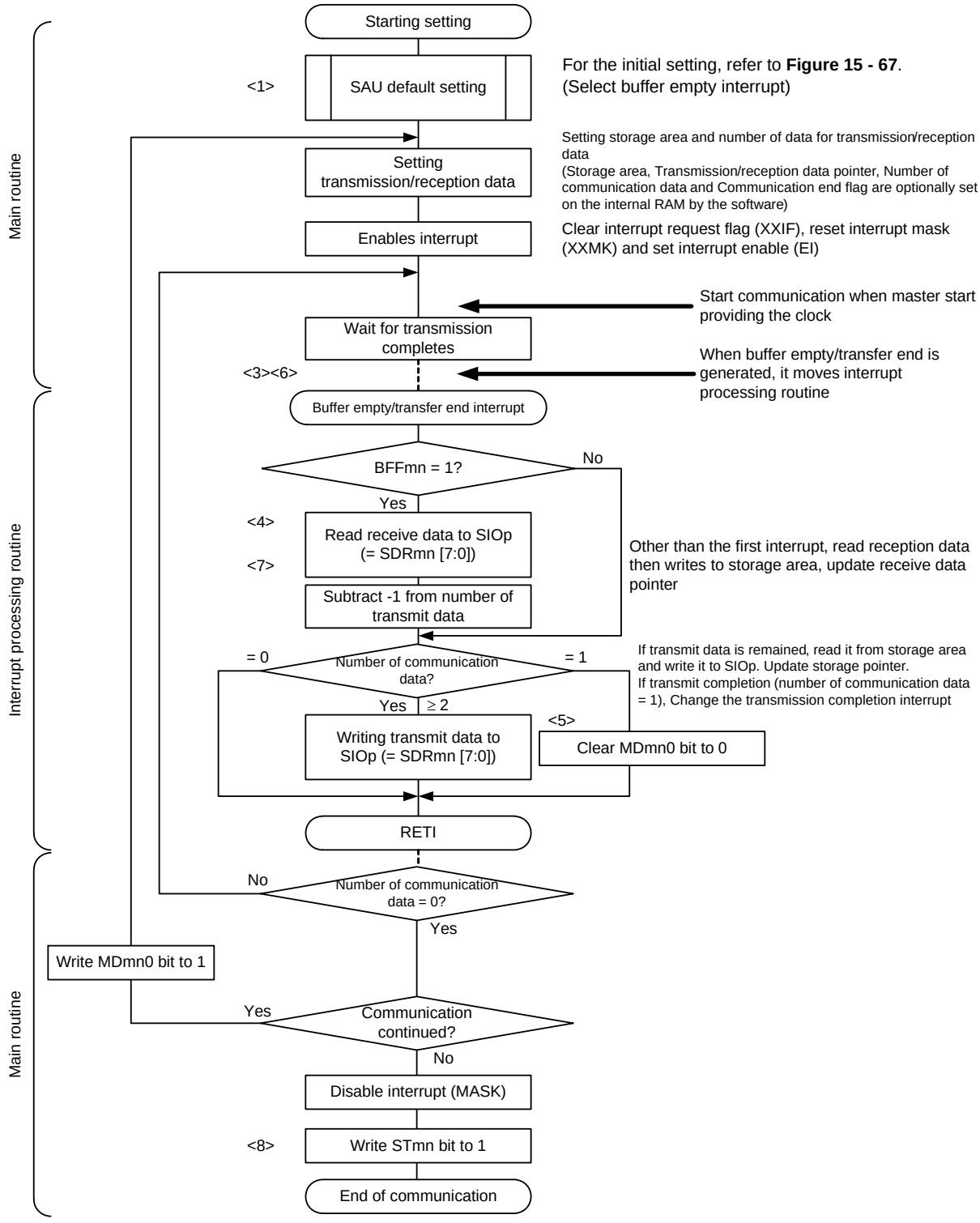
Note 2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in Figure 15 - 73 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode).

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20, 30), mn = 00, 02, 10, 12

Figure 15 - 73 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 - 72 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

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 and CSI20 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 - 75 Flowchart of SNOOZE Mode Operation (once startup)** and **Figure 15 - 77 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 SSm0 bit of serial channel start register m (SSm) to 1.
- The CPU shifts to the SNOOZE mode on detecting the valid edge of the SCKp signal following a transition to the STOP mode. A CSIp starts reception on detecting input of the serial clock on the SCKp pin.

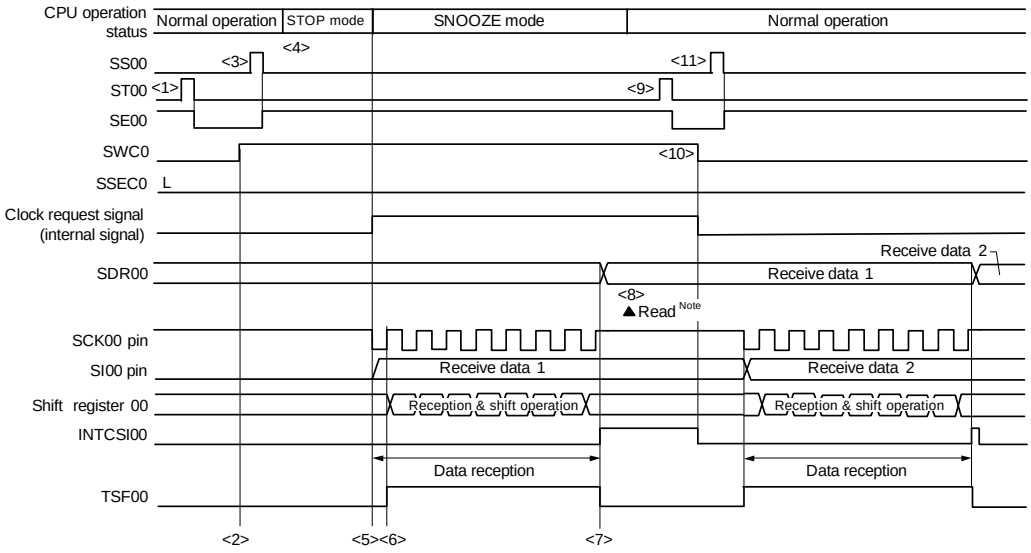
Caution 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for f_{CLK}.

Caution 2. The maximum transfer rate when using CSIp in the SNOOZE mode is 1 Mbps.

(1) SNOOZE mode operation (once startup)

<R>

Figure 15 - 74 Timing Chart of SNOOZE Mode Operation (once startup) (Type 1: DAPmn = 0, CKPmn = 0)



Note Only read received data while SWCm = 1 and before the next valid edge of the SCKp pin input is detected.

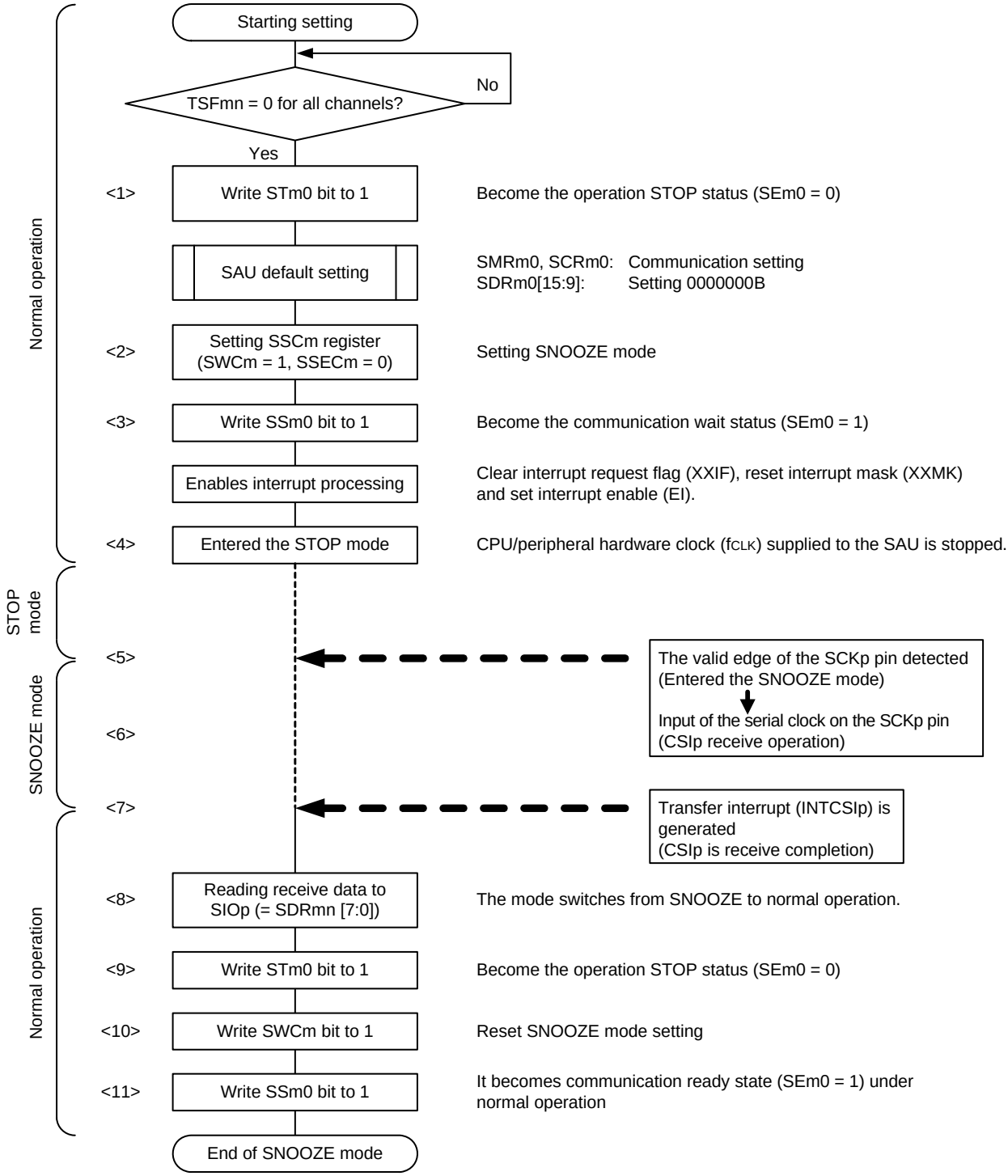
Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (clear the SEm0 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

Caution 2. When SWCm = 1, the BFFm1 and OVFm1 flags will not change.

Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 15 - 75 Flowchart of SNOOZE Mode Operation (once startup).

Remark 2. m = 0, 1; p = 00, 20

Figure 15 - 75 Flowchart of SNOOZE Mode Operation (once startup)

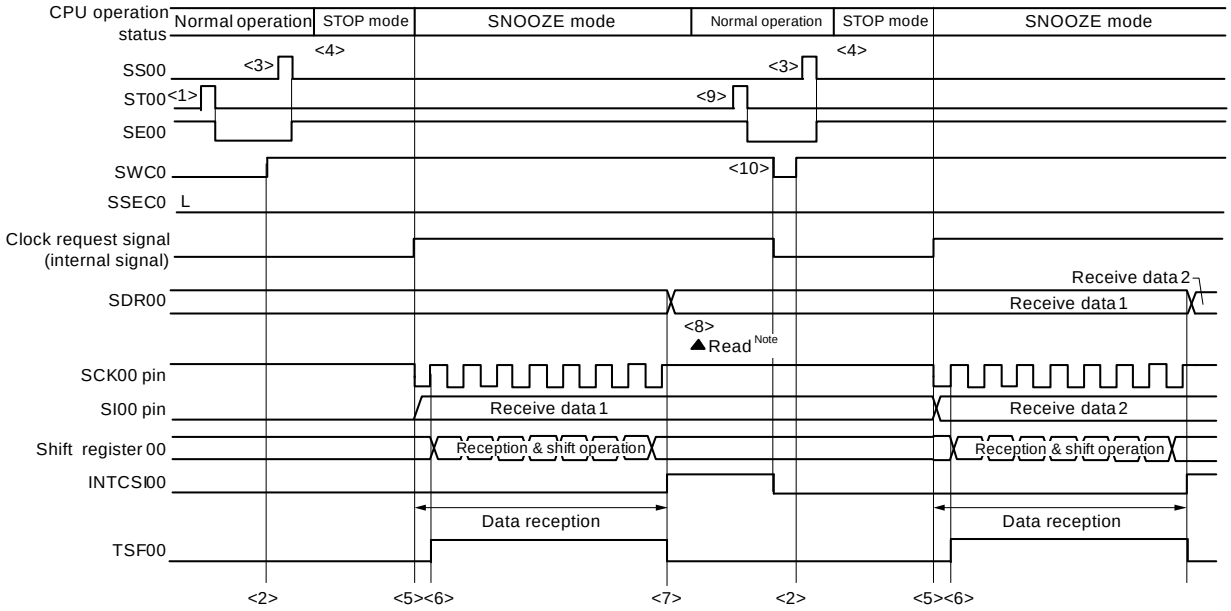


Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 15 - 74 Timing Chart of SNOOZE Mode Operation (once startup).

Remark 2. m = 0, 1; p = 00, 20

(2) SNOOZE mode operation (continuous startup)

<R> **Figure 15 - 76 Timing Chart of SNOOZE Mode Operation (continuous startup) (Type 1: DAPmn = 0, CKPmn = 0)**



Note Only read received data while SWCm = 1 and before the next valid edge of the SCKp pin input is detected.

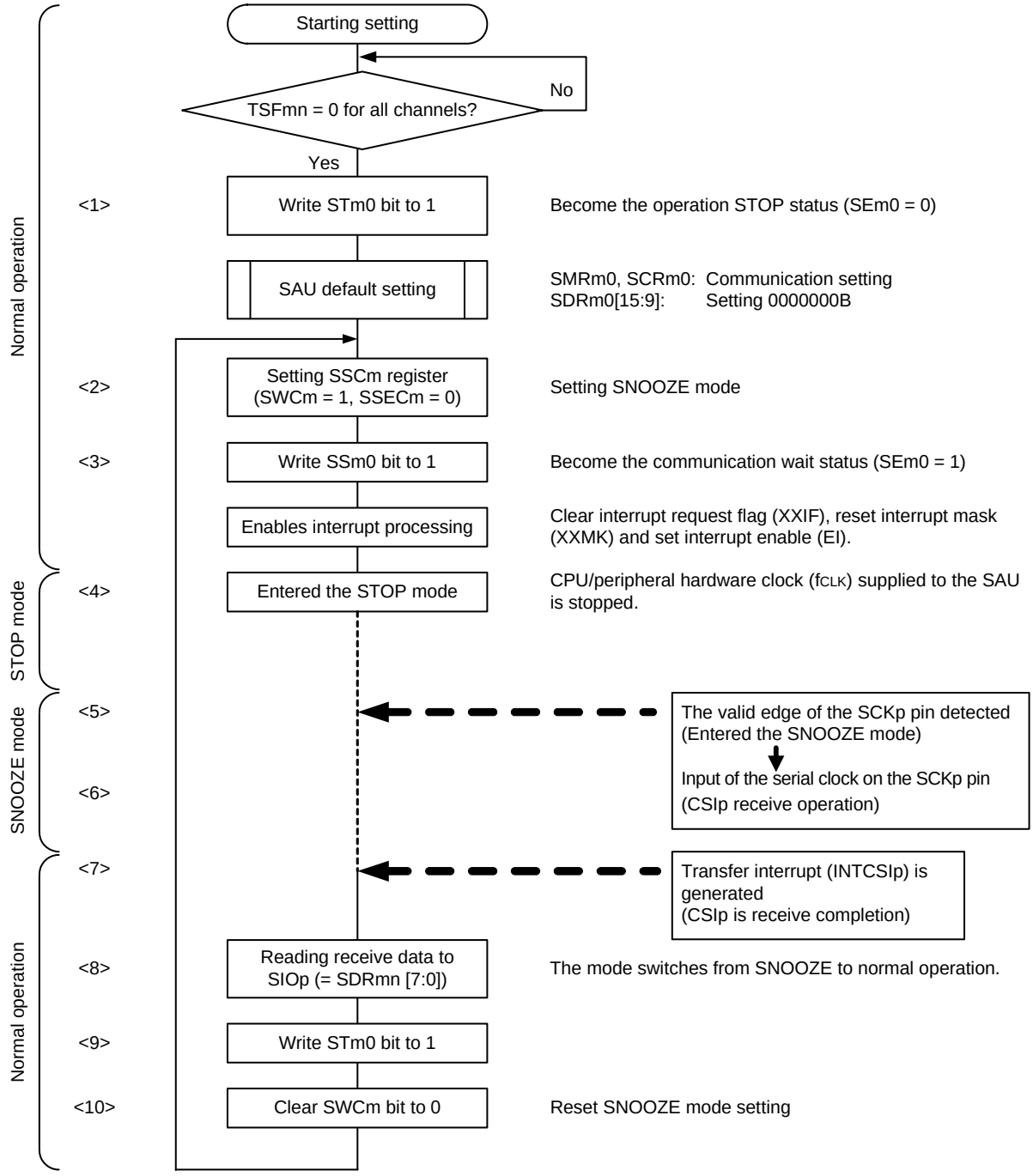
Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (clear the SEm0 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE release).

Caution 2. When SWCm = 1, the BFFm1 and OVFm1 flags will not change.

Remark 1. <1> to <10> in the figure correspond to <1> to <10> in Figure 15 - 77 Flowchart of SNOOZE Mode Operation (continuous startup).

Remark 2. m = 0, 1; p = 00, 20

Figure 15 - 77 Flowchart of SNOOZE Mode Operation (continuous startup)



Remark 1. <1> to <10> in the figure correspond to <1> to <10> in Figure 15 - 76 Timing Chart of SNOOZE Mode Operation (continuous startup).

Remark 2. m = 0, 1; p = 00, 20

15.5.8 Calculating transfer clock frequency

The transfer clock frequency for simplified SPI (CSI00, CSI10) 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} = 24 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	24 MHz
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	12 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	6 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	3 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	1.5 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	750 kHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	375 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	187.5 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	93.8 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	46.9 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	23.4 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	11.7 kHz
	×	×	×	×	1	1	0	0	$f_{CLK}/2^{12}$	5.86 kHz
	×	×	×	×	1	1	0	1	$f_{CLK}/2^{13}$	2.93 kHz
	×	×	×	×	1	1	1	0	$f_{CLK}/2^{14}$	1.46 kHz
	×	×	×	×	1	1	1	1	$f_{CLK}/2^{15}$	732 Hz
1	0	0	0	0	×	×	×	×	f_{CLK}	24 MHz
	0	0	0	1	×	×	×	×	$f_{CLK}/2$	12 MHz
	0	0	1	0	×	×	×	×	$f_{CLK}/2^2$	6 MHz
	0	0	1	1	×	×	×	×	$f_{CLK}/2^3$	3 MHz
	0	1	0	0	×	×	×	×	$f_{CLK}/2^4$	1.5 MHz
	0	1	0	1	×	×	×	×	$f_{CLK}/2^5$	750 kHz
	0	1	1	0	×	×	×	×	$f_{CLK}/2^6$	375 kHz
	0	1	1	1	×	×	×	×	$f_{CLK}/2^7$	187.5 kHz
	1	0	0	0	×	×	×	×	$f_{CLK}/2^8$	93.8 kHz
	1	0	0	1	×	×	×	×	$f_{CLK}/2^9$	46.9 kHz
	1	0	1	0	×	×	×	×	$f_{CLK}/2^{10}$	23.4 kHz
	1	0	1	1	×	×	×	×	$f_{CLK}/2^{11}$	11.7 kHz
	1	1	0	0	×	×	×	×	$f_{CLK}/2^{12}$	5.86 kHz
	1	1	0	1	×	×	×	×	$f_{CLK}/2^{13}$	2.93 kHz
	1	1	1	0	×	×	×	×	$f_{CLK}/2^{14}$	1.46 kHz
	1	1	1	1	×	×	×	×	$f_{CLK}/2^{15}$	732 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remark 1. ×: Don't care

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12

15.5.9 Procedure for processing errors that occurred during simplified SPI (CSI00, CSI10, CSI20, CSI30) communication

The procedure for processing errors that occurred during simplified SPI (CSI00, CSI10, CSI20, CSI30) communication is described in Figure 15 - 78.

Figure 15 - 78 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, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12

15.6 Operation of UART (UART0 to UART3) 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 can be implemented by using UART0 and 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 and UART2 reception support the SNOOZE mode. When RxD pin input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only UART0, UART2 can be specified for the reception baud rate adjustment function.

The LIN-bus is accepted in UART0 (channels 0 and 1 of unit 0).

[LIN-bus functions]
- Wakeup signal detection
- Break field (BF) detection
- Sync field measurement, baud rate calculation

} Using the external interrupt (INTP0) and timer array unit 0 (channel 7)

Note Only UART0, UART2 can be specified for the 9-bit data length.

UART0 uses channels 0 and 1 of SAU0.
UART1 uses channels 2 and 3 of SAU0.
UART2 uses channels 0 and 1 of SAU1.
UART3 uses channels 2 and 3 of SAU1.

• 80-pin, 85-pin, and 100-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0 (supporting LIN-bus)	IIC00
	1	—		—
	2	CSI10	UART1	IIC10
	3	—		—
1	0	CSI20	UART2	IIC20
	1	—		—
	2	CSI30	UART3	IIC30
	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 or 3 of the same unit can be used for a function other than UART0, such as CSI10, UART1, and IIC10.

Caution When using a serial array unit for UART, both the transmitter side (even-numbered channel) and the receiver side (odd-numbered channel) can only be used for UART.

UART performs the following four types of communication operations.

• UART transmission (See **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**.)

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	UART2	UART3
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1	Channel 2 of SAU1
Pins used	TxD0	TxD1	TxD2	TxD3
Interrupt	INTST0	INTST1	INTST2	INTST3
	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			

Note 1. Only UART0, UART2 can be specified for the 9-bit data length.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 34** or **CHAPTER 35 ELECTRICAL SPECIFICATIONS**).

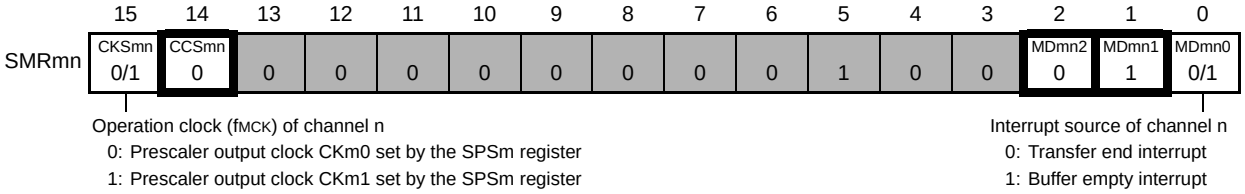
Remark 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12

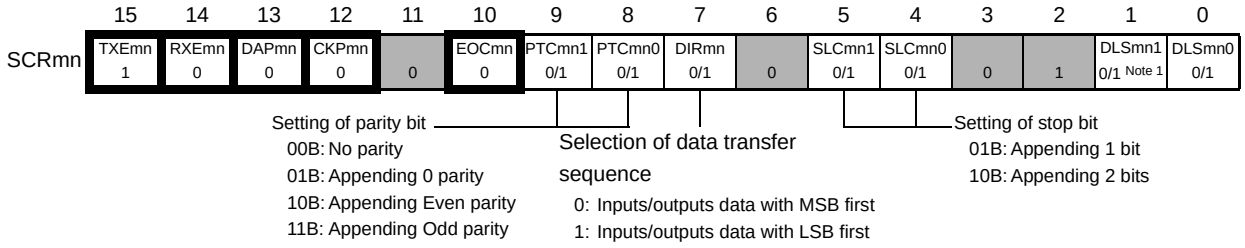
(1) Register setting

Figure 15 - 79 Example of Contents of Registers for UART Transmission of UART (UART0 to UART3) (100-pin products) (1/2)

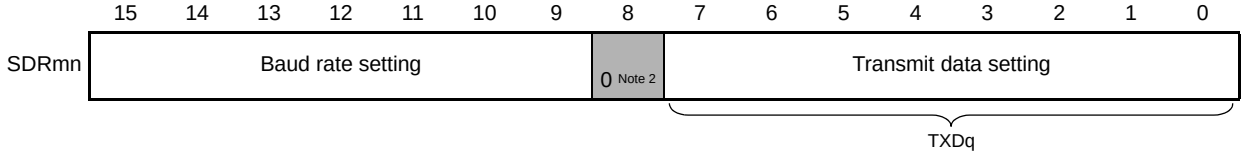
(a) Serial mode register mn (SMRmn)



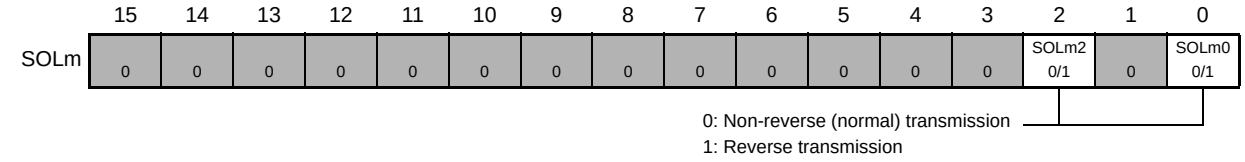
(b) Serial communication operation setting register mn (SCRmn)



(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.



Note 1. Only provided for the SCR00, SCR01, SCR10 and SCR11 registers. This bit is fixed to 1 for the other registers.

Note 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, UART2 can be specified for the 9-bit data length.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 3), mn = 00, 02, 10, 12

Remark 2. ■: Setting is fixed in the UART transmission mode,
■: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 15 - 80 Example of Contents of Registers for UART Transmission of UART (UART0 to UART3) (100-pin products) (2/2)

(e) Serial output register m (SOm)... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 Note 2 ×	1	CKOm0 Note 2 ×	0	0	0	0	1	SOm2 0/1 Note 1	1	SOm0 0/1 Note 1

0: Serial data output value is “0”
1: Serial data output value is “1”

(f) Serial output enable register m (SOEm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	0	SOEm0 0/1

(g) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

Note 1. 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.

Note 2. Serial array unit 0 only.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12

Remark 2. ■: Setting is fixed in the UART transmission mode,
▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 15 - 81 Initial Setting Procedure for UART Transmission

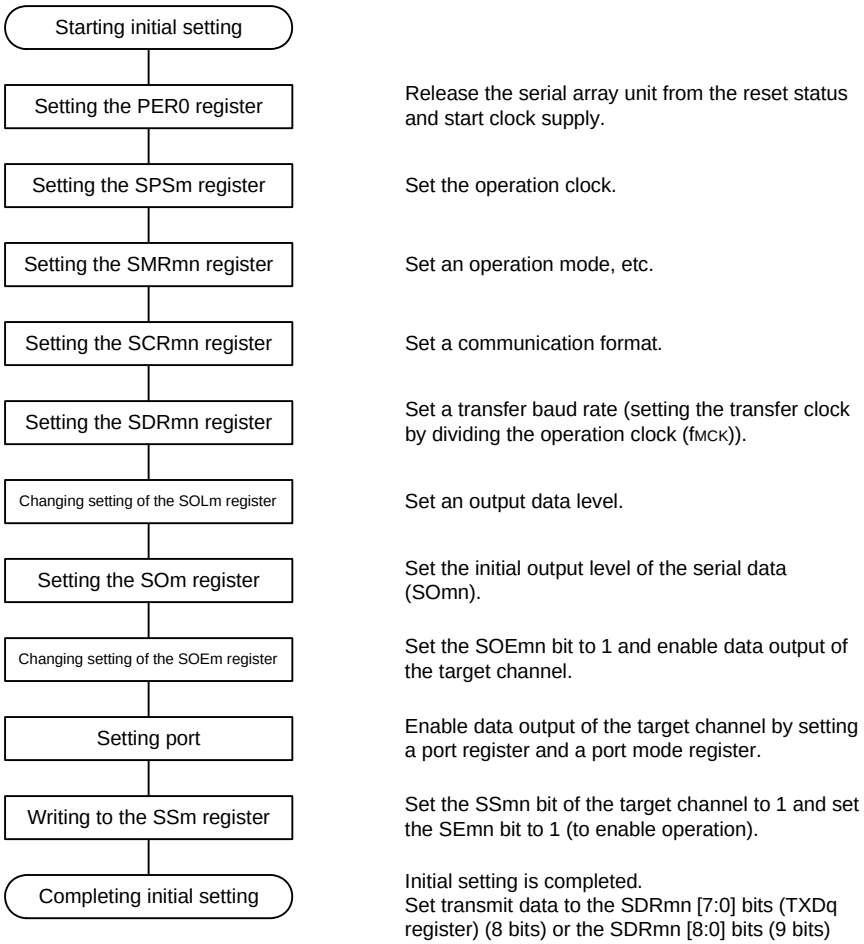


Figure 15 - 82 Procedure for Stopping UART Transmission

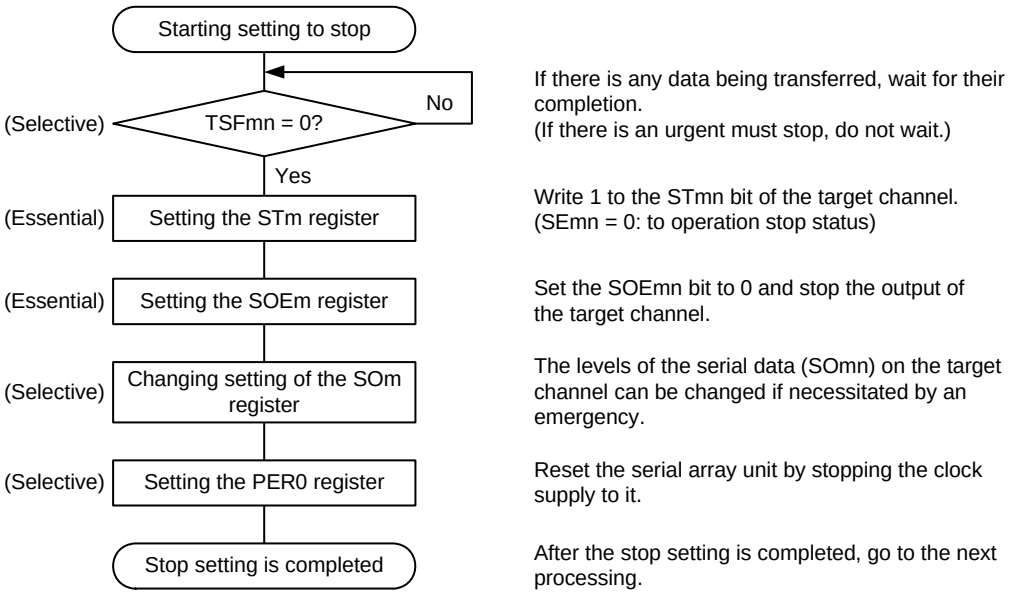
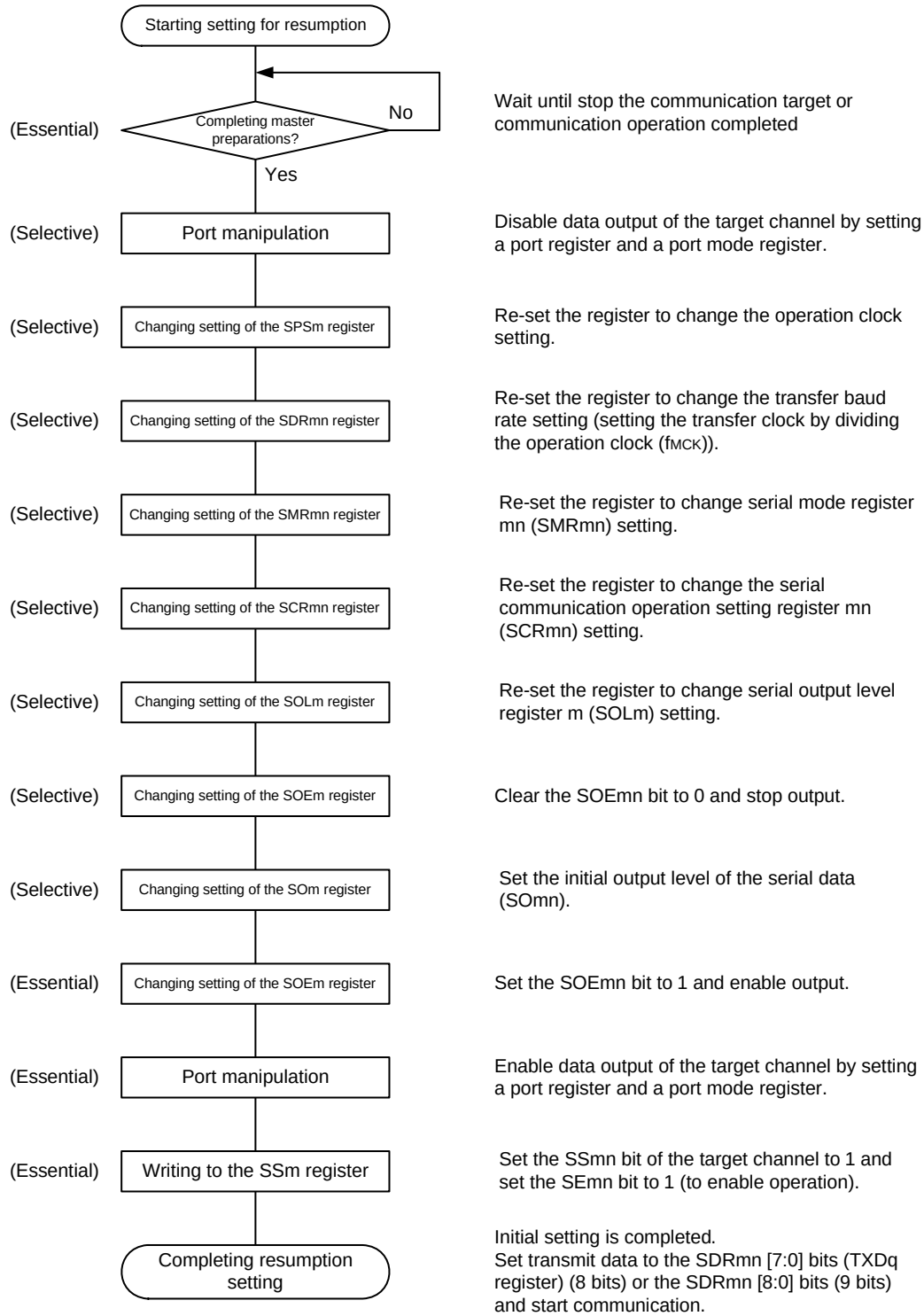


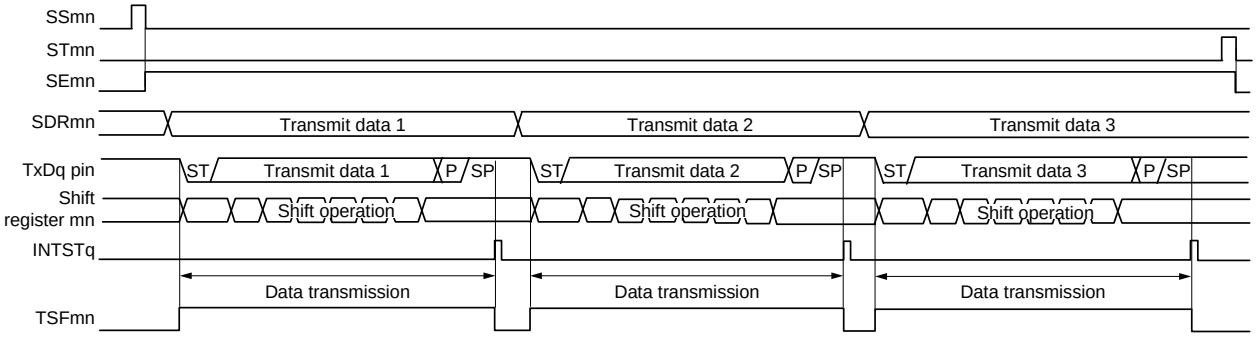
Figure 15 - 83 Procedure for Resuming UART Transmission



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target stops or transmission finishes, and then perform initialization instead of restarting the transmission.

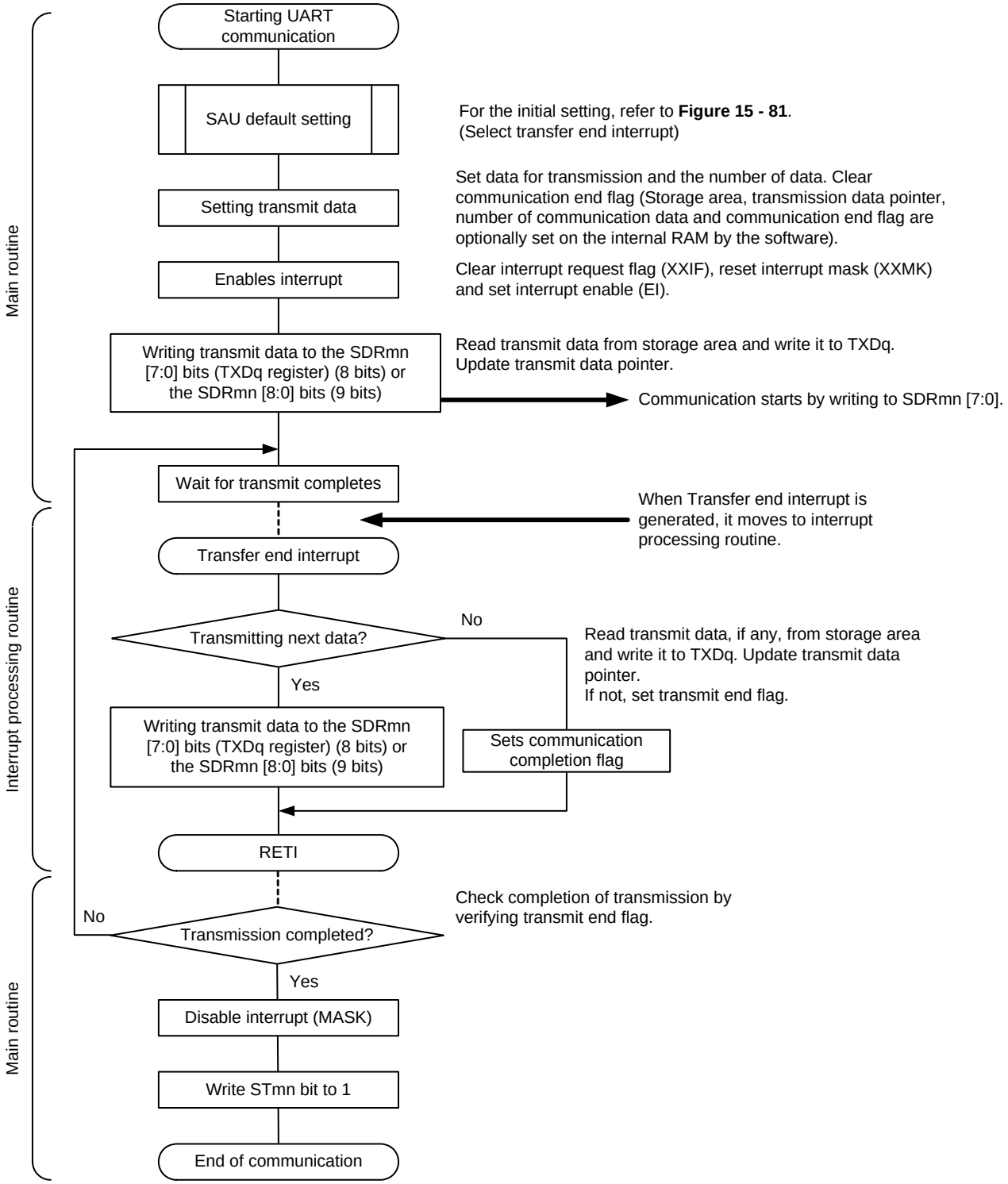
(3) Processing flow (in single-transmission mode)

Figure 15 - 84 Timing Chart of UART Transmission (in Single-Transmission Mode)



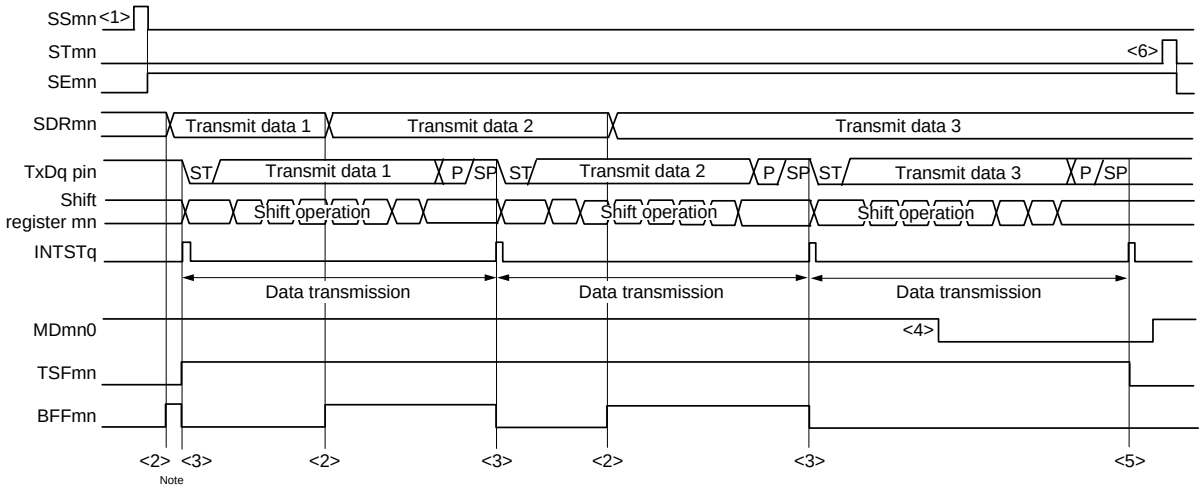
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 3)
mn = 00, 02, 10, 12

Figure 15 - 85 Flowchart of UART Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 15 - 86 Timing Chart of UART Transmission (in Continuous Transmission Mode)

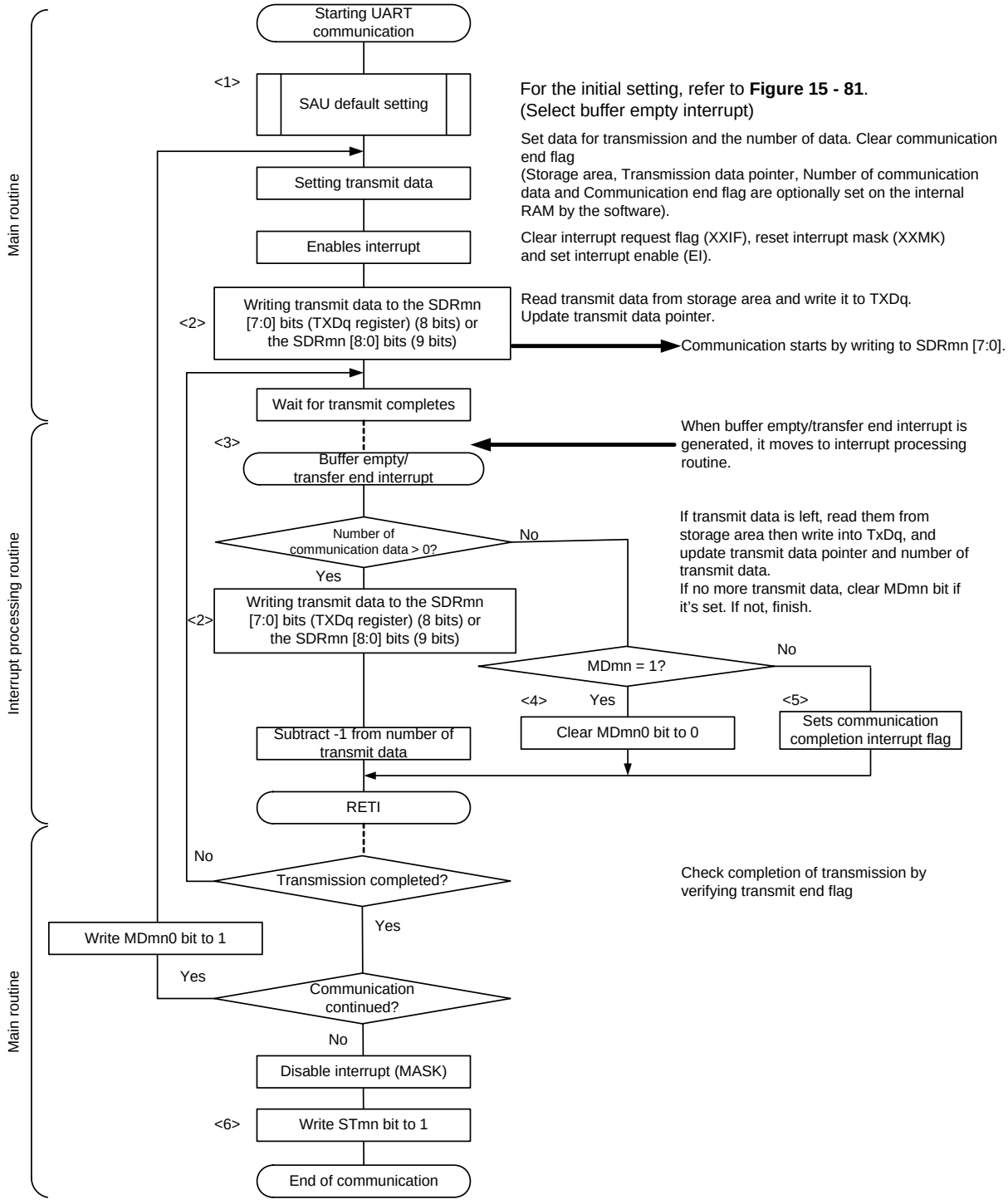


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

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, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 3)
mn = 00, 02, 10, 12

Figure 15 - 87 Flowchart of UART Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 15 - 86 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	UART2	UART3
Target channel	Channel 1 of SAU0	Channel 3 of SAU0	Channel 1 of SAU1	Channel 3 of SAU1
Pins used	RxD0	RxD1	RxD2	RxD3
Interrupt	INTST0	INTST1	INTST2	INTST3
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error interrupt	INTSRE0	INTSRE1	INTSRE2	INTSRE3
Error detection flag	• Framing error detection flag (FEFmn) • Parity error detection flag (PEFmn) • Overrun error detection flag (OVFmn)			
Transfer data length	7, 8 or 9 bits Note 1			
Transfer rate Note 2	Max. f_{MCK}/6 [bps] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]			
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)			
Parity bit	The following selectable • No parity bit (no parity check) • No parity judgment (0 parity) • Even parity check • Odd parity check			
Stop bit	Appending 1 bit			
Data direction	MSB or LSB first			

Note 1. Only UART0, UART2 can be specified for the 8-bit data length.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 34** or **CHAPTER 35 ELECTRICAL SPECIFICATIONS**).

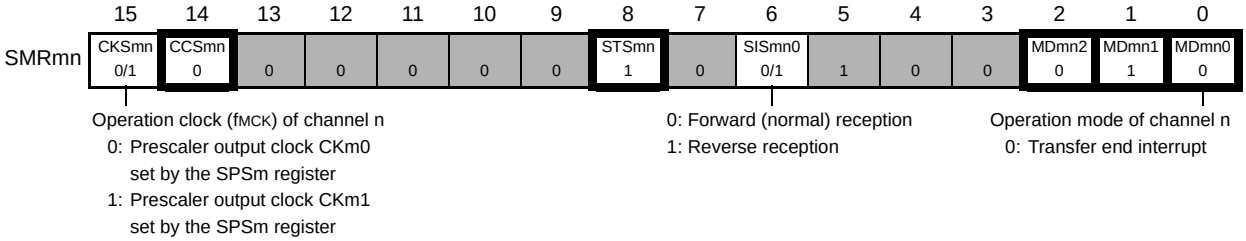
Remark 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11, 13

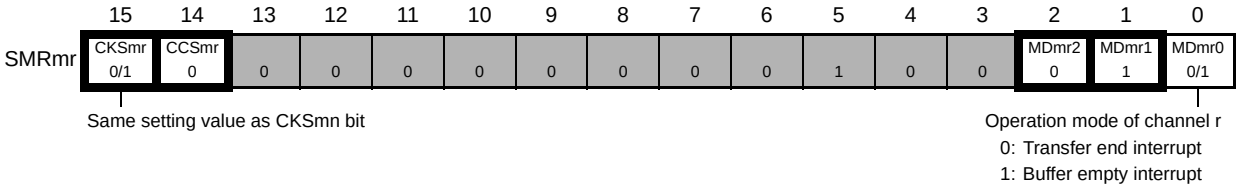
(1) Register setting

Figure 15 - 88 Example of Contents of Registers for UART Reception of UART (UART0 to UART3) (100-pin products) (1/2)

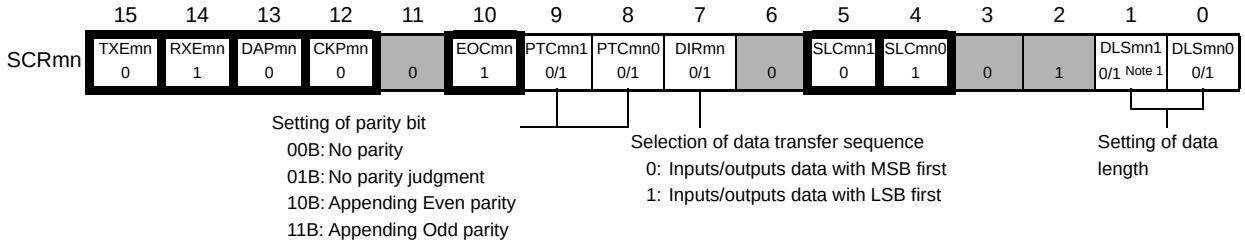
(a) Serial mode register mn (SMRmn)



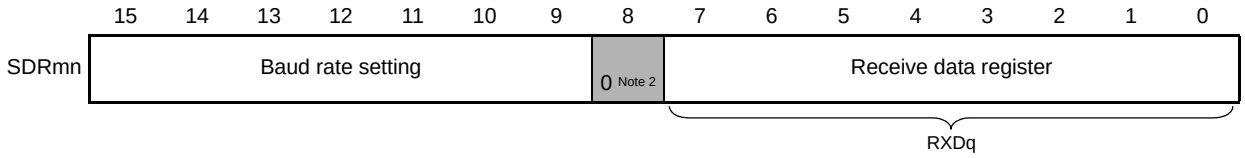
(b) Serial mode register mr (SMRmr)



(c) Serial communication operation setting register mn (SCRmn)



(d) Serial data register mn (SDRmn) (lower 8 bits: RXDq)



Note 1. Only provided for the SCR00, SCR01, SCR10 and SCR11 registers. This bit is fixed to 1 for the other registers.

Note 2. When UART performs 9-bit communication, bits 0 to 8 of the SDRm1 register are used as the transmission data specification area. Only UART0, UART2 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.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11, 13
r: Channel number (r = n - 1), q: UART number (q = 0 to 3)

Remark 2. ■: Setting is fixed in the UART reception mode,
▒: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 15 - 89 Example of Contents of Registers for UART Reception of UART (UART0 to UART3) (100-pin products) (2/2)

(e) Serial output register m (SOm)... The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 Note ×	1	CKOm0 Note ×	0	0	0	0	1	SOm2 ×	1	SOm0 ×

(f) Serial output enable register m (SOEm)... The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	0	SOEm0 ×

(g) Serial channel start register m (SSm)... Sets only the bits of the target channel is 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 ×	SSm1 0/1	SSm0 ×

Note Serial array unit 0 only.

Remark 1. m: Unit number (m = 0, 1)

Remark 2. ■: Setting is fixed in the UART reception mode,

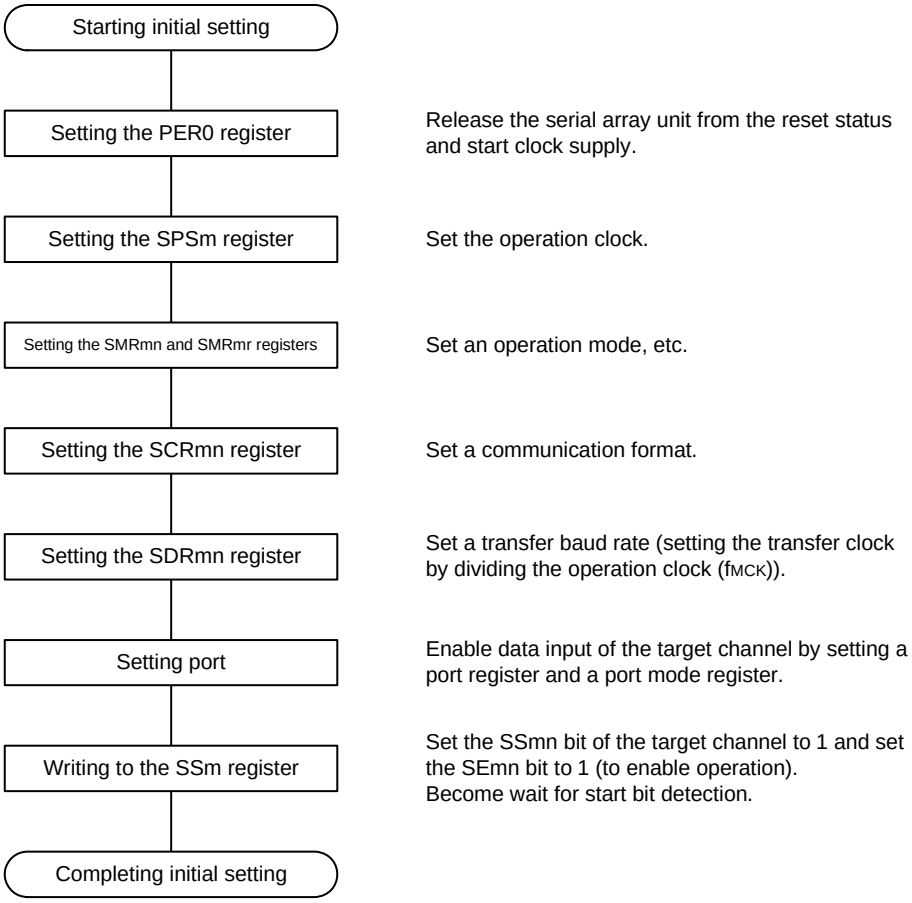
▒: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 15 - 90 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 fMCK clocks have elapsed.

Figure 15 - 91 Procedure for Stopping UART Reception

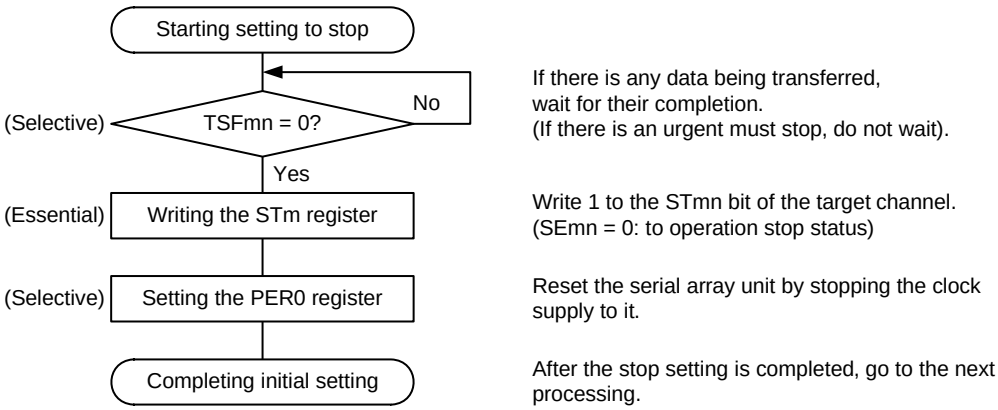
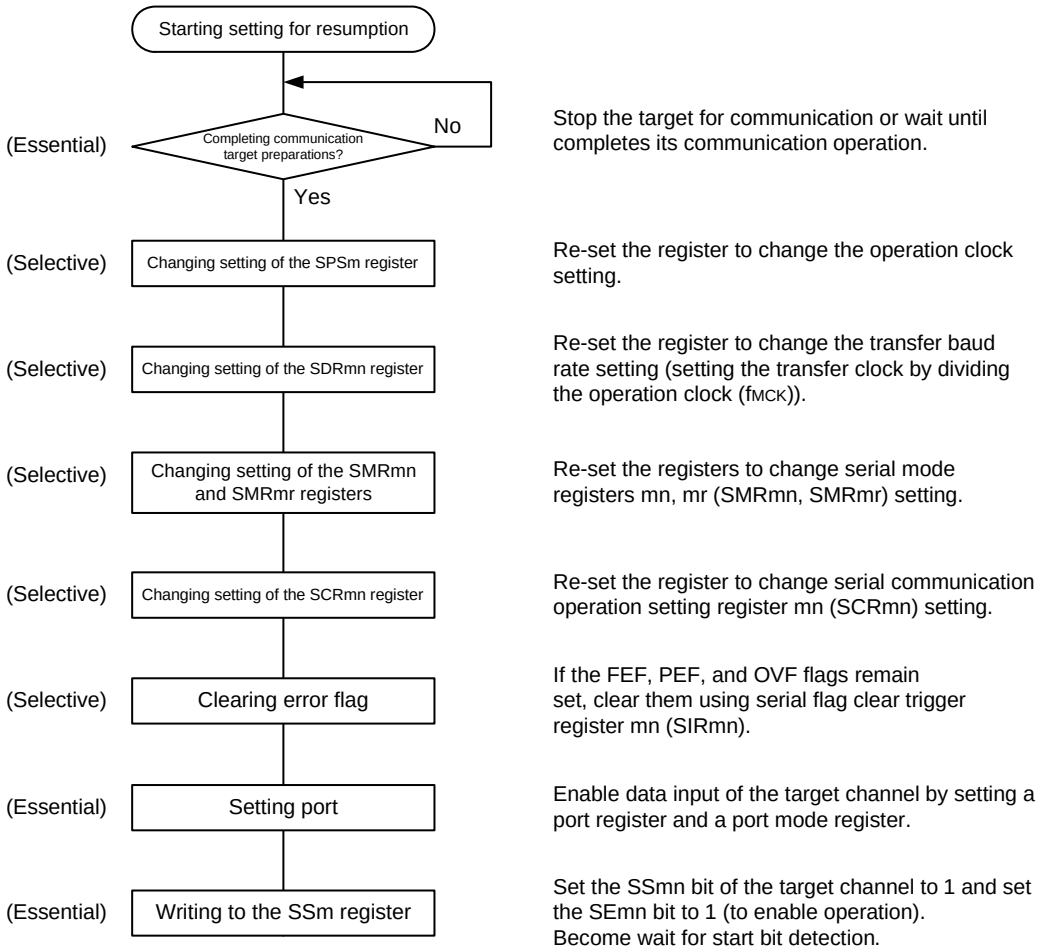


Figure 15 - 92 Procedure for Resuming UART Reception

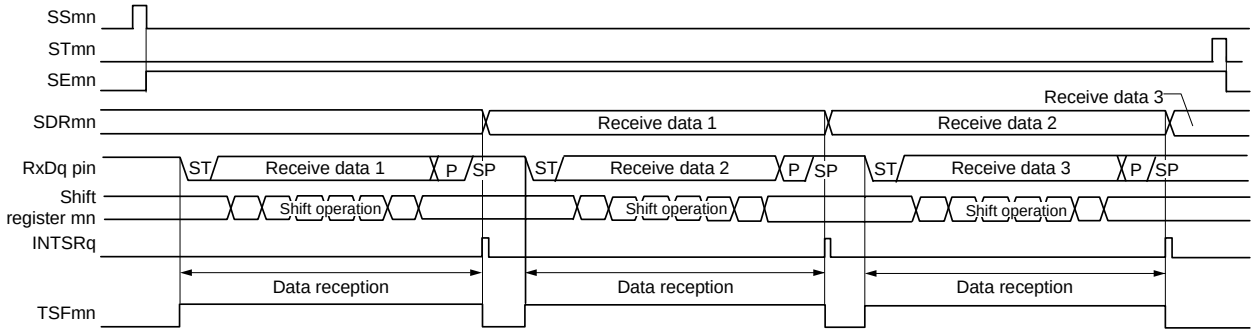


Caution **After is set RXEmn bit to 1 of SCRmn register, set the SSmn = 1 from an interval of at least four clocks of fMCK.**

Remark If PER0 is rewritten while stopping the communication target and the clock supply is stopped, wait until the communication target stops or communication finishes, and then perform initialization instead of restarting the communication.

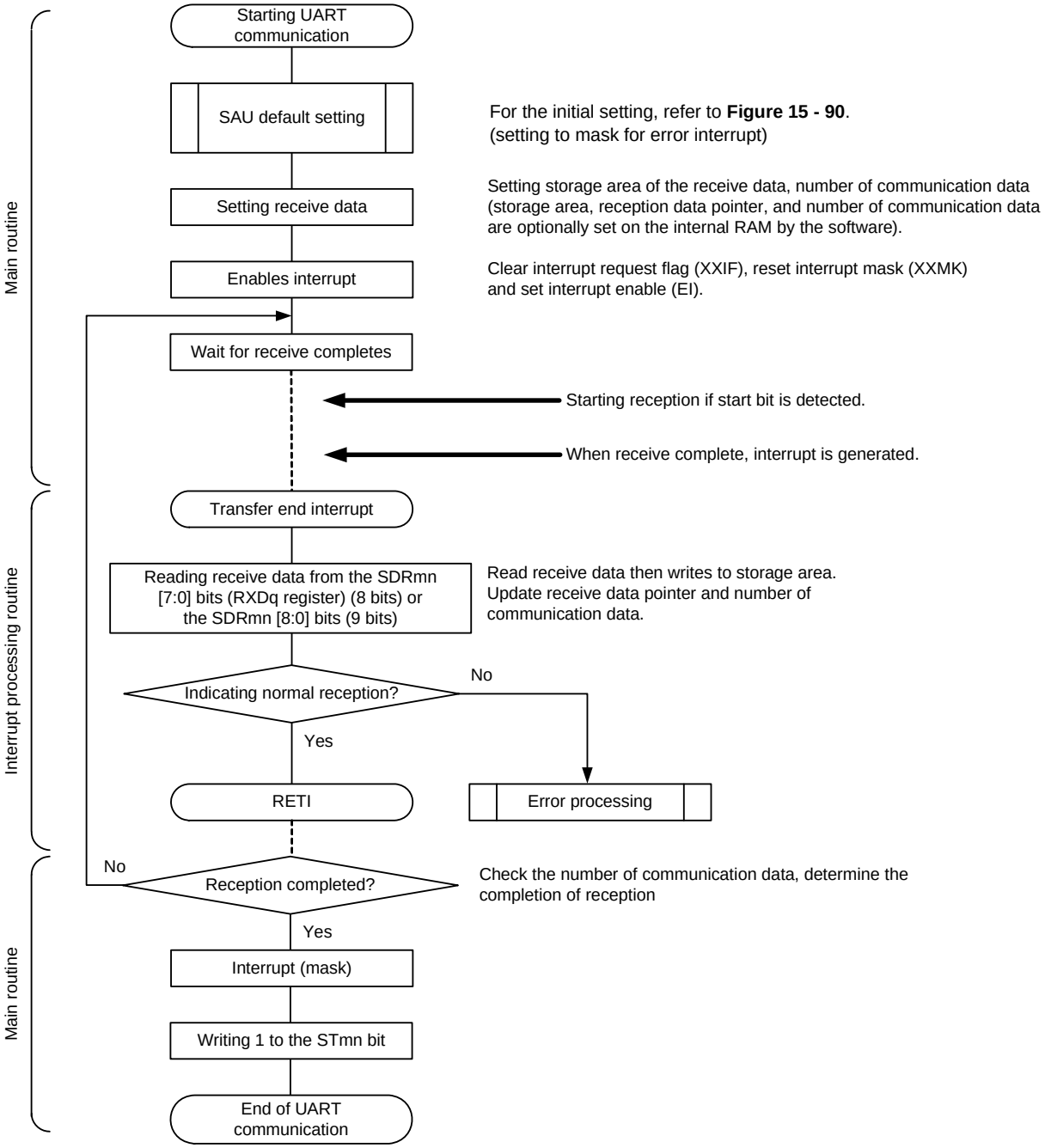
(3) Processing flow

Figure 15 - 93 Timing Chart of UART Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11, 13
r: Channel number (r = n - 1), q: UART number (q = 0 to 3)

Figure 15 - 94 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 the UART0, UART2 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 - 97 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)** and **Figure 15 - 99 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 SSm0 bit of serial channel start register m (SSm) to 1.
- A UARTq starts reception in SNOOZE mode on detecting input of the start bit on the RxDq pin following a transition of the CPU to the STOP mode.

Caution 1. The SNOOZE mode can only be used when the high-speed on-chip oscillator clock (f_{HOCO}) is selected for f_{CLK}.

Caution 2. The transfer rate in the SNOOZE mode is only 4800 bps.

Caution 3. When SWCm = 1, UARTq can be used only when the reception operation is started in the STOP mode. When used simultaneously with another SNOOZE mode function or interrupt, if the reception operation is started in a state other than the STOP mode, such as those given below, data may not be received correctly and a framing error or parity error may be generated.

- **When after the SWCm bit has been set to 1, the reception operation is started before the STOP mode is entered**
- **When the reception operation is started while another function is in the SNOOZE mode**
- **When after returning from the STOP mode to normal operation due to an interrupt or other cause, the reception operation is started before the SWCm bit is returned to 0**

Caution 4. If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFmn, FEFmn, or OVFmn flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFmn, FEFmn, or OVFmn flag before setting the SWC0 bit to 1 and read the value in bits 7 to 0 (RxDq register) of the SDRm1 register.

Caution 5. The CPU shifts from the STOP mode to the SNOOZE mode on detecting the valid edge of the RxDq signal. Note, however, that transfer through the UART channel may not start and the CPU may remain in the SNOOZE mode if an input pulse on the RxDq pin is too short to be detected as a start bit.
In such cases, data may not be received correctly, and this may lead to a framing error or parity error in the next UART transfer.

Table 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
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.

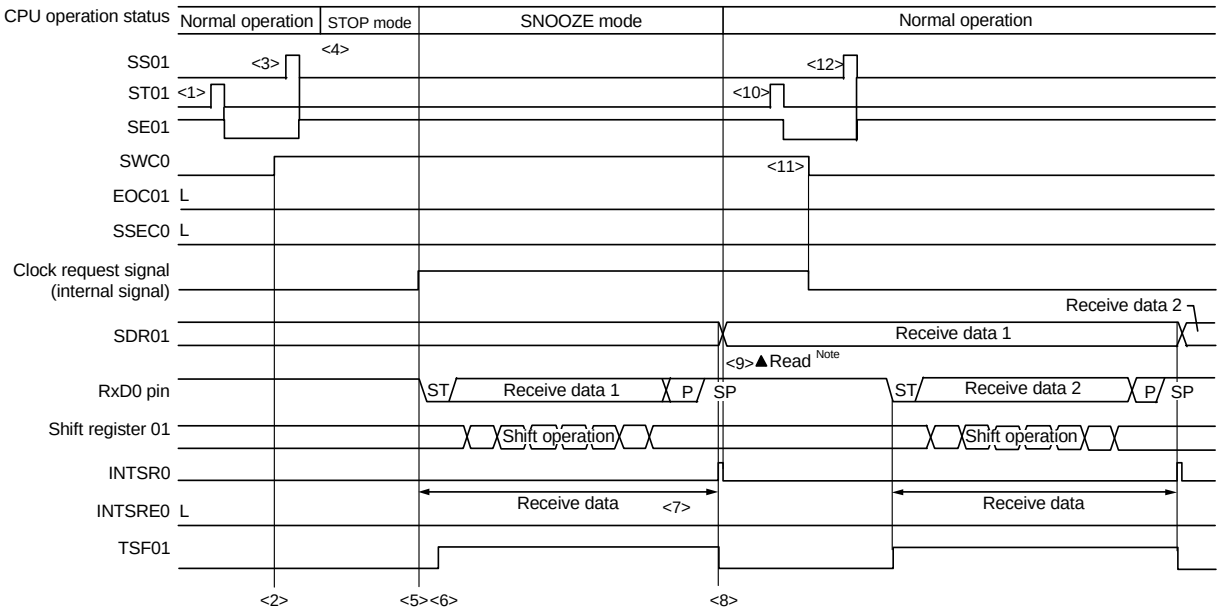
- In the case of f_{IH} ± 1.5%, perform (Maximum permissible value - 0.5%) and (Minimum permissible value + 0.5%) to the values in the above table.
- In the case of f_{IH} ± 2.0%, perform (Maximum permissible value - 1.0%) and (Minimum permissible value + 1.0%) to the values in the above table.

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 - 95 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)



Note Read the received data when SWCm is 1.

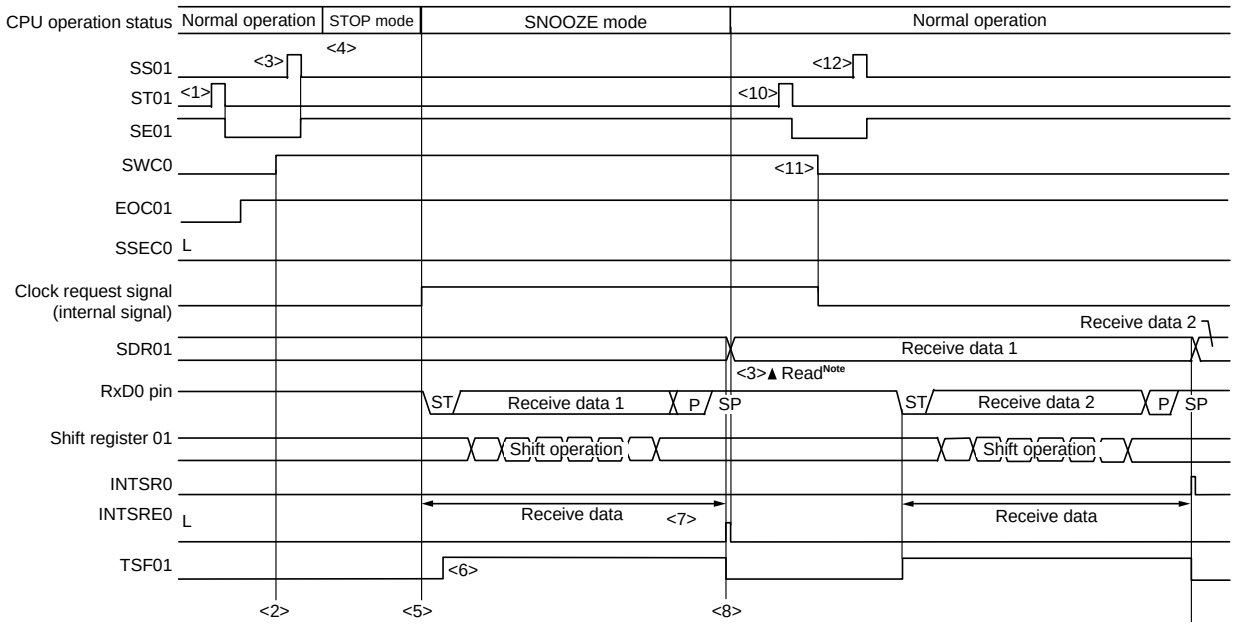
Caution **Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit, and stop the operation).**
After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).

Remark 1. <1> to <12> in the figure correspond to <1> to <12> in Figure 15 - 97 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0).

Remark 2. m = 0, 1; q = 0, 2

(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 - 96 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)



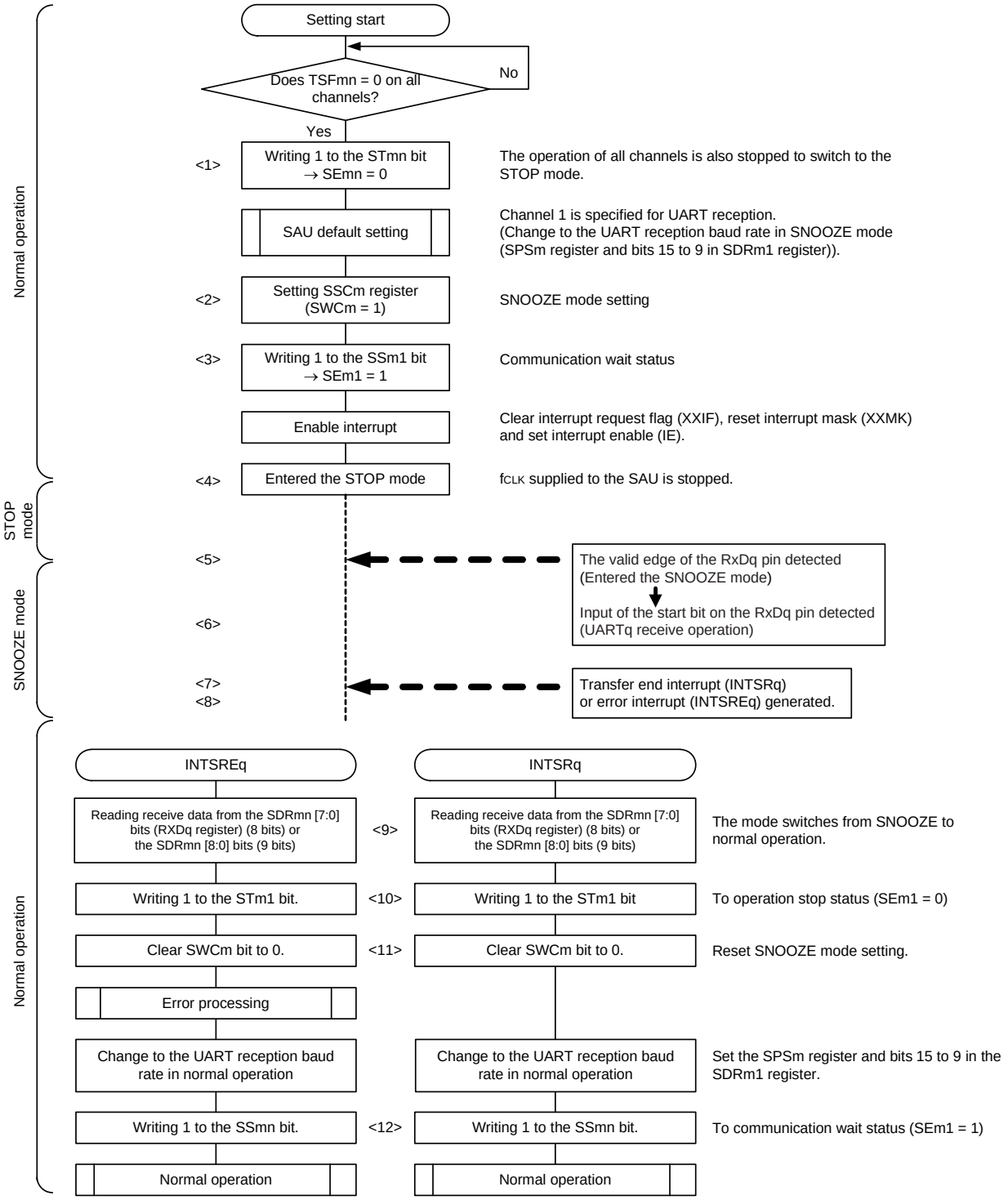
Note Read the received data when SWCm is 1.

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).**
After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).

Remark 1. <1> to <12> in the figure correspond to <1> to <12> in Figure 15 - 97 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0).

Remark 2. m = 0, 1; q = 0, 2

Figure 15 - 97 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)



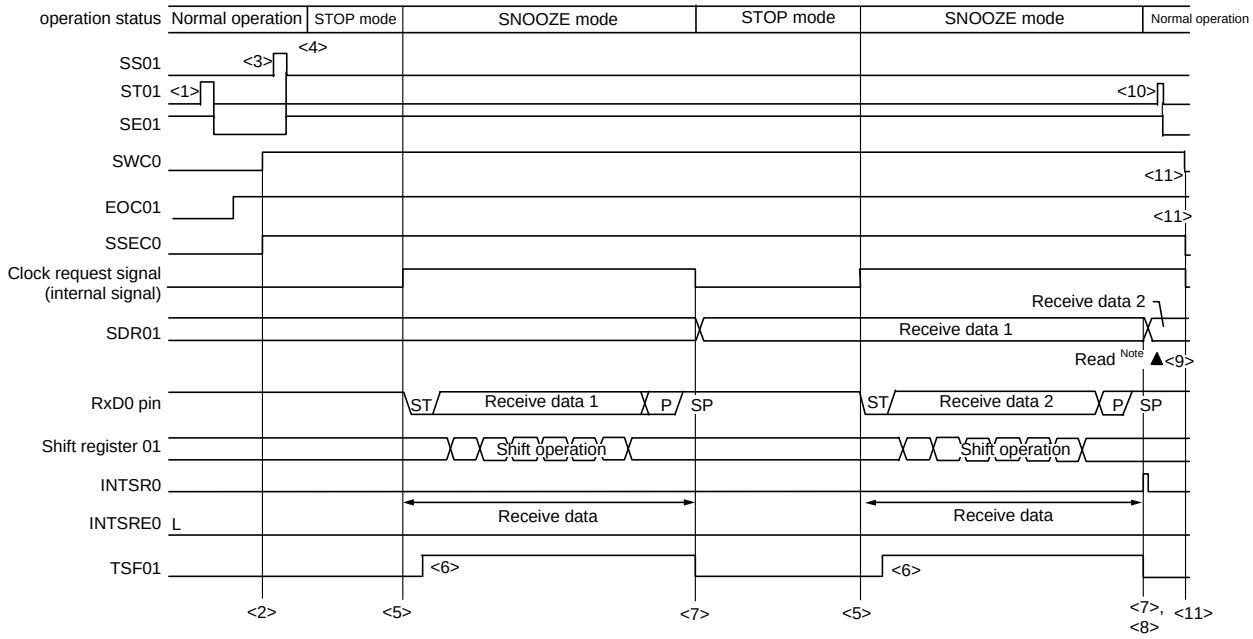
Remark 1. <1> to <12> in the figure correspond to <1> to <12> in Figure 15 - 95 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1) and Figure 15 - 96 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0).

Remark 2. m = 0, 1; q = 0, 2

(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 - 98 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



Note Read the received data when SWCm = 1.

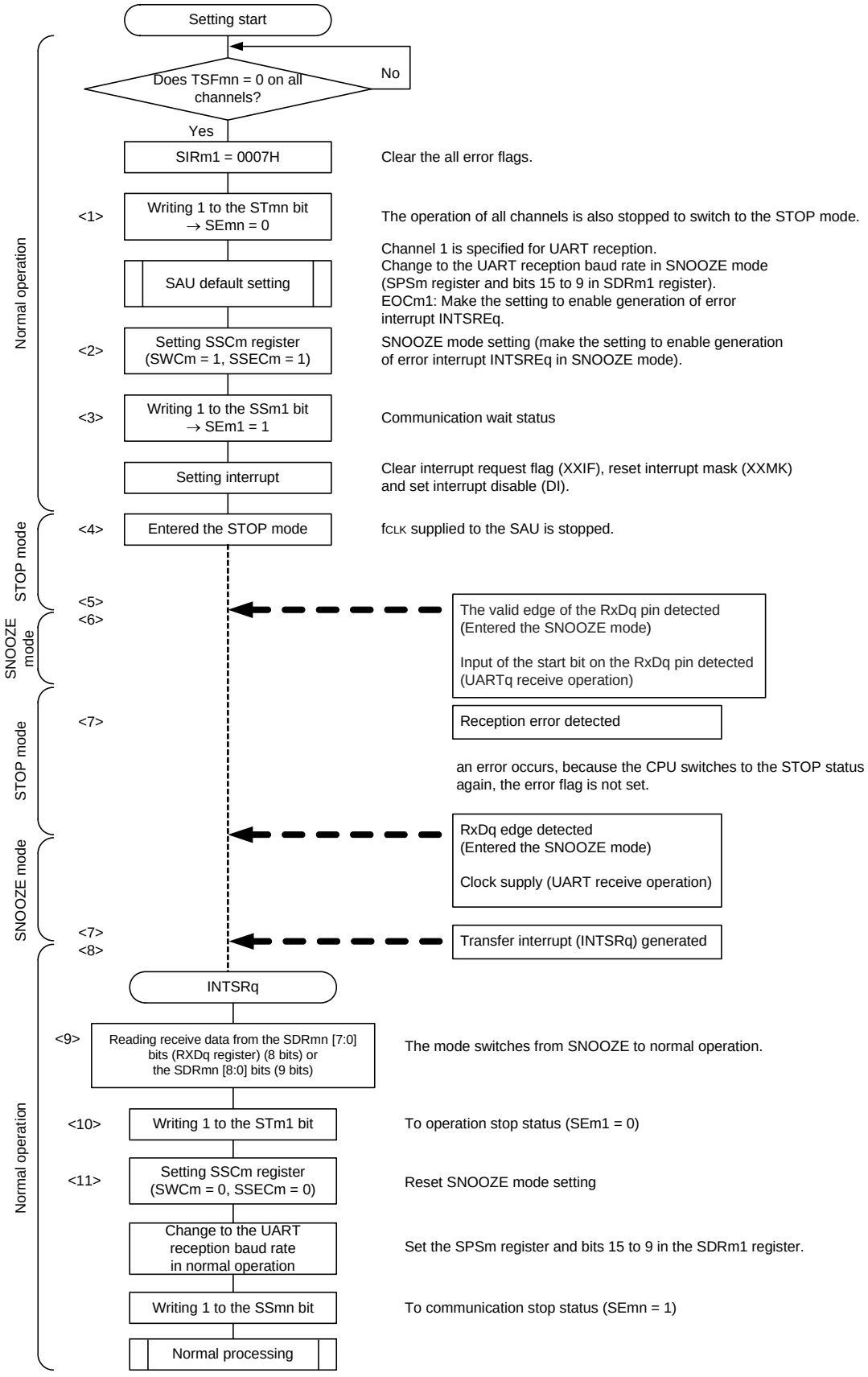
Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit and stop the operation).
After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).

Caution 2. If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFm1, FEFm1, or OVFm1 flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFm1, FEFm1, or OVFm1 flag before setting the SWCm bit to 1 and read the value in SDRm1[7:0] (RxDq register) (8 bits) or SDRm1[8:0] (9 bits).

Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 15 - 99 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1).

Remark 2. m = 0, 1; q = 0, 2

Figure 15 - 99 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



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

Caution **If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFm1, FEFm1, or OVFm1 flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFm1, FEFm1, or OVFm1 flag before setting the SWCm bit to 1 and read the value in SDRm1[7:0] (RxDq register) (8 bits) or SDRm1[8:0] (9 bits).**

Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 15 - 98 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1).

Remark 2. m = 0, 1; q = 0, 2

15.6.4 Calculating baud rate

(1) Baud rate calculation expression

The baud rate for UART (UART0 to UART3) communication can be calculated by the following expressions.

(Baud rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [bps]

Caution Setting serial data register mn (SDRmn) SDRmn[15:9] = (0000000B, 0000001B) is prohibited.

Remark 1. When UART is used, the value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000010B to 1111111B) and therefore is 2 to 127.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

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} = 24 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	24 MHz
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	12 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	6 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	3 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	1.5 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	750 kHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	375 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	187.5 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	93.8 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	46.9 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	23.4 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	11.7 kHz
	×	×	×	×	1	1	0	0	$f_{CLK}/2^{12}$	5.86 kHz
	×	×	×	×	1	1	0	1	$f_{CLK}/2^{13}$	2.93 kHz
	×	×	×	×	1	1	1	0	$f_{CLK}/2^{14}$	1.46 kHz
	×	×	×	×	1	1	1	1	$f_{CLK}/2^{15}$	732 Hz
1	0	0	0	0	×	×	×	×	f_{CLK}	24 MHz
	0	0	0	1	×	×	×	×	$f_{CLK}/2$	12 MHz
	0	0	1	0	×	×	×	×	$f_{CLK}/2^2$	6 MHz
	0	0	1	1	×	×	×	×	$f_{CLK}/2^3$	3 MHz
	0	1	0	0	×	×	×	×	$f_{CLK}/2^4$	1.5 MHz
	0	1	0	1	×	×	×	×	$f_{CLK}/2^5$	750 kHz
	0	1	1	0	×	×	×	×	$f_{CLK}/2^6$	375 kHz
	0	1	1	1	×	×	×	×	$f_{CLK}/2^7$	187.5 kHz
	1	0	0	0	×	×	×	×	$f_{CLK}/2^8$	93.8 kHz
	1	0	0	1	×	×	×	×	$f_{CLK}/2^9$	46.9 kHz
	1	0	1	0	×	×	×	×	$f_{CLK}/2^{10}$	23.4 kHz
	1	0	1	1	×	×	×	×	$f_{CLK}/2^{11}$	11.7 kHz
	1	1	0	0	×	×	×	×	$f_{CLK}/2^{12}$	5.86 kHz
	1	1	0	1	×	×	×	×	$f_{CLK}/2^{13}$	2.93 kHz
	1	1	1	0	×	×	×	×	$f_{CLK}/2^{14}$	1.46 kHz
	1	1	1	1	×	×	×	×	$f_{CLK}/2^{15}$	732 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remark 1. ×: Don't care

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

(2) Baud rate error during transmission

The baud rate error of UART (UART0 to UART3) communication during transmission can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

(Baud rate error) = (Calculated baud rate value) ÷ (Target baud rate) ×100 - 100 [%]

Here is an example of setting a UART baud rate at f_{CLK} = 24 MHz.

UART Baud Rate (Target Baud Rate)	f_{CLK} = 24 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{CLK}/2^9$	77	300.48 bps	+0.16%
600 bps	$f_{CLK}/2^8$	77	600.96 bps	+0.16%
1200 bps	$f_{CLK}/2^7$	77	1201.92 bps	+0.16%
2400 bps	$f_{CLK}/2^6$	77	2403.85 bps	+0.16%
4800 bps	$f_{CLK}/2^5$	77	4807.69 bps	+0.16%
9600 bps	$f_{CLK}/2^4$	77	9615.38 bps	+0.16%
19200 bps	$f_{CLK}/2^3$	77	19230.8 bps	+0.16%
31250 bps	$f_{CLK}/2^3$	47	31250.0 bps	±0.0%
38400 bps	$f_{CLK}/2^2$	77	38461.5 bps	+0.16%
76800 bps	$f_{CLK}/2$	77	76923.1 bps	+0.16%
153600 bps	f_{CLK}	77	153846 bps	+0.16%
312500 bps	f_{CLK}	37	315789 bps	±1.05%

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0 to UART3) 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.

$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

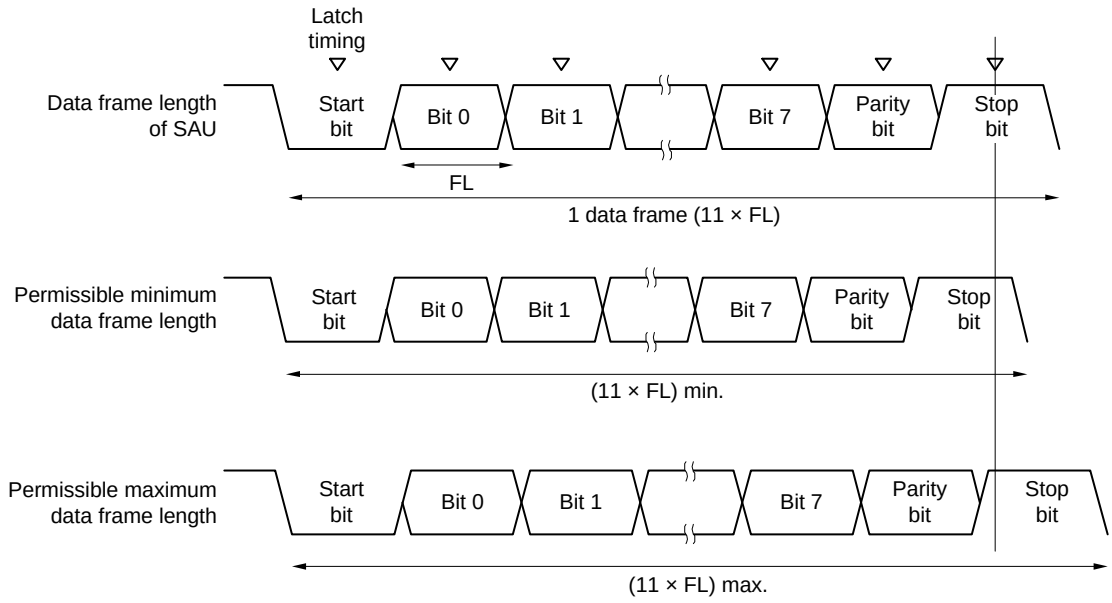
Brate: Calculated baud rate value at the reception side (See **15.6.4 (1) Baud rate calculation expression**.)

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, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11, 13

Figure 15 - 100 Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in Figure 15 - 100, 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 to UART3) communication

The procedure for processing errors that occurred during UART (UART0 to UART3) communication is described in Figures 15 - 101 and 15 - 102.

Figure 15 - 101 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 - 102 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, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

15.7 LIN Communication Operation

15.7.1 LIN transmission

Of UART transmission, UART0 support LIN communication.

For LIN transmission, channel 0 of unit 0 is used.

UART	UART0	UART1	UART2	UART3
Support of LIN communication	Supported	Not supported	Not supported	Not supported
Target channel	Channel 0 of SAU0	—	—	—
Pins used	TxD0	—	—	—
Interrupt	INTST0	—	—	—
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	None			
Transfer data length	8 bits			
Transfer rate	Max. f_{MCK}/6 [bps] (SDR00 [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps] Note			
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 34** or **CHAPTER 35 ELECTRICAL SPECIFICATIONS**). In general, 2.4/9.6/19.2 kbps is often used in LIN communication.

Remark f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

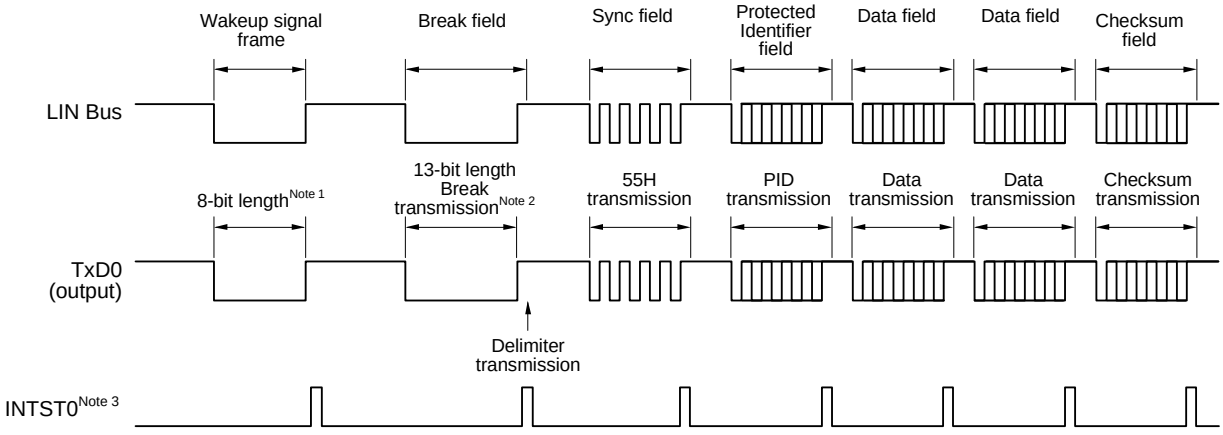
LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol designed to reduce the cost of an automobile network. Communication of LIN is single-master communication and up to 15 slaves can be connected to one master. The slaves are used to control switches, actuators, and sensors, which are connected to the master via LIN.

Usually, the master is connected to a network such as CAN (Controller Area Network). A LIN bus is a single-wire bus to which nodes are connected via transceiver conforming to ISO9141.

According to the protocol of LIN, the master transmits a frame by attaching baud rate information to it. A slave receives this frame and corrects a baud rate error from the master. If the baud rate error of a slave is within ±15%, communication can be established.

Figure 15 - 103 outlines a transmission operation of LIN.

Figure 15 - 103 Transmission Operation of LIN



Note 1. Set the baud rate in accordance with the wakeup signal regulations and transmit data of 80H.

Note 2. A break field is defined to have a width of 13 bits and output a low level. Where the baud rate for main transfer is N [bps], therefore, the baud rate of the break field is calculated as follows.

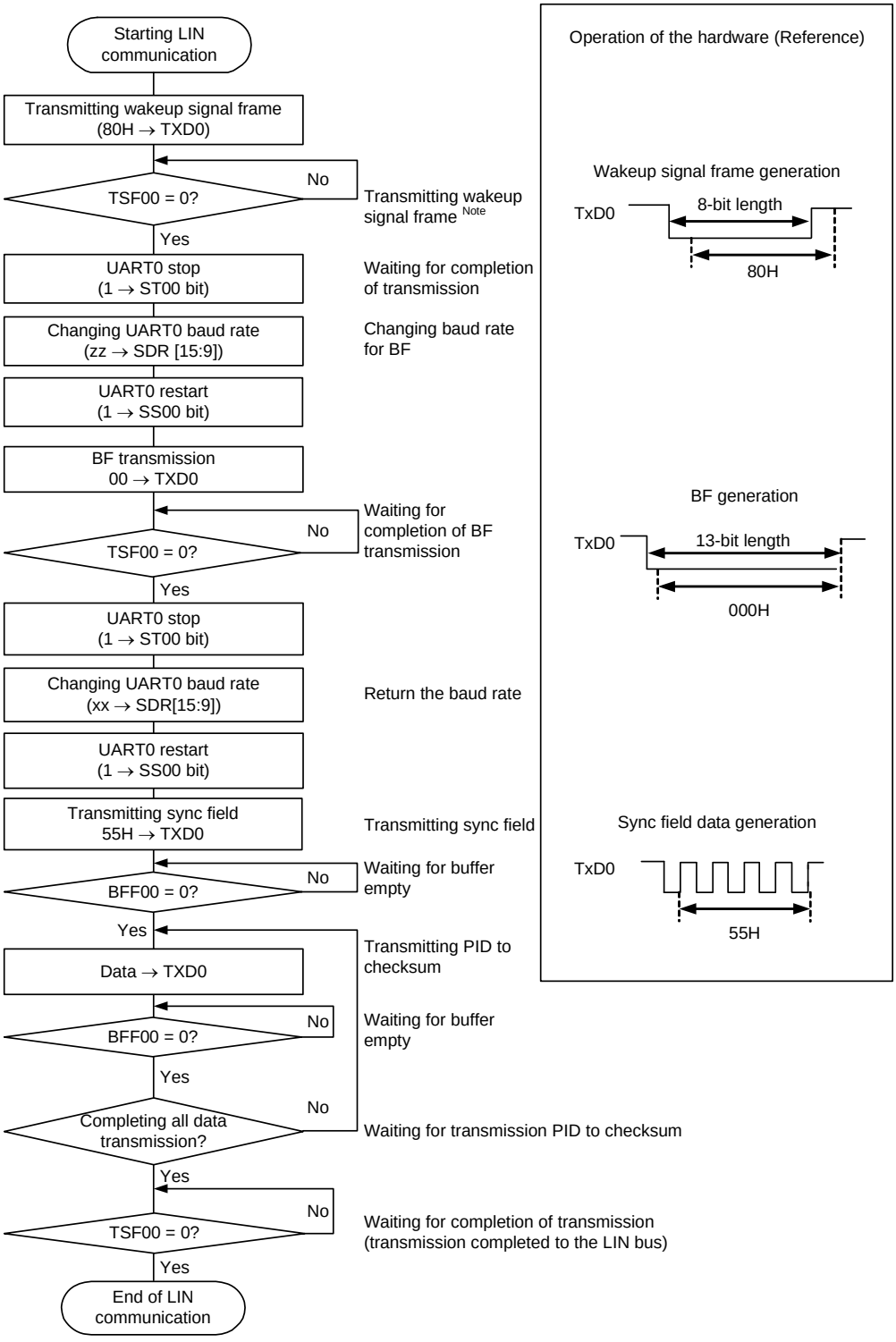
(Baud rate of sync break field) = 9/13 × N

By transmitting data of 00H at this baud rate, a break field is generated.

Note 3. INTST0 is output upon completion of transmission. INTST0 is also output at BF transmission.

Remark The interval between fields is controlled by software.

Figure 15 - 104 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 support LIN communication.

For LIN reception, channel 1 of unit 1 is used.

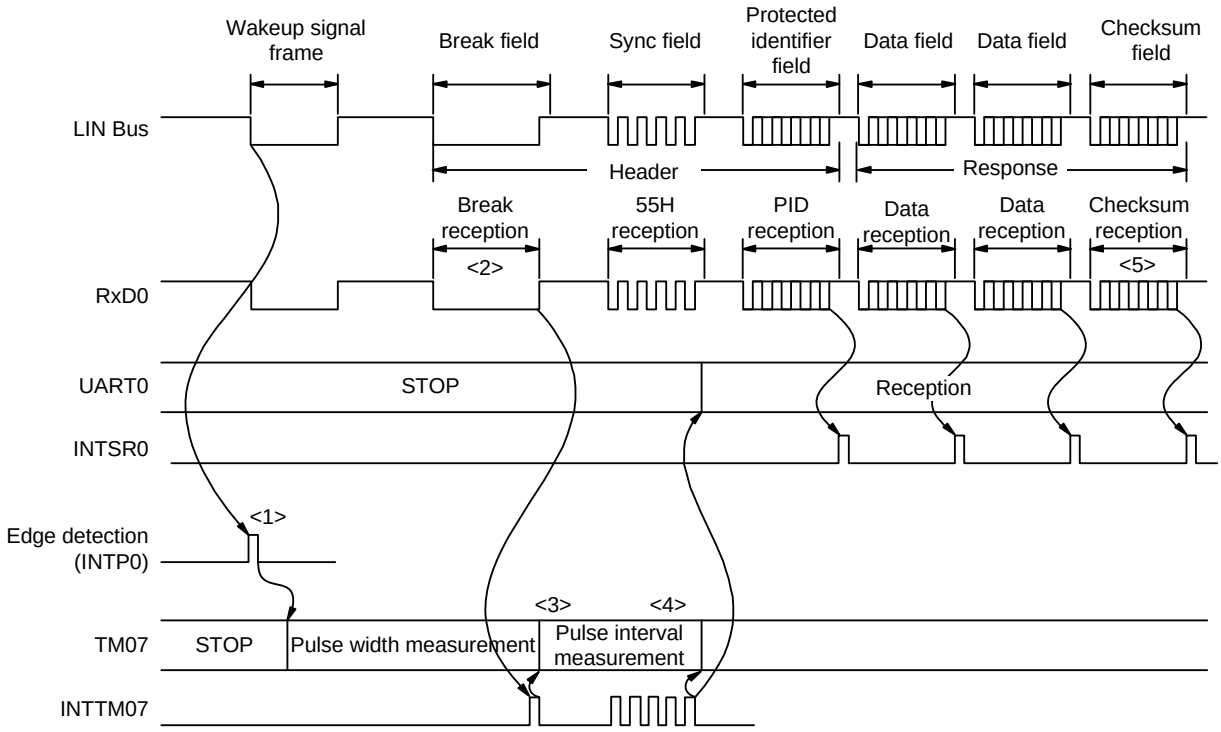
UART	UART0	UART1	UART2	UART3
Support of LIN communication	Supported	Not supported	Not supported	Not supported
Target channel	Channel 1 of SAU0	—	—	—
Pins used	RxD0	—	—	—
Interrupt	INTSR0	—	—	—
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error interrupt	INTSRE0	—	—	—
Error detection flag	• Framing error detection flag (FEF01) • Overrun error detection flag (OVF01)			
Transfer data length	8 bits			
Transfer rate Note	Max. f_{MCK}/6 [bps] (SDR01 [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]			
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)			
Parity bit	No parity bit (The parity bit is not checked.)			
Stop bit	Check the first bit			
Data direction	LSB first			

Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 34** or **CHAPTER 35 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

Figure 15 - 105 outlines a reception operation of LIN.

Figure 15 - 105 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.

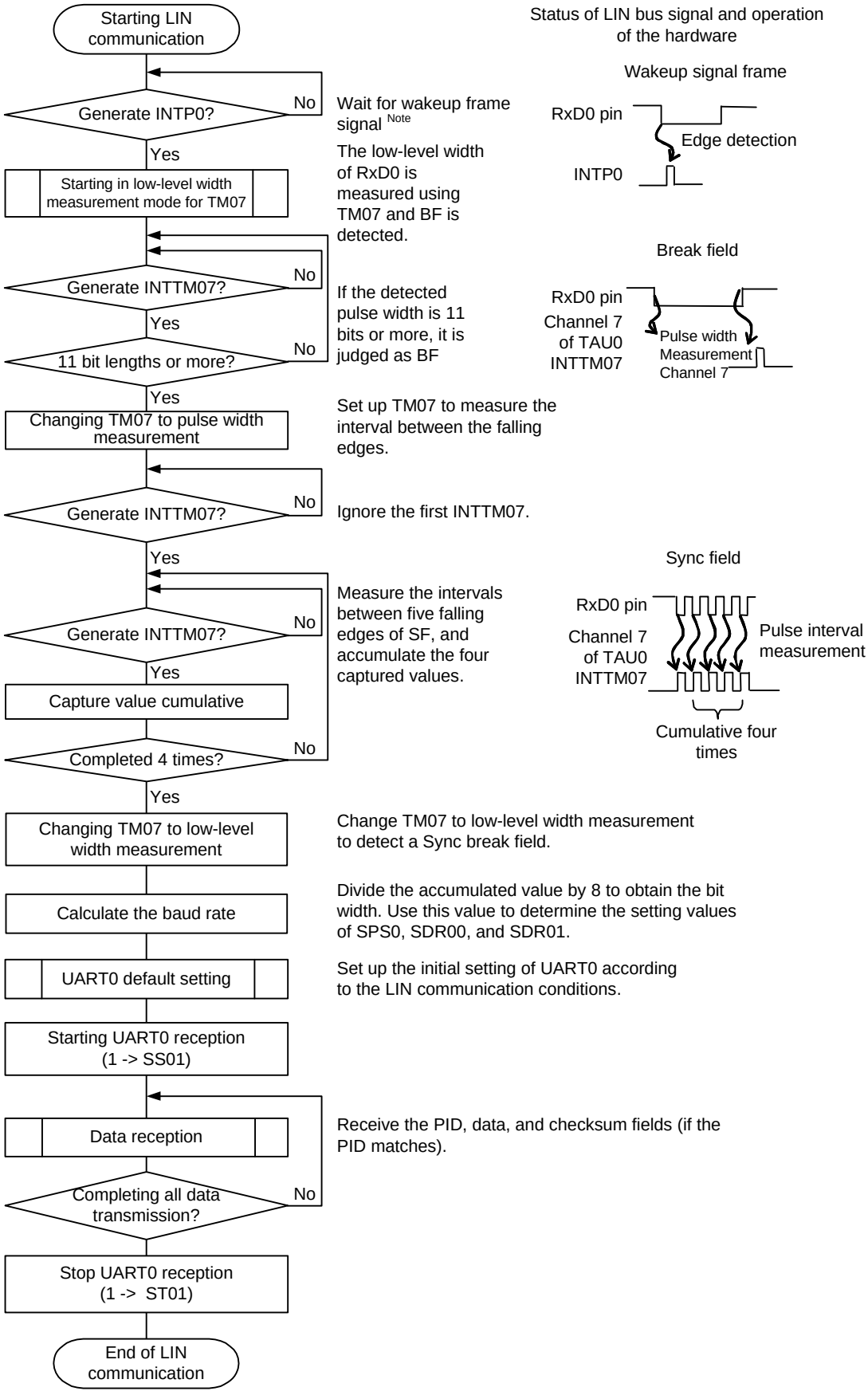
<2> TM07 starts measuring the low-level width upon detection of the falling edge of the BF signal, and then captures the data upon detection of the rising edge of the BF signal. The captured data is used to judge whether it is the BF signal.

<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 (see **6.8.3 Operation as input pulse interval measurement**).

<4> Calculate a baud rate error from the bit interval of sync field (SF). Stop UART0 once and adjust (re-set) the baud rate.

<5> The checksum field should be distinguished by software. In addition, processing to initialize UART0 after the checksum field is received and to wait for reception of BF should also be performed by software.

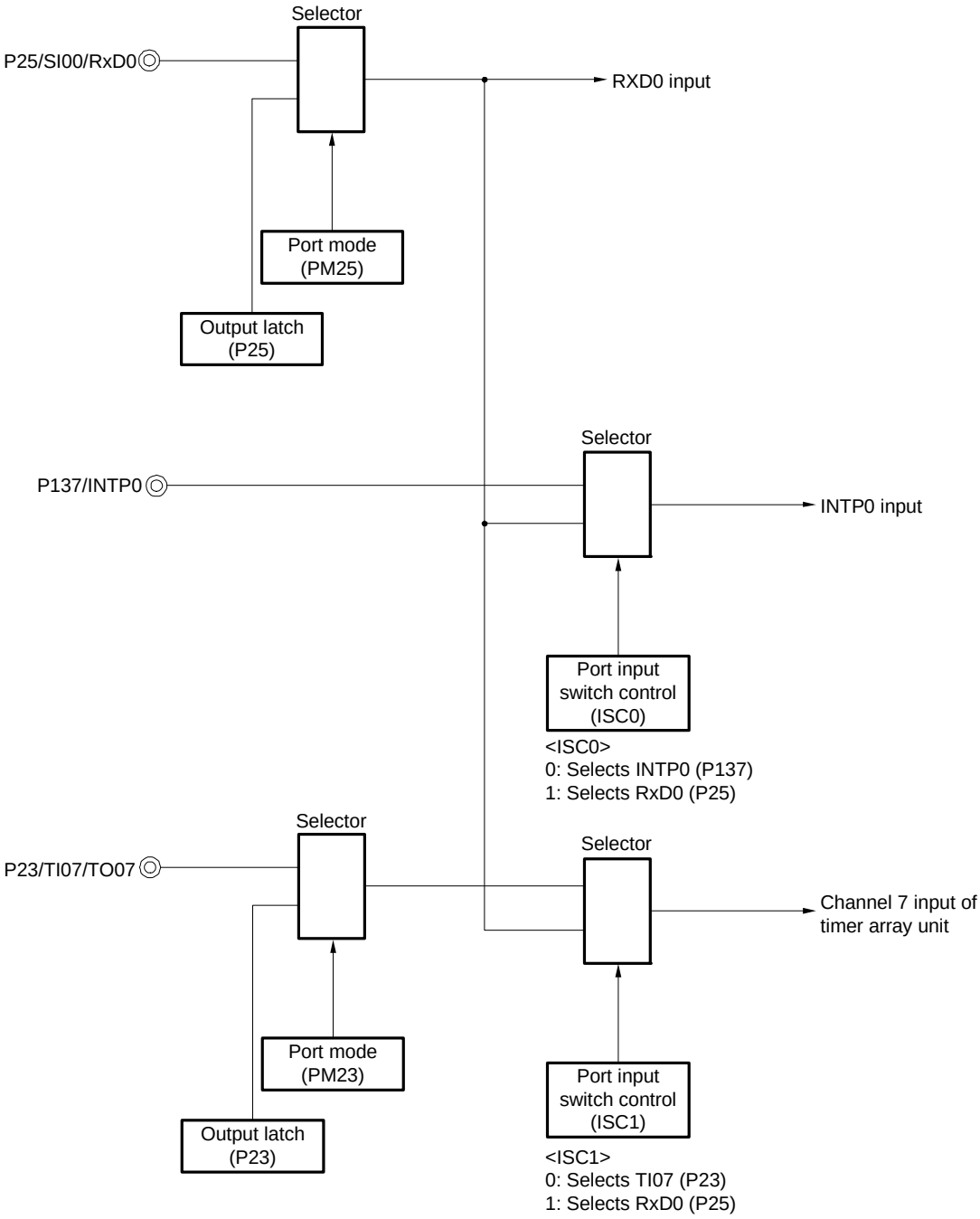
Figure 15 - 106 Flowchart of LIN Reception



Note Required in the sleep status only.

Figure 15 - 107 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 - 107 Port Configuration for Manipulating Reception of LIN



Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (See **Figure 15 - 24**.)

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 0 (SAU0)

15.8 Operation of Simplified I²C (IIC00, IIC10, IIC20, IIC30) Communication

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This communication function is designed to execute single communication with devices such as EEPROM, flash memory, and A/D converter, and therefore, can be used only by the master.

Operate the control registers by software for setting the start and stop conditions while observing the specifications of the I²C bus line

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function Note and ACK detection function
- Data length of 8 bits
  (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Generation of start condition and stop condition for software

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- Overrun error
- ACK error

* [Functions not supported by simplified I²C]

- Slave transmission, slave reception
- Multi-master function (arbitration loss detection function)
- Clock stretch detection function

Note When receiving the last data, ACK will not be output if 0 is written to the SOEmn (SOEm register) bit and serial communication data output is stopped. See the processing flow in **15.8.3 (2)** for details.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12

The channel supporting simplified I2C (IIC00, IIC10, IIC20, IIC30) is channels 0 and 2 of SAU0, and channels 0 and 2 of SAU1

• 80-pin, 85-pin, and 100-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I2C
0	0	CSI00	UART0 (supporting LIN-bus)	IIC00
	1	—		—
	2	CSI10	UART1	IIC10
	3	—		—
1	0	CSI20	UART2	IIC20
	1	—		—
	2	CSI30	UART3	IIC30
	3	—		—

Simplified I2C (IIC00, IIC10, IIC20, IIC30) performs the following four types of communication operations.

- Address field transmission (See **15.8.1**.)
- Data transmission (See **15.8.2**.)
- Data reception (See **15.8.3**.)
- Stop condition generation (See **15.8.4**.)

15.8.1 Address field transmission

Address field transmission is a transmission operation that first executes in I^2C communication to identify the target for transfer (slave). After a start condition is generated, an address (7 bits) and a transfer direction (1 bit) are transmitted in one frame.

Simplified I^2C	IIC00	IIC10	IIC20	IIC30
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1	Channel 2 of SAU1
Pins used	SCL00, SDA00 Note 1	SCL10, SDA10 Note 1	SCL20, SDA20 Note 1	SCL30, SDA30 Note 1
Interrupt	INTIIC00	INTIIC10	INTIIC20	INTIIC30
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error detection flag	ACK error detection flag (PEFmn)			
Transfer data length	8 bits (transmitted with specifying the higher 7 bits as address and the least significant bit as R/W control)			
Transfer rate Note 2	Max. f_{MCK}/4 [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^2C. • Max. 1 MHz (fast mode plus) • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)			
Data level	Non-reversed output (default: high level)			
Parity bit	No parity bit			
Stop bit	Appending 1 bit (for ACK reception timing)			
Data direction	MSB first			

Note 1. To perform communication via simplified I^2C, set the N-ch open-drain output (V_{DD} tolerance mode (POMxx = 1) with the port output mode register (POMxx). For details, see **4.3 Registers Controlling Port Function** and **4.5 Register Settings When Using Alternate Function**.
When IIC00, IIC10, IIC20, IIC30 is communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance mode (POMxx = 1) also for the clock input/output pins (SCL00, SCL10, SCL20, SCL30).
For details, see **4.4.4 Handling different potential (1.8 V, 2.5 V) by using I/O buffers**.

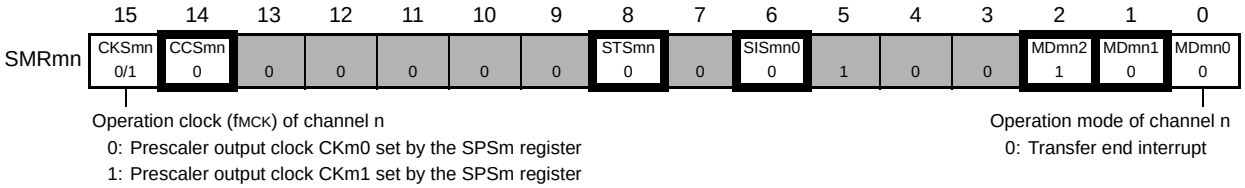
Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 34** or **CHAPTER 35 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12

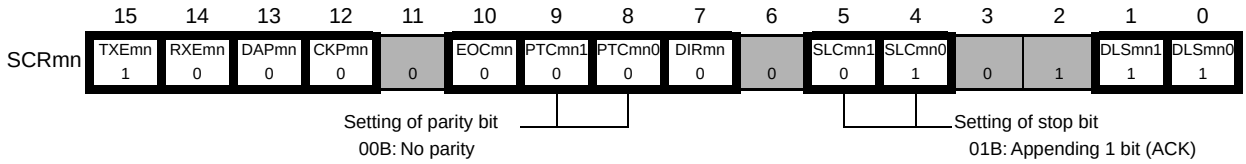
(1) Register setting

Figure 15 - 108 Example of Contents of Registers for Address Field Transmission of Simplified I²C (IIC00, IIC10, IIC20, IIC30)

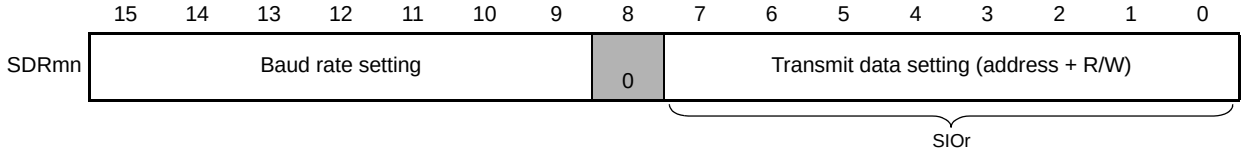
(a) Serial mode register mn (SMRmn)



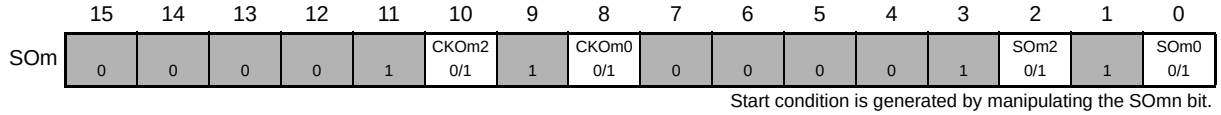
(b) Serial communication operation setting register mn (SCRmn)



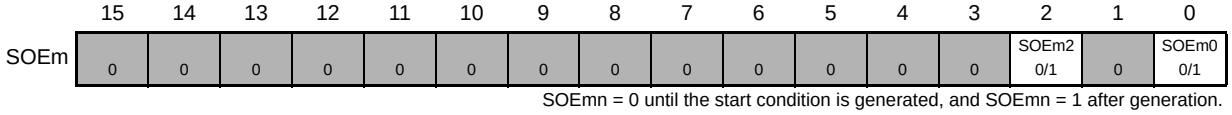
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)



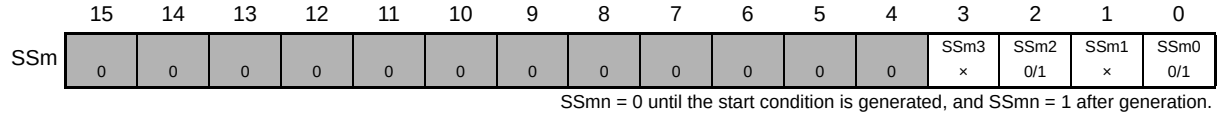
(d) Serial output register m (SOm)



(e) Serial output enable register m (SOEm)



(f) Serial channel start register m (SSm)... Sets only the bits of the target channel is 1.



Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 00, 10, 20, 30), mn = 00, 02, 10, 12

Remark 2. ■: Setting is fixed in the IIC mode,

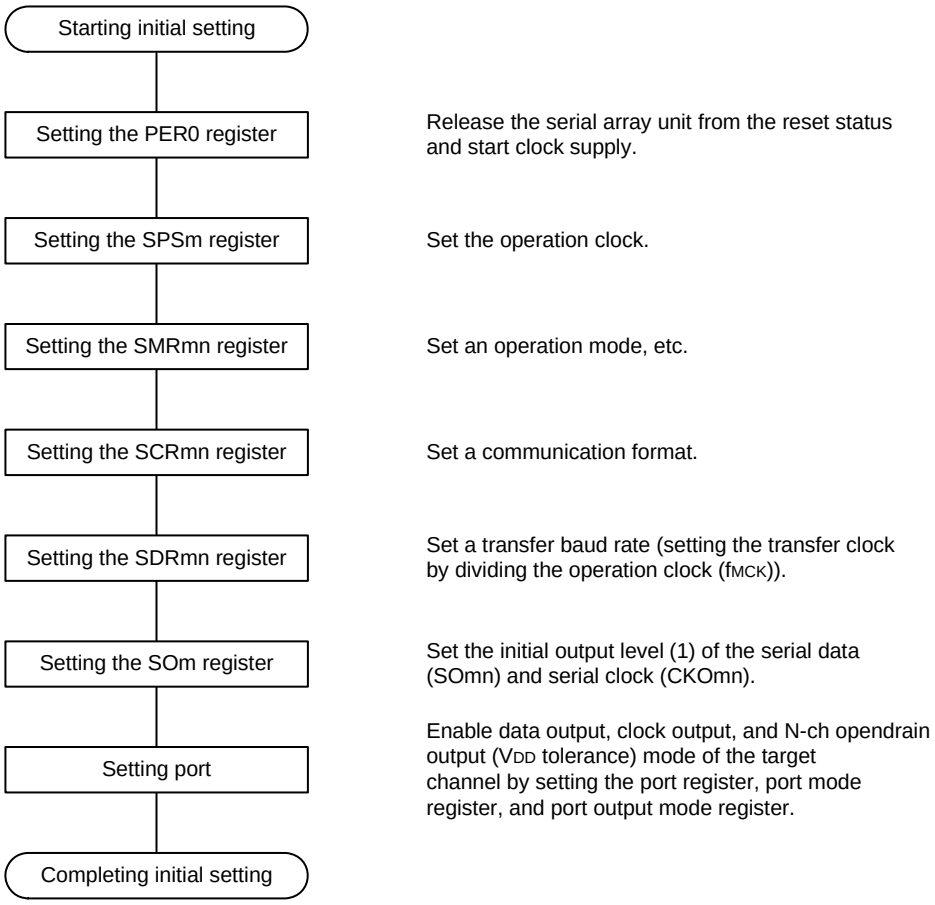
▒: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

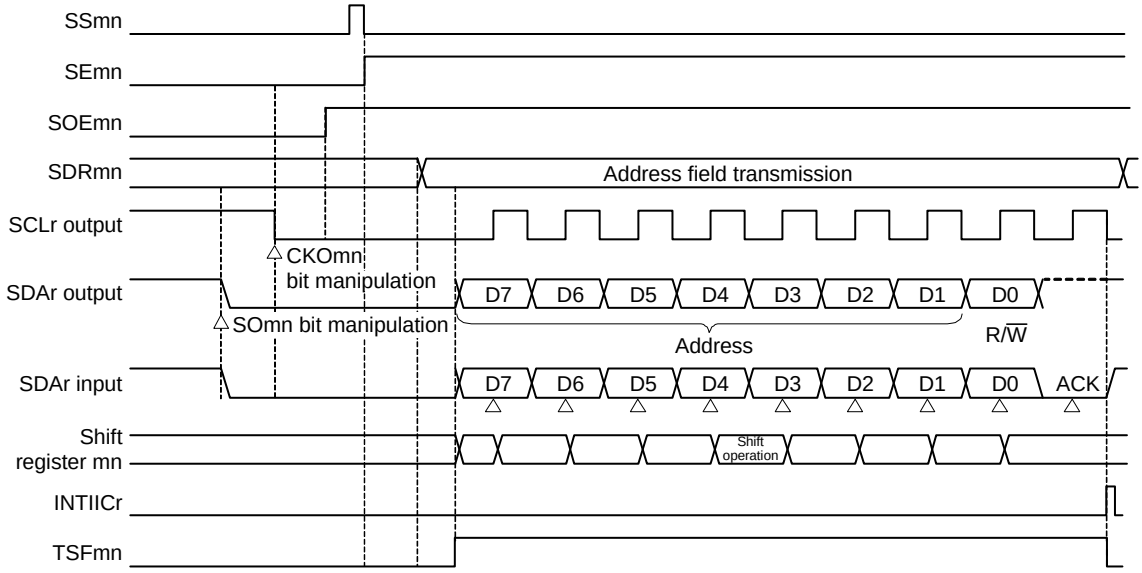
(2) Operation procedure

Figure 15 - 109 Initial Setting Procedure for Address Field Transmission



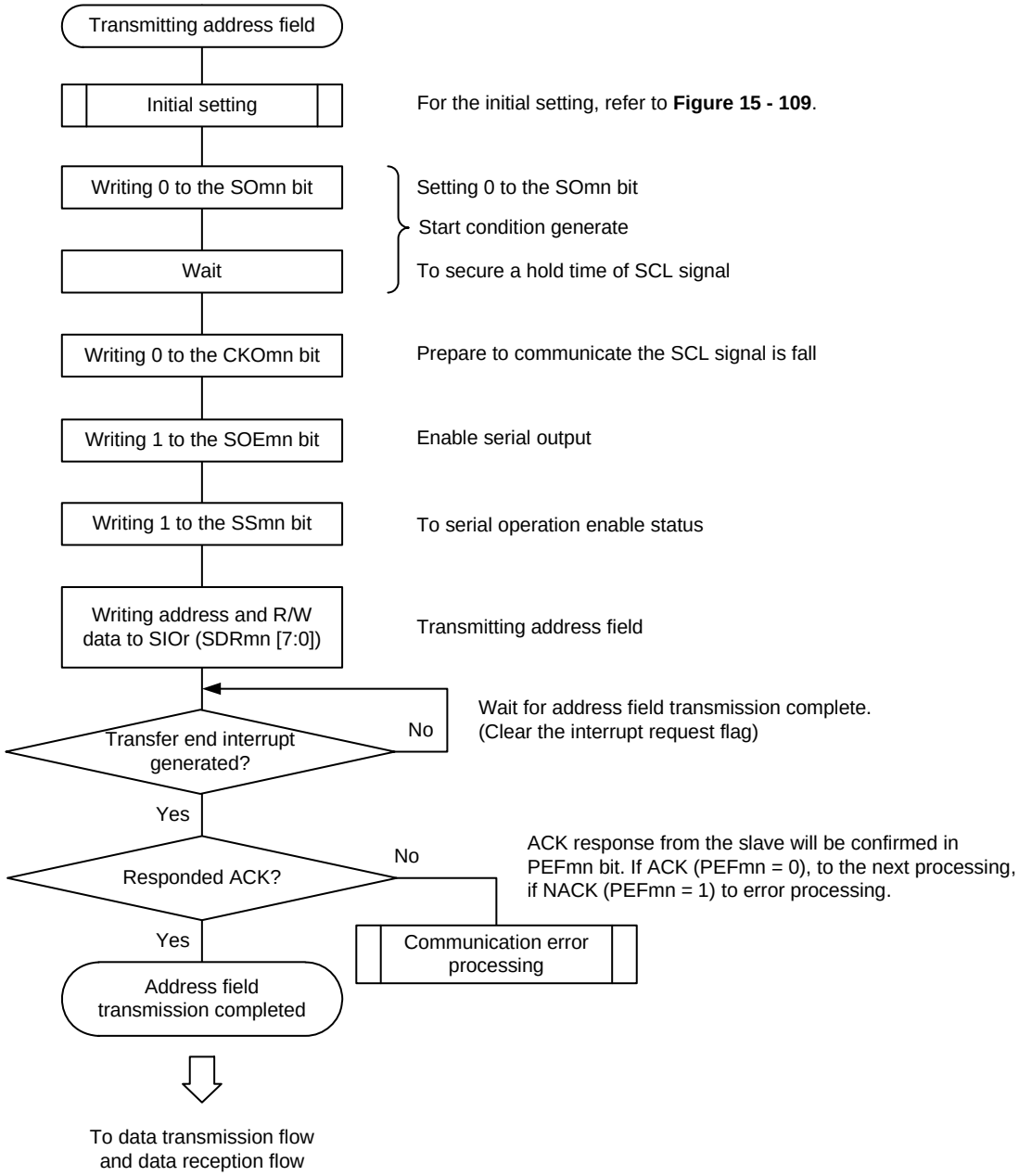
(3) Processing flow

Figure 15 - 110 Timing Chart of Address Field Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 00, 10, 20, 30), mn = 00, 02, 10, 12

Figure 15 - 111 Flowchart of Address Field Transmission



15.8.2 Data transmission

Data transmission is an operation to transmit data to the target for transfer (slave) after transmission of an address field. After all data are transmitted to the slave, a stop condition is generated and the bus is released.

Simplified I²C	IIC00	IIC10	IIC20	IIC30
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1	Channel 2 of SAU1
Pins used	SCL00, SDA00 Note 1	SCL10, SDA10 Note 1	SCL20, SDA20 Note 1	SCL30, SDA30 Note 1
Interrupt	INTIIC00	INTIIC10	INTIIC20	INTIIC30
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error detection flag	ACK error flag (PEFmn)			
Transfer data length	8 bits			
Transfer rate Note 2	Max. fMCK/4 [Hz] (SDRmn[15:9] = 1 or more) fMCK: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I²C. • Max. 1 MHz (fast mode plus) • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)			
Data level	Non-reversed output (default: high level)			
Parity bit	No parity bit			
Stop bit	Appending 1 bit (for ACK reception timing)			
Data direction	MSB first			

Note 1. To perform communication via simplified I²C, set the N-ch open-drain output (VDD tolerance mode (POMxx = 1) with the port output mode register (POMxx). For details, see **4.3 Registers Controlling Port Function** and **4.5 Register Settings When Using Alternate Function**.
When IIC00, IIC10, IIC20, IIC30 is communicating with an external device with a different potential, set the N-ch open-drain output (VDD tolerance mode (POMxx = 1) also for the clock input/output pins (SCL00, SCL10, SCL20, SCL30).
For details, see **4.4.4 Handling different potential (1.8 V, 2.5 V) by using I/O buffers**.

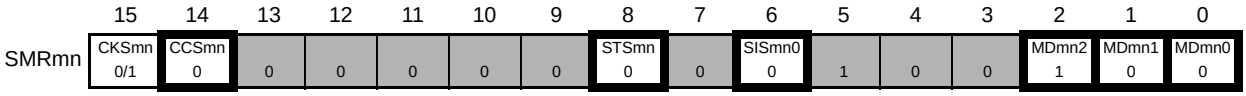
Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 34** or **CHAPTER 35 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12

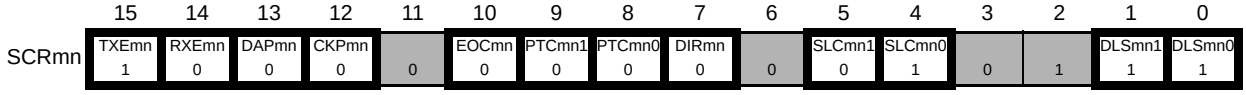
(1) Register setting

Figure 15 - 112 Example of Contents of Registers for Data Transmission of Simplified I²C (IIC00, IIC10, IIC20, IIC30)

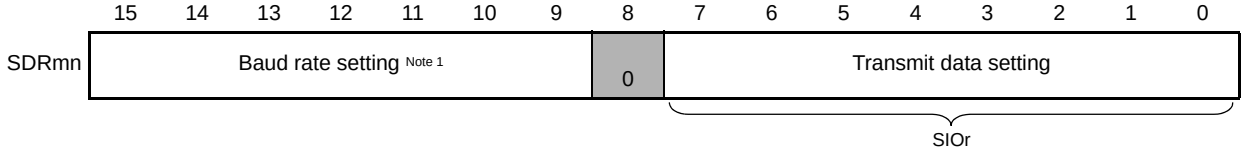
(a) Serial mode register mn (SMRmn)... Do not manipulate this register during data transmission/reception.



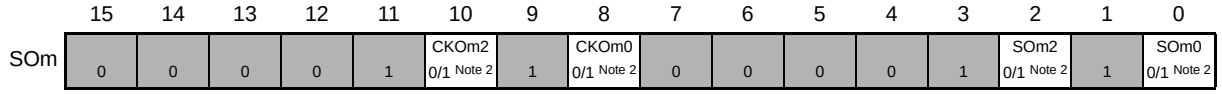
(b) Serial communication operation setting register mn (SCRmn)... Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.



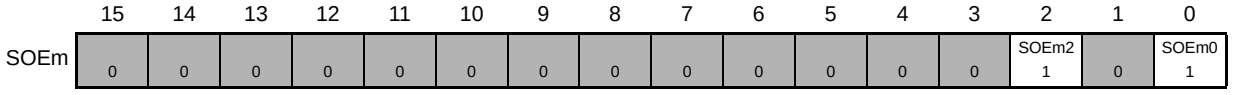
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)... During data transmission/reception, valid only lower 8-bits (SIOr)



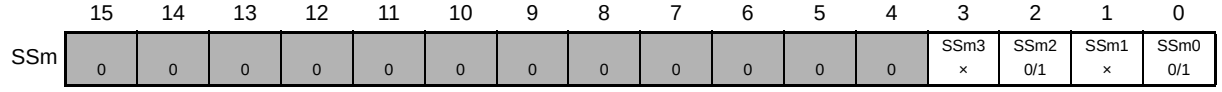
(d) Serial output register m (SOm)... Do not manipulate this register during data transmission/reception.



(e) Serial output enable register m (SOEm)... Do not manipulate this register during data transmission/reception



(f) Serial channel start register m (SSm)... Do not manipulate this register during data transmission/reception.



Note 1. Because the setting is completed by address field transmission, setting is not required.

Note 2. The value varies depending on the communication data during communication operation.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 00, 10, 20, 30), mn = 00, 02, 10, 12

Remark 2. ■: Setting is fixed in the IIC mode,

▒: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Processing flow

Figure 15 - 113 Timing Chart of Data Transmission

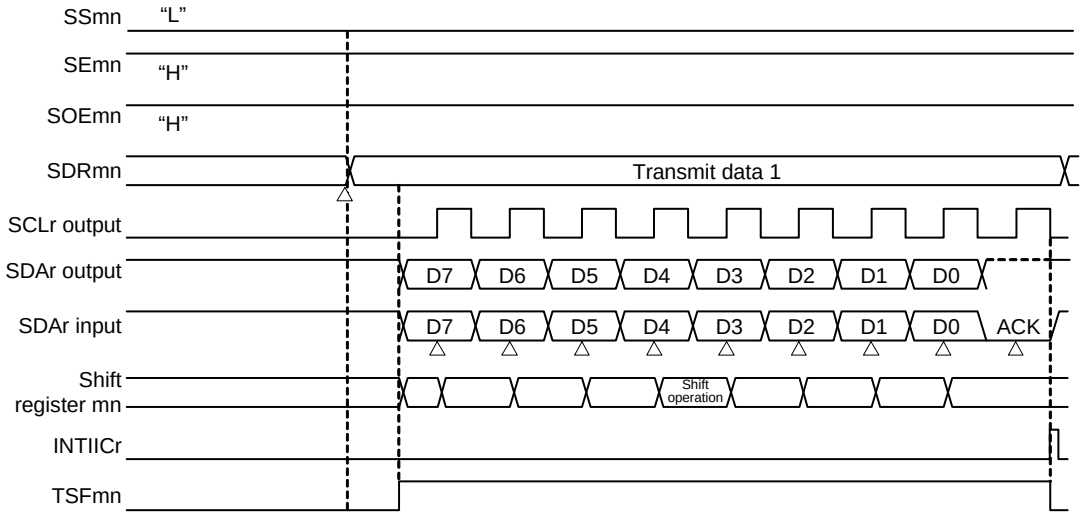
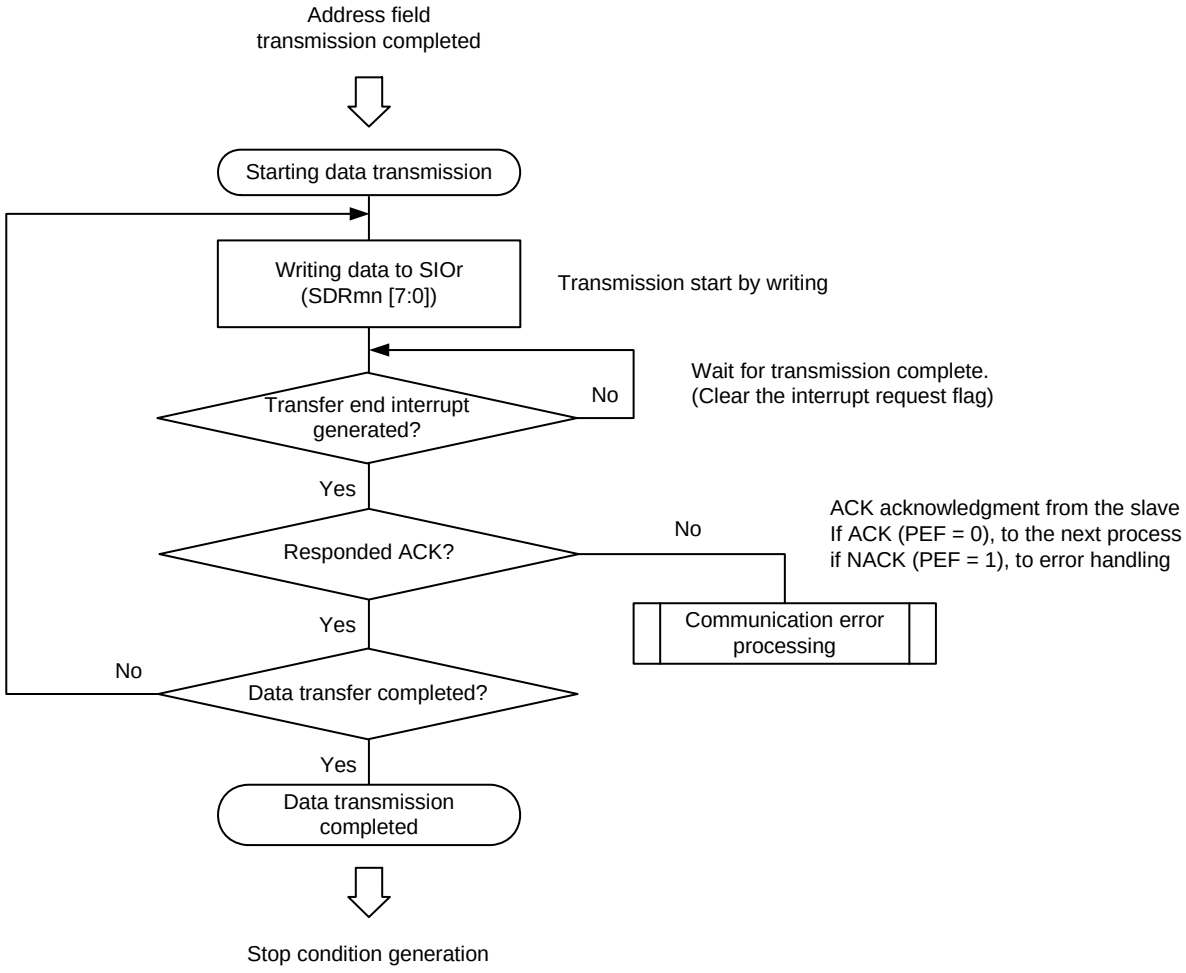


Figure 15 - 114 Flowchart of Simplified I²C Data Transmission



15.8.3 Data reception

Data reception is an operation to receive data to the target for transfer (slave) after transmission of an address field. After all data are received to the slave, a stop condition is generated and the bus is released.

Simplified I²C	IIC00	IIC10	IIC20	IIC30
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1	Channel 2 of SAU1
Pins used	SCL00, SDA00 Note 1	SCL10, SDA10 Note 1	SCL20, SDA20 Note 1	SCL30, SDA30 Note 1
Interrupt	INTIIC00	INTIIC10	INTIIC20	INTIIC30
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error detection flag	Overrun error detection flag (OVFmn) only			
Transfer data length	8 bits			
Transfer rate Note 2	Max. fMCK/4 [Hz] (SDRmn[15:9] = 1 or more) fMCK: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I²C. • Max. 1 MHz (fast mode plus) • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)			
Data level	Non-reversed output (default: high level)			
Parity bit	No parity bit			
Stop bit	Appending 1 bit (ACK transmission)			
Data direction	MSB first			

Note 1. To perform communication via simplified I²C, set the N-ch open-drain output (VDD tolerance mode (POMxx = 1) with the port output mode register (POMxx). For details, see **4.3 Registers Controlling Port Function** and **4.5 Register Settings When Using Alternate Function**.
When IIC00, IIC10, IIC20, IIC30 is communicating with an external device with a different potential, set the N-ch open-drain output (VDD tolerance mode (POMxx = 1) also for the clock input/output pins (SCL00, SCL10, SCL20, SCL30).
For details, see **4.4.4 Handling different potential (1.8 V, 2.5 V) by using I/O buffers**.

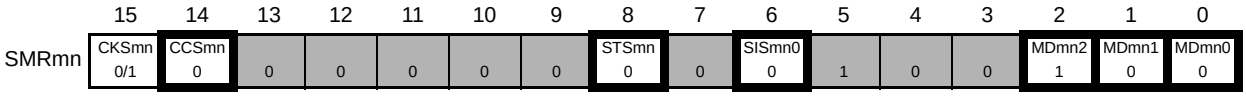
Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 34** or **CHAPTER 35 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12

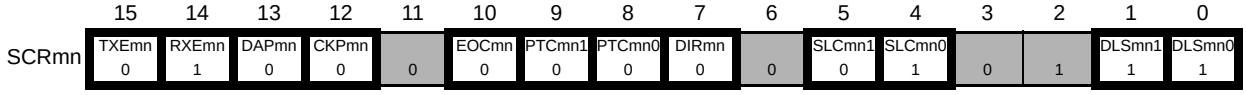
(1) Register setting

Figure 15 - 115 Example of Contents of Registers for Data Reception of Simplified I²C (IIC00, IIC10, IIC20, IIC30)

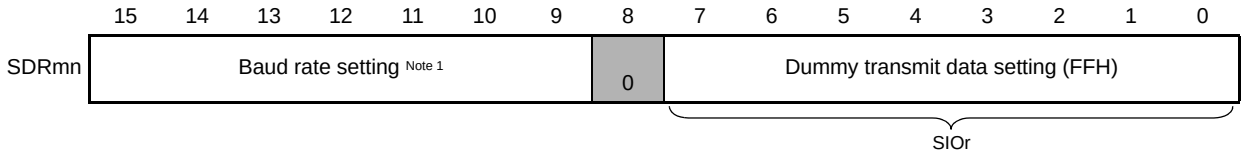
(a) Serial mode register mn (SMRmn)... Do not manipulate this register during data transmission/reception.



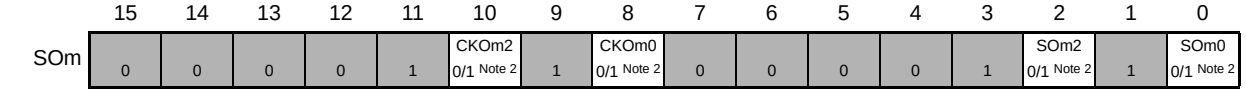
(b) Serial communication operation setting register mn (SCRmn)... Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.



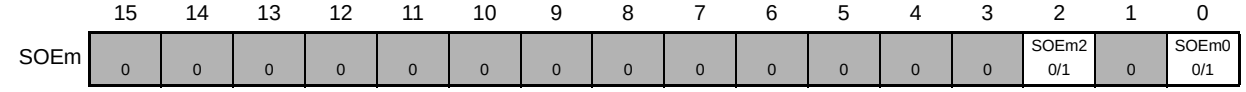
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)



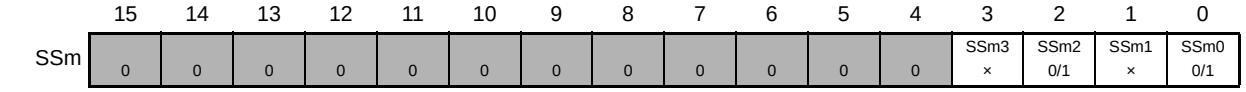
(d) Serial output register m (SOm)... Do not manipulate this register during data transmission/reception.



(e) Serial output enable register m (SOEm)... Do not manipulate this register during data transmission/reception.



(f) Serial channel start register m (SSm)... Do not manipulate this register during data transmission/reception.



Note 1. The baud rate setting is not required because the baud rate has already been set when the address field was transmitted.

Note 2. The value varies depending on the communication data during communication operation.

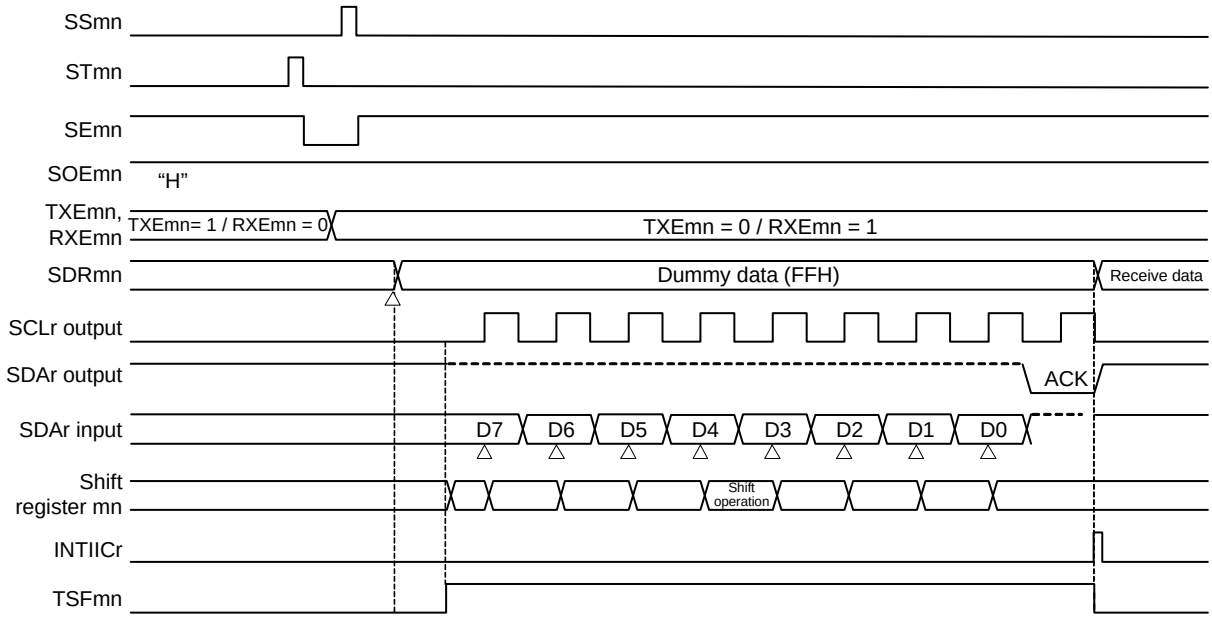
Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 00, 10, 20, 30), mn = 00, 02, 10, 12

Remark 2. ■: Setting is fixed in the IIC mode,
■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

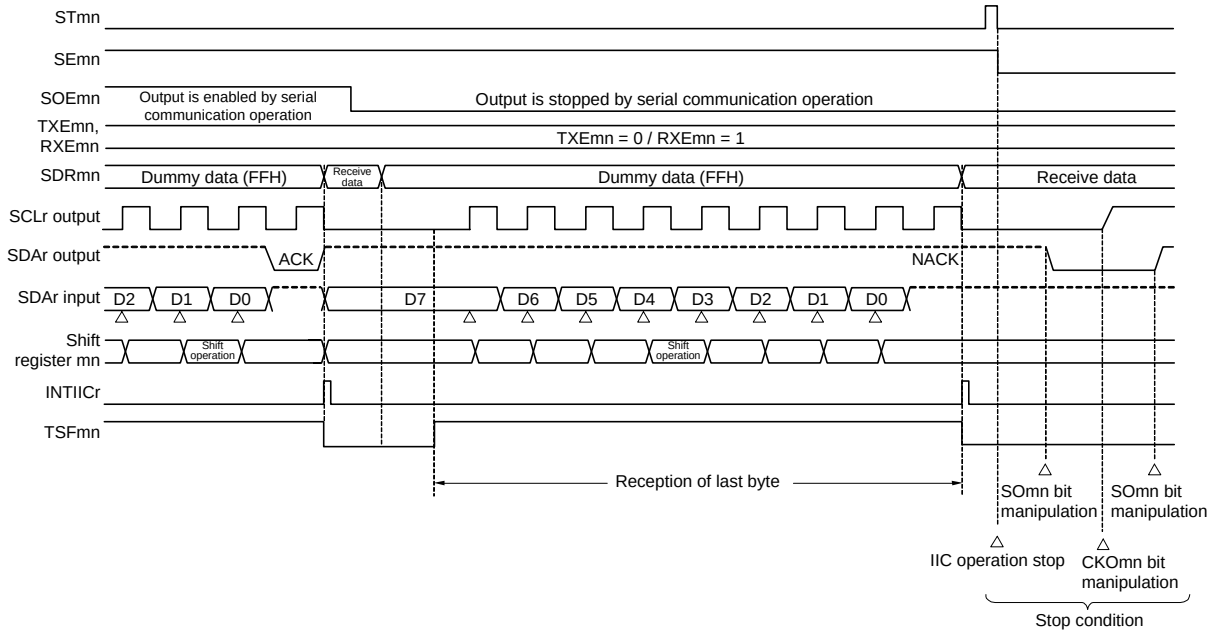
(2) Processing flow

Figure 15 - 116 Timing Chart of Data Reception

(a) When starting data reception

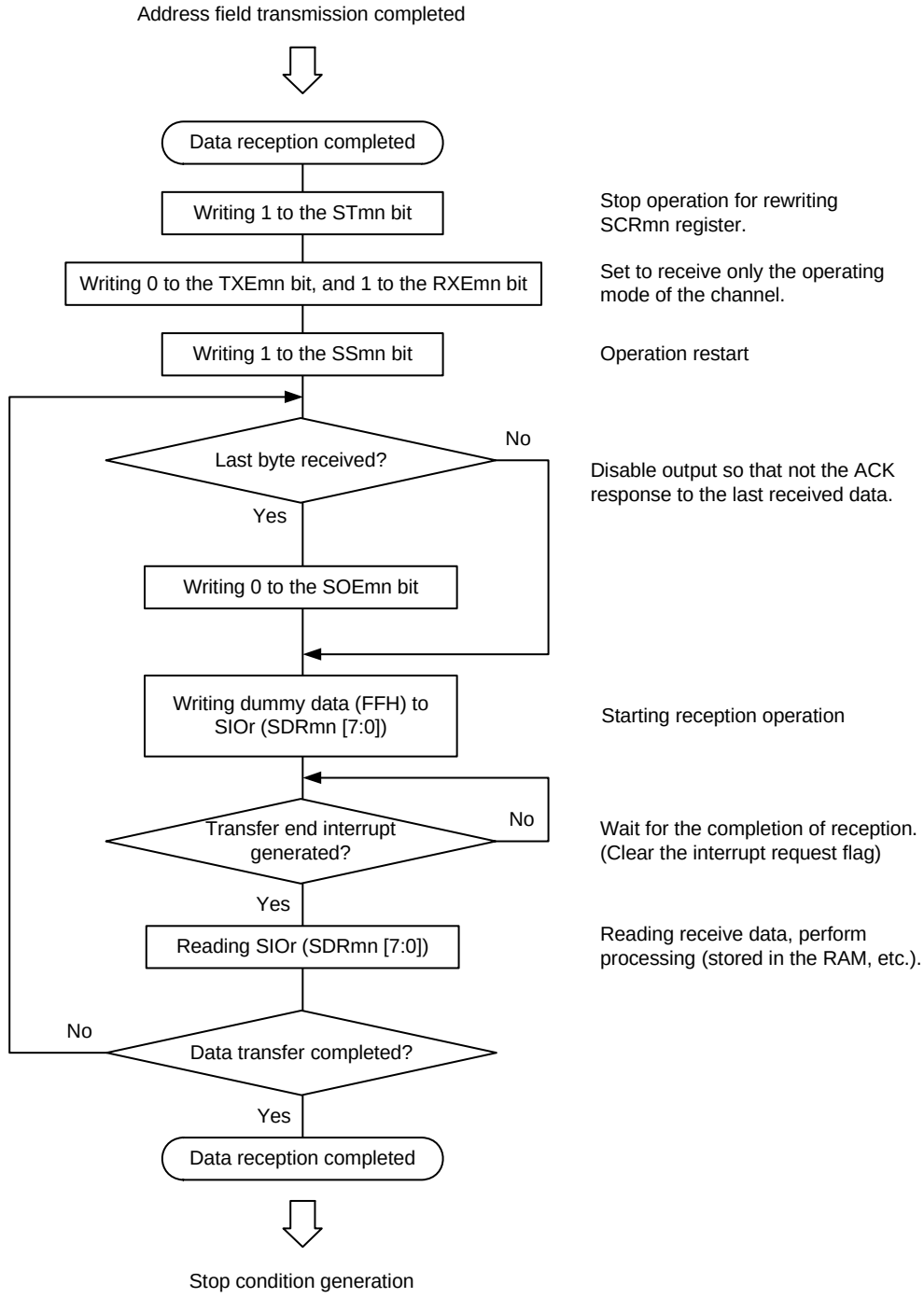


(b) When receiving last data



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 00, 10, 20, 30), mn = 00, 02, 10, 12

Figure 15 - 117 Flowchart of Data Reception



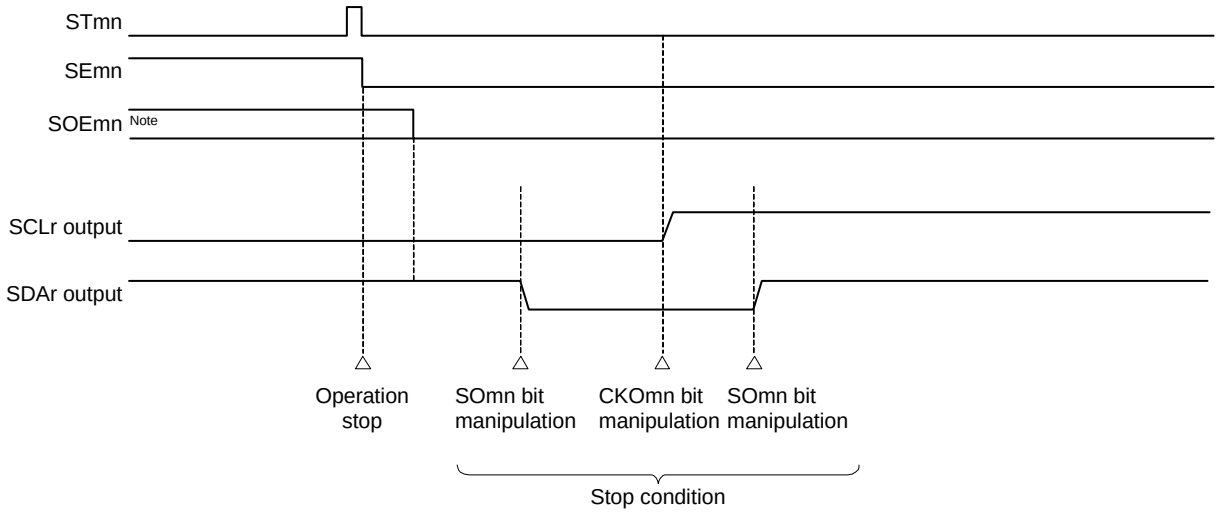
Caution ACK is not output when the last data is received (NACK). Communication is then completed by setting "1" to the STmn bit of serial channel stop register m (STm) to stop operation and generating a stop condition.

15.8.4 Stop condition generation

After all data are transmitted to or received from the target slave, a stop condition is generated and the bus is released.

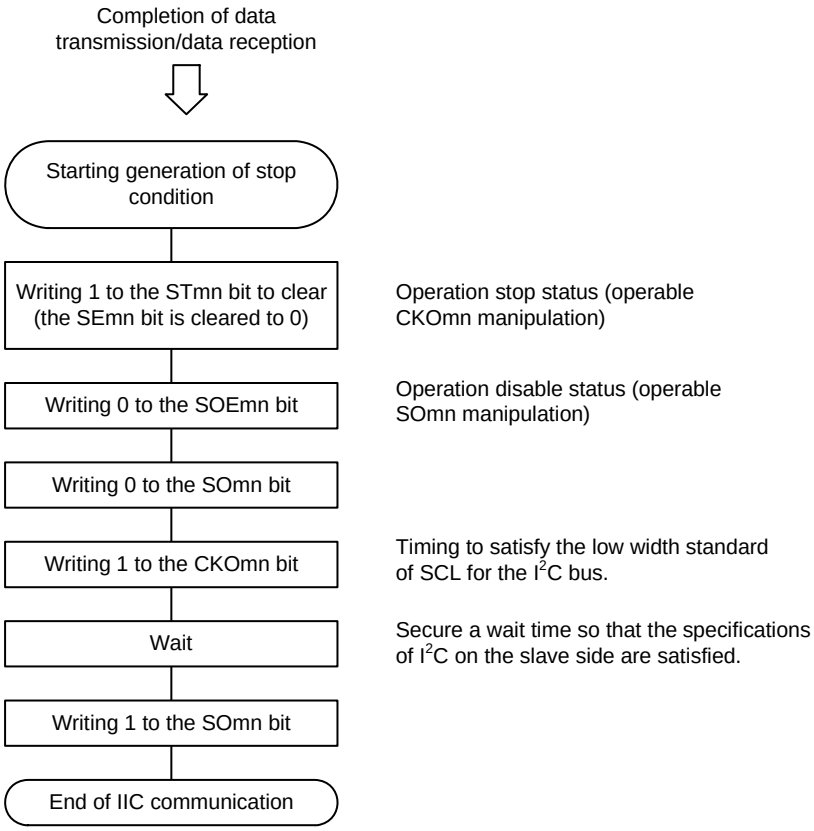
(1) Processing flow

Figure 15 - 118 Timing Chart of Stop Condition Generation



Note During a receive operation, the SOEmn bit of serial output enable register m (SOEm) is cleared to 0 before receiving the last data.

Figure 15 - 119 Flowchart of Stop Condition Generation



15.8.5 Calculating transfer rate

The transfer rate for simplified I²C (IIC00, IIC10) communication can be calculated by the following expressions.

(Transfer rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2

Caution SDRmn[15:9] must not be set to 00000000B. Be sure to set a value of 00000001B or greater for SDRmn[15:9]. The duty ratio of the SCL signal output by the simplified I²C is 50%. The I²C bus specifications define that the low-level width of the SCL signal is longer than the high-level width. If 400 kbps (fast mode) or 1 Mbps (fast mode plus) is specified, therefore, the low-level width of the SCL output signal becomes shorter than the value specified in the I²C bus specifications. Make sure that the SDRmn[15:9] value satisfies the I²C bus specifications.

Remark 1. The value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000001B to 1111111B) and therefore is 1 to 127.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12

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 - 5 Selection of Operation Clock For Simplified I²C

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 24 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	24 MHz
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	12 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	6 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	3 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	1.5 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	750 kHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	375 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	187.5 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	93.8 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	46.9 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	23.4 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	11.7 kHz
1	0	0	0	0	×	×	×	×	f_{CLK}	24 MHz
	0	0	0	1	×	×	×	×	$f_{CLK}/2$	12 MHz
	0	0	1	0	×	×	×	×	$f_{CLK}/2^2$	6 MHz
	0	0	1	1	×	×	×	×	$f_{CLK}/2^3$	3 MHz
	0	1	0	0	×	×	×	×	$f_{CLK}/2^4$	1.5 MHz
	0	1	0	1	×	×	×	×	$f_{CLK}/2^5$	750 kHz
	0	1	1	0	×	×	×	×	$f_{CLK}/2^6$	375 kHz
	0	1	1	1	×	×	×	×	$f_{CLK}/2^7$	187.5 kHz
	1	0	0	0	×	×	×	×	$f_{CLK}/2^8$	93.8 kHz
	1	0	0	1	×	×	×	×	$f_{CLK}/2^9$	46.9 kHz
	1	0	1	0	×	×	×	×	$f_{CLK}/2^{10}$	23.4 kHz
	1	0	1	1	×	×	×	×	$f_{CLK}/2^{11}$	11.7 kHz
Other than above									Setting prohibited	

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remark 1. ×: Don't care

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12

Here is an example of setting an I²C transfer rate where f_{MCK} = f_{CLK} = 24 MHz.

I²C Transfer Mode (Desired Transfer Rate)	f_{CLK} = 24 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Transfer Rate	Error from Desired Transfer Rate
100 kHz	$f_{CLK}/2$	59	100 kHz	0.0%
400 kHz	f_{CLK}	31	375 kHz	6.25% Note
1 MHz	f_{CLK}	14	0.80 MHz	20.0% Note

Note The error cannot be set to about 0% because the duty ratio of the SCL signal is 50%.

15.8.6 Procedure for processing errors that occurred during simplified I²C (IIC00, IIC10, IIC20, IIC30) communication

The procedure for processing errors that occurred during simplified I²C (IIC00, IIC10, IIC20, IIC30) communication is described in Figures 15 - 120 and 15 - 121.

Figure 15 - 120 Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		The error type is identified and the read value is used to clear the error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	The error flag is cleared.	The error only during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

Figure 15 - 121 Processing Procedure in Case of ACK error in Simplified I²C Mode

Software Manipulation	Hardware Status	Remark
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STmn bit of serial channel stop register m (STm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operation.	Slave is not ready for reception because ACK is not returned. Therefore, a stop condition is created, the bus is released, and communication is started again from the start condition. Or, a restart condition is generated and transmission can be redone from address transmission.
Creates stop condition.		
Creates start condition.		
Sets the SSmn bit of serial channel start register m (SSm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 00, 10, 20, 30), mn = 00, 02, 10, 12

CHAPTER 16 SERIAL INTERFACE IICA

16.1 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 (SCLAn) line and a serial data bus (SDAAn) 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 SCLAn and SDAAn pins are used for open drain outputs, serial interface IICA requires pull-up resistors for the serial clock line and the serial data bus line.

(3) Wakeup mode

The STOP mode can be released by generating an interrupt request signal (INTIICAn) when an extension code from the master device or a local address has been received while in STOP mode. This can be set by using the WUPn bit of IICA control register n1 (IICCTLn1).

Figure 16 - 1 shows a block diagram of serial interface IICA

Remark n = 0

Figure 16 - 1 Block Diagram of Serial Interface IICA

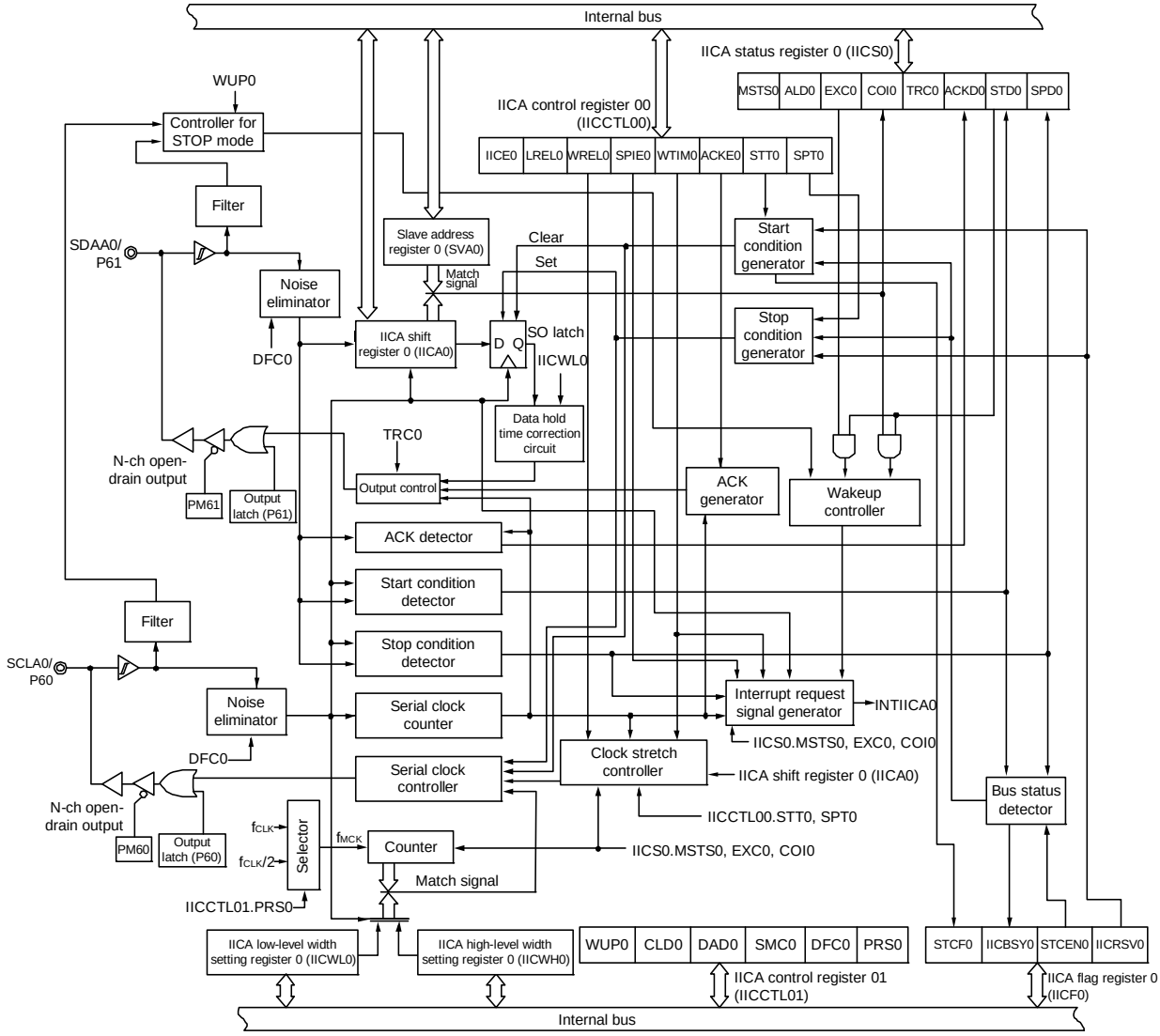
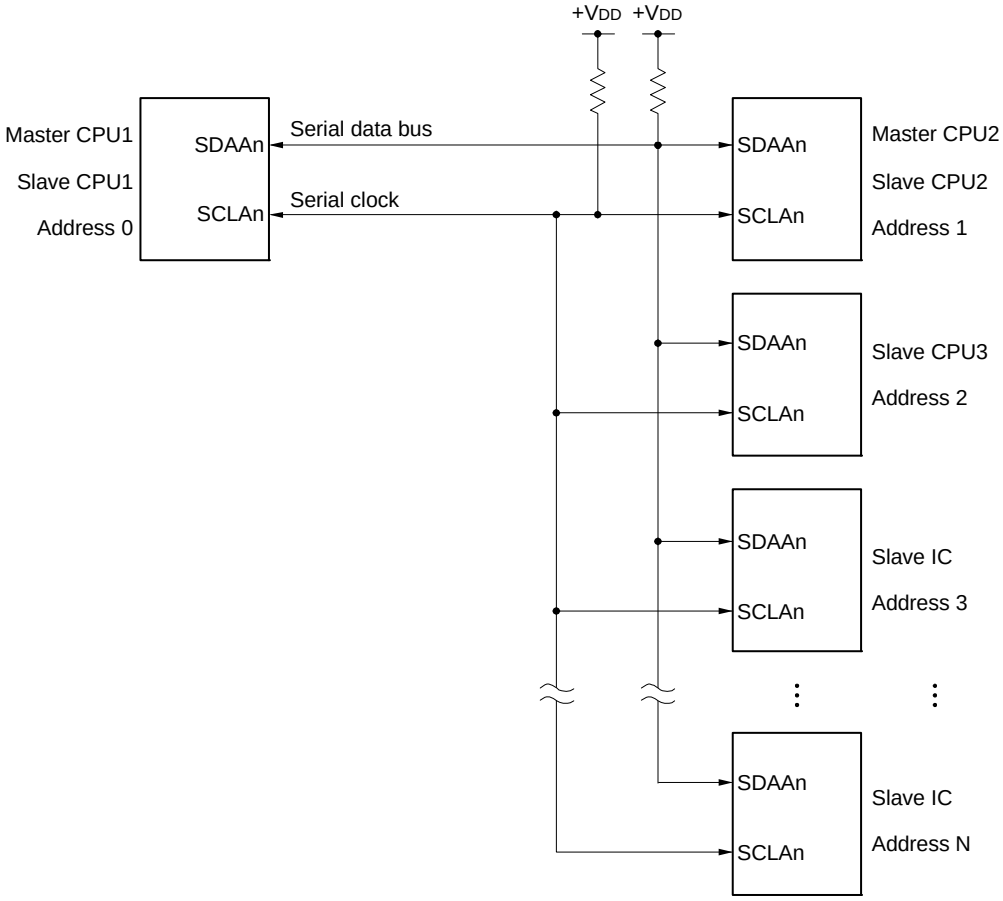


Figure 16 - 2 shows a serial bus configuration example.

Figure 16 - 2 Serial Bus Configuration Example Using I²C Bus



Remark n = 0

16.2 Configuration of Serial Interface IICA

Serial interface IICA includes the following hardware.

Table 16 - 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 6 (PM6) Port register 6 (P6)

Remark n = 0

(1) IICA shift register 0 (IICA0)

The IICAn register is used to convert 8-bit serial data to 8-bit parallel data and vice versa in synchronization with the serial clock. The IICAn register can be used for both transmission and reception.

The actual transmit and receive operations can be controlled by writing and reading operations to the IICAn register.

Cancel the clock stretch state and start data transfer by writing data to the IICAn register during the clock stretch period.

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

Reset signal generation clears IICAn to 00H.

Figure 16 - 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								

Caution 1. Do not write data to the IICAn register during data transfer.

Caution 2. Write or read the IICAn register only during the clock stretch period. Accessing the IICAn register in a communication state other than during the clock stretch period is prohibited. When the device serves as the master, however, the IICAn register can be written only once after the communication trigger bit (STTn) is set to 1.

Caution 3. When communication is reserved, write data to the IICAn register after the interrupt triggered by a stop condition is detected.

Remark n = 0

(2) Slave address register 0 (SVA0)

This register stores seven bits of local addresses {A6, A5, A4, A3, A2, A1, A0} when in slave mode.

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

However, rewriting to this register is prohibited while STDn = 1 (while the start condition is detected).

Reset signal generation clears the SVAn register to 00H.

Figure 16 - 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	0 Note

Note Bit 0 is fixed to 0.

(3) SO latch

The SO latch is used to retain the SDAAn pin's output level.

(4) Wakeup controller

This circuit generates an interrupt request (INTIICAn) when the address received by this register matches the address value set to the slave address register n (SVAn) 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 (INTIICAn).

An I^2C interrupt request is generated by the following two triggers.

- Falling edge of eighth or ninth clock of the serial clock (set by the WTIMn bit)
- Interrupt request generated when a stop condition is detected (set by the SPIEn bit)

Remark WTIMn bit: Bit 3 of IICA control register n0 (IICCTLn0)
SPIEn bit: Bit 4 of IICA control register n0 (IICCTLn0)

(7) Serial clock controller

In master mode, this circuit generates the clock output via the SCLAn pin from a sampling clock.

(8) Clock stretch controller

This circuit controls the clock stretch timing.

Remark n = 0

(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 STTn bit is set to 1.

However, in the communication reservation disabled status (IICRSVn bit = 1), when the bus is not released (IICBSYn bit = 1), start condition requests are ignored and the STCFn bit is set to 1.

(12) Stop condition generator

This circuit generates a stop condition when the SPTn 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 STCENn bit.

Remark 1. STTn bit: Bit 1 of IICA control register n0 (IICCTLn0)
SPTn bit: Bit 0 of IICA control register n0 (IICCTLn0)
IICRSVn bit: Bit 0 of IICA flag register n (IICFn)
IICBSYn bit: Bit 6 of IICA flag register n (IICFn)
STCFn bit: Bit 7 of IICA flag register n (IICFn)
STCENn bit: Bit 1 of IICA flag register n (IICFn)

Remark 2. n = 0

16.3 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 6 (PM6)
- Port register 6 (P6)

Remark n = 0

16.3.1 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 IICAn is used, be sure to set bit 4 (IICAnEN) 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 16 - 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	RTCWEN	0	ADCEN	IICAnEN	SAU1EN	SAU0EN	0	TAU0EN

IICAnEN	Control of serial interface IICAn input clock supply
0	Stops input clock supply. • SFR used by serial interface IICAn cannot be written. • Serial interface IICAn is in the reset status.
1	Enables input clock supply. • SFR used by serial interface IICAn can be read/written.

Caution 1. When setting serial interface IICA, be sure to set the following registers first while the IICAnEN bit is set to 1. If IICAnEN = 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 6 (PM6) and port register 6 (P6)).

- **IICA control register n0 (IICCTLn0)**
- **IICA flag register n (IICFn)**
- **IICA status register n (IICSn)**
- **IICA control register n1 (IICCTLn1)**
- **IICA low-level width setting register n (IICWLn)**
- **IICA high-level width setting register n (IICWHn)**

Caution 2. Be sure to clear bits 1, 6 to 0.

Remark n = 0

16.3.2 IICA control register 00 (IICCTL00)

This register is used to enable/stop I^2C operations, set clock stretch timing, and set other I^2C operations.

The IICCTLn0 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, set the SPIEn, WTIMn, and ACKEn bits while IICEn = 0 or during the clock stretch period. These bits can be set at the same time when the IICEn bit is set from “0” to “1”.

Reset signal generation clears this register to 00H.

Remark n = 0

Figure 16 - 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	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

IICEn	I^2C operation enable
0	Stop operation. Reset the IICA status register n (IICSn) Note 1. Stop internal operation.
1	Enable operation.
Be sure to set this bit (1) while the SCLAn and SDAAn lines are at high level.	
Condition for clearing (IICEn = 0)	Condition for setting (IICEn = 1)
• Cleared by instruction • Reset	• Set by instruction

LRELn 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 SCLAn and SDAAn lines are set to high impedance. The following flags of IICA control register n0 (IICCTLn0) and the IICA status register n (IICSn) are cleared to 0. • STTn • SPTn • MSTSn • EXCn • COIn • TRCn • ACKDn • STDn
The standby mode following exit from communications remains in effect until the following communications entry conditions are met. • After a stop condition is detected, restart is in master mode. • An address match or extension code reception occurs after the start condition.	
Condition for clearing (LRELn = 0)	Condition for setting (LRELn = 1)
• Automatically cleared after execution • Reset	• Set by instruction

WRELn Notes 2, 3	Clock stretch cancellation
0	Do not cancel clock stretch
1	Cancel clock stretch. This setting is automatically cleared after clock stretch is canceled.
When the WRELn bit is set (clock stretch canceled) during the clock stretch period at the ninth clock pulse in the transmission status (TRCn = 1), the SDAAn line goes into the high impedance state (TRCn = 0).	
Condition for clearing (WRELn = 0)	Condition for setting (WRELn = 1)
• Automatically cleared after execution • Reset	• Set by instruction

Note 1. The IICA shift register n (IICAn), the STCFn and IICBSYn bits of the IICA flag register n (IICFn), and the CLDn and DADn bits of IICA control register n1 (IICCTLn1) are reset.

Note 2. The signal of this bit is invalid while IICEn is 0.

Note 3. When the LRELn and WRELn bits are read, 0 is always read.

Caution **If the operation of I^2C is enabled (IICEn = 1) when the SCLAn line is high level, the SDAAn line is low level, and the digital filter is turned on (DFCn bit of IICCTLn1 register = 1), a start condition will be inadvertently detected immediately. In this case, set (1) the LRELn bit by using a 1-bit memory manipulation instruction immediately after enabling operation of I^2C (IICEn = 1).**

Remark n = 0

Figure 16 - 7 Format of IICA control register 00 (IICCTL00) (2/4)

SPIEn Note 1	Enable/disable generation of interrupt request when stop condition is detected
0	Disable
1	Enable
If the WUPn bit of IICA control register n1 (IICCTLn1) is 1, no stop condition interrupt will be generated even if SPIEn = 1.	
Condition for clearing (SPIEn = 0)	Condition for setting (SPIEn = 1)
• Cleared by instruction • Reset	• Set by instruction

WTIMn Note 1	Control of clock stretch 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 stretch is set. Slave mode: After input of eight clocks, the clock is set to low level and clock stretch 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 stretch is set. Slave mode: After input of nine clocks, the clock is set to low level and clock stretch 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 stretch 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 stretch 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 stretch is inserted at the falling edge of the eighth clock.	
Condition for clearing (WTIMn = 0)	Condition for setting (WTIMn = 1)
• Cleared by instruction • Reset	• Set by instruction

ACKEn Notes 1, 2	Acknowledgment control
0	Disable acknowledgment.
1	Enable acknowledgment. During the ninth clock period, the SDAAn line is set to low level.
Condition for clearing (ACKEn = 0)	Condition for setting (ACKEn = 1)
• Cleared by instruction • Reset	• Set by instruction

Note 1. The signal of this bit is invalid while IICEn is 0. Set this bit during that period.

Note 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.

Remark n = 0

Figure 16 - 8 Format of IICA control register 00 (IICCTL00) (3/4)

STTn Notes 1, 2	Start condition trigger
0	Do not generate a start condition.
1	When bus is released (in standby state, when IICBSYn = 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 (IICRSVn = 0) Functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus is released. • When communication reservation function is disabled (IICRSVn = 1) Even if this bit is set (1), the STTn bit is cleared and the STTn clear flag (STCFn) is set (1). No start condition is generated. In the clock stretch state (when master device): Generates a restart condition after releasing the clock stretch.

Cautions concerning set timing
- For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the clock stretching period when the ACKEn bit has been cleared to 0 and slave has been notified of final reception.
- For master transmission: A start condition cannot be generated normally during the acknowledge period. Set to 1 during the clock stretch period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as stop condition trigger (SPTn).
- Once STTn is set (1), setting it again (1) before the clear condition is met is not allowed.

Condition for clearing (STTn = 0)	Condition for setting (STTn = 1)
• Cleared by setting the STTn bit to 1 while communication reservation is prohibited. • Cleared by loss in arbitration • Cleared after start condition is generated by master device • Cleared by LRELn = 1 (exit from communications) • When IICEn = 0 (operation stop) • Reset	• Set by instruction

Note 1. The signal of this bit is invalid while IICEn is 0.
Note 2. The STTn bit is always read as 0.

Remark 1. IICRSVn: Bit 0 of IICA flag register n (IICFn)
STCFn: Bit 7 of IICA flag register n (IICFn)
Remark 2. n = 0

Figure 16 - 9 Format of IICA control register 00 (IICCTL00) (4/4)

SPTn 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 stretching period when the ACKEn bit has been cleared to 0 and slave has been notified of final reception.
- For master transmission: A stop condition cannot be generated normally during the acknowledge period. Therefore, set it during the clock stretch period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as start condition trigger (STTn).
- The SPTn bit can be set to 1 only when in master mode.
- When the WTIMn bit has been cleared to 0, if the SPTn bit is set to 1 during the clock stretch period that follows output of eight clocks, note that a stop condition will be generated during the high-level period of the ninth clock. The WTIMn bit should be changed from 0 to 1 during the clock stretch period following the output of eight clocks, and the SPTn bit should be set to 1 during the clock stretch period that follows the output of the ninth clock.
- Once SPTn is set (1), setting it again (1) before the clear condition is met is not allowed.

Condition for clearing (SPTn = 0)	Condition for setting (SPTn = 1)
• Cleared by loss in arbitration • Automatically cleared after stop condition is detected • Cleared by LRELn = 1 (exit from communications) • When IICEn = 0 (operation stop) • Reset	• Set by instruction

Note The SPTn bit is always read as 0.

Caution **When bit 3 (TRCn) of the IICA status register n (IICSn) is set to 1 (transmission status), bit 5 (WRELn) of IICA control register n0 (IICCTLn0) is set to 1 during the ninth clock and clock stretch is canceled, after which the TRCn bit is cleared (reception status) and the SDAAn line is set to high impedance. Release the clock stretch performed while the TRCn bit is 1 (transmission status) by writing to the IICA shift register n.**

Remark 1. Bit 0 (SPTn) becomes 0 when it is read after data setting.
Remark 2. n = 0

16.3.3 IICA status register 0 (IICS0)

This register indicates the status of I^2C.

The IICSn register is read by a 1-bit or 8-bit memory manipulation instruction only when STTn = 1 and during the clock stretch period.

Reset signal generation clears this register to 00H.

Caution Reading the IICSn register while the address match wakeup function is enabled (WUPn = 1) in STOP mode is prohibited. When the WUPn bit is changed from 1 to 0 (wakeup operation is stopped), regardless of the INTIICAn 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 (SPIEn = 1) the interrupt generated by detecting a stop condition and read the IICSn register after the interrupt has been detected.

Remark STTn: bit 1 of IICA control register n0 (IICCTLn0)
WUPn: bit 7 of IICA control register n1 (IICCTLn1)

Figure 16 - 10 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	MSTSn	ALDn	EXCn	COIn	TRCn	ACKDn	STDn	SPDn

MSTSn	Master status check flag
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTSn = 0)	Condition for setting (MSTSn = 1)
• When a stop condition is detected • When ALDn = 1 (arbitration loss) • Cleared by LRELn = 1 (exit from communications) • When the IICEn bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is generated

ALDn	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 MSTSn bit is cleared.

Condition for clearing (ALDn = 0)	Condition for setting (ALDn = 1)
• Automatically cleared after the IICSn register is read Note • When the IICEn 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 IICSn register. Therefore, when using the ALDn bit, read the data of this bit before the data of the other bits.

Remark 1. LRELn: Bit 6 of IICA control register n0 (IICCTLn0)
IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0

Figure 16 - 11 Format of IICA status register 0 (IICS0) (2/3)

EXCn	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXCn = 0)	Condition for setting (EXCn = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LRELn = 1 (exit from communications) • When the IICEn bit changes from 1 to 0 (operation stop) • Reset	• When the higher four bits of the received address data is either “0000” or “1111” (set at the rising edge of the eighth clock).

COIn	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COIn = 0)	Condition for setting (COIn = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LRELn = 1 (exit from communications) • When the IICEn bit changes from 1 to 0 (operation stop) • Reset	• When the received address matches the local address (slave address register n (SVAn)) (set at the rising edge of the eighth clock).

TRCn	Detection of transmit/receive status
0	Receive status (other than transmit status). The SDAAn line is set for high impedance.
1	Transmit status. The value in the SOn latch is enabled for output to the SDAAn line (valid starting at the falling edge of the first byte’s ninth clock).

Condition for clearing (TRCn = 0)	Condition for setting (TRCn = 1)
<Both master and slave> • When a stop condition is detected • Cleared by LRELn = 1 (exit from communications) • When the IICEn bit changes from 1 to 0 (operation stop) • Cleared by WRELn = 1 Note (clock stretch cancel) • When the ALDn bit changes from 0 to 1 (arbitration loss) • Reset • When not used for communication (MSTSn, EXCn, COIn = 0) <Master> • When “1” is output to the first byte’s LSB (transfer direction specification bit) <Slave> • When a start condition is detected • When “0” is input to the first byte’s LSB (transfer direction specification bit)	<Master> • When a start condition is generated • When 0 (master transmission) is output to the LSB (transfer direction specification bit) of the first byte (during address transfer) <Slave> • When 1 (slave transmission) is input to the LSB (transfer direction specification bit) of the first byte from the master (during address transfer)

Note When bit 3 (TRCn) of the IICA status register n (IICSn) is set to 1 (transmission status), bit 5 (WRELn) of IICA control register n0 (IICCTLn0) is set to 1 during the ninth clock and clock stretch is canceled, after which the TRCn bit is cleared (reception status) and the SDAAn line is set to high impedance. Release the clock stretch performed while the TRCn bit is 1 (transmission status) by writing to the IICA shift register n.

Remark 1. LRELn: Bit 6 of IICA control register n0 (IICCTLn0)
IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0

Figure 16 - 12 Format of IICA status register 0 (IICS0) (3/3)

ACKDn	Detection of acknowledge (ACK)
0	Acknowledge was not detected.
1	Acknowledge was detected.

Condition for clearing (ACKDn = 0)	Condition for setting (ACKDn = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock • Cleared by LRELn = 1 (exit from communications) • When the IICEn bit changes from 1 to 0 (operation stop) • Reset	• After the SDAAn line is set to low level at the rising edge of SCLAn line's ninth clock

STDn	Detection of start condition
0	Start condition was not detected.
1	Start condition was detected. This indicates that the address transfer period is in effect.

Condition for clearing (STDn = 0)	Condition for setting (STDn = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by LRELn = 1 (exit from communications) • When the IICEn bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is detected

SPDn	Detection of stop condition
0	Stop condition was not detected.
1	Stop condition was detected. The master device's communication is terminated and the bus is released.

Condition for clearing (SPDn = 0)	Condition for setting (SPDn = 1)
• At the rising edge of the address transfer byte's first clock following setting of this bit and detection of a start condition • When the WUPn bit changes from 1 to 0 • When the IICEn bit changes from 1 to 0 (operation stop) • Reset	• When a stop condition is detected

Remark 1. LRELn: Bit 6 of IICA control register n0 (IICCTLn0)
IICEn: Bit 7 of IICA control register n0 (IICCTLn0)
Remark 2. n = 0

16.3.4 IICA flag register 0 (IICF0)

This register sets the operation mode of I^2C and indicates the status of the I^2C bus.
The IICFn register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the STTn clear flag (STCFn) and I^2C bus status flag (IICBSYn) bits are read-only.
The IICRSVn bit can be used to enable/disable the communication reservation function.
The STCENn bit can be used to set the initial value of the IICBSYn bit.
The IICRSVn and STCENn bits can be written only when the operation of I^2C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) = 0). When operation is enabled, the IICFn register can be read.
Reset signal generation clears this register to 00H.

Figure 16 - 13 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	STCFn	IICBSYn	0	0	0	0	STCENn	IICRSVn

STCFn	STTn clear flag
0	Generate start condition
1	Start condition generation unsuccessful: clear the STTn flag

Condition for clearing (STCFn = 0)	Condition for setting (STCFn = 1)
• Cleared by STTn = 1 • When IICEn = 0 (operation stop) • Reset	• Generating start condition unsuccessful and the STTn bit cleared to 0 when communication reservation is disabled (IICRSVn = 1).

IICBSYn	I^2C bus status flag
0	Bus release status (communication initial status when STCENn = 1)
1	Bus communication status (communication initial status when STCENn = 0)

Condition for clearing (IICBSYn = 0)	Condition for setting (IICBSYn = 1)
• Detection of stop condition • When IICEn = 0 (operation stop) • Reset	• Detection of start condition • Setting of the IICEn bit when STCENn = 0

STCENn	Initial start enable trigger
0	After operation is enabled (IICEn = 1), enable generation of a start condition upon detection of a stop condition.
1	After operation is enabled (IICEn = 1), enable generation of a start condition without detecting a stop condition.

Condition for clearing (STCENn = 0)	Condition for setting (STCENn = 1)
• Cleared by instruction • Detection of start condition • Reset	• Set by instruction

IICRSVn	Communication reservation function disable bit
0	Enable communication reservation
1	Disable communication reservation

Condition for clearing (IICRSVn = 0)	Condition for setting (IICRSVn = 1)
• Cleared by instruction • Reset	• Set by instruction

Note Bits 6 and 7 are read-only.

Caution 1. Write to the STCENn bit only when the operation is stopped (IICEn = 0).

Caution 2. As the bus release status (IICBSYn = 0) is recognized regardless of the actual bus status when STCENn = 1, when generating the first start condition (STTn = 1), it is necessary to verify that no third party communications are in progress in order to prevent such communications from being destroyed.

Caution 3. Write to IICRSVn only when the operation is stopped (IICEn = 0).

Remark 1. STTn: Bit 1 of IICA control register n0 (IICCTLn0)
IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0

16.3.5 IICA control register 01 (IICCTL01)

This register is used to set the operation mode of I^2C and detect the statuses of the SCLAn and SDAAn pins.

The IICCTLn1 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the CLDn and DADn bits are read-only.

Set the IICCTLn1 register, except the WUPn bit, while operation of I^2C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) is 0).

Reset signal generation clears this register to 00H.

Figure 16 - 14 Format of IICA control register 01 (IICCTL01) (1/2)

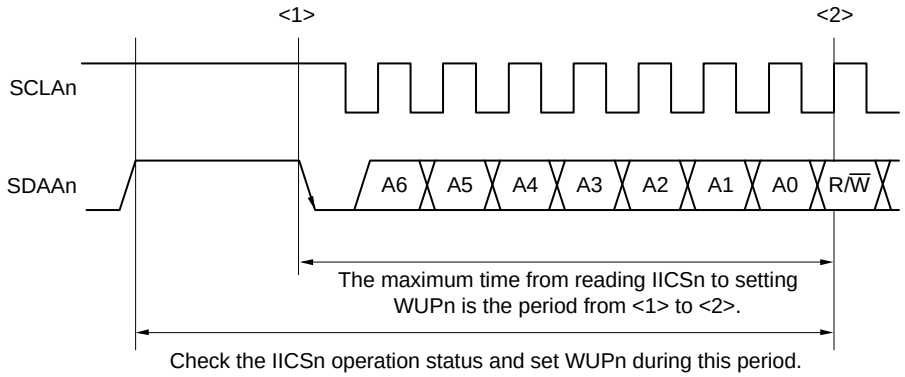
Address: F0231H After reset: 00H R/W Note 1

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
IICCTL01	WUPn	0	CLDn	DADn	SMCn	DFCn	0	PRSn

WUPn	Control of address match wakeup
0	Stops operation of address match wakeup function in STOP mode.
1	Enables operation of address match wakeup function in STOP mode.
To shift to STOP mode when WUPn = 1, execute the STOP instruction at least three clocks of f_{MCK} after setting (1) the WUPn bit (see **Figure 16 - 29 Flow When Setting WUPn = 1**). Clear (0) the WUPn bit after the address has matched or an extension code has been received. The subsequent communication can be entered by the clearing (0) WUPn bit. (The clock stretch must be released and transmit data must be written after the WUPn bit has been cleared (0).) The interrupt timing when the address has matched or when an extension code has been received, while WUPn = 1, is identical to the interrupt timing when WUPn = 0. (A delay of the difference of sampling by the clock will occur.) Furthermore, when WUPn = 1, a stop condition interrupt is not generated even if the SPIEn bit is set to 1.	
Condition for clearing (WUPn = 0)	Condition for setting (WUPn = 1)
• Cleared by instruction (after address match or extension code reception)	• Set by instruction (when the MSTSn, EXCn, and COIn bits are “0”, and the STDn bit also “0” (communication not entered)) Note 2

Note 1. Bits 4 and 5 are read-only.

Note 2. The status of the IICA status register n (IICSn) must be checked and the WUPn bit must be set during the period shown below.



Remark n = 0

Figure 16 - 15 Format of IICA control register 01 (IICCTL01) (2/2)

CLDn	Detection of SCLAn pin level (valid only when IICEn = 1)
0	The SCLAn pin was detected at low level.
1	The SCLAn pin was detected at high level.

Condition for clearing (CLDn = 0)	Condition for setting (CLDn = 1)
• When the SCLAn pin is at low level • When IICEn = 0 (operation stop) • Reset	• When the SCLAn pin is at high level

DADn	Detection of SDAAn pin level (valid only when IICEn = 1)
0	The SDAAn pin was detected at low level.
1	The SDAAn pin was detected at high level.

Condition for clearing (DADn = 0)	Condition for setting (DADn = 1)
• When the SDAAn pin is at low level • When IICEn = 0 (operation stop) • Reset	• When the SDAAn pin is at high level

SMCn	Operation mode switching
0	Operates in standard mode (fastest transfer rate: 100 kbps).
1	Operates in fast mode (fastest transfer rate: 400 kbps) or fast mode plus (fastest transfer rate: 1 Mbps).

DFCn	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
Use the digital filter only in fast mode and fast mode plus. The digital filter is used for noise elimination. The transfer clock does not vary, regardless of the DFCn bit being set (1) or cleared (0).	

PRSn	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})

Caution 1. The fastest operation frequency of the IICA operation clock (f_{MCK}) is 20 MHz (Max.).
Set bit 0 (PRSn) of the IICA control register n1 (IICCTLn1) to "1" only when the f_{CLK} exceeds 20 MHz.

Caution 2. Note the minimum f_{CLK} operation frequency when setting the transfer clock.
The minimum f_{CLK} operation frequency for serial interface IICA is determined according to the mode.
Fast mode: f_{CLK} = 3.5 MHz (MIN.)
Fast mode plus: f_{CLK} = 10 MHz (MIN.)
Normal mode: f_{CLK} = 1 MHz (MIN.)

Caution 3. The fast mode plus is only available in the products for "A: Consumer applications (T_A = -40°C to +85°C)".

Remark 1. IICEn: Bit 7 of IICA control register n0 (IICCTLn0)
Remark 2. n = 0

16.3.6 IICA low-level width setting register 0 (IICWL0)

This register is used to set the low-level width (tLOW) of the SCLAn pin signal that is output by serial interface IICA and to control the SDAAn pin signal.

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

Set the IICWLn register while operation of I^2C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) is 0).

Reset signal generation sets this register to FFH.

For details about setting the IICWLn register, see **16.4.2 Setting transfer clock by using IICWLn and IICWHn registers**.

The data hold time is one-quarter of the time set by the IICWLn register.

Figure 16 - 16 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								

16.3.7 IICA high-level width setting register 0 (IICWH0)

This register is used to set the high-level width of the SCLAn pin signal that is output by serial interface IICA and to control the SDAAn pin signal.

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

Set the IICWHn register while operation of I^2C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) is 0).

Reset signal generation sets this register to FFH.

Figure 16 - 17 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 1. For setting procedures of the transfer clock on master side and of the IICWLn and IICWHn registers on slave side, see **16.4.2 (1)** and **16.4.2 (2)**, respectively.

Remark 2. n = 0

16.3.8 Port mode register 6 (PM6)

This register sets the input/output of port 6 in 1-bit units.

When using the P60/SCLA0 pin as clock I/O and the P61/SDAA0 pin as serial data I/O, clear PM60 and PM61, and the output latches of P60 and P61 to 0.

Set the IICEn bit (bit 7 of IICA control register n0 (IICCTLn0)) to 1 before setting the output mode because the P60/SCLA0 and P61/SDAA0 pins output a low level (fixed) when the IICEn bit is 0.

The PM6 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 16 - 18 Format of Port mode register 6 (PM6)

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	1	1	1	1	1	PM61	PM60

PM6n	P6n pin I/O mode selection (n = 0, 1)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

16.4 I²C Bus Mode Functions

16.4.1 Pin configuration

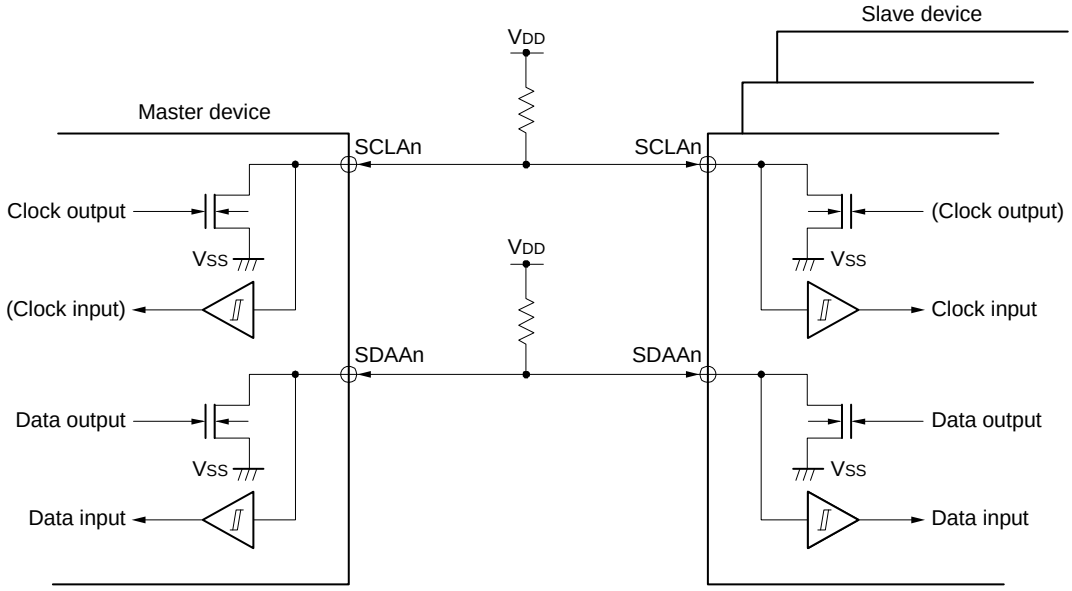
The serial clock pin (SCLAn) and the serial data bus pin (SDAAn) are configured as follows.

(1) SCLAn..... This pin is used for serial clock input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

(2) SDAAn ..... This pin is used for serial data input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

Figure 16 - 19 Pin Configuration Diagram



Remark n = 0

16.4.2 Setting transfer clock by using IICWLn and IICWHn registers

(1) Setting transfer clock on master side

$$\text{Transfer clock} = \frac{f_{MCK}}{IICWL + IICWH + f_{MCK}(t_R + t_F)}$$

At this time, the optimal setting values of the IICWLn and IICWHn registers are as follows.
(The fractional parts of all setting values are rounded up.)

• When the fast mode

$$IICWLn = \frac{0.52}{\text{Transfer clock}} \times f_{MCK}$$

$$IICWHn = \left(\frac{0.48}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

• When the normal mode

$$IICWLn = \frac{0.47}{\text{Transfer clock}} \times f_{MCK}$$

$$IICWHn = \left(\frac{0.53}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

• When the fast mode plus

$$IICWLn = \frac{0.50}{\text{Transfer clock}} \times f_{MCK}$$

$$IICWHn = \left(\frac{0.50}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

(2) Setting IICWLn and IICWHn registers on slave side
(The fractional parts of all setting values are truncated.)

• When the fast mode

$IICWLn = 1.3\ \mu s \times f_{MCK}$

$IICWHn = (1.2\ \mu s - t_R - t_F) \times f_{MCK}$

• When the normal mode

$IICWLn = 4.7\ \mu s \times f_{MCK}$

$IICWHn = (5.3\ \mu s - t_R - t_F) \times f_{MCK}$

• When the fast mode plus

$IICWLn = 0.50\ \mu s \times f_{MCK}$

$IICWHn = (0.50\ \mu s - t_R - t_F) \times f_{MCK}$

Caution 1. The fastest operation frequency of the IICA operation clock (f_{MCK}) is 20 MHz (Max.).
Set bit 0 (PRSn) of the IICA control register n1 (IICCTLn1) to "1" only when the f_{CLK} exceeds 20 MHz.

Caution 2. Note the minimum f_{CLK} operation frequency when setting the transfer clock.
The minimum f_{CLK} operation frequency for serial interface IICA is determined according to the mode.
Fast mode: f_{CLK} = 3.5 MHz (MIN.)
Fast mode plus: f_{CLK} = 10 MHz (MIN.)
Normal mode: f_{CLK} = 1 MHz (MIN.)

(**Remarks** are listed on the next page.)

Remark 1. Calculate the rise time (t_R) and fall time (t_F) of the SDAAn and SCLAn signals separately, because they differ depending on the pull-up resistance and wire load.

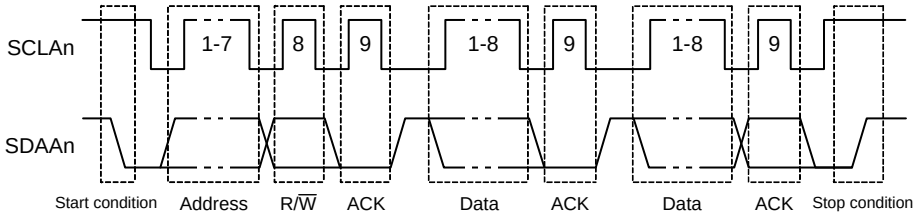
Remark 2. IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
t_F: SDAAn and SCLAn signal falling times
t_R: SDAAn and SCLAn signal rising times
f_{MCK}: IICA operation clock frequency

Remark 3. n = 0

16.5 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 16 - 20 shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I²C bus's serial data bus.

Figure 16 - 20 I²C Bus Serial Data Transfer Timing

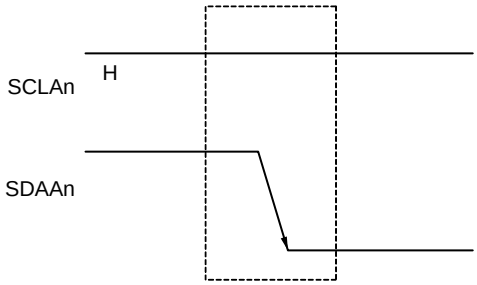


The master device generates the start condition, slave address, and stop condition.
The acknowledge (ACK) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).
The serial clock (SCLAn) is continuously output by the master device. However, in the slave device, the SCLAn pin low level period can be extended and a clock stretch can be inserted.

16.5.1 Start conditions

A start condition is met when the SCLAn pin is at high level and the SDAAn pin changes from high level to low level. The start conditions for the SCLAn pin and SDAAn 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 16 - 21 Start Conditions



A start condition is output when bit 1 (STTn) of IICA control register n0 (IICCTLn0) is set (1) after a stop condition has been detected (SPDn: Bit 0 of the IICA status register n (IICSn) = 1). When a start condition is detected, bit 1 (STDn) of the IICSn register is set (1).

Remark n = 0

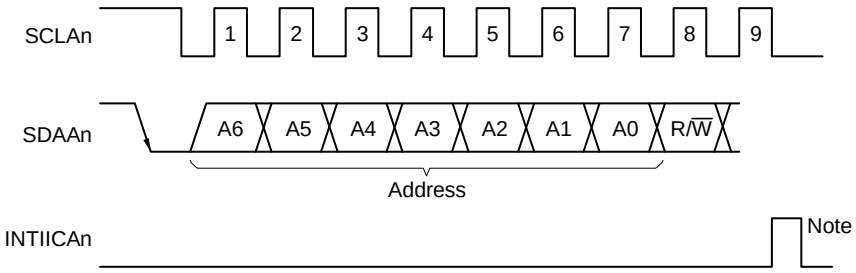
16.5.2 Addresses

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 n (SVAn). If the address data matches the SVAn register values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

Figure 16 - 22 Address



Note INTIICAn 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 **16.5.3 Transfer direction specification** are written to the IICA shift register n (IICAn). The received addresses are written to the IICAn register.

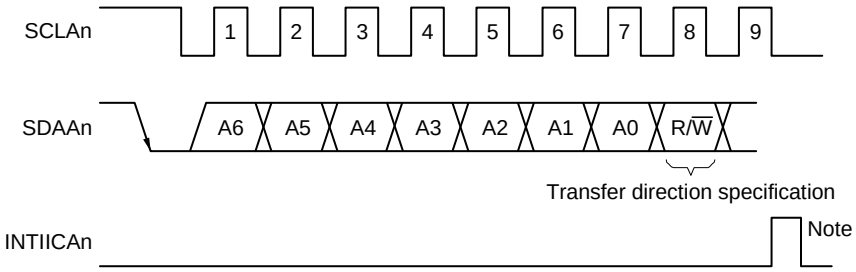
The slave address is assigned to the higher 7 bits of the IICAn register.

16.5.3 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 16 - 23 Transfer Direction Specification



Note INTIICAn is not issued if data other than a local address or extension code is received during slave device operation.

Remark n = 0

16.5.4 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 (ACKDn) of the IICA status register n (IICSn).

When the master receives the last data item, it does not return ACK and instead generates a stop condition. If a slave does not return ACK after receiving data, the master outputs a stop condition or restart condition and stops transmission. If ACK is not returned, the possible causes are as follows.

<1> Reception was not performed normally.
<2> The final data item was received.
<3> The reception side specified by the address does not exist.

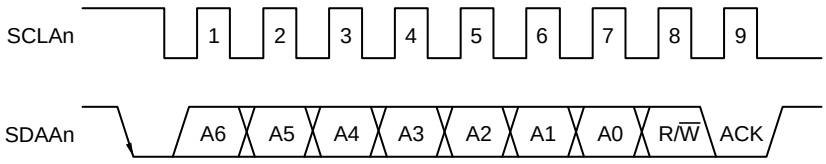
To generate ACK, the reception side makes the SDAAn line low at the ninth clock (indicating normal reception).

Automatic generation of ACK is enabled by setting bit 2 (ACKEn) of IICA control register n0 (IICCTLn0) to 1. Bit 3 (TRCn) of the IICSn register is set by the data of the eighth bit that follows 7-bit address information. Usually, set the ACKEn bit to 1 for reception (TRCn = 0).

If a slave can receive no more data during reception (TRCn = 0) or does not require the next data item, then the slave must inform the master, by clearing the ACKEn bit to 0, that it will not receive any more data.

When the master does not require the next data item during reception (TRCn = 0), it must clear the ACKEn bit to 0 so that ACK is not generated. In this way, the master informs a slave at the transmission side that it does not require any more data (transmission will be stopped).

Figure 16 - 24 ACK



When the local address is received, ACK is automatically generated, regardless of the value of the ACKEn bit.

When an address other than that of the local address is received, ACK is not generated (NACK).

When an extension code is received, ACK is generated if the ACKEn 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 stretch timing.

- When 8-clock clock stretch state is selected (bit 3 (WTIMn) of IICCTLn0 register = 0):
  By setting the ACKEn bit to 1 before releasing the clock stretch state, ACK is generated at the falling edge of the eighth clock of the SCLAn pin.
- When 9-clock clock stretch state is selected (bit 3 (WTIMn) of IICCTLn0 register = 1):
  ACK is generated by setting the ACKEn bit to 1 in advance.

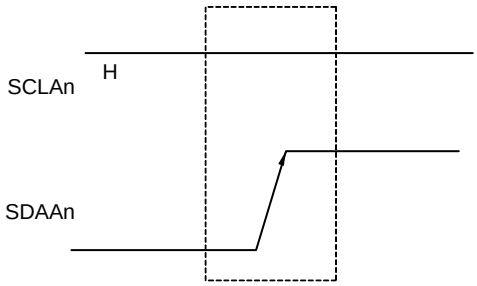
Remark n = 0

16.5.5 Stop condition

When the SCLAn pin is at high level, changing the SDAAn pin from low level to high level generates a stop condition.

A stop condition is a signal that the master device generates to the slave device when serial transfer has been completed. When the device is used as a slave, stop conditions can be detected.

Figure 16 - 25 Stop Condition



A stop condition is generated when bit 0 (SPTn) of IICA control register n0 (IICCTLn0) is set to 1. When the stop condition is detected, bit 0 (SPDn) of the IICA status register n (IICSn) is set to 1 and INTIICAn is generated when bit 4 (SPIEn) of the IICCTLn0 register is set to 1.

Remark n = 0

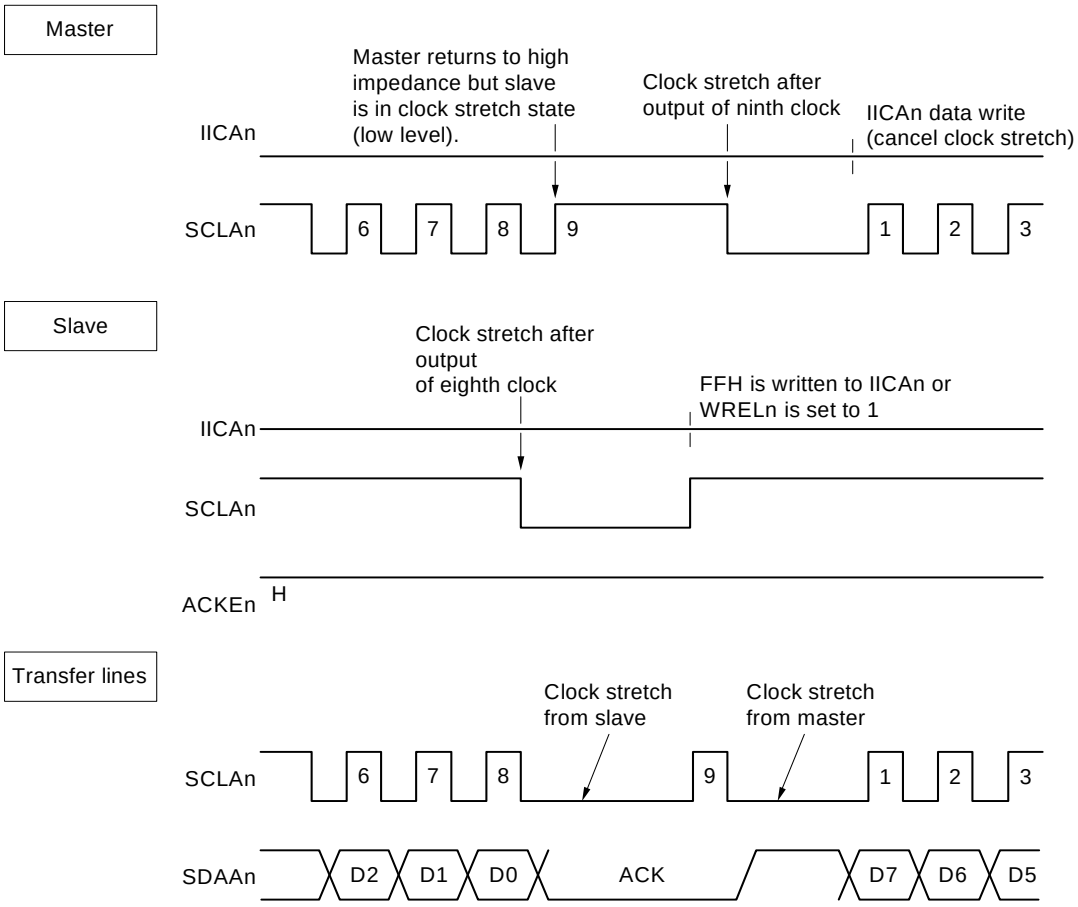
16.5.6 Clock stretch

The clock stretch 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 SCLAn 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 16 - 26 Clock stretch (1/2)

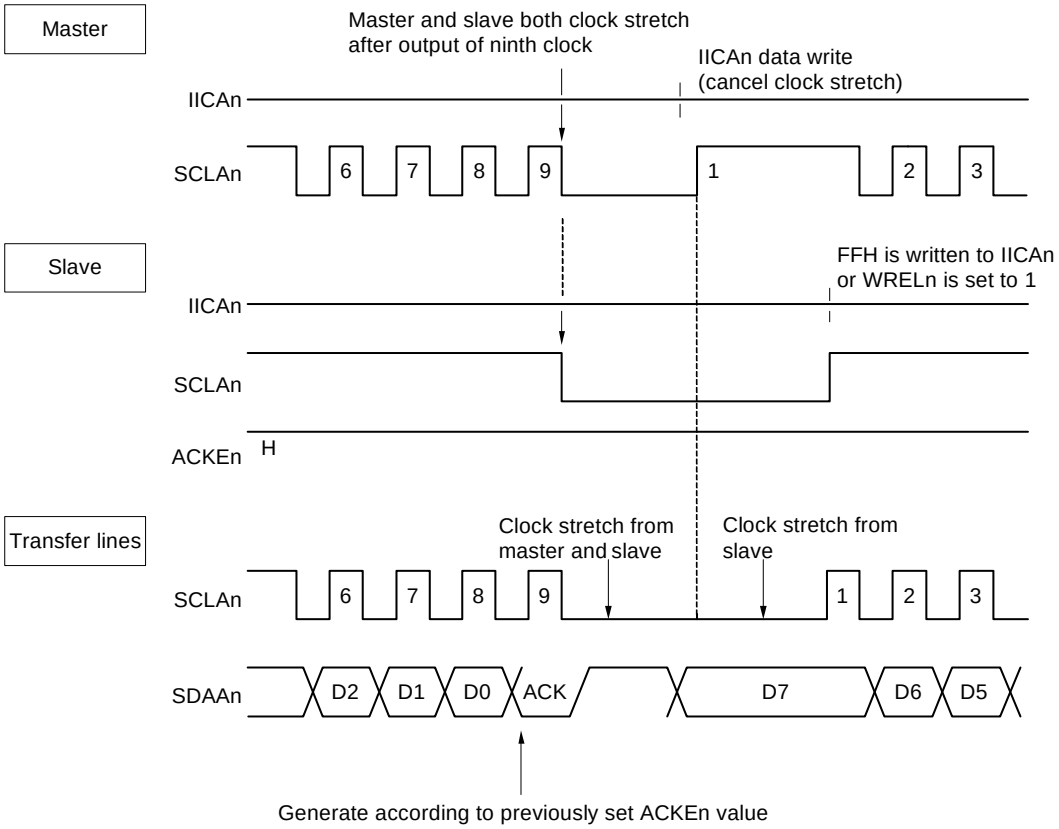
(1) When master device has a nine-clock clock stretch and slave device has an eight-clock clock stretch (master transmits, slave receives, and ACKEn = 1)



Remark n = 0

Figure 16 - 27 Clock stretch (2/2)

(2) When master and slave devices both have a nine-clock clock stretch
(master transmits, slave receives, and ACKEn = 1)



Remark ACKEn: Bit 2 of IICA control register n0 (IICCTLn0)
WRELn: Bit 5 of IICA control register n0 (IICCTLn0)

A clock stretch may be automatically generated depending on the setting of bit 3 (WTIMn) of IICA control register n0 (IICCTLn0).
Normally, the receiving side cancels the clock stretch state when bit 5 (WRELn) of the IICCTLn0 register is set to 1 or when FFH is written to the IICA shift register n (IICAn), and the transmitting side cancels the clock stretch state when data is written to the IICAn register.
The master device can also cancel the clock stretch state via either of the following methods.

- By setting bit 1 (STTn) of the IICCTLn0 register to 1
- By setting bit 0 (SPTn) of the IICCTLn0 register to 1

Remark n = 0

16.5.7 Canceling clock stretch

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 (WRELn) of IICA control register 00 (IICCTL00) (canceling clock stretch)
- Setting bit 1 (STTn) of the IICCTLn0 register (generating start condition) Note
- Setting bit 0 (SPTn) of the IICCTLn0 register (generating stop condition) Note

Note Master only

When the above clock stretch 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 IICAn register.
To receive data after canceling a clock stretch state, or to complete data transmission, set bit 5 (WRELn) of the IICCTLn0 register to 1.
To generate a restart condition after canceling a clock stretch state, set bit 1 (STTn) of the IICCTLn0 register to 1.
To generate a stop condition after canceling a clock stretch state, set bit 0 (SPTn) of the IICCTLn0 register to 1.
Execute the canceling processing only once for one clock stretch state.
If, for example, data is written to the IICAn register after canceling a clock stretch state by setting the WRELn bit to 1, an incorrect value may be output to SDAAn line because the timing for changing the SDAAn line conflicts with the timing for writing the IICAn register.
In addition to the above, communication is stopped if the IICEn 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 (LRELn) of the IICCTLn0 register, so that the clock stretch state can be canceled.

Caution **If a processing to cancel a clock stretch state is executed when WUPn = 1, the clock stretch state will not be canceled.**

Remark n = 0

16.5.8 Interrupt request (INTIICAn) generation timing and clock stretch control

The setting of bit 3 (WTIMn) of IICA control register n0 (IICCTLn0) determines the timing by which INTIICAn is generated and the corresponding clock stretch control, as shown in Table 16 - 2.

Table 16 - 2 INTIICAn Generation Timing and Clock Stretch Control

WTIMn	During Slave Device Operation			During Master Device Operation		
	Address	Data Reception	Data Transmission	Address	Data Reception	Data Transmission
0	9 Notes 1, 2	8 Note 2	8 Note 2	9	8	8
1	9 Notes 1, 2	9 Note 2	9 Note 2	9	9	9

Note 1. The slave device's INTIICAn signal and clock 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 n (SVAn).
At this point, ACK is generated regardless of the value set to the IICCTLn0 register's bit 2 (ACKEn). For a slave device that has received an extension code, INTIICAn occurs at the falling edge of the eighth clock.
However, if the address does not match after restart, INTIICAn is generated at the falling edge of the 9th clock, but clock stretch does not occur.

Note 2. If the received address does not match the contents of the slave address register n (SVAn) and extension code is not received, neither INTIICAn nor a clock stretch occurs.

Remark The numbers in the table indicate the number of the serial clock's clock signals. Interrupt requests and clock stretch control are both synchronized with the falling edge of these clock signals.

(1) During address transmission/reception
- Slave device operation: Interrupt and clock stretch timing are determined depending on the conditions described in Notes 1 and 2 above, regardless of the WTIMn bit.
- Master device operation: Interrupt and clock stretch timing occur at the falling edge of the ninth clock regardless of the WTIMn bit.

(2) During data reception
- Master/slave device operation: Interrupt and clock stretch timing are determined according to the WTIMn bit.

(3) During data transmission
- Master/slave device operation: Interrupt and clock stretch timing are determined according to the WTIMn bit.

Remark n = 0

(4) Clock stretch cancellation method

The four clock stretch cancellation methods are as follows.

- Writing data to the IICA shift register 0 (IICA0)
- Setting bit 5 (WRELn) of IICA control register 00 (IICCTL00) (canceling clock stretch)
- Setting bit 1 (STTn) of IICCTLn0 register (generating start condition) Note
- Setting bit 0 (SPTn) of IICCTLn0 register (generating stop condition) Note

Note Master only.

When an 8-clock clock stretch has been selected (WTIMn = 0), the presence/absence of ACK generation must be determined prior to clock stretch cancellation.

(5) Stop condition detection

INTIICAn is generated when a stop condition is detected (only when SPIEn = 1).

16.5.9 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 (INTIICAn) occurs when the address set to the slave address register n (SVAn) matches the slave address sent by the master device, or when an extension code has been received.

16.5.10 Error detection

In I^2C bus mode, the status of the serial data bus (SDAAn) during data transmission is captured by the IICA shift register n (IICAn) of the transmitting device, so the IICA data prior to transmission can be compared with the transmitted IICA data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

Remark n = 0

16.5.11 Extension code

(1) When the higher 4 bits of the receive address are either “0000” or “1111”, the extension code reception flag (EXCn) is set to 1 for extension code reception and an interrupt request (INTIICAn) is issued at the falling edge of the eighth clock. The local address stored in the slave address register n (SVAn) is not affected.

(2) The settings below are specified if 11110xx0 is transferred from the master by using a 10-bit address transfer when the SVAn register is set to 11110xx0. Note that INTIICAn occurs at the falling edge of the eighth clock.

- Higher four bits of data match: EXCn = 1
- Seven bits of data match: COIn = 1

Remark EXCn: Bit 5 of IICA status register n (IICSn)
COIn: Bit 4 of IICA status register n (IICSn)

(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 (LRELn) of IICA control register n0 (IICCTLn0) to 1 to set the standby mode for the next communication operation.

Table 16 - 3 Bit Definitions of Major Extension Codes

Slave Address	R/W Bit	Description
0000 000	0	General call address
1111 0xx	0	10-bit slave address specification (during address authentication)
1111 0xx	1	10-bit slave address specification (after address match, when read command is issued)

Remark 1. See the I^2C bus specifications issued by NXP Semiconductors for details of extension codes other than those described above.

Remark 2. n = 0

16.5.12 Arbitration

When several master devices simultaneously generate a start condition (when the STTn bit is set to 1 before the STDn 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.

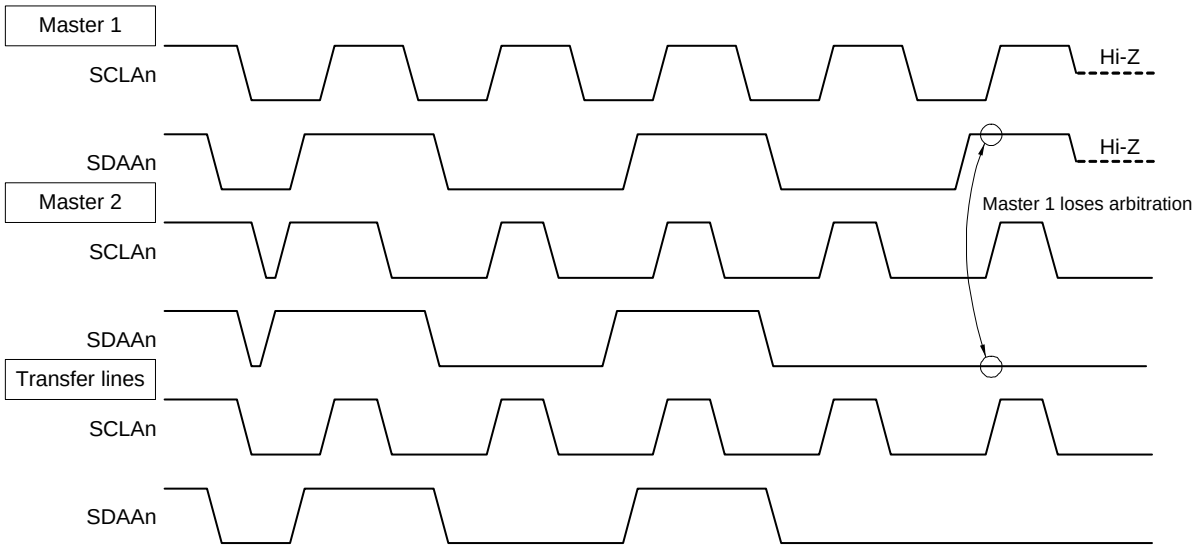
When one of the master devices loses in arbitration, an arbitration loss flag (ALDn) in the IICA status register n (IICSn) is set (1) via the timing by which the arbitration loss occurred, and the SCLAn and SDAAn lines are both set to high impedance, which releases the bus.

The arbitration loss is detected based on the timing of the next interrupt request (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALDn = 1 setting that has been made by software.

For details of interrupt request timing, see **16.5.8 Interrupt request (INTIICAn) generation timing and clock stretch control**.

Remark STDn: Bit 1 of IICA status register n (IICSn)
STTn: Bit 1 of IICA control register n0 (IICCTLn0)

Figure 16 - 28 Arbitration Timing Example



Remark n = 0

Table 16 - 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 SPIEn = 1) Note 2
When data is at low level while attempting to generate a restart condition	At falling edge of eighth or ninth clock following byte transfer Note 1
When stop condition is detected while attempting to generate a restart condition	When stop condition is generated (when SPIEn = 1) Note 2
When data is at low level while attempting to generate a stop condition	At falling edge of eighth or ninth clock following byte transfer Note 1
When SCLAn is at low level while attempting to generate a restart condition	

Note 1. When the WTIMn bit (bit 3 of IICA control register n0 (IICCTLn0)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIMn = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock.

Note 2. When there is a chance that arbitration will occur, set SPIEn = 1 for master device operation.

Remark 1. SPIEn: Bit 4 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0

16.5.13 Wakeup function

The I^2C bus slave function is a function that generates an interrupt request signal (INTIICAn) when a local address and extension code have been received.

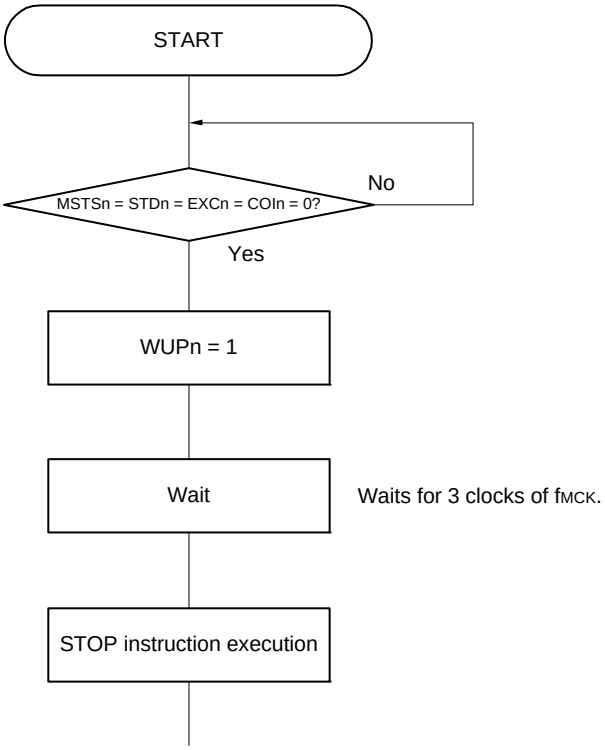
This function makes processing more efficient by preventing unnecessary INTIICAn signal from occurring when addresses do not match.

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 WUPn bit to 1. Addresses can be received regardless of the operation clock. An interrupt request signal (INTIICAn) 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 WUPn bit after this interrupt has been generated.

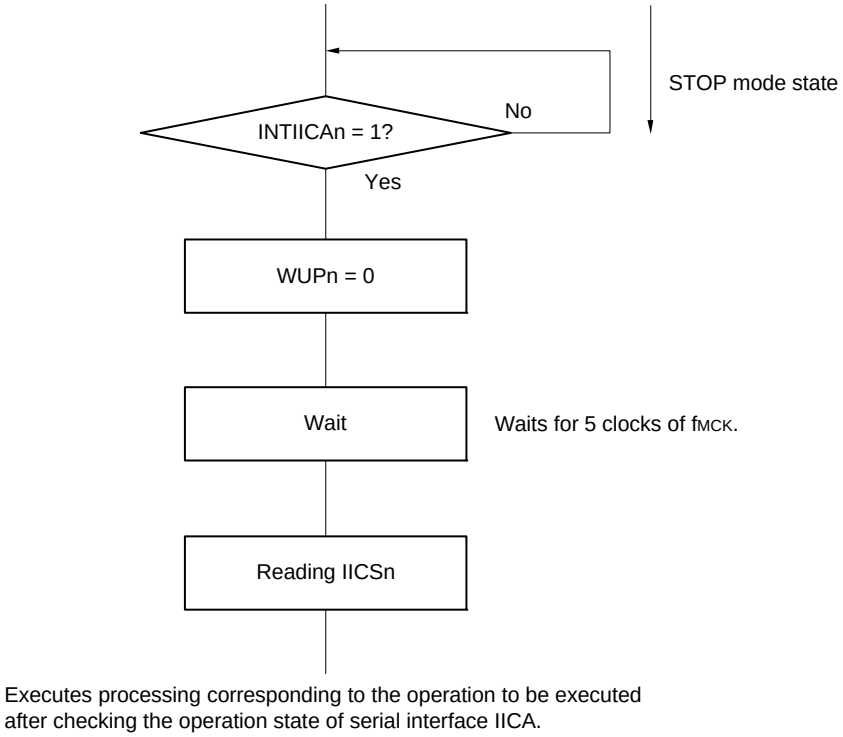
Figure 16 - 29 shows the flow for setting WUPn = 1 and Figure 16 - 30 shows the flow for setting WUPn = 0 upon an address match.

Figure 16 - 29 Flow When Setting WUPn = 1



Remark n = 0

Figure 16 - 30 Flow When Setting WUPn = 0 upon Address Match (Including Extension Code Reception)

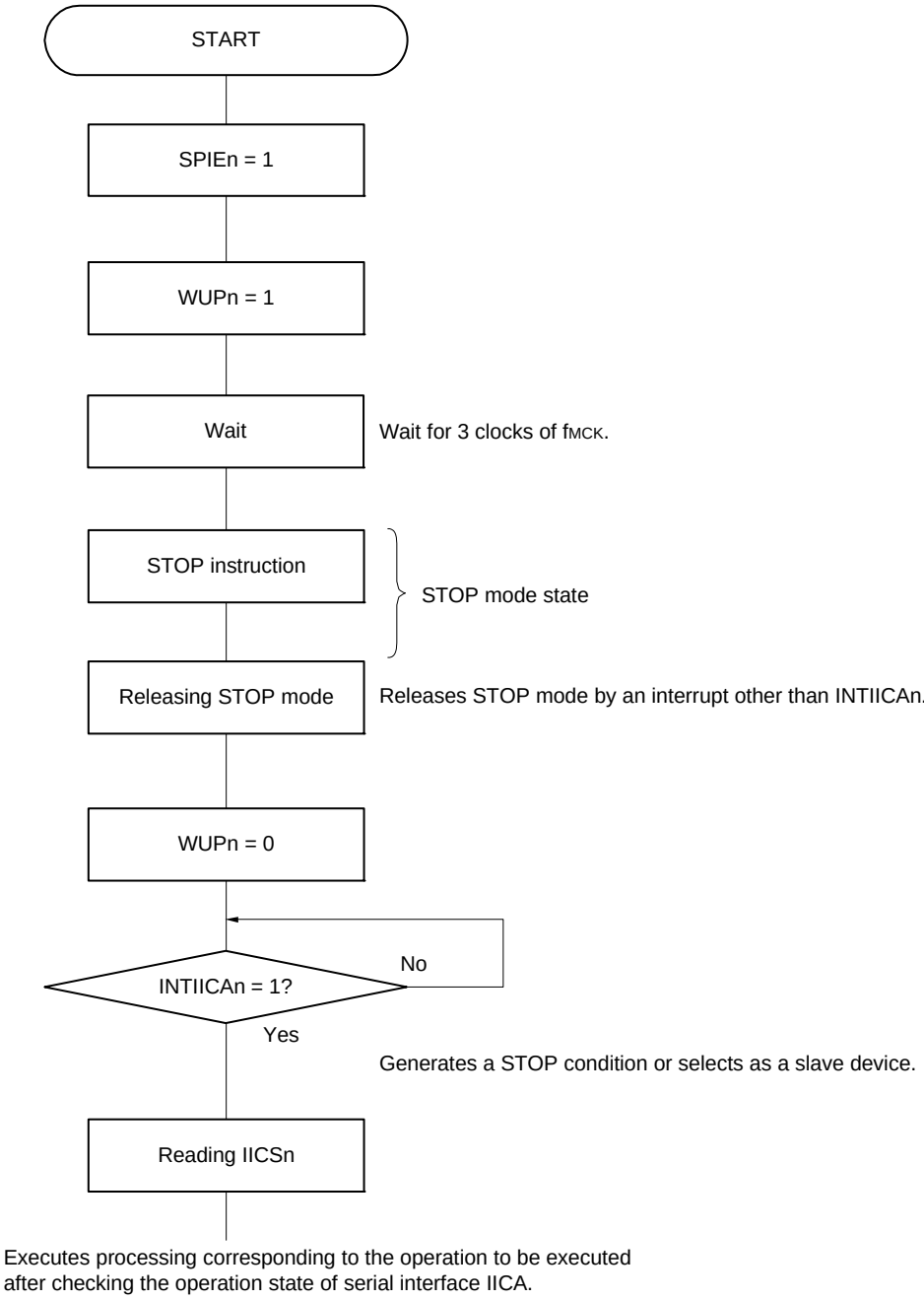


Use the following flows to perform the processing to release the STOP mode other than by an interrupt request (INTIICAn) generated from serial interface IICA.

- When operating as the master device for the next IIC communication: Flow shown in Figure 16 - 31
- When operating as a slave device for the next IIC communication:
  - When the INTIICAn interrupt is used to return from the mode:
    - Same as the flow in Figure 16 - 30
  - When an interrupt other than the INTIICAn interrupt is used to return from the mode:
    - Continue operation while WUPn = 1 until an INTIICAn interrupt is generated.

Remark n = 0

Figure 16 - 31 When Operating as Master Device after Releasing STOP Mode other than by INTIICAn



Remark n = 0

16.5.14 Communication reservation

(1) When communication reservation function is enabled (bit 0 (IICRSVn) of IICA flag register n (IICFn) = 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 (LRELn) of IICA control register n0 (IICCTLn0) to 1 and saving communication).

If bit 1 (STTn) of the IICCTLn0 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 n (IICAn) after bit 4 (SPIEn) of the IICCTLn0 register was set to 1, and it was detected by generation of an interrupt request signal (INTIICAn) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to the IICAn register before the stop condition is detected is invalid.
When the STTn bit has been set to 1, the operation mode (as start condition or as communication reservation) is determined according to the 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 MSTSn bit (bit 7 of the IICA status register n (IICSn)) after the STTn bit is set to 1 and the wait time elapses.
Use software to secure the wait time calculated by the following expression.

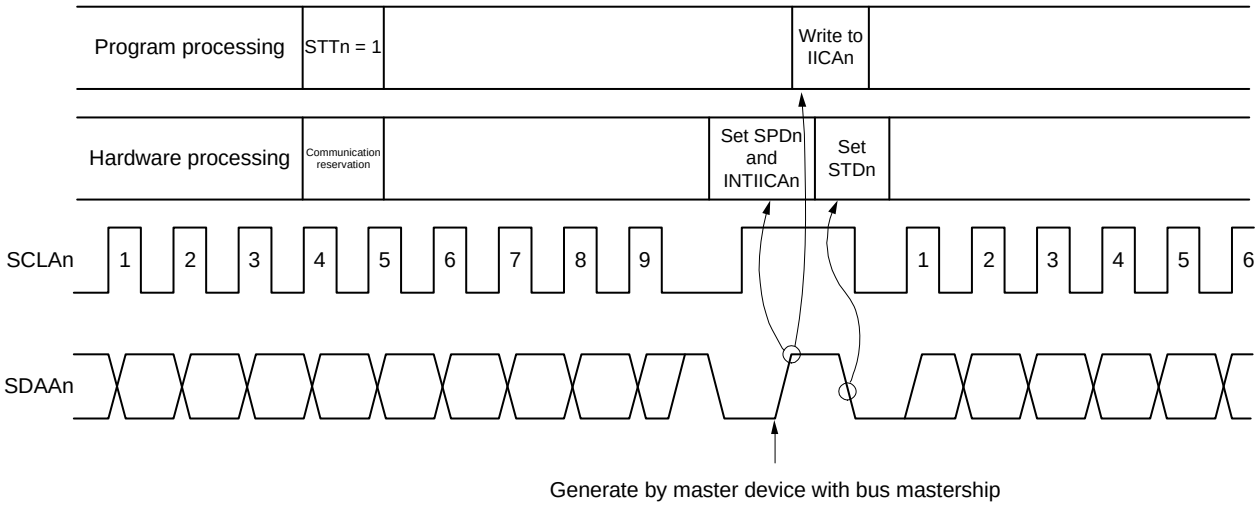
Wait time from setting STTn = 1 to checking the MSTSn flag: (IICWLn setting value + IICWHn setting value + 4) / f_{CLK} + t_F × 2

Remark 1. IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
t_F: SDAAn and SCLAn signal falling times
f_{MCK}: IICA operation clock frequency

Remark 2. n = 0

Figure 16 - 32 shows the Communication Reservation Timing.

Figure 16 - 32 Communication Reservation Timing



Remark IICAn: IICA shift register n
STTn: Bit 1 of IICA control register n0 (IICCTLn0)
STDn: Bit 1 of IICA status register n (IICSn)
SPDn: Bit 0 of IICA status register n (IICSn)

Communication reservations are accepted via the timing shown in Figure 16 - 33. After bit 1 (STDn) of the IICA status register n (IICSn) is set to 1, a communication reservation can be made by setting bit 1 (STTn) of IICA control register n0 (IICCTLn0) to 1 before a stop condition is detected.

Figure 16 - 33 Timing for Accepting Communication Reservations

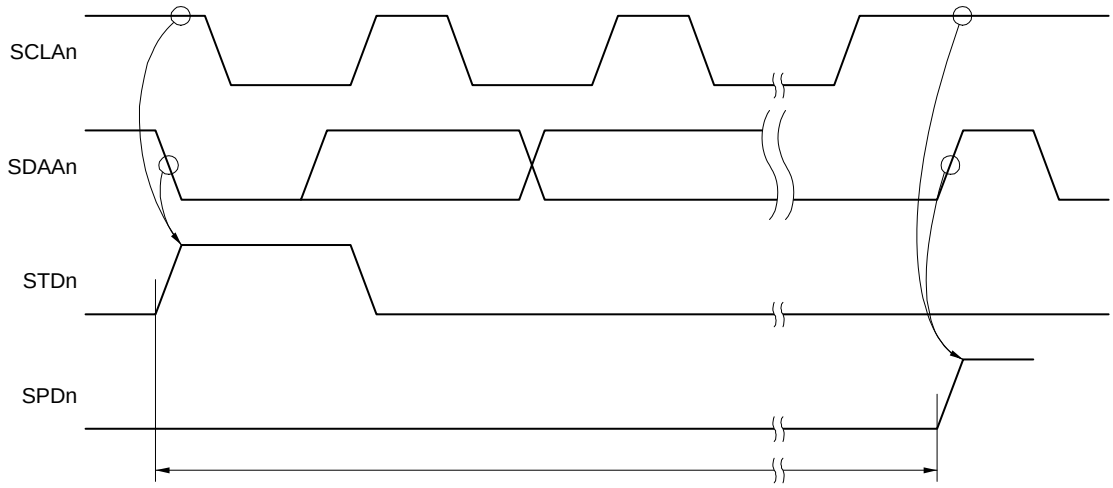
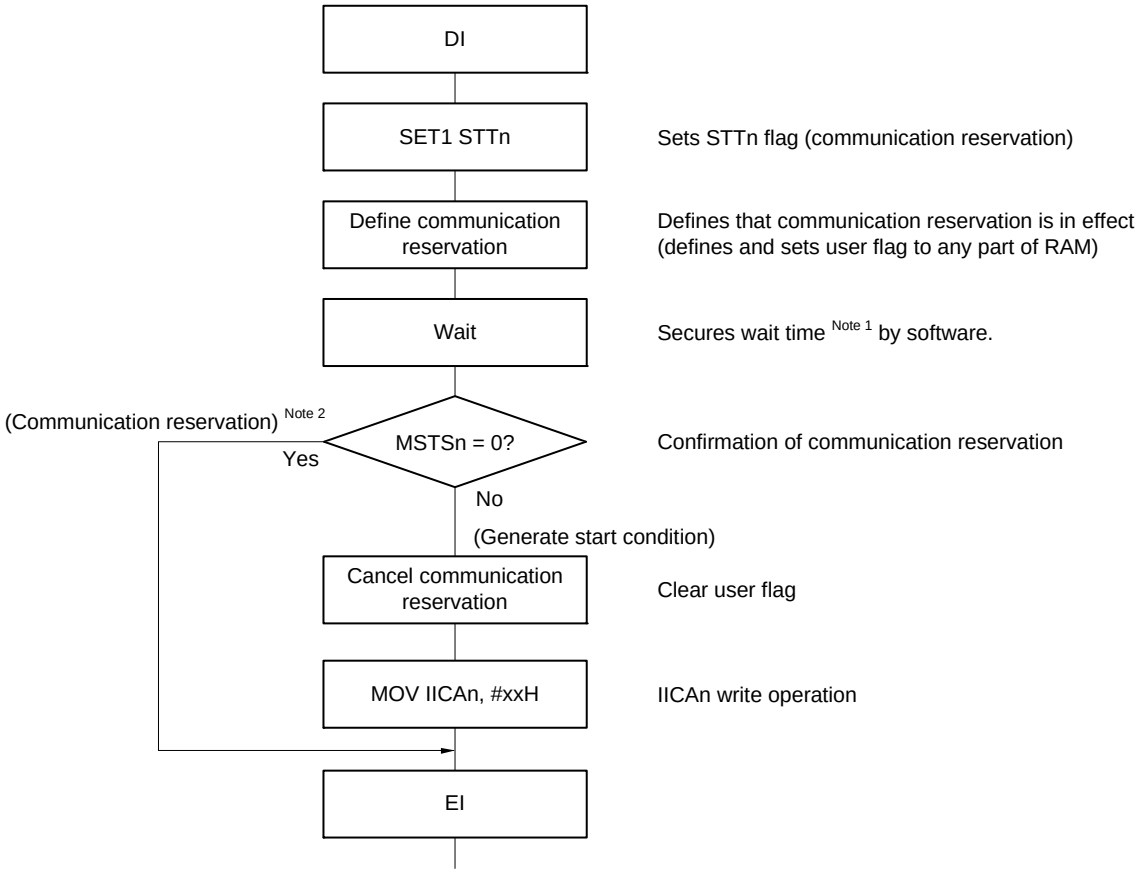


Figure 16 - 34 shows the Communication Reservation Protocol.

Remark n = 0

Figure 16 - 34 Communication Reservation Protocol



Note 1. The wait time is calculated as follows.
(IICWLn setting value + IICWHn setting value + 4) / f_{MCK} + t_F × 2

Note 2. The communication reservation operation executes a write to the IICA shift register n (IICAn) when a stop condition interrupt request occurs.

Remark1. STTn: Bit 1 of IICA control register n0 (IICCTLn0)
MSTSn: Bit 7 of IICA status register n (IICSn)
IICAn: IICA shift register n
IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
t_F: SDAAn and SCLAn signal falling times
f_{MCK}: IICA operation clock frequency

Remark2. n = 0

(2) When communication reservation function is disabled (bit 0 (IICRSVn) of IICA flag register n (IICFn) = 1)

When bit 1 (STTn) of IICA control register n0 (IICCTLn0) is set to 1 when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The 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 (LRELn) of the IICCTLn0 register to 1 and saving communication)

To confirm whether the start condition was generated or request was rejected, check STCFn (bit 7 of the IICFn register). It takes up to 5 clocks of f_{MCK} until the STCFn bit is set to 1 after setting STTn = 1. Therefore, secure the time by software.

Remark n = 0

16.5.15 Cautions

(1) When STCENn = 0

Immediately after I^2C operation is enabled (IICEn = 1), the bus communication status (IICBSYn = 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 n1 (IICCTLn1).
<2> Set bit 7 (IICEn) of IICA control register n0 (IICCTLn0) to 1.
<3> Set bit 0 (SPTn) of the IICCTLn0 register to 1.

(2) When STCENn = 1

Immediately after I^2C operation is enabled (IICEn = 1), the bus released status (IICBSYn = 0) is recognized regardless of the actual bus status. To generate the first start condition (STTn = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

(3) If other I^2C communications are already in progress

If I^2C operation is enabled and the device participates in communication already in progress when the SDAAn pin is low and the SCLAn pin is high, the macro of I^2C recognizes that the SDAAn pin has gone low (detects a start condition). If the value on the bus at this time can be recognized as an extension code, 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 (SPIEn) of the IICCTLn0 register to 0 to disable generation of an interrupt request signal (INTIICAn) when the stop condition is detected.
<2> Set bit 7 (IICEn) of the IICCTLn0 register to 1 to enable the operation of I^2C.
<3> Wait for detection of the start condition.
<4> Set bit 6 (LRELn) of the IICCTLn0 register to 1 before ACK is returned (4 to 80 clocks of f_{MCK} after setting the IICEn bit to 1), to forcibly disable detection.

(4) Setting the STTn and SPTn bits (bits 1 and 0 of the IICCTLn0 register) again after they are set and before they are cleared to 0 is prohibited.

(5) When transmission is reserved, set the SPIEn bit (bit 4 of the IICCTLn0 register) to 1 so that an interrupt request is generated when the stop condition is detected. Transfer is started when communication data is written to the IICA shift register n (IICAn) after the interrupt request is generated. Unless the interrupt is generated when the stop condition is detected, the device stops in the wait state because the interrupt request is not generated when communication is started. However, it is not necessary to set the SPIEn bit to 1 when the MSTSn bit (bit 7 of the IICA status register n (IICSn)) is detected by software.

Remark n = 0

16.5.16 Communication operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single master system

The flowchart when using the RL78/L1C 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/L1C 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/L1C 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/L1C is used as the I^2C bus slave is shown below.

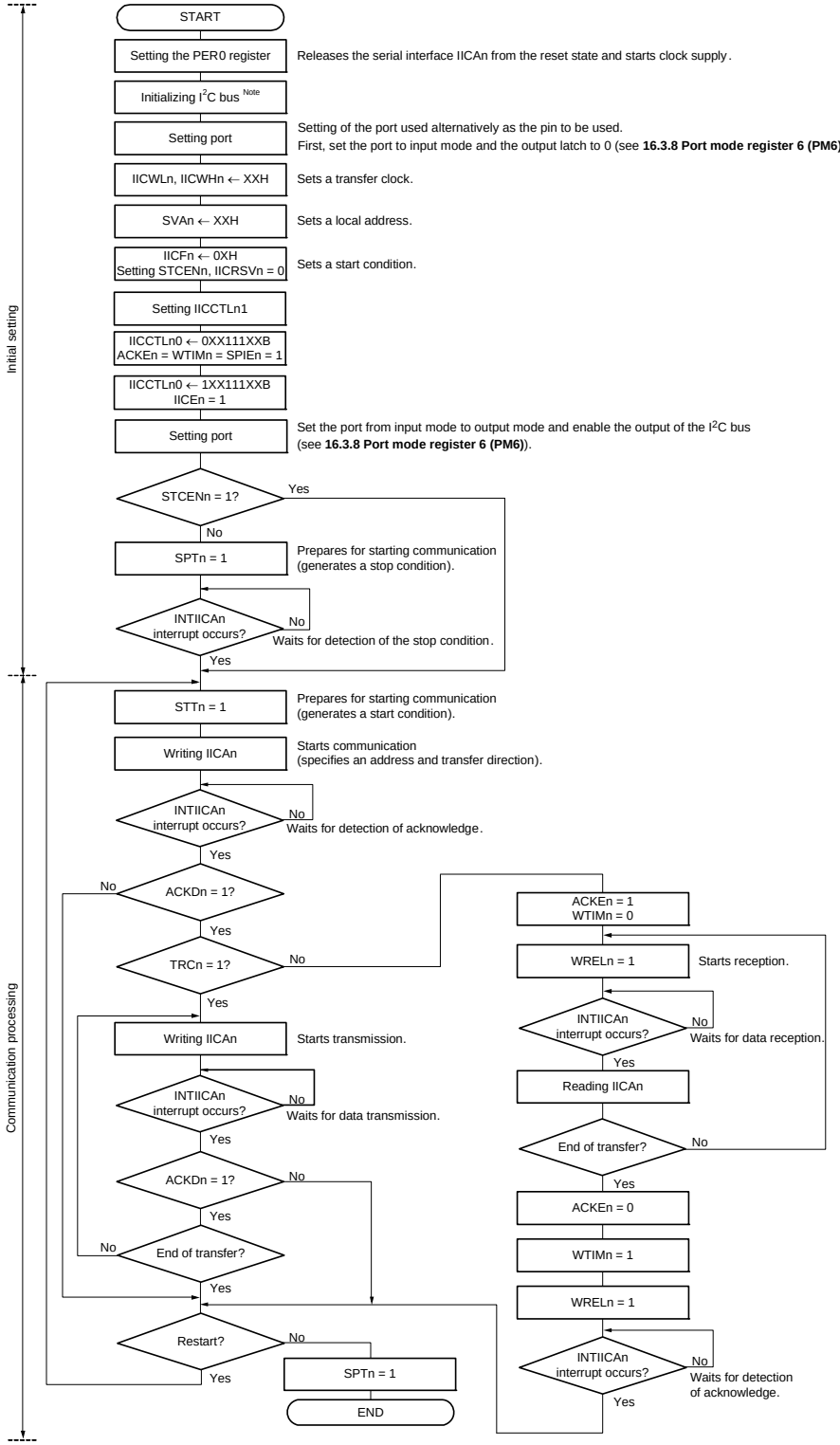
When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIICAn interrupt occurrence (communication waiting). When an INTIICAn interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.

By checking the flags, necessary communication processing is performed.

Remark n = 0

(1) Master operation in single master system

Figure 16 - 35 Master Operation in Single-Master System



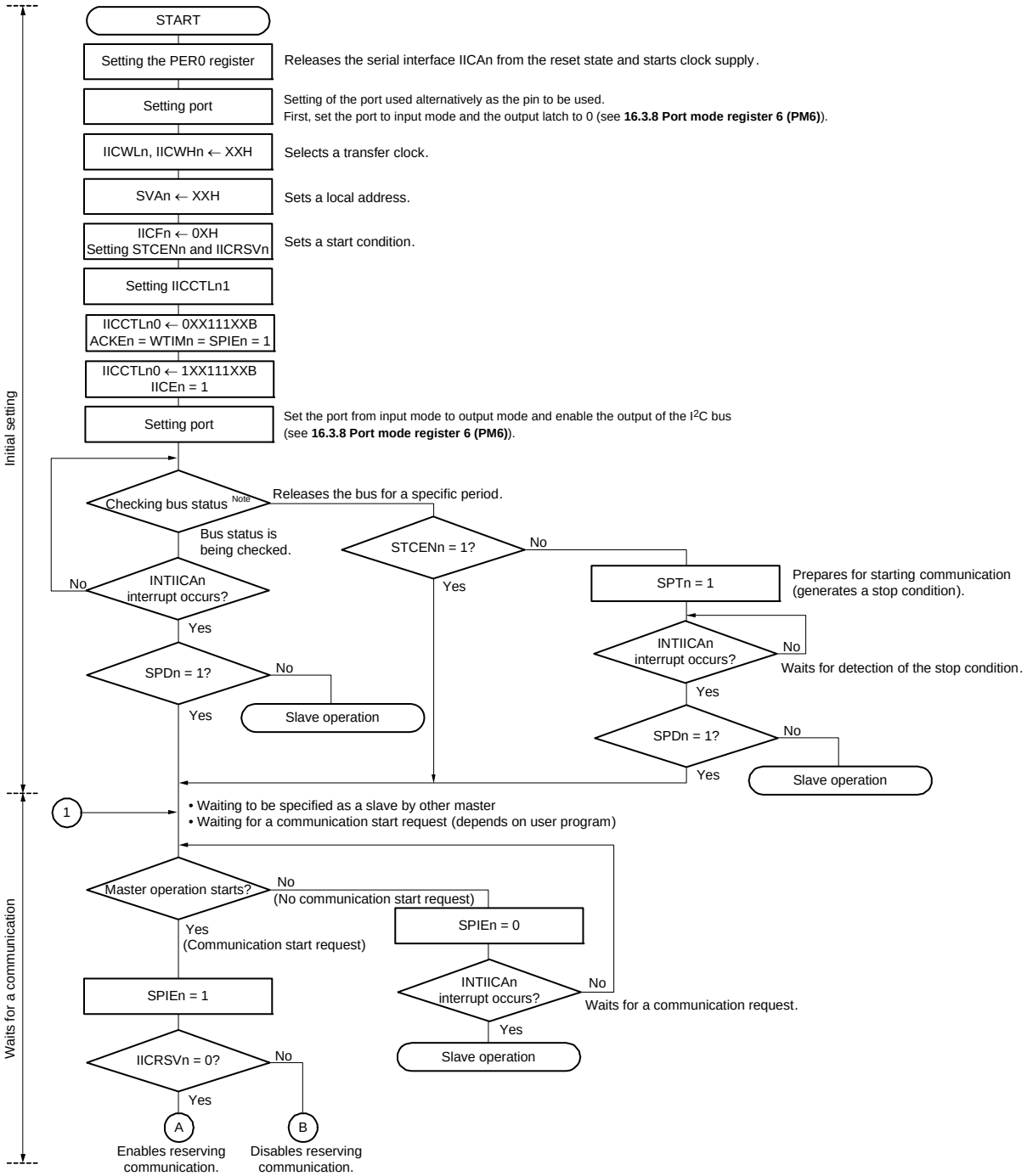
Note Release (SCLAn and SDAAn pins = high level) the I2C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAAn pin, for example, set the SCLAn pin in the output port mode, and output a clock pulse from the output port until the SDAAn pin is constantly at high level.

Remark1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

Remark2. n = 0

(2) Master operation in multimaster system

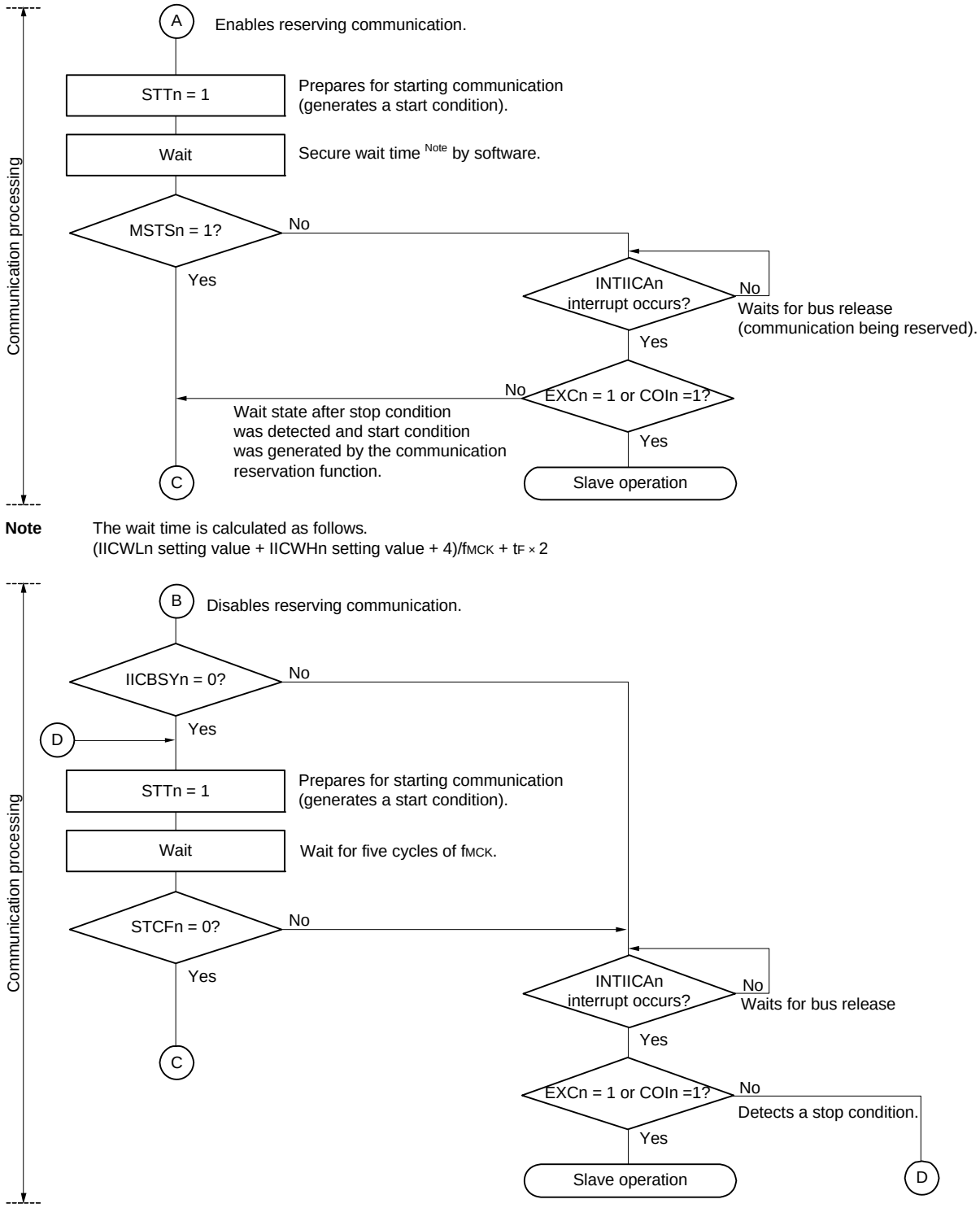
Figure 16 - 36 Master Operation in Multi-Master System (1/3)



Note Confirm that the bus is released (CLDn bit = 1, DADn bit = 1) for a specific period (for example, for a period of one frame). If the SDAAn pin is constantly at low level, decide whether to release the I^2C bus (SCLAn and SDAAn pins = high level) in conformance with the specifications of the product that is communicating.

Remark n = 0

Figure 16 - 37 Master Operation in Multi-Master System (2/3)

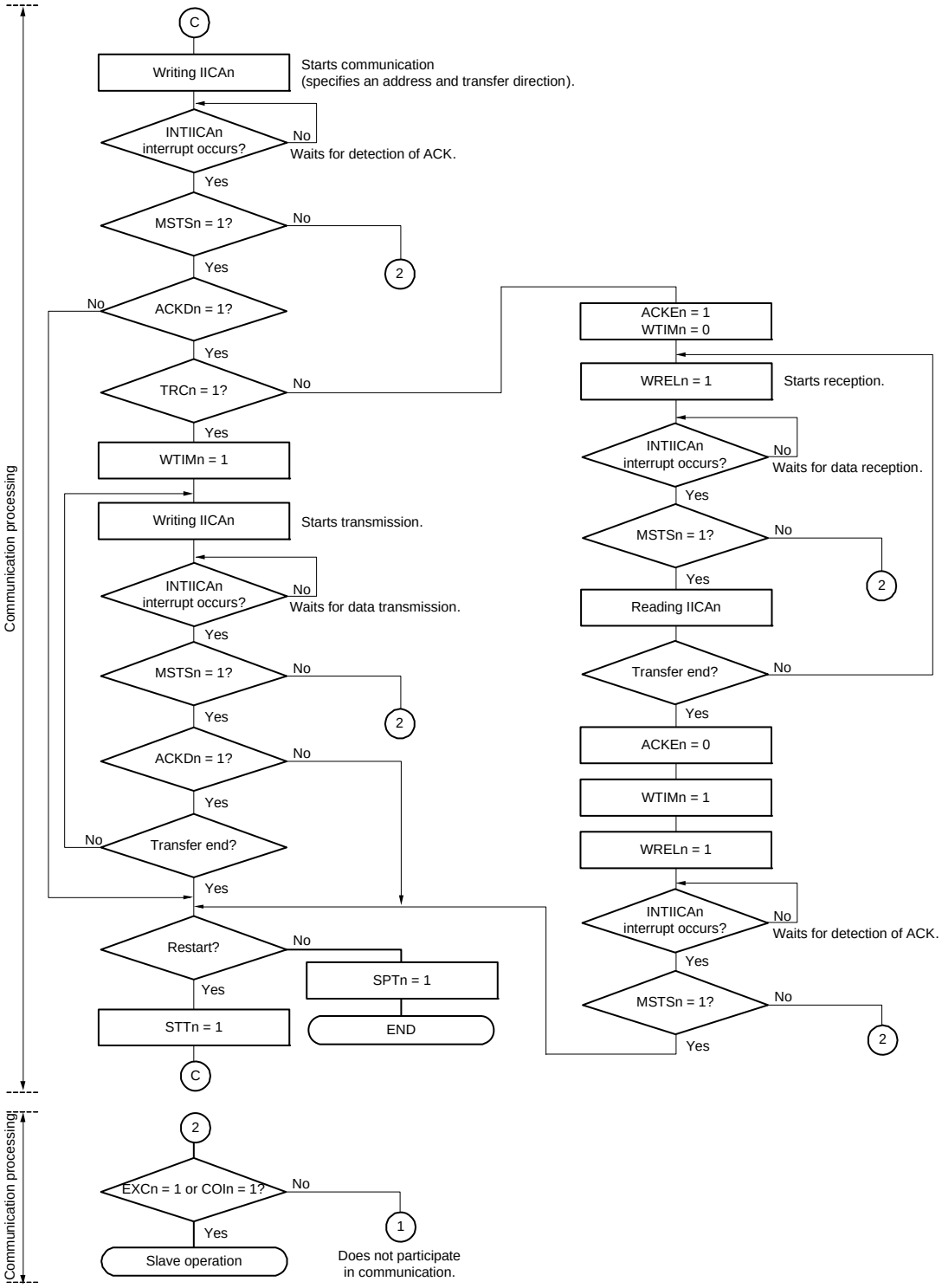


Note The wait time is calculated as follows.
(IICWLn setting value + IICWHn setting value + 4)/f_{MCK} + $t_F \times 2$

Remark1. IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
t_F: SDAAn and SCLAn signal falling times
f_{MCK}: IICA operation clock frequency

Remark2. n = 0

Figure 16 - 38 Master Operation in Multi-Master System (3/3)



Remark 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

Remark 2. To use the device as a master in a multi-master system, read the MSTSn bit each time interrupt INTIICAn has occurred to check the arbitration result.

Remark 3. To use the device as a slave in a multi-master system, check the status by using the IICA status register n (IICSn) and IICA flag register n (IICFn) each time interrupt INTIICAn has occurred, and determine the processing to be performed next.

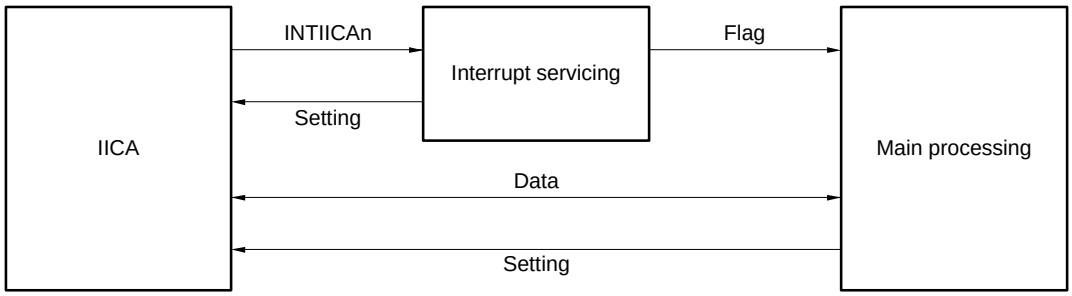
Remark 4. n = 0

(3) Slave operation

The processing procedure of the slave operation is as follows.

Basically, the slave operation is event-driven. Therefore, processing by the INTIICAn interrupt (processing that must substantially change the 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 INTIICAn 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 INTIICAn.

<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 INTIICAn interrupt for ordinary data communication. This flag is set by interrupt servicing and cleared by the main processing. Clear this flag by interrupt servicing when communication is started. However, the ready flag is not set by interrupt servicing when the first data is transmitted. Therefore, the first data is transmitted without the flag being cleared (an address match is interpreted as a request for the next data).

<3> Communication direction flag

This flag indicates the direction of communication. Its value is the same as the TRCn bit.

Remark n = 0

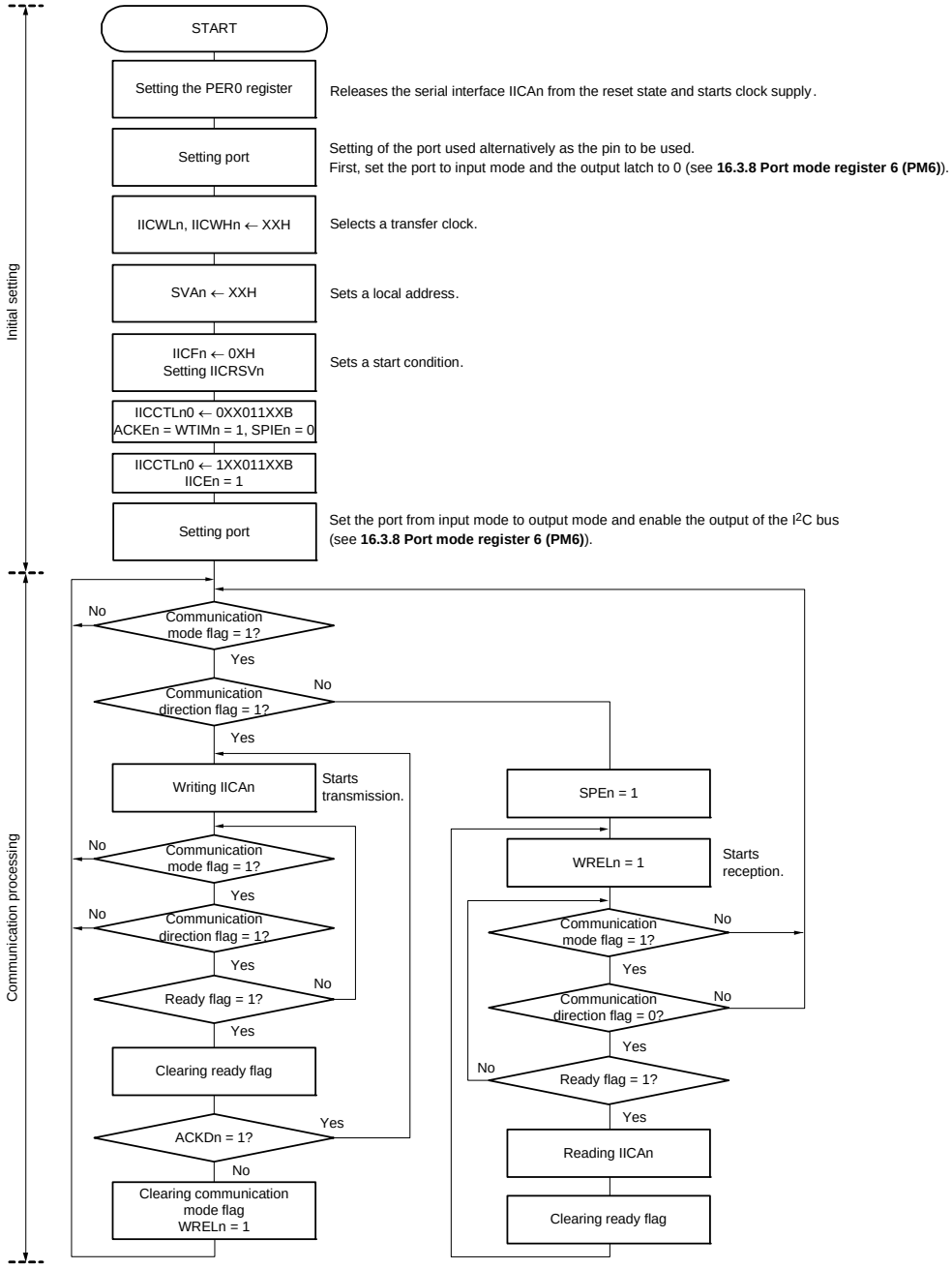
The main processing of the slave operation is explained next.

Start serial interface IICA and wait until communication is enabled. When communication is enabled, execute communication by using the communication mode flag and ready flag (processing of the stop condition and start condition is performed by an interrupt. Here, check the 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 16 - 39 Slave Operation Flowchart (1)



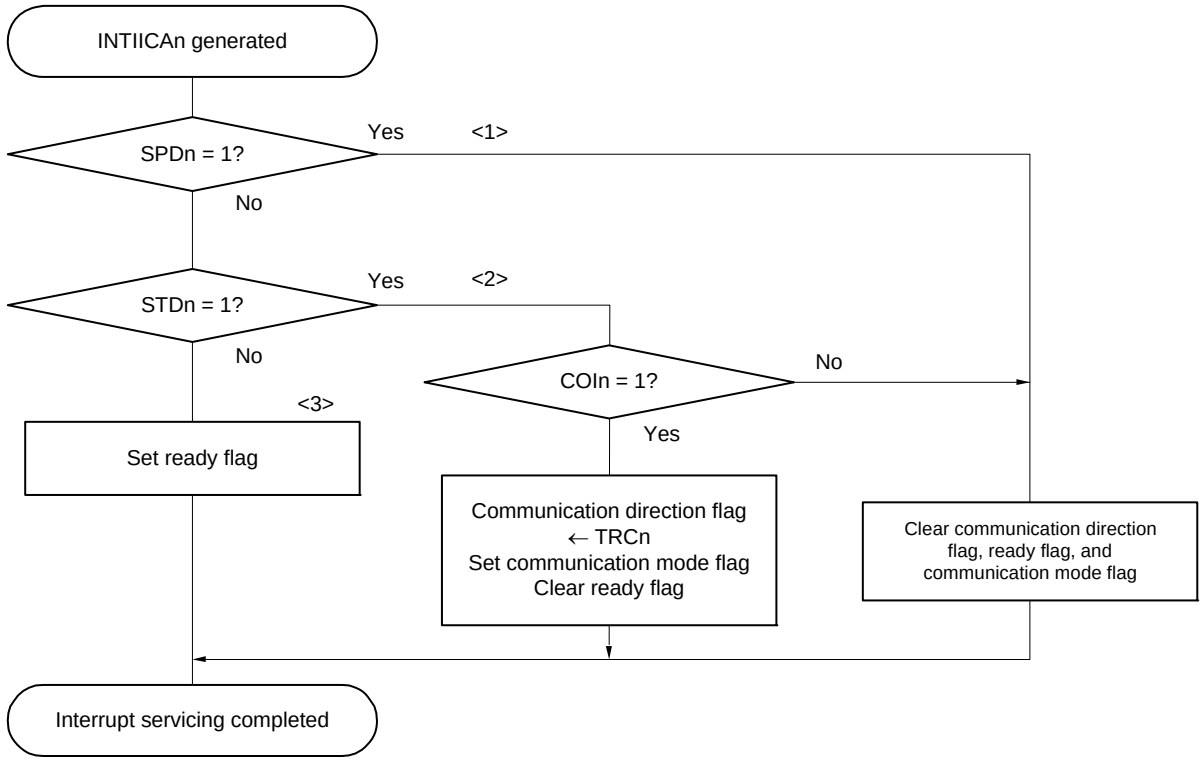
Remark1. Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.
Remark2. n = 0

An example of the processing procedure of the slave with the INTIICAn interrupt is explained below (processing is performed assuming that no extension code is used). The INTIICAn 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 16 - 40 Slave Operation Flowchart (2).

Figure 16 - 40 Slave Operation Flowchart (2)



Remark n = 0

16.5.17 Timing of I²C interrupt request (INTIICAn) occurrence

The timing of transmitting or receiving data and generation of interrupt request signal INTIICAn, and the value of the IICA status register n (IICSn) when the INTIICAn signal is generated are shown below.

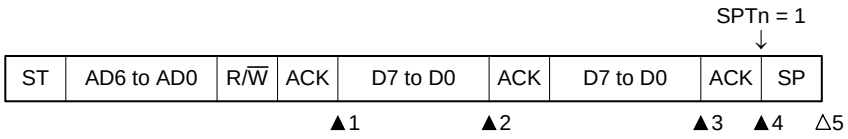
Remark 1. ST: Start condition
AD6 to AD0: Address
R/$\overline{W}$: Transfer direction specification
ACK: Acknowledge
D7 to D0: Data
SP: Stop condition

Remark 2. n = 0

(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (transmission/reception)

(i) When WTIMn = 0



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×000B

▲3: IICSn = 1000×000B (Sets the WTIMn bit to 1) Note

▲4: IICSn = 1000××00B (Sets the SPTn bit to 1)

△5: IICSn = 00000001B

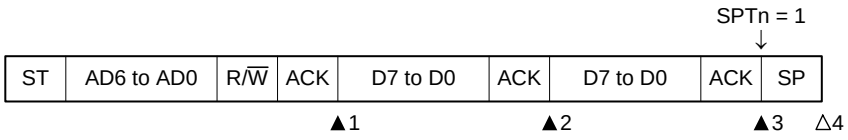
Note To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×100B

▲3: IICSn = 1000××00B (Sets the SPTn bit to 1)

△4: IICSn = 00000001B

Remark ▲: Always generated

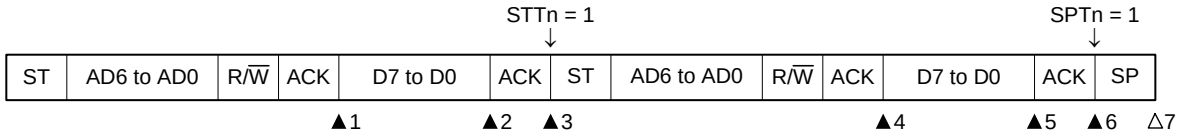
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

(i) When WTIMn = 0



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1) Note 1

▲3: IICSn = 1000××00B (Clears the WTIMn bit to 0 Note 2, sets the STTn bit to 1)

▲4: IICSn = 1000×110B

▲5: IICSn = 1000×000B (Sets the WTIMn bit to 1) Note 3

▲6: IICSn = 1000××00B (Sets the SPTn bit to 1)

△7: IICSn = 00000001B

Note 1. To generate a start condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

Note 2. Clear the WTIMn bit to 0 to restore the original setting.

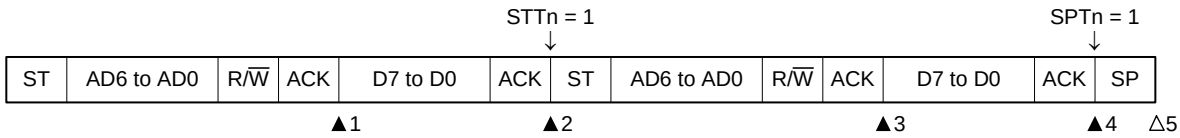
Note 3. To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1



▲1: IICSn = 1000×110B

▲2: IICSn = 1000××00B (Sets the STTn bit to 1)

▲3: IICSn = 1000×110B

▲4: IICSn = 1000××00B (Sets the SPTn bit to 1)

△5: IICSn = 00000001B

Remark ▲: Always generated

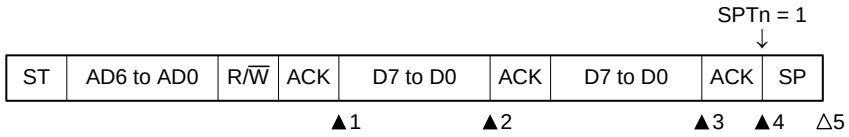
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

(i) When WTIMn = 0



▲1: IICSn = 1010×110B

▲2: IICSn = 1010×000B

▲3: IICSn = 1010×000B (Sets the WTIMn bit to 1) Note

▲4: IICSn = 1010××00B (Sets the SPTn bit to 1)

△5: IICSn = 00000001B

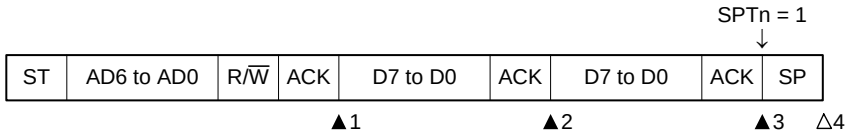
Note To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1



▲1: IICSn = 1010×110B

▲2: IICSn = 1010×100B

▲3: IICSn = 1010××00B (Sets the SPTn bit to 1)

△4: IICSn = 00001001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

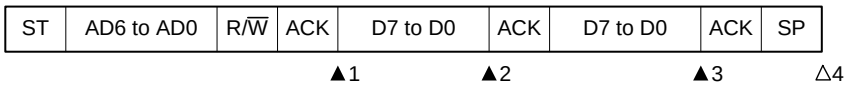
×: Don't care

Remark n = 0

(2) Slave device operation (slave address data reception)

(a) Start ~ Address ~ Data ~ Data ~ Stop

(i) When WTIMn = 0



▲1: IICSn = 0001×110B

▲2: IICSn = 0001×000B

▲3: IICSn = 0001×000B

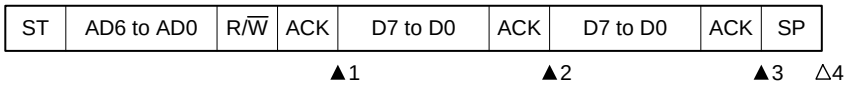
△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1



▲1: IICSn = 0001×110B

▲2: IICSn = 0001×100B

▲3: IICSn = 0001××00B

△4: IICSn = 00000001B

Remark ▲: Always generated

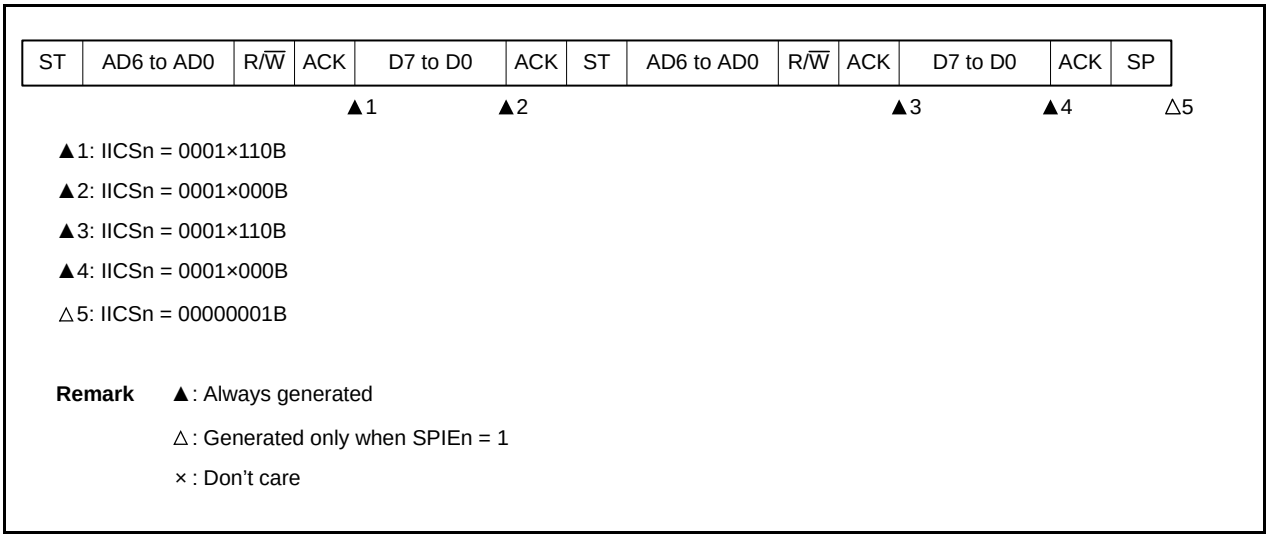
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, matches with SVAn)



▲1: IICSn = 0001×110B

▲2: IICSn = 0001×000B

▲3: IICSn = 0001×110B

▲4: IICSn = 0001×000B

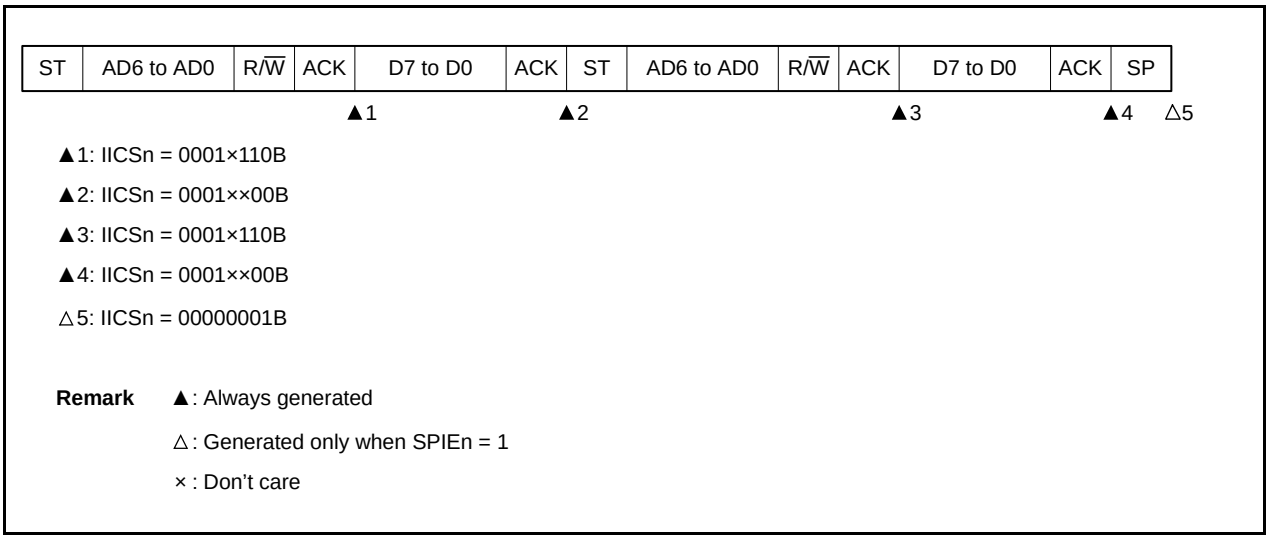
△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1 (after restart, matches with SVAn)



▲1: IICSn = 0001×110B

▲2: IICSn = 0001××00B

▲3: IICSn = 0001×110B

▲4: IICSn = 0001××00B

△5: IICSn = 00000001B

Remark ▲: Always generated

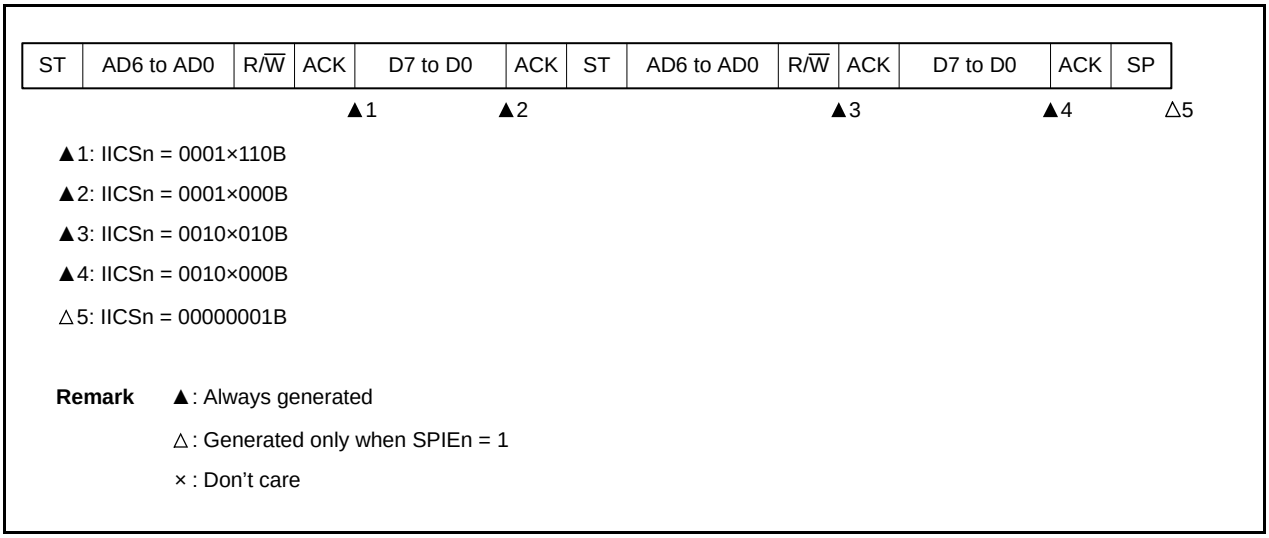
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, does not match address (= extension code))



▲1: IICSn = 0001×110B

▲2: IICSn = 0001×000B

▲3: IICSn = 0010×010B

▲4: IICSn = 0010×000B

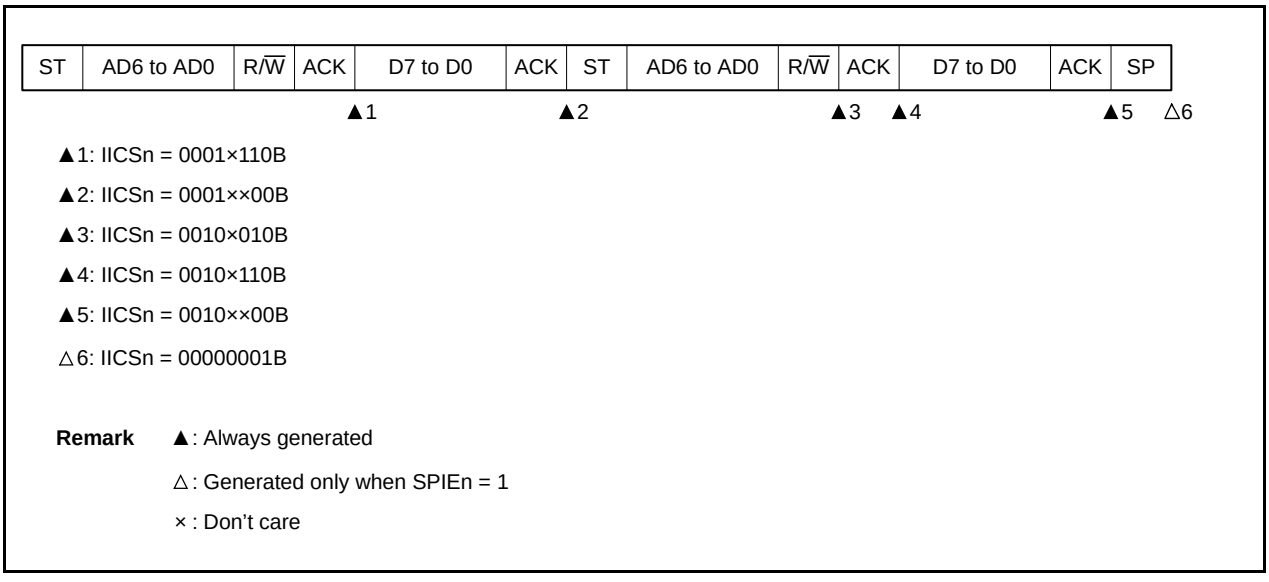
△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1 (after restart, does not match address (= extension code))



▲1: IICSn = 0001×110B

▲2: IICSn = 0001××00B

▲3: IICSn = 0010×010B

▲4: IICSn = 0010×110B

▲5: IICSn = 0010××00B

△6: IICSn = 00000001B

Remark ▲: Always generated

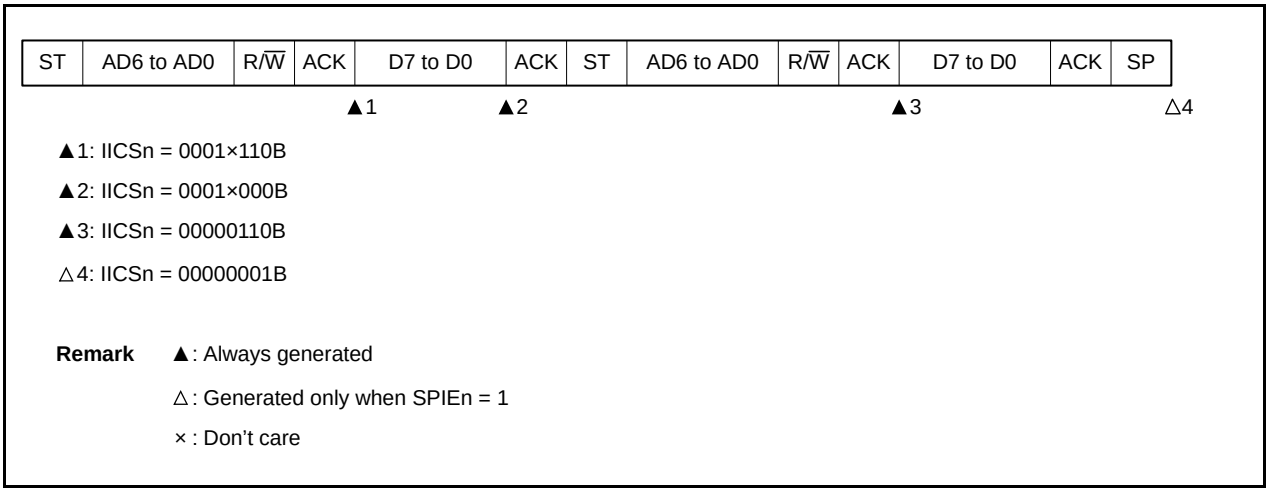
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, does not match address (= not extension code))



▲1: IICSn = 0001×110B

▲2: IICSn = 0001×000B

▲3: IICSn = 00000110B

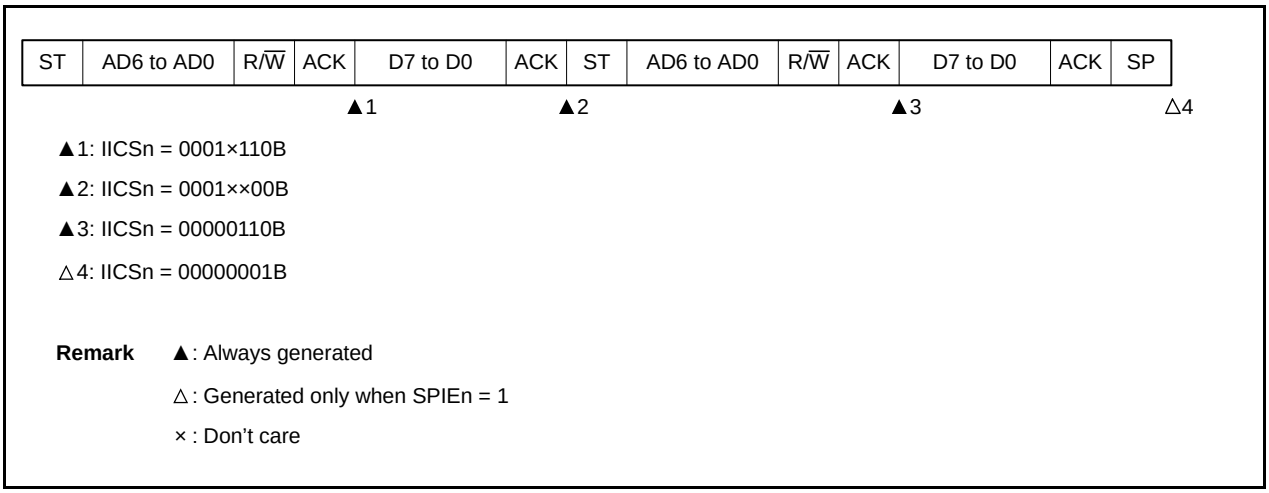
△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1 (after restart, does not match address (= not extension code))



▲1: IICSn = 0001×110B

▲2: IICSn = 0001××00B

▲3: IICSn = 00000110B

△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

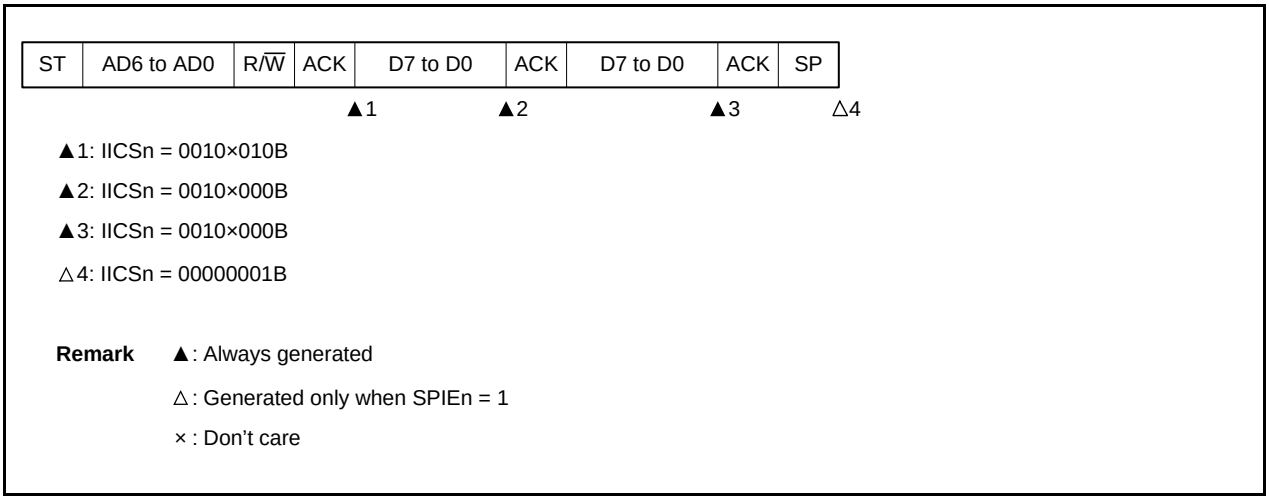
Remark n = 0

(3) Slave device operation (when receiving extension code)

The device is always participating in communication when it receives an extension code.

(a) Start ~ Code ~ Data ~ Data ~ Stop

(i) When WTIMn = 0



▲1: IICSn = 0010×010B

▲2: IICSn = 0010×000B

▲3: IICSn = 0010×000B

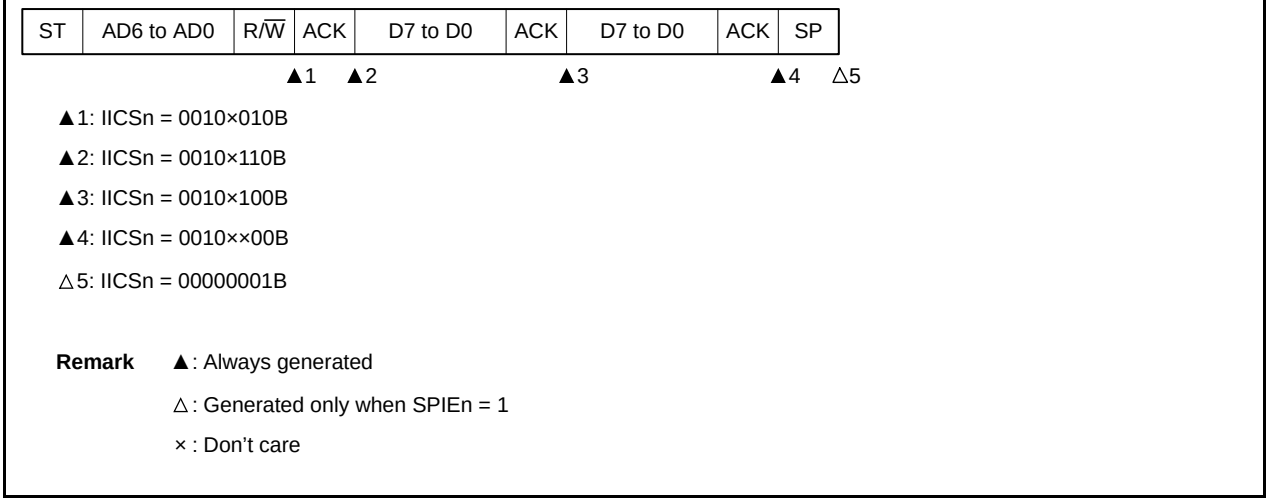
△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1



▲1: IICSn = 0010×010B

▲2: IICSn = 0010×110B

▲3: IICSn = 0010×100B

▲4: IICSn = 0010××00B

△5: IICSn = 00000001B

Remark ▲: Always generated

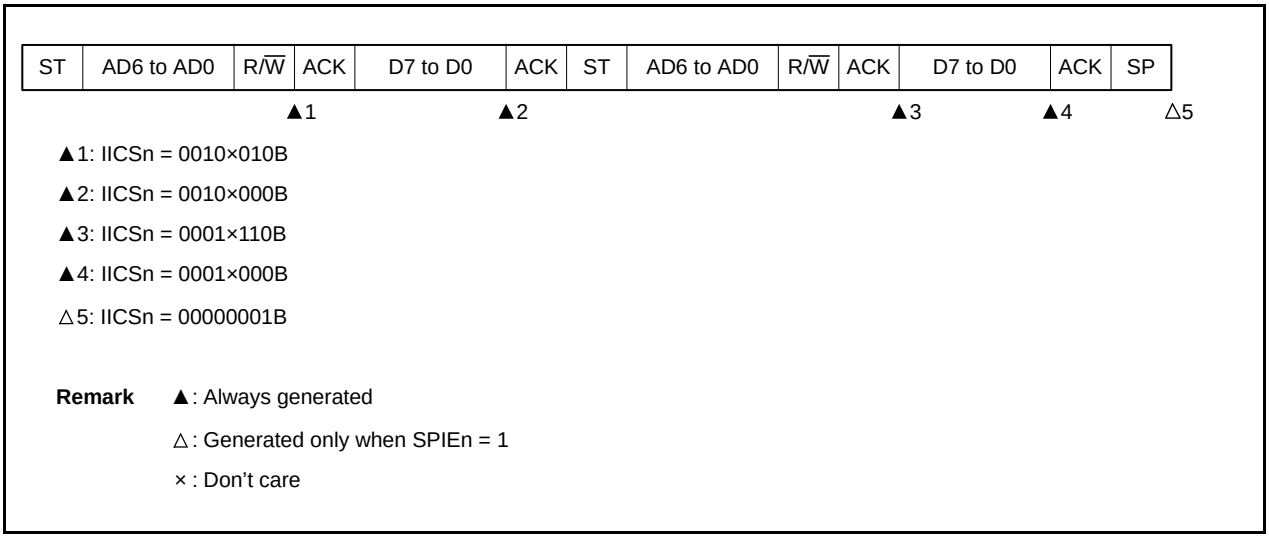
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, matches SVAn)



▲1: IICSn = 0010×010B

▲2: IICSn = 0010×000B

▲3: IICSn = 0001×110B

▲4: IICSn = 0001×000B

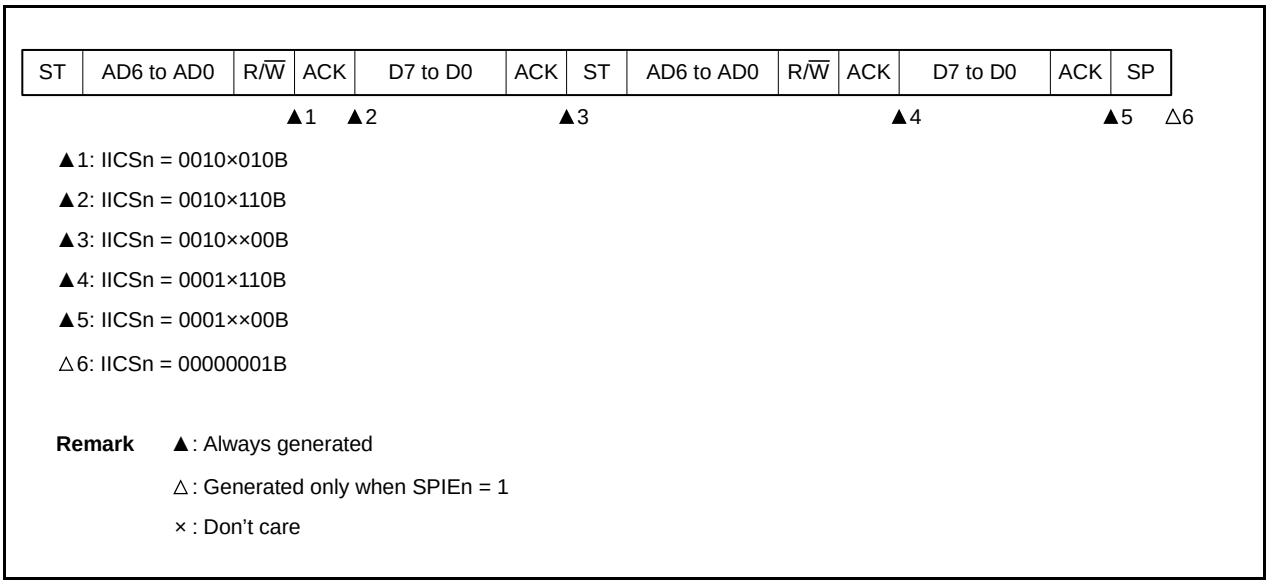
△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1 (after restart, matches SVAn)



▲1: IICSn = 0010×010B

▲2: IICSn = 0010×110B

▲3: IICSn = 0010××00B

▲4: IICSn = 0001×110B

▲5: IICSn = 0001××00B

△6: IICSn = 00000001B

Remark ▲: Always generated

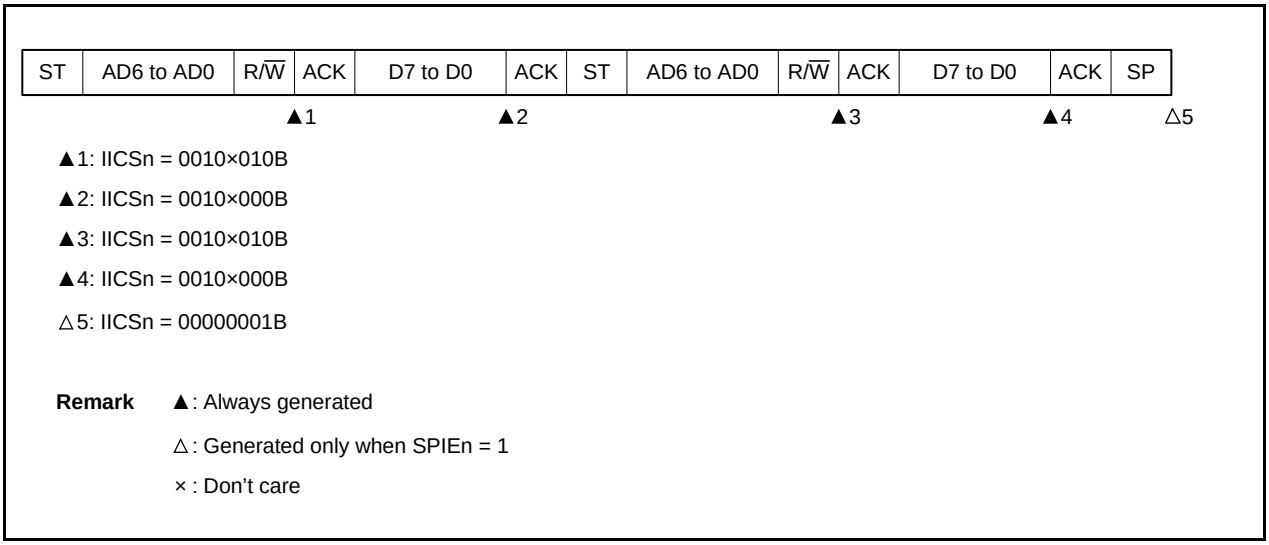
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, extension code reception)



▲1: IICSn = 0010×010B

▲2: IICSn = 0010×000B

▲3: IICSn = 0010×010B

▲4: IICSn = 0010×000B

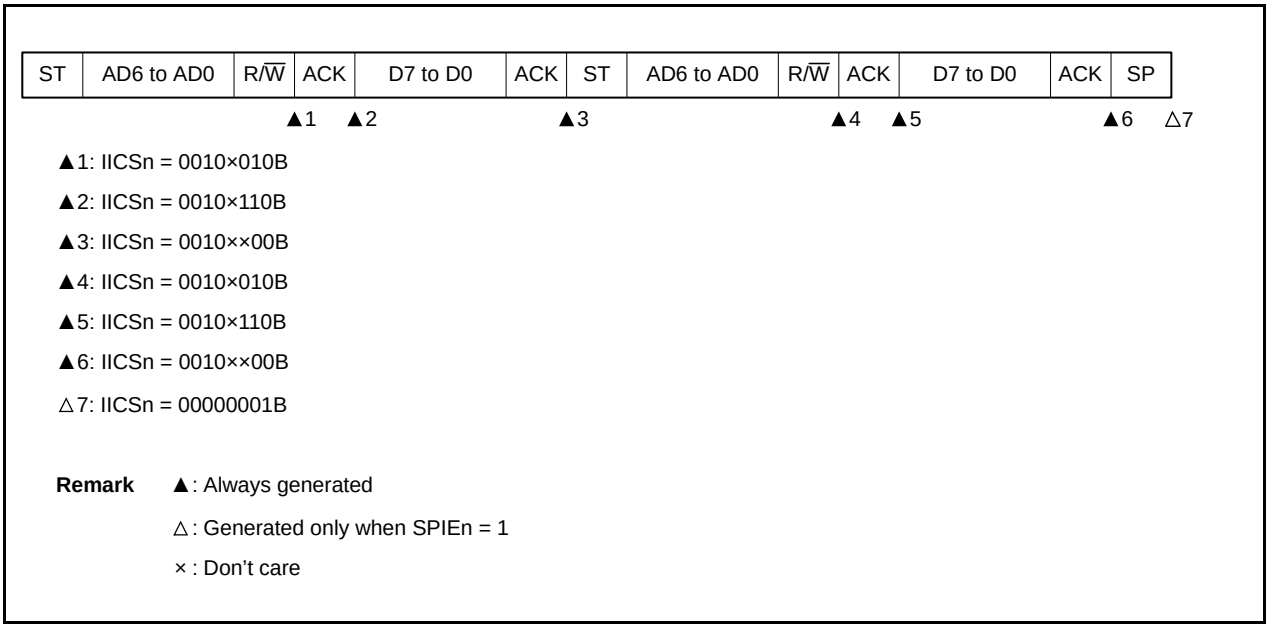
△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1 (after restart, extension code reception)



▲1: IICSn = 0010×010B

▲2: IICSn = 0010×110B

▲3: IICSn = 0010××00B

▲4: IICSn = 0010×010B

▲5: IICSn = 0010×110B

▲6: IICSn = 0010××00B

△7: IICSn = 00000001B

Remark ▲: Always generated

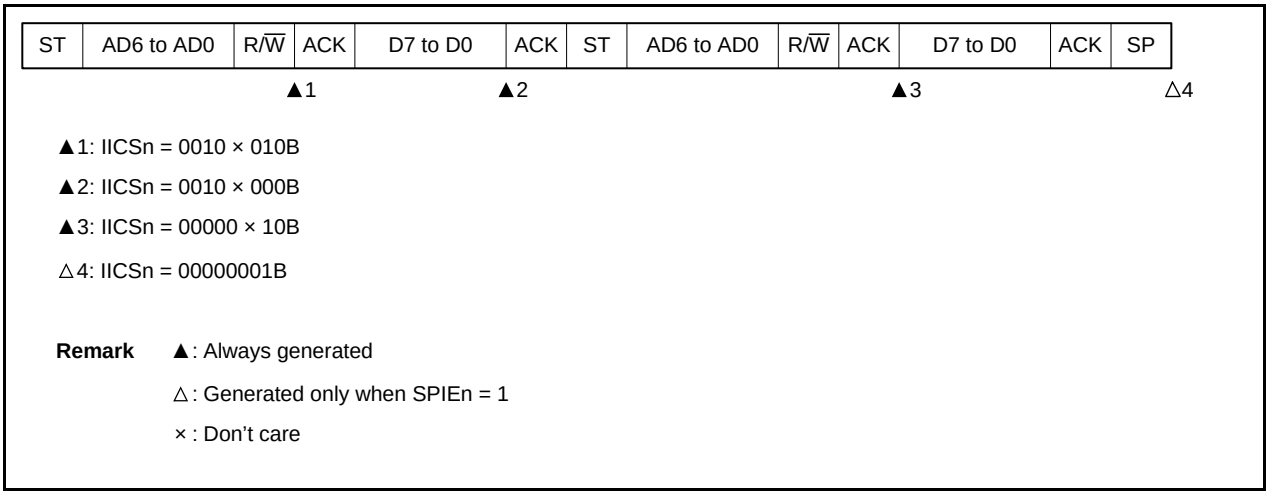
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, does not match address (= not extension code))



▲1: IICSn = 0010 × 010B

▲2: IICSn = 0010 × 000B

▲3: IICSn = 00000 × 10B

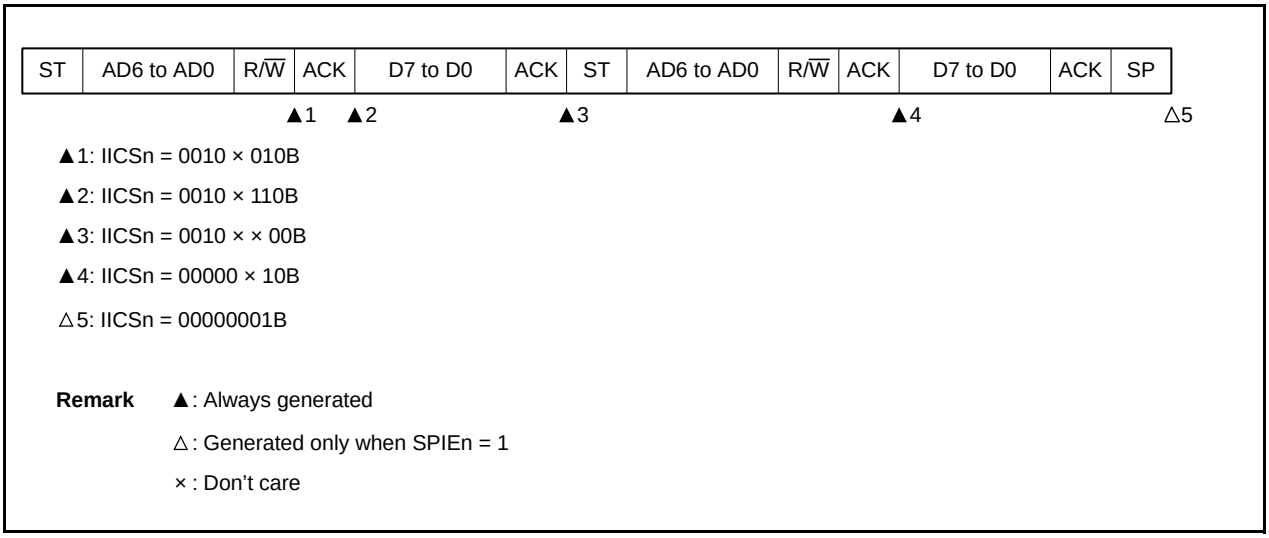
△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1 (after restart, does not match address (= not extension code))



▲1: IICSn = 0010 × 010B

▲2: IICSn = 0010 × 110B

▲3: IICSn = 0010 × × 00B

▲4: IICSn = 00000 × 10B

△5: IICSn = 00000001B

Remark ▲: Always generated

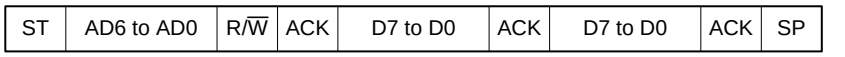
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop



△1: IICSn = 00000001B

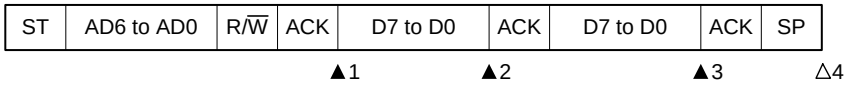
Remark △: Generated only when SPIEn = 1

(5) Arbitration loss operation (operation as slave after arbitration loss)

When the device is used as a master in a multi-master system, read the MSTSn bit each time interrupt request signal INTIICAn has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data

(i) When WTIMn = 0



▲1: IICSn = 0101×110B

▲2: IICSn = 0001×000B

▲3: IICSn = 0001×000B

△4: IICSn = 00000001B

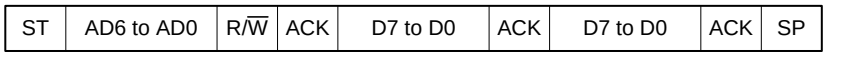
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0

(ii) When WTIMn = 1



▲1 ▲2 ▲3 △4

▲1: IICSn = 0101×110B

▲2: IICSn = 0001×100B

▲3: IICSn = 0001××00B

△4: IICSn = 00000001B

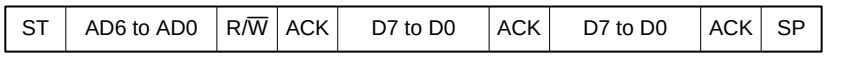
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(b) When arbitration loss occurs during transmission of extension code

(i) When WTIMn = 0



▲1 ▲2 ▲3 △4

▲1: IICSn = 0110×010B

▲2: IICSn = 0010×000B

▲3: IICSn = 0010×000B

△4: IICSn = 00000001B

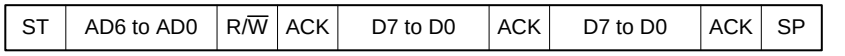
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0

(ii) When WTIMn = 1



▲1 ▲2 ▲3 ▲4 △5

▲1: IICSn = 0110×010B

▲2: IICSn = 0010×110B

▲3: IICSn = 0010×100B

▲4: IICSn = 0010××00B

△5: IICSn = 00000001B

Remark ▲: Always generated

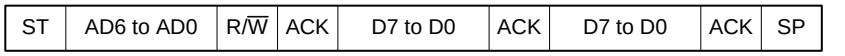
△: Generated only when SPIEn = 1

×: Don't care

(6) Operation when arbitration loss occurs (no communication after arbitration loss)

When the device is used as a master in a multi-master system, read the MSTSn bit each time interrupt request signal INTIICAn has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data (when WTIMn = 1)



▲1 △2

▲1: IICSn = 01000110B

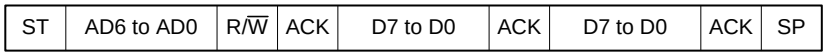
△2: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

Remark n = 0

(b) When arbitration loss occurs during transmission of extension code



▲1 △2

▲1: IICSn = 0110×010B

Sets LRELn = 1 by software

△2: IICSn = 00000001B

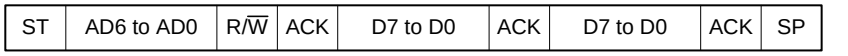
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(c) When arbitration loss occurs during transmission of data

(i) When WTIMn = 0



▲1 ▲2 △3

▲1: IICSn = 10001110B

▲2: IICSn = 01000000B

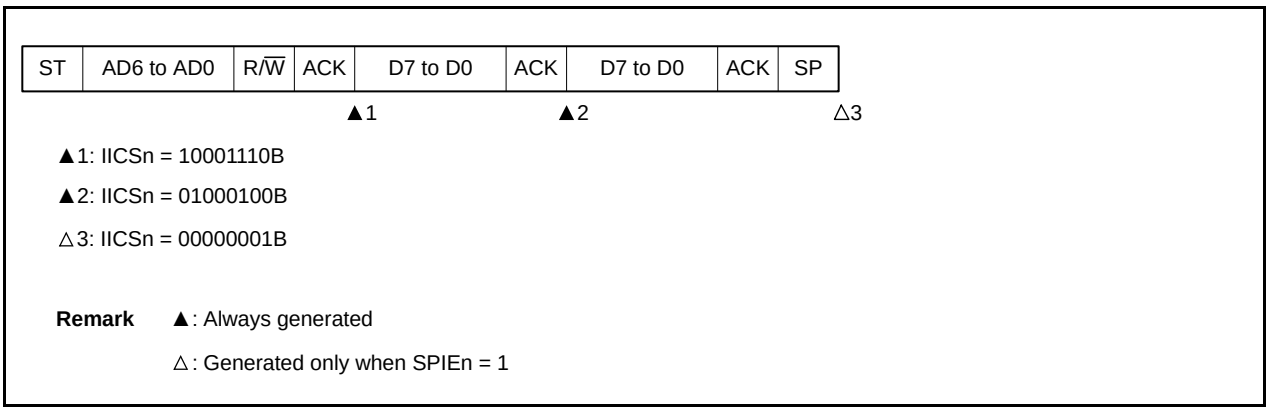
△3: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

Remark n = 0

(ii) When WTIMn = 1



▲1: IICSn = 10001110B

▲2: IICSn = 01000100B

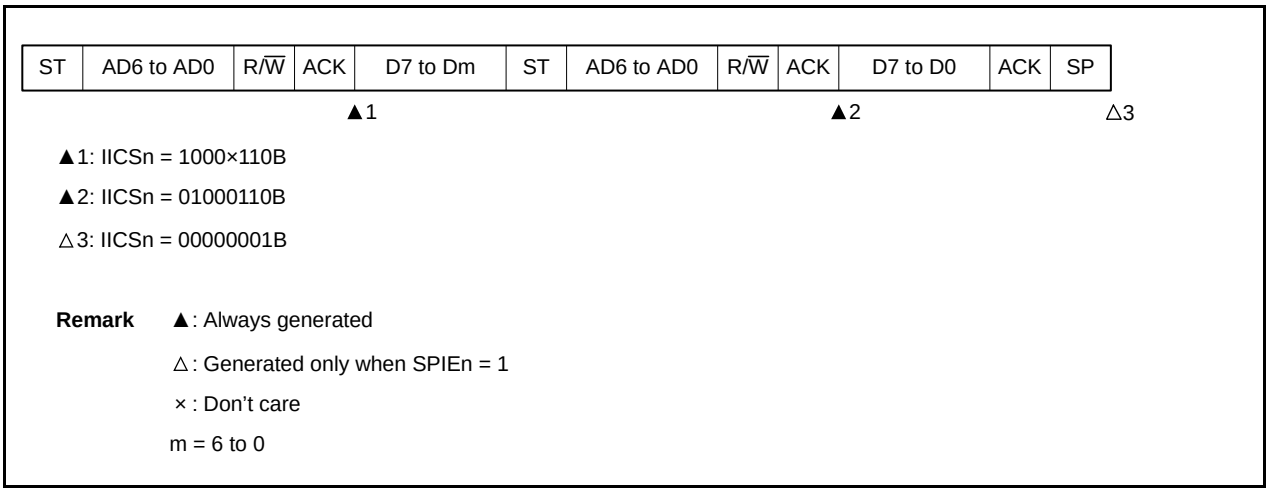
△3: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

(d) When loss occurs due to restart condition during data transfer

(i) Not extension code (Example: unmatches with SVAn)



▲1: IICSn = 1000×110B

▲2: IICSn = 01000110B

△3: IICSn = 00000001B

Remark ▲: Always generated

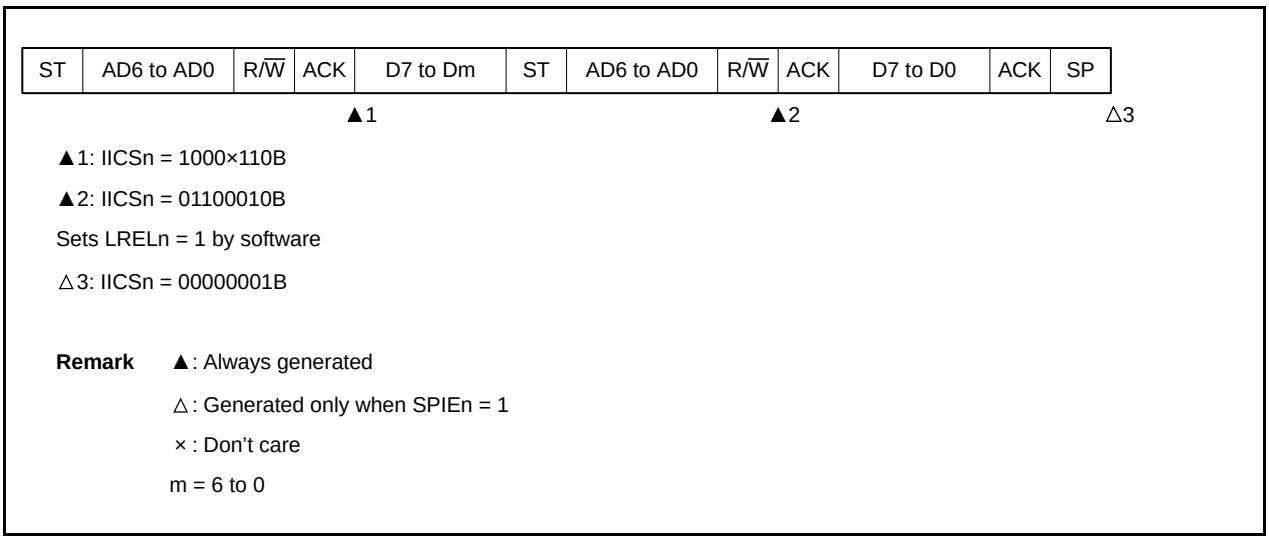
△: Generated only when SPIEn = 1

×: Don't care

m = 6 to 0

Remark n = 0

(ii) Extension code



▲1: IICSn = 1000×110B

▲2: IICSn = 01100010B

Sets LRELn = 1 by software

△3: IICSn = 00000001B

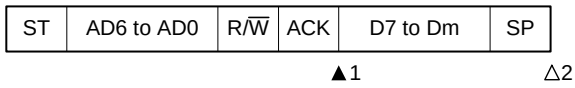
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

m = 6 to 0

(e) When loss occurs due to stop condition during data transfer



▲1: IICSn = 10000110B

△2: IICSn = 01000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

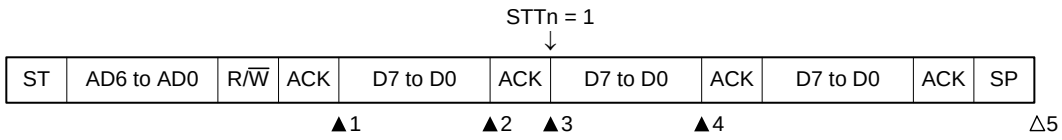
×: Don't care

m = 6 to 0

Remark n = 0

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

(i) When WTIMn = 0



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1)

▲3: IICSn = 1000×100B (Clears the WTIMn bit to 0)

▲4: IICSn = 01000000B

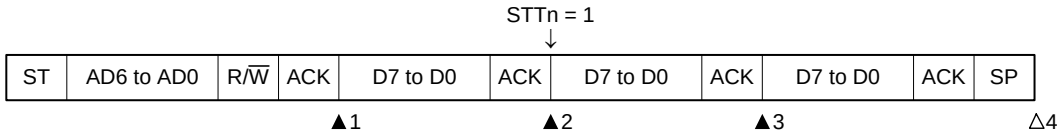
△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×100B (Sets the STTn bit to 1)

▲3: IICSn = 01000100B

△4: IICSn = 00000001B

Remark ▲: Always generated

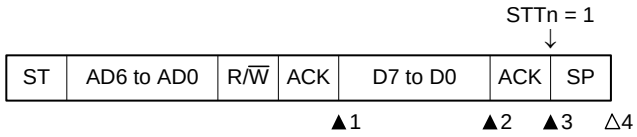
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0

(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

(i) When WTIMn = 0



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1)

▲3: IICSn = 1000××00B (Sets the STTn bit to 1)

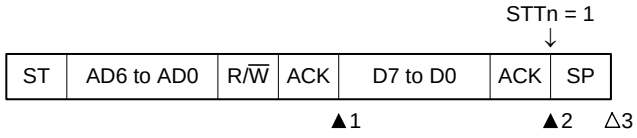
△4: IICSn = 01000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1



▲1: IICSn = 1000×110B

▲2: IICSn = 1000××00B (Sets the STTn bit to 1)

△3: IICSn = 01000001B

Remark ▲: Always generated

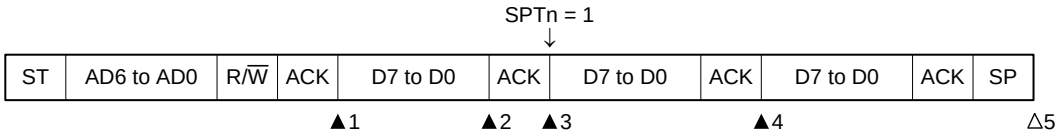
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

(i) When WTIMn = 0



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1)

▲3: IICSn = 1000×100B (Clears the WTIMn bit to 0)

▲4: IICSn = 01000100B

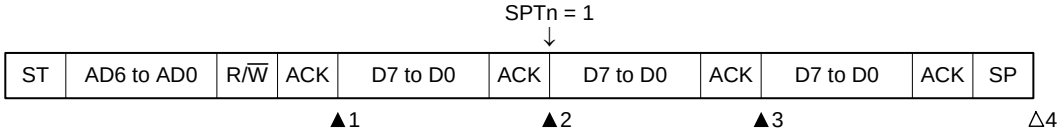
△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×100B (Sets the SPTn bit to 1)

▲3: IICSn = 01000100B

△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0

16.6 Timing Charts

When using the I^2C bus mode, the master device outputs an address via the serial bus to select one of several slave devices as its communication partner.

After outputting the slave address, the master device transmits the TRCn bit (bit 3 of the IICA status register n (IICSn)), which specifies the data transfer direction, and then starts serial communication with the slave device.

Figures 16 - 41 to 16 - 47 show timing charts of the data communication.

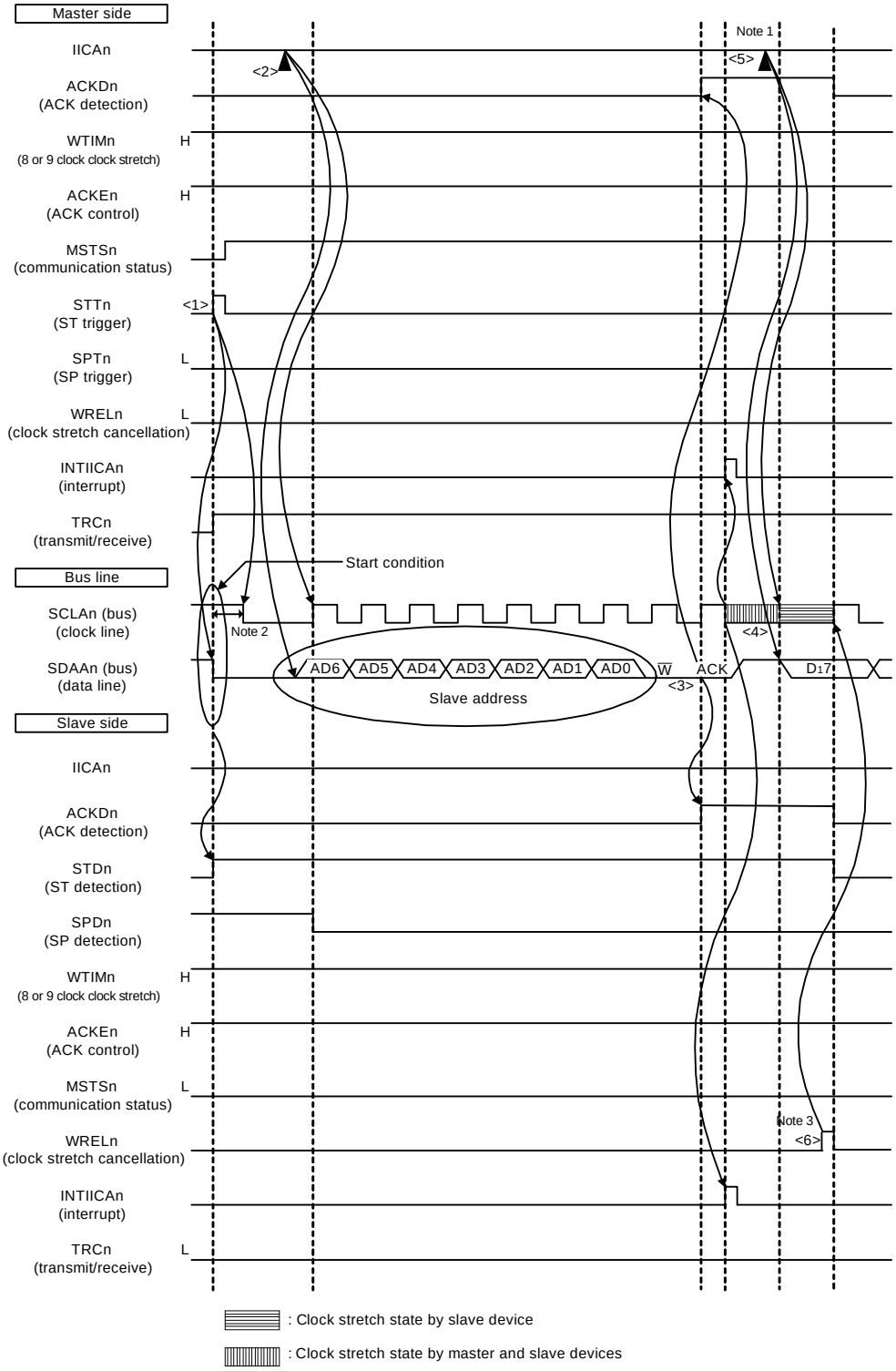
The IICA shift register n (IICAn)'s shift operation is synchronized with the falling edge of the serial clock (SCLAn). The transmit data is transferred to the SO latch and is output (MSB first) via the SDAAn pin.

Data input via the SDAAn pin is captured into IICAn at the rising edge of SCLAn.

Remark n = 0

Figure 16 - 41 Example of Master to Slave Communication
(When 9-Clock Clock Stretch Is Selected for Master, 9-Clock Clock Stretch Is Selected for Slave) (1/4)

(1) Start condition ~ address ~ data



Note 1. Write data to IICAn, not setting the WRELn bit, in order to cancel a clock stretch state during transmission by a master device.
Note 2. Make sure that the time between the fall of the SDAAn pin signal and the fall of the SCLAn pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.
Note 3. For releasing clock stretch state during reception of a slave device, write “FFH” to IICAn or set the WRELn bit.
Remark n = 0

The meanings of <1> to <6> in (1) Start condition ~ address ~ data in Figure 16 - 41 are explained below.

<1> The start condition trigger is set by the master device (STTn = 1) and a start condition (i.e. SCLAn = 1 changes SDAAn from 1 to 0) is generated once the bus data line goes low (SDAAn). When the start condition is subsequently detected, the master device enters the master device communication status (MSTSn = 1). The master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.

<2> The master device writes the address + W (transmission) to the IICA shift register n (IICAn) and transmits the slave address.

<3> In the slave device if the address received matches the address (SVAn value) of a slave device Note, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a clock stretch status (SCLAn = 0) and issues an interrupt (INTIICAn: address match) Note.

<5> The master device writes the data to transmit to the IICAn register and releases the clock stretch status that it set by the master device.

<6> If the slave device releases the clock stretch status (WRELn = 1), the master device starts transferring data to the slave device.

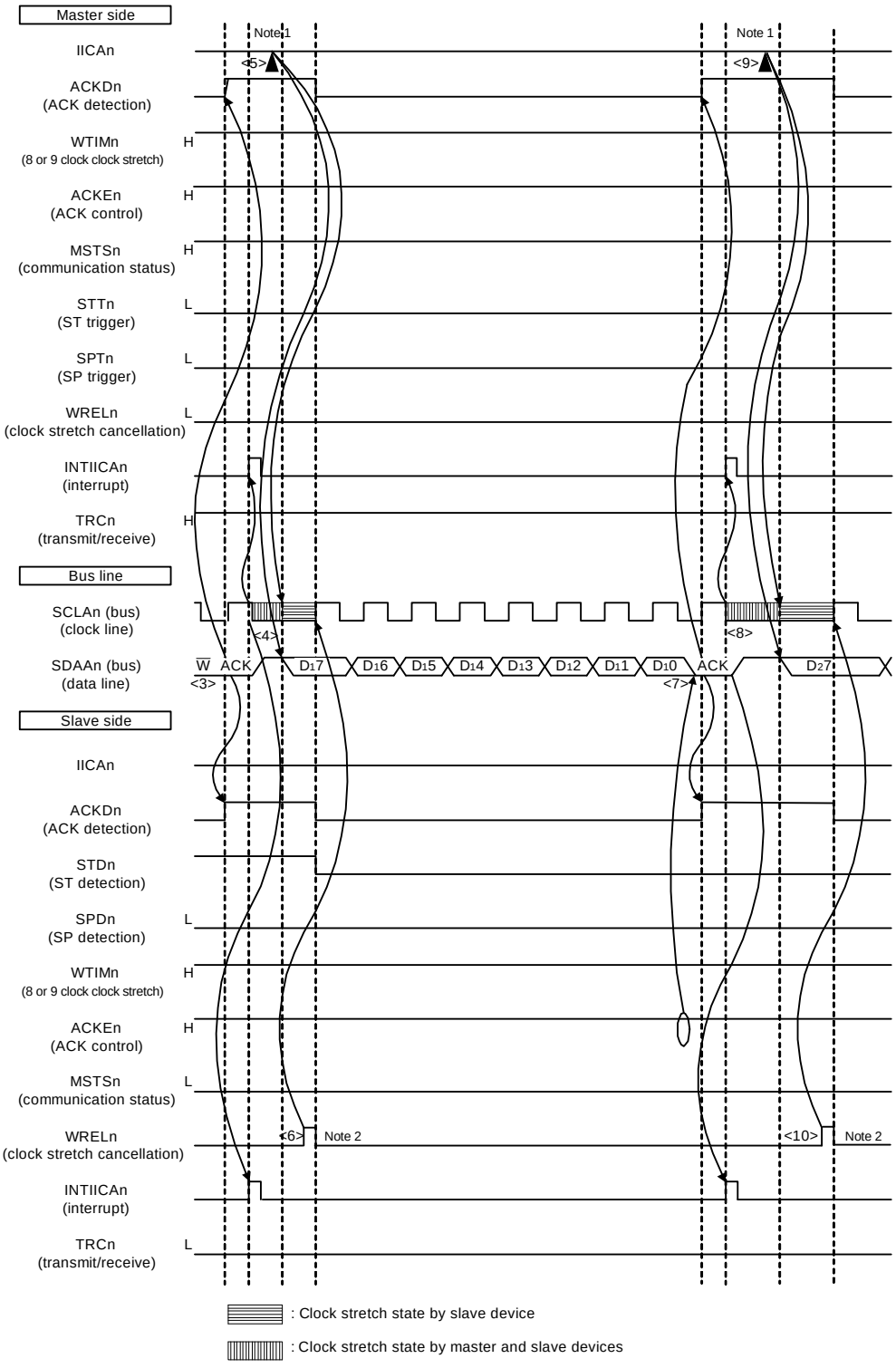
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set a clock stretch status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <15> in Figures 16 - 41 to 16 - 43 represent the entire procedure for communicating data using the I^2C bus. Figure 16 - 41 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 16 - 42 (3) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 16 - 43 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Remark 2. n = 0

Figure 16 - 42 Example of Master to Slave Communication
(When 9-Clock Clock Stretch Is Selected for Master, 9-Clock Clock Stretch Is Selected for Slave) (2/4)

(3) Address ~ data ~ data



Note 1. Write data to IICAn, not setting the WRELn bit, in order to cancel a clock stretch state during transmission by a master device.
Note 2. For releasing clock stretch state during reception of a slave device, write “FFH” to IICAn or set the WRELn bit.
Remark n = 0

The meanings of <3> to <10> in (3) Address ~ data ~ data in Figure 16 - 42 are explained below.

<3> In the slave device if the address received matches the address (SVAn value) of a slave device Note, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a clock stretch status (SCLAn = 0) and issues an interrupt (INTIICAn: address match) Note.

<5> The master device writes the data to transmit to the IICA shift register n (IICAn) and releases the clock stretch status that it set by the master device.

<6> If the slave device releases the clock stretch status (WRELn = 1), the master device starts transferring data to the slave device.

<7> After data transfer is completed, because of ACKEn = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a clock stretch status (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<9> The master device writes the data to transmit to the IICAn register and releases the clock stretch status that it set by the master device.

<10>The slave device reads the received data and releases the clock stretch status (WRELn = 1). The master device then starts transferring data to the slave device.

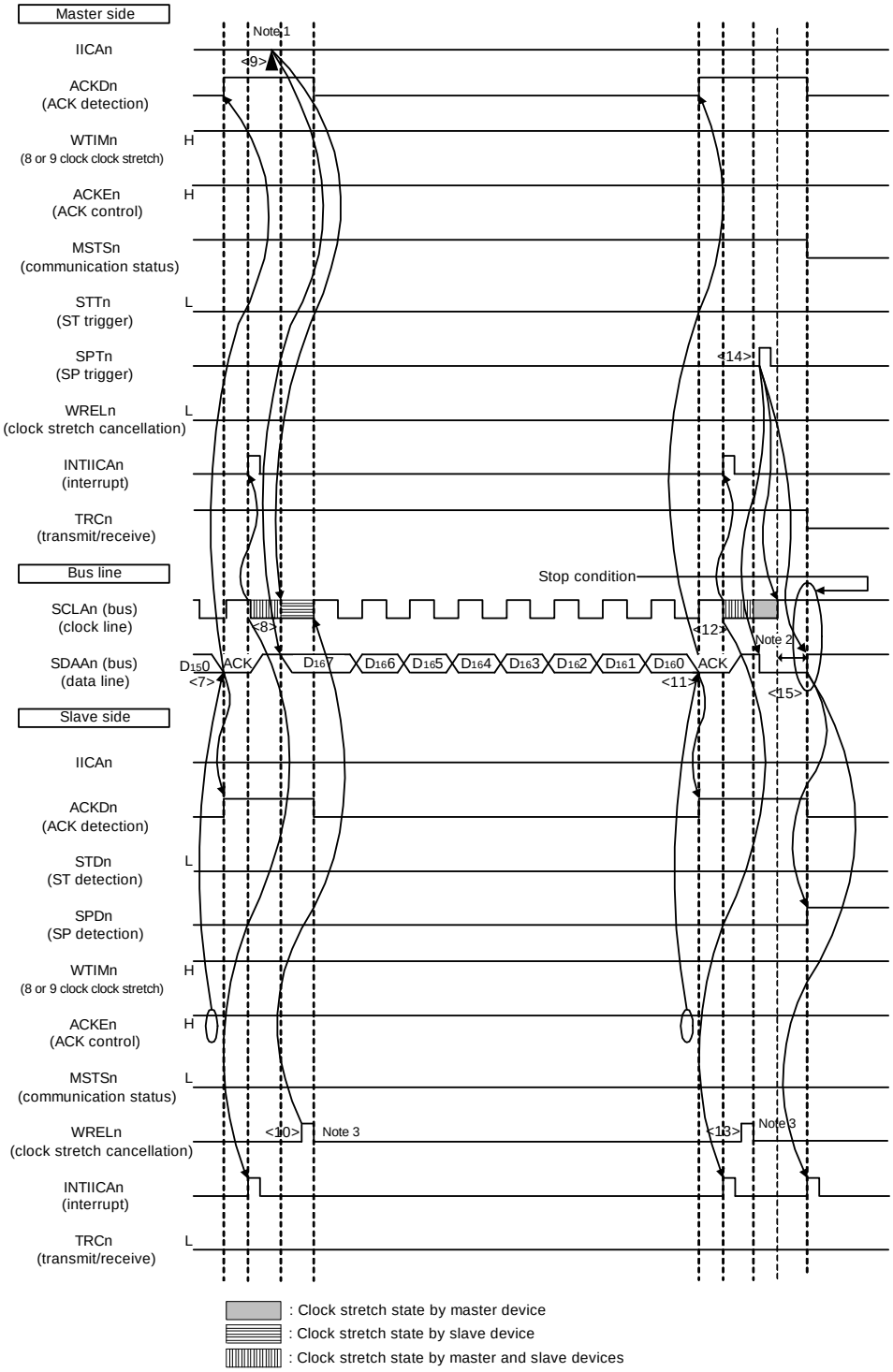
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set a clock stretch status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <15> in Figures 16 - 41 to 16 - 43 represent the entire procedure for communicating data using the I^2C bus. Figure 16 - 41 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 16 - 42 (3) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 16 - 43 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Remark 2. n = 0

Figure 16 - 43 Example of Master to Slave Communication
(When 9-Clock Clock Stretch Is Selected for Master, 9-Clock Clock Stretch Is Selected for Slave) (3/4)

(3) Data ~ data ~ stop condition



Note 1. Write data to IICAn, not setting the WRELn bit, in order to cancel a clock stretch state during transmission by a master device.

Note 2. Make sure that the time between the rise of the SCLAn pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

Note 3. For releasing clock stretch state during reception of a slave device, write “FFH” to IICAn or set the WRELn bit.

Remark n = 0

The meanings of <7> to <15> in (3) Data ~ data ~ stop condition in Figure 16 - 43 are explained below.

<7> After data transfer is completed, because of ACKEn = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a clock stretch status (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<9> The master device writes the data to transmit to the IICA shift register n (IICAn) and releases the clock stretch status that it set by the master device.

<10>The slave device reads the received data and releases the clock stretch status (WRELn = 1). The master device then starts transferring data to the slave device.

<11>When data transfer is complete, the slave device (ACKEn =1) sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<12>The master device and slave device set a clock stretch status (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<13>The slave device reads the received data and releases the clock stretch status (WRELn = 1).

<14> By the master device setting a stop condition trigger (SPTn = 1), the bus data line is cleared (SDAAn = 0) and the bus clock line is set (SCLAn = 1). After the stop condition setup time has elapsed, by setting the bus data line (SDAAn = 1), the stop condition is then generated (i.e. SCLAn =1 changes SDAAn from 0 to 1).

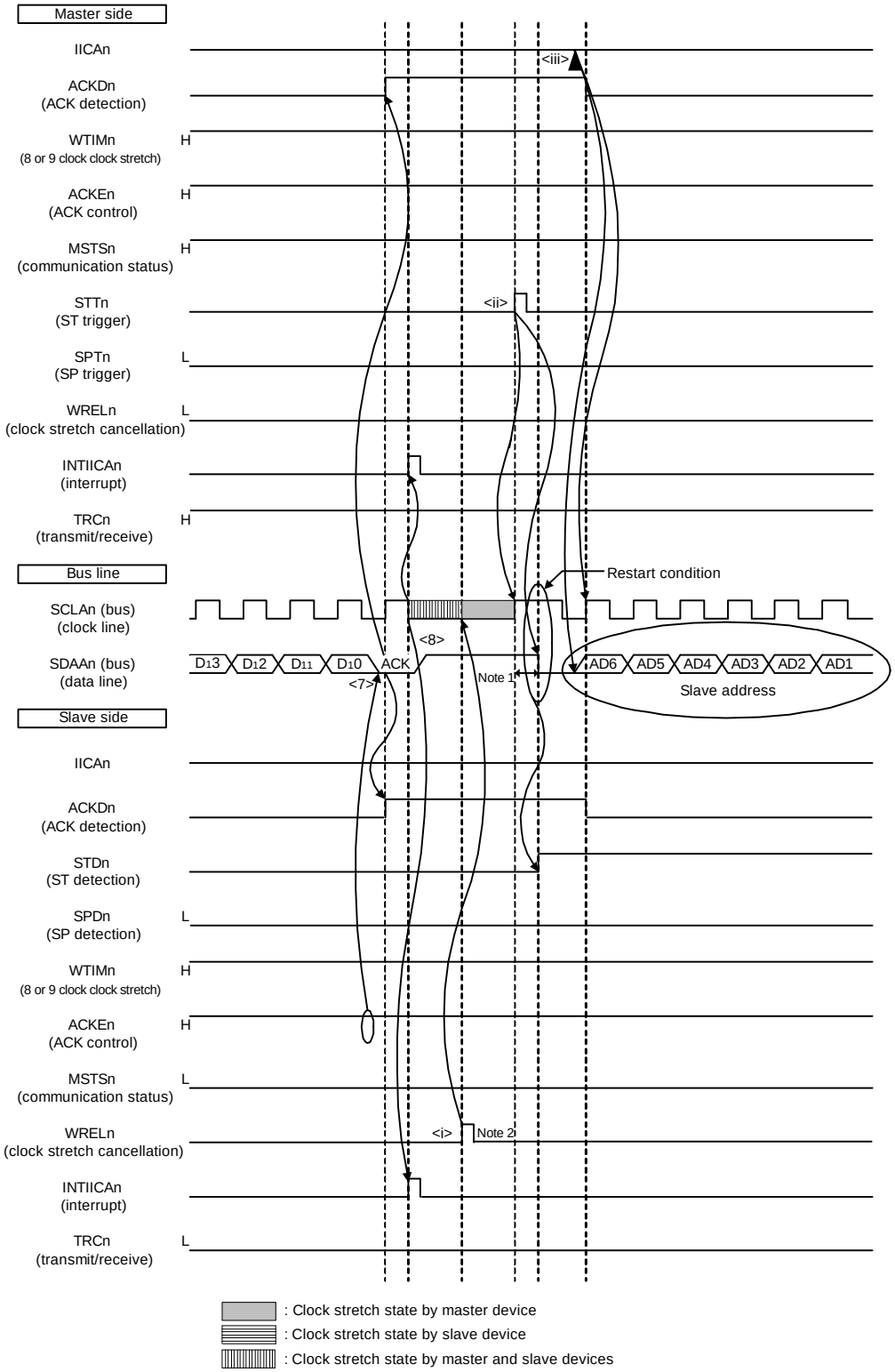
<15> When a stop condition is generated, the slave device detects the stop condition and issues an interrupt (INTIICAn: stop condition).

Remark 1. <1> to <15> in Figures 16 - 41 to 16 - 43 represent the entire procedure for communicating data using the I^2C bus. Figure 16 - 41 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 16 - 42 (3) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 16 - 43 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Remark 2. n = 0

Figure 16 - 44 Example of Master to Slave Communication
(When 9-Clock Clock Stretch Is Selected for Master, 9-Clock Clock Stretch Is Selected for Slave) (4/4)

(3) Data ~ restart condition ~ address



Note 1. Make sure that the time between the rise of the SCLAn pin signal and the generation of the start condition after a restart condition has been issued is at least 4.7 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

Note 2. For releasing clock stretch state during reception of a slave device, write “FFH” to IICAn or set the WRELn bit.

Remark n = 0

The following describes the operations in Figure 16 - 44 (3) Data ~ restart condition ~ address. After the operations in steps <7> and <8>, the operations in steps <i> to <iii> are performed. These steps return the processing to step <iii>, the data transmission step.

<7> After data transfer is completed, because of ACKEn = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a clock stretch status (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<i> The slave device reads the received data and releases the clock stretch status (WRELn = 1).

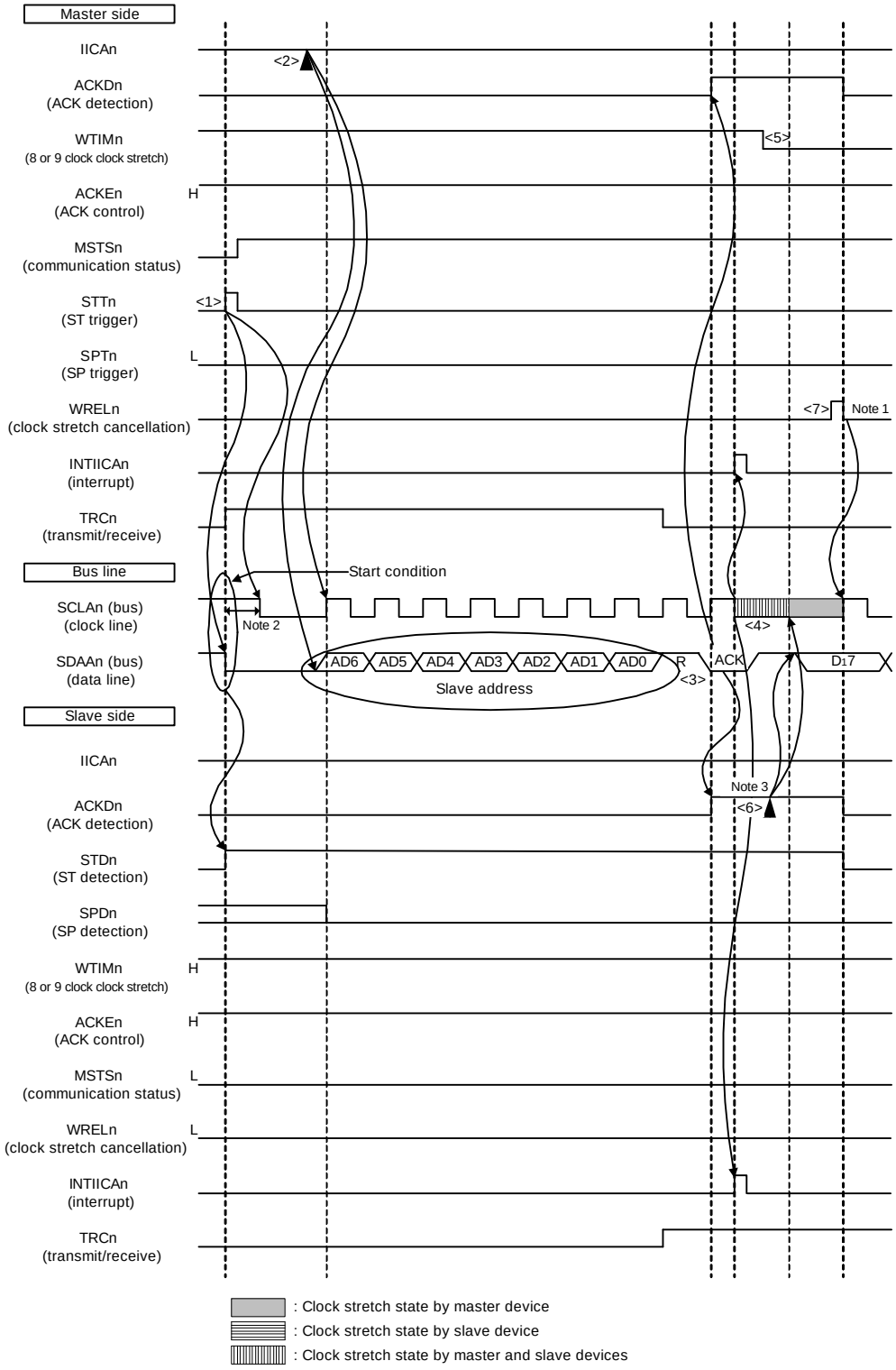
<ii> The start condition trigger is set again by the master device (STTn = 1) and a start condition (i.e. SCLAn =1 changes SDAAn from 1 to 0) is generated once the bus clock line goes high (SCLAn = 1) and the bus data line goes low (SDAAn = 0) after the restart condition setup time has elapsed. When the start condition is subsequently detected, the master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.

<iii> The master device writing the address + R/W (transmission) to the IICA shift register (IICAn) enables the slave address to be transmitted.

Remark n = 0

Figure 16 - 45 Example of Slave to Master Communication
(When 8-Clock Clock Stretch Is Selected for Master, 9-Clock Clock Stretch Is Selected for Slave) (1/3)

(1) Start condition ~ address ~ data



Note 1. For releasing clock stretch state during reception of a master device, write “FFH” to IICAn or set the WRELn bit.
Note 2. Make sure that the time between the fall of the SDAAn pin signal and the fall of the SCLAn pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.
Note 3. Write data to IICAn, not setting the WRELn bit, in order to cancel a clock stretch state during transmission by a slave device.
Remark n = 0

The meanings of <1> to <7> in (1) Start condition ~ address ~ data in Figure 16 - 45 are explained below.

<1> The start condition trigger is set by the master device (STTn = 1) and a start condition (i.e. SCLAn =1 changes SDAAn from 1 to 0) is generated once the bus data line goes low (SDAAn). When the start condition is subsequently detected, the master device enters the master device communication status (MSTSn = 1). The master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.

<2> The master device writes the address + R (reception) to the IICA shift register n (IICAn) and transmits the slave address.

<3> In the slave device if the address received matches the address (SVAn value) of a slave device Note, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a clock stretch status (SCLAn = 0) and issues an interrupt (INTIICAn: address match) Note.

<5> The timing at which the master device sets the clock stretch status changes to the 8th clock (WTIMn = 0).

<6> The slave device writes the data to transmit to the IICAn register and releases the clock stretch status that it set by the slave device.

<7> The master device releases the clock stretch status (WRELn = 1) and starts transferring data from the slave device to the master device.

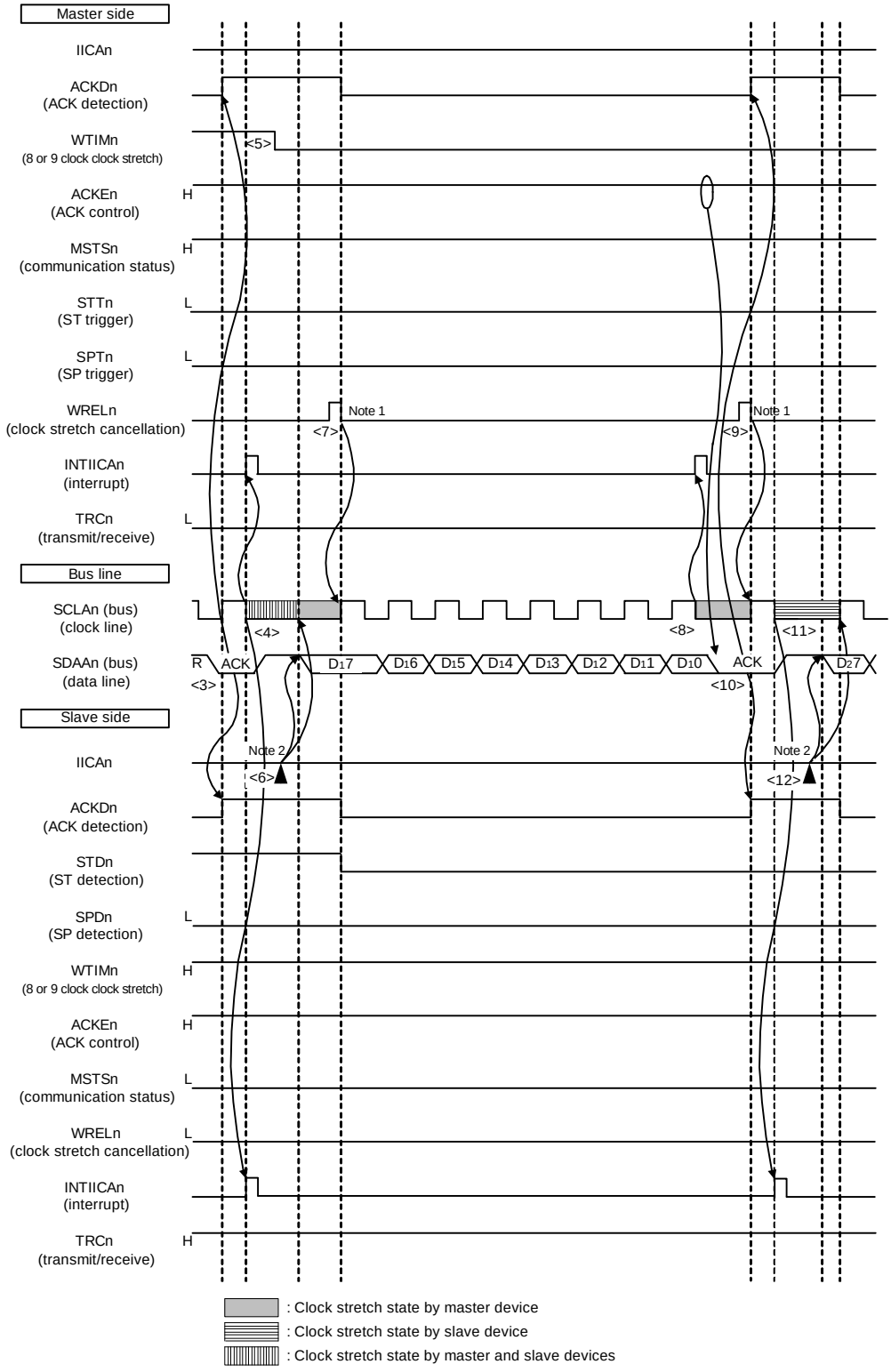
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set a clock stretch status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <19> in Figures 16 - 45 to 16 - 47 represent the entire procedure for communicating data using the I^2C bus. Figure 16 - 45 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 16 - 46 (3) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 16 - 47 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Remark 2. n = 0

Figure 16 - 46 Example of Slave to Master Communication
(When 8-Clock Clock Stretch Is Selected for Master, 9-Clock Clock Stretch Is Selected for Slave) (2/3)

(3) Address ~ data ~ data



Note 1. For releasing clock stretch state during reception of a master device, write “FFH” to IICAn or set the WRELn bit.
Note 2. Write data to IICAn, not setting the WRELn bit, in order to cancel a clock stretch state during transmission by a slave device.
Remark n = 0

The meanings of <3> to <12> in (3) Address ~ data ~ data in Figure 16 - 46 are explained below.

<3> In the slave device if the address received matches the address (SVAn value) of a slave device Note, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a clock stretch status (SCLAn = 0) and issues an interrupt (INTIICAn: address match) Note.

<5> The master device changes the timing of the clock stretch status to the 8th clock (WTIMn = 0).

<6> The slave device writes the data to transmit to the IICA shift register n (IICAn) and releases the clock stretch status that it set by the slave device.

<7> The master device releases the clock stretch status (WRELn = 1) and starts transferring data from the slave device to the master device.

<8> The master device sets a clock stretch status (SCLAn = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICAn: end of transfer). Because of ACKEn = 1 in the master device, the master device then sends an ACK by hardware to the slave device.

<9> The master device reads the received data and releases the clock stretch status (WRELn = 1).

<10>The ACK is detected by the slave device (ACKDn = 1) at the rising edge of the 9th clock.

<11>The slave device set a clock stretch status (SCLAn = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICAn: end of transfer).

<12>By the slave device writing the data to transmit to the IICAn register, the clock stretch status set by the slave device is released. The slave device then starts transferring data to the master device.

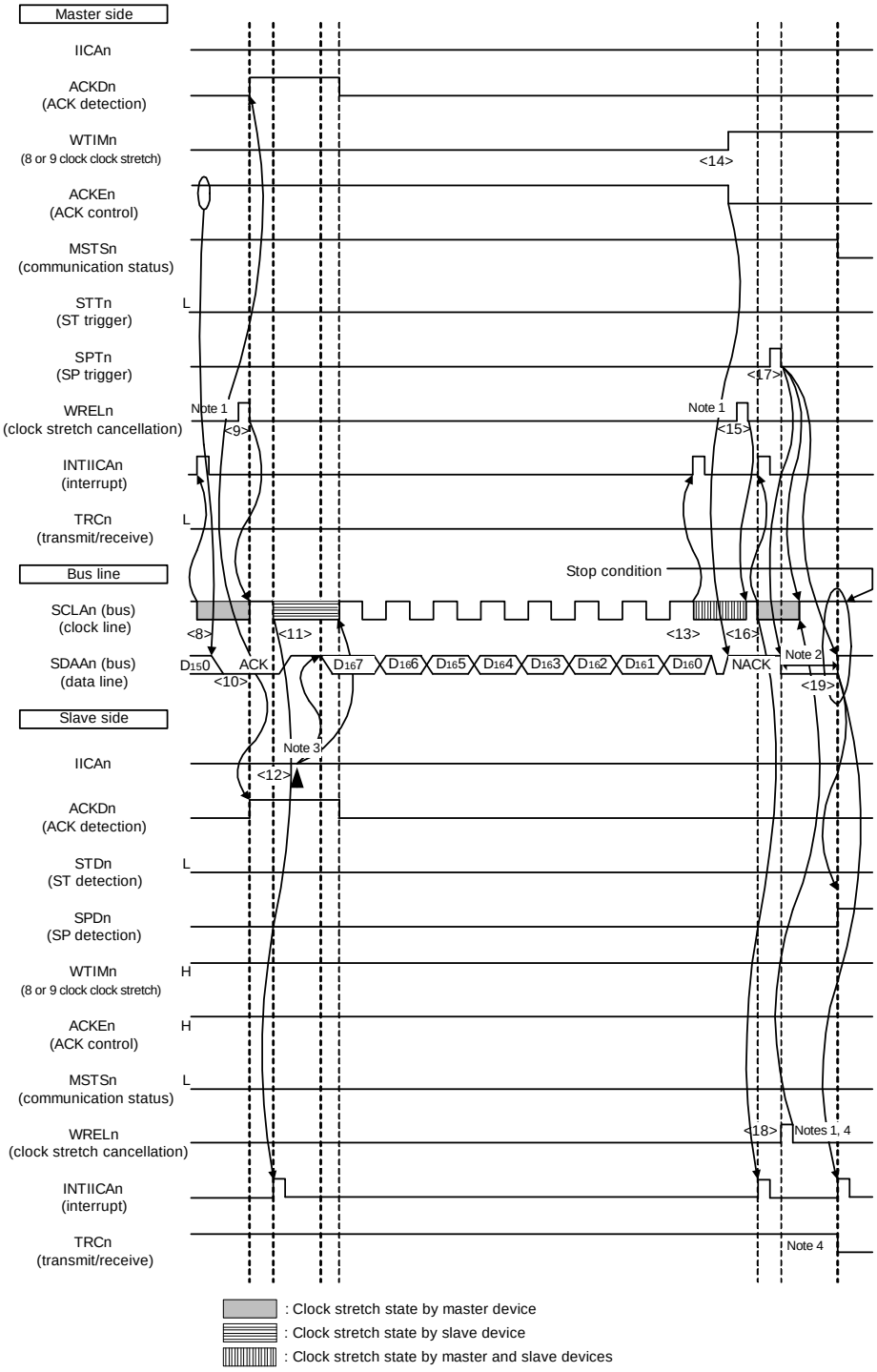
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set a clock stretch status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <19> in Figures 16 - 45 to 16 - 47 represent the entire procedure for communicating data using the I^2C bus. Figure 16 - 45 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 16 - 46 (3) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 16 - 47 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Remark 2. n = 0

Figure 16 - 47 Example of Slave to Master Communication
(When 8-Clock and 9-Clock Clock Stretch Is Selected for Master, 9-Clock Clock Stretch Is Selected for Slave) (3/3)

(3) Data ~ data ~ stop condition



Note 1. To cancel a clock stretch state, write “FFH” to IICAn or set the WRELn bit.

Note 2. Make sure that the time between the rise of the SCLAn pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

Note 3. Write data to IICAn, not setting the WRELn bit, in order to cancel a clock stretch state during transmission by a slave device.

Note 4. If a clock stretch state during transmission by a slave device is canceled by setting the WRELn bit, the TRCn bit will be cleared.

Remark n = 0

The meanings of <8> to <19> in (3) Data ~ data ~ stop condition in Figure 16 - 47 are explained below.

<8> The master device sets a clock stretch status (SCLAn = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICAn: end of transfer). Because of ACKEn = 0 in the master device, the master device then sends an ACK by hardware to the slave device.

<9> The master device reads the received data and releases the clock stretch status (WRELn = 1).

<10>The ACK is detected by the slave device (ACKDn = 1) at the rising edge of the 9th clock.

<11>The slave device set a clock stretch status (SCLAn = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICAn: end of transfer).

<12>By the slave device writing the data to transmit to the IICA register, the clock stretch status set by the slave device is released. The slave device then starts transferring data to the master device.

<13>The master device issues an interrupt (INTIICAn: end of transfer) at the falling edge of the 8th clock, and sets a clock stretch status (SCLAn = 0). Because ACK control (ACKEn = 1) is performed, the bus data line is at the low level (SDAAn = 0) at this stage.

<14>The master device sets NACK as the response (ACKEn = 0) and changes the timing at which it sets the clock stretch status to the 9th clock stretch (WTIMn = 1).

<15>If the master device releases the clock stretch status (WRELn = 1), the slave device detects the NACK (ACKDn = 0) at the rising edge of the 9th clock stretch.

<16>The master device and slave device set a clock stretch status (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<17> When the master device issues a stop condition (SPTn = 1), the bus data line is cleared (SDAAn = 0) and the master device releases the clock stretch status. The master device then waits until the bus clock line is set (SCLAn = 1).

<18> The slave device acknowledges the NACK, halts transmission, and releases the clock stretch status (WRELn = 1) to end communication. Once the slave device releases the clock stretch status, the bus clock line is set (SCLAn = 1).

<19> Once the master device recognizes that the bus clock line is set (SCLAn = 1) and after the stop condition setup time has elapsed, the master device sets the bus data line (SDAAn = 1) and issues a stop condition (i.e. SCLAn =1 changes SDAAn from 0 to 1). The slave device detects the generated stop condition and slave device issue an interrupt (INTIICAn: stop condition).

Remark 1. <1> to <19> in Figures 16 - 45 to 16 - 47 represent the entire procedure for communicating data using the I^2C bus. Figure 16 - 45 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 16 - 46 (3) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 16 - 47 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Remark 2. n = 0

CHAPTER 17 USB 2.0 FUNCTION MODULE (USB)

17.1 Functions of USB 2.0 Function Module

The RL78/L1C incorporates a USB 2.0 function module (USB module) compliant to USB (Universal Serial Bus) Specification 2.0. The USB module provides capabilities as a function controller which supports full-speed (12 Mbps) and low-speed (1.5 Mbps) transfer.

The USB module can also detect battery charging (hereafter BC) connection during function controller operation compliant to Battery Charging Specification Revision 1.2.

Table 17 - 1 lists the USB Module Specifications.

Table 17 - 1 USB Module Specifications

Item	Specifications
Features	• USB Device Controller (UDC) and transceiver for USB 2.0 are incorporated. • The USB function controller is incorporated.
	• One port is provided. • Full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps) are supported • Control transfer stage control function • Device state control function • Auto response function for SET_ADDRESS request • SOF interpolation function • On-chip D+/D- pin pull-up resistor
Communication data transfer type	• Control transfer • Bulk transfer • Interrupt transfer
Pipe configuration	• 448 bytes of buffer memory for USB communications is provided. • Up to five pipes can be selected (including the default control pipe). • Usable pipe numbers are 0, 4 to 7. • Endpoint numbers can be assigned flexibly to PIPE4 to PIPE7.
	Transfer conditions that can be set for each pipe: • PIPE0: Control transfer only (default control pipe: DCP) Buffer size: 8, 16, 32, or 64 bytes (single buffer) • PIPE4, PIPE5: Bulk transfer only Buffer size: 8, 16, 32, or 64 bytes (double buffer can be specified) • PIPE6, PIPE7: Interrupt transfer only Buffer size: 1 to 64 bytes (single buffer)
Others	• Reception ending function using transaction count • Function that changes the BRDY interrupt event notification timing (BFRE) • NAK setting function for response PID generated by end of transfer (SHTNAK) • Automatic buffer memory clearing function after reading data in the pipe specified by the DnFIFO (n = 0, 1) port (DCLRM) • Battery charging is supported (Battery Charging Specification Revision 1.2) - Function (portable device) BC connection detection function (one port) is supported • Optional functions for battery charging connection detection are provided - USB port voltage detection function (16 stages)

17.2 Configuration of USB 2.0 Function Module

The USB module consists of the following hardware.

- USB 2.0 function controller
  This controls the function supporting full-speed (12 Mbps) and low-speed (1.5 Mbps) transfer.

- USB transceiver
  This is a USB transceiver of a dedicated port (USB port) for the function.
  This transceiver contains a pull-up resistor (for the function) for transfer rate detection.
  An external power supply (via the UV_{BUS} pin) is used as the power supply for the USB transceiver (hereafter USB power supply).

- Buffer memory for USB communications and FIFO/memory controller
  Up to five pipes can be used. End point numbers can be assigned flexibly to PIPE4 to PIPE7 according to peripheral devices and a user system for communication.

- Battery charging detection/controller
  This processes BC connection detection during function controller operation compliant to Battery Charging Specification Revision 1.2.

- Various registers
  There are various registers for control, monitoring, and transmitting/receiving data.
  See **Tables 17 - 3** and **17 - 4**.

- Various pins
  USB port I/O pins (UDP, UDM)
  VBUS input pin (UV_{BUS})
  USB power supply stabilization capacitance connection pin (U_{REGC})
  See **Table 17 - 2** for details.

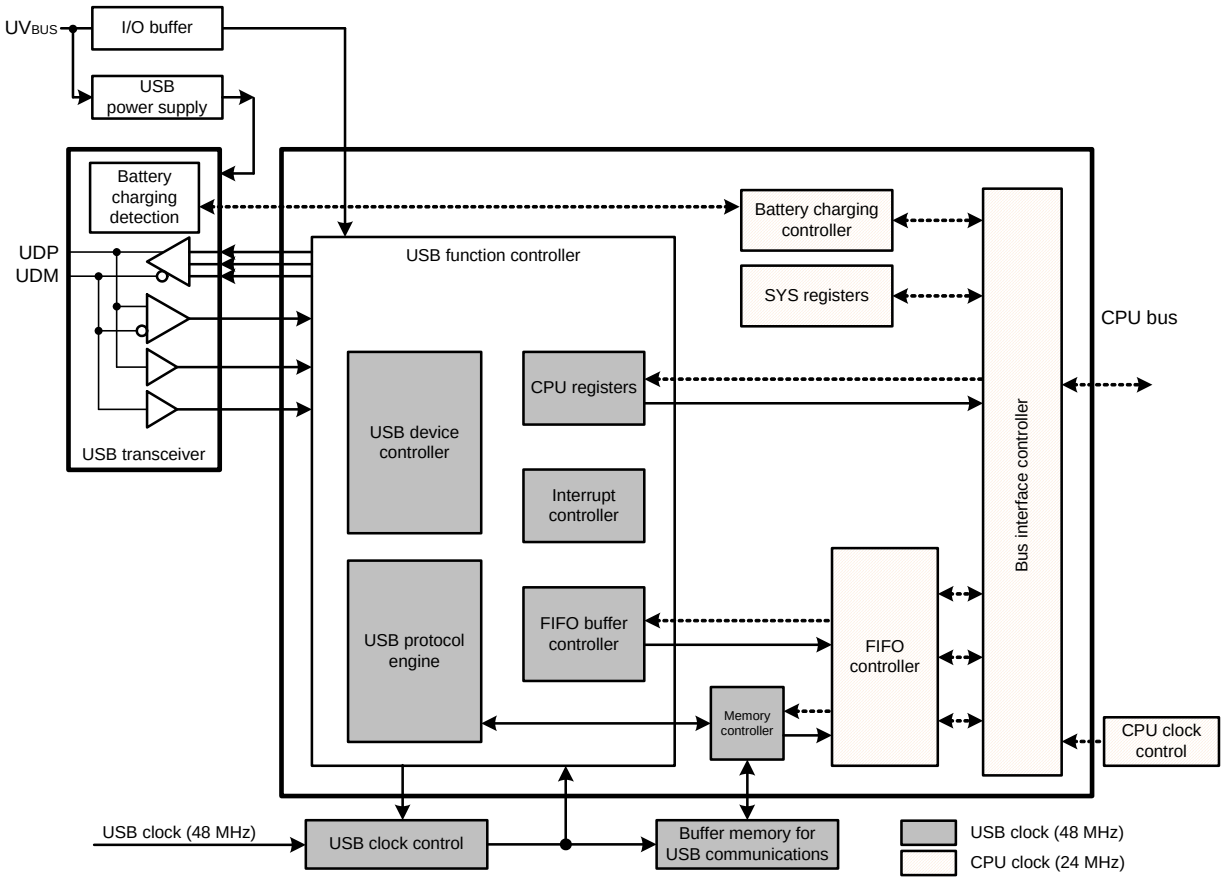
- Others
  The clock controller operates or stops various clocks to be used by the USB module and divides their frequencies.
  The bus interface controller controls access between the CPU, DTC, and each register of the USB module.

Table 17 - 2 lists the USB Module I/O Pins and Figure 17 - 1 shows the Block Diagram of USB Module.

Table 17 - 2 USB Module I/O Pins

Pin Name	I/O	Function
UDP	I/O	D+ I/O pin of USB port This pin should be connected to the D+ pin of the USB bus.
UDM	I/O	D- I/O pin of USB port This pin should be connected to the D- pin of the USB bus.
UV_{BUS}	Input	USB cable connection monitor pin and positive power supply pin for the USB transceiver This pin should be connected to VBUS of the USB bus.
U_{REGC}	Output	USB power supply stabilization capacitance connection pin A capacitor (0.33 µF) should be connected between this pin and V_{SS}.

Figure 17 - 1 Block Diagram of USB Module



17.3 Registers Used in USB 2.0 Function Module

Tables 17 - 3 and 17 - 4 list the USB Registers.

Table 17 - 3 USB Registers (1/2)

Register Name	Symbol
System configuration control register	SYSCFG
System configuration status register 0	SYSSTS0
Device state control register 0	DVSTCTR0
DTC0 to FIFO pin configuration register	DTC0PCFG
DTC1 to FIFO pin configuration register	DTC1PCFG
CFIFO port register	CFIFOM
D0FIFO port register	D0FIFOM
D1FIFO port register	D1FIFOM
CFIFO port select register	CFIFOSEL
CFIFO port control register	CFIFOCTR
D0FIFO port selection register	D0FIFOSEL
D0FIFO port control register	D0FIFOCTR
D1FIFO port selection register	D1FIFOSEL
D1FIFO port control register	D1FIFOCTR
Interrupt enable register 0	INTENB0
Interrupt enable register 1	INTENB1
BRDY interrupt enable register	BRDYENB
NRDY interrupt enable register	NRDYENB
BEMP interrupt enable register	BEMPENB
SOF output configuration register	SOFCFG
Interrupt status register 0	INTSTS0
Interrupt status register 1	INTSTS1
BRDY interrupt status register	BRDYSTS
NRDY interrupt status register	NRDYSTS
BEMP interrupt status register	BEMPSTS
Frame number register	FRMNUM
USB address register	USBADDR
USB request type register	USBREQ
USB request value register	USBVAL
DTC transfer D0FIFO port register	D0FIFOD00
DTC transfer D1FIFO port register	D1FIFOD00
USB request index register	USBINDX
USB request length register	USBLENG
DCP configuration register	DCPCFG
DCP maximum packet size register	DCPMAXP
DCP control register	DCPCTR
Pipe window selection register	PIPESEL
Pipe configuration register	PIPECFG
Pipe maximum packet size register	PIPEMAXP
Pipe 4 control register	PIPE4CTR
Pipe 5 control register	PIPE5CTR

Table 17 - 4 USB Registers (2/2)

Register Name	Symbol
Pipe 6 control register	PIPE6CTR
Pipe 7 control register	PIPE7CTR
Pipe 4 transaction counter enable register	PIPE4TRE
Pipe 4 transaction counter register	PIPE4TRN
Pipe 5 transaction counter enable register	PIPE5TRE
Pipe 5 transaction counter register	PIPE5TRN
BC control register 0	USBBCCTRL0
BC option control register 0	USBBCOPT0
USB clock selection register	UCKSEL
USB module control register	USBMC

Table 17 - 5 Registers Initialized by Writing USBE = 0

Register	Symbol
SYSSTS0	LNST1, LNST0
DVSTCTR0	RHST2 to RHST0
INTSTS0	DVSQ2 to DVSQ0
USBADDR	USBADDR
USBREQ	BREQUEST, BMREQUESTTYPE
USBVAL	WVALUE
USBINDX	WINDEX
USBLENG	WLENGTH

17.3.1 System configuration control register (SYSCFG)

Figure 17 - 2 Format of System configuration control register (SYSCFG)

Address: F0600H, F0601H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYSCFG	0	0	0	0	0	SCKE	0	CNEN	0	0	0	DPRPU	DMRPU	0	0	USBE

SCKE	USB module clock enable
0	Stops supplying the clock signal to the USB module.
1	Enables supplying the clock signal to the USB module.
Stops or enables supplying 48-MHz clock signals to the USB module. When this bit is 0, only the SYSCFG, DTC0PCFG, and DTC1PCFG registers can be read from and written to. The other registers in the USB module cannot be read from or written to. After writing 1 to the SCKE bit, be sure to read this bit to confirm that it is set to 1.	

CNEN	USB port single end receiver enable
0	Single end receiver operation is disabled.
1	Single end receiver operation is enabled.
Enables or disables the single end receiver. Setting the CNEN bit to 1 allows the USB module to enable the single end receiver of USB port and set the LNST bit to monitor the status of the D+/D- lines. The CNEN bit is used to monitor LNST when the USB module operates as a portable device for battery charging.	

DPRPU	USB port D+ line resistor control Note
0	Pulling up the line is disabled.
1	Pulling up the line is enabled.
Enables or disables pulling up the D+ line. Setting the DPRPU bit to 1 allows the USB module to enable pulling up the D+ line of USB port, thus notifying the USB host of connection as a full-speed device. Modifying the DPRPU bit from 1 to 0 allows the USB module to disable pulling up the D+ line of USB port, thus notifying the USB host of disconnection.	

DMRPU	USB port D- line resistor control Note
0	Pulling up the line is disabled.
1	Pulling up the line is enabled.
Enables or disables pulling up the D- line. Setting the DMRPU bit to 1 allows the USB module to enable pulling up the D- line of USB port, thus notifying the USB host of connection as a low-speed device. Modifying the DMRPU bit from 1 to 0 allows the USB module to disable pulling up the D- line of USB port, thus notifying the USB host of disconnection.	

USBE	USB module operation enable
0	USB module operation is disabled.
1	USB module operation is enabled.
Enables or disables operation of the USB module. Modifying the USBE bit from 1 to 0 initializes some register bits as listed in Tables 17 - 5. This bit should be modified while SCKE is 1.	

Note Setting the DMRPU and DPRPU bits simultaneously to 1 (pulling up the line is enabled) is prohibited.

17.3.2 System configuration status register 0 (SYSSTS0)

Figure 17 - 3 Format of System configuration status register 0 (SYSSTS0)

Address: F0604H, F0605H After reset: 00000000 000000XXB R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYSSTS0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	LNST1	LNST0

LNST1	LNST0	USB port USB data line status monitor
0	0	SE0
0	1	J-State (full speed)/K-State (low speed)
1	0	K-State (full speed)/J-State (low speed)
1	1	SE1
These bits indicate the status of the USB data bus lines (D+ and D-).		

17.3.3 Device state control register 0 (DVSTCTR0)

Figure 17 - 4 Format of Device state control register 0 (DVSTCTR0)

Address: F0608H, F0609H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DVSTCTR0	0	0	0	0	0	0	0	WKUP	0	0	0	0	0	RHST2	RHST1	RHST0

WKUP	USB port wakeup output
0	Remote wakeup signal is not output.
1	Remote wakeup signal is output. Note

Enables or disables outputting the remote wakeup signal (resume signal) to the USB bus.
The USB module controls the output time of a remote wakeup signal. When this bit is set to 1, the USB module clears this bit to 0 after outputting the 10-ms K-state.
According to the USB Specifications, the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is sent. Even if the USB module writes 1 to this bit right after detection of the suspended state, the K-state will be output after 2 ms.
Do not write 1 to this bit, unless the device state is in the suspended state (DVSQ2 to DVSQ0 in the INTSTS0 register = 1xxB) and the USB host enables the remote wakeup signal. When this bit is set to 1, the internal clock must not be stopped even in the suspended state (write 1 to WKUP while SCKE in the SYSCFG register = 1).

RHST2	RHST1	RHST0	Port 0 USB bus reset status
0	0	0	Communication speed not determined
1	x	x	USB bus reset in progress
0	0	1	Low-speed connection
0	1	0	Full-speed connection

The RHST[2:0] bits indicate the status of the USB bus reset.
A DVST interrupt is generated as soon as the USB module detects the USB bus reset and then these bits are fixed to 010B.

Note Only 1 can be written.

17.3.4 DTCn to FIFO Pin Configuration Register (DTCnPCFG)

Figure 17 - 5 Format of DTCn to FIFO Pin Configuration Register (DTCnPCFG)

Address: F06A8H, F06A9H (DTC0PCFG), F06ACH, F06ADH (DTC1PCFG) After reset: 0000H R/W

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

DnDBLK	DTC block transfer mode
0	Cycle steal transfer mode
1	Block transfer mode

17.3.5 CFIFO port register (CFIFOM), DnFIFO port register (DnFIFOM) (n = 0, 1)

Figure 17 - 6 Format of CFIFO port register (CFIFOM)

Address: F0614H, F0615H After reset: 0000H R/W

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

CFIFO[15:0]	CFIFO port
Accessing the CFIFO bits allow reading the received data from the FIFO buffer or writing the transmit data to the FIFO buffer. The CFIFO port register can be accessed only while the FRDY bit in the CFIFO port control register (CFIFOCTR) is 1. The valid bits in the CFIFO port register depend on the settings of the corresponding MBW and BIGEND bits as shown in Tables 17 - 6 and 17 - 7.	

Caution 1. The FIFO buffer for DCP (control transfer) cannot be accessed using a DTC transfer.

Caution 2. When using functions specific to the FIFO port, the pipe number (selected pipe) specified by the CURPIPE bits cannot be changed.

Caution 3. The same pipe should not be assigned to different FIFO ports.

Caution 4. There are two FIFO buffer states: the access right is on the CPU side and it is on the SIE side. When the FIFO buffer access right is on the SIE side, the FIFO buffer cannot be accessed from the CPU.

Table 17 - 6 Endian Operation in 16-Bit Access

BIGEND Bit in CFIFO Port Register, DnFIFO Port Register	Bits 15 to 8	Bits 7 to 0
0	N + 1 data	N + 0 data
1	N + 0 data	N + 1 data

Table 17 - 7 Endian Operation in 8-Bit Access

BIGEND Bit in CFIFO Port Register, DnFIFO Port Register	Bits 15 to 8	Bits 7 to 0
0	Access prohibited	N + 0 data
1	Access prohibited	N + 0 data

Figure 17 - 7 Format of DnFIFO Port Register (DnFIFOM) (n = 0, 1)

Address: F0618H, F0619H (D0FIFOM), F061CH, F061DH (D1FIFOM) After reset: 0000H R/W

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

DnFIFO[15:0]	FIFO port
Addresses for CPU transfers using the DnFIFO port. Accessing the DnFIFO bits allow reading the received data from the FIFO buffer or writing the transmit data to the FIFO buffer. The DnFIFO port register can be accessed only while the FRDY bit in the DnFIFO port control register (DnFIFOCTR) is 1. The valid bits in the DnFIFO port register depend on the settings of the corresponding MBW and BIGEND bits as shown in Tables 17 - 6 and 17 - 7.	

Caution 1. The FIFO buffer for DCP (control transfer) cannot be accessed. This register cannot be used for the addresses for DTC transfers.

Caution 2. When using functions specific to the FIFO port, the pipe number (selected pipe) specified by the CURPIPE bits cannot be changed.

Caution 3. The same pipe should not be assigned to different FIFO ports.

Caution 4. There are two FIFO buffer states: the access right is on the CPU side and it is on the SIE side. When the FIFO buffer access right is on the SIE side, the FIFO buffer cannot be accessed from the CPU.

17.3.6 DTC transfer DnFIFO port register (DnFIFOD00) (n = 0, 1)

Figure 17 - 8 Format of DTC transfer DnFIFO port register (DnFIFOD00) (n = 0, 1)

Address: F0580H, F0581H (D0FIFOD00), F05C0H, F05C1H (D1FIFOD00) After reset: 0000H R/W

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

DnFIFO[15:0]	DTC transfer DnFIFO port
Addresses for DTC transfers using the DnFIFO port. The valid bits in the DTC transfer DnFIFO port register depend on the settings of the corresponding MBW and BIGEND bits as shown in Tables 17 - 6 and 17 - 7.	

Caution 1. Accessing this register from the CPU is prohibited.

Caution 2. When using functions specific to the FIFO port, the pipe number (selected pipe) specified by the CURPIPE bits cannot be changed.

Caution 3. The same pipe should not be assigned to different FIFO ports.

17.3.7 CFIFO port select register (CFIFOSEL), DnFIFO port select register (DnFIFOSEL) (n = 0, 1)

Figure 17 - 9 Format of CFIFO port select register (CFIFOSEL)

Address: F0620H, F0621H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFIFOSEL	RCNT	REW	0	0	0	MBW	0	BIGEND	0	0	ISEL	0	CURPIPE3	CURPIPE2	CURPIPE1	CURPIPE0

RCNT	Read count mode
0	When all of the receive data has been read from the CFIFO, 0 is written to the DTLN bit. (In double buffer mode, the DTLN bit value is cleared when all the data has been read from only a single plane.)
1	The DTLN bit is decremented each time the receive data is read from the CFIFO.
Specifies the read mode for the value in the DTLN[8:0] bits in the CFIFOCTR register.	

REW	Buffer pointer rewind
0	Disable (the buffer pointer is not rewound.) Note 1
1	The buffer pointer is rewound.
The buffer pointer is rewound when REW = 1. When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed). Do not set the REW bit to 1 simultaneously with modifying the CURPIPE3 to CURPIPE0 bits. Before setting the REW bit to 1, be sure to check that the FRDY bit is 1. To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.	

MBW	CFIFO port access bit width
0	8-bit width
1	16-bit width
Specifies the bit width for accessing the CFIFO port. When the selected pipe is in the receiving direction, once reading data is started after setting this bit, this bit should not be modified until all the data has been read. When the selected pipe is in the receiving direction, set the CURPIPE3 to CURPIPE0 and MBW bits simultaneously. If the size of data to be read is an odd number of bytes when 16-bit width is selected, delete unnecessary bytes after reading the data in words. When the selected pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the buffer memory. An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.	

BIGEND	CFIFO port endian control
0	Little endian
1	Big endian
Specifies the byte endian for the CFIFO port.	

ISEL	CFIFO port access direction when DCP is selected
0	Reading from the buffer memory is selected
1	Writing to the buffer memory is selected
After writing to the ISEL bit with the DCP being a selected pipe, read this bit to check that the written value agrees with the read value before proceeding to the next process. Set this bit and the CURPIPE3 to CURPIPE0 bits simultaneously.	

CURPIPE3	CURPIPE2	CURPIPE1	CURPIPE0	CFIFO port access pipe specification Note 2
0	0	0	0	DCP (Default control pipe)
0	1	0	0	Pipe 4
0	1	0	1	Pipe 5
0	1	1	0	Pipe 6
0	1	1	1	Pipe 7
Other than above				Do not set.
The CURPIPE3 to CURPIPE0 bits specify the pipe number using which data is read or written through the CFIFO port. After writing to the CURPIPE3 to CURPIPE0 bits, read these bits to check that the written value agrees with the read value before proceeding to the next process. Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective, thus enabling continuous access.				

Note 1. Only 0 can be read.

Note 2. The same pipe number should not be set by the CURPIPE3 to CURPIPE0 bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers.

Figure 17 - 10 Format of DnFIFO port select register (DnFIFOSEL) (n = 0, 1)

Address: F0628H, F0629H (D0FIFOSEL), F062CH, F062DH (D1FIFOSEL) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DnFIFOSEL	RCNT	REW	DCLRM	DREQE	0	MBW	0	BIGEND	0	0	0	0	CURPIPE3	CURPIPE2	CURPIPE1	CURPIPE0

RCNT	Read count mode
0	When all of the receive data has been read from the DnFIFO, 0 is written to the DTLN[8:0] bits. (In double buffer mode, the DTLN[8:0] bit value is cleared when all the data has been read from only a single plane.)
1	The DTLN[8:0] bits are decremented each time the receive data is read from the D0FIFO.
Specifies the read mode for the value in the DTLN[8:0] bits in the DnFIFOCTR register. When accessing DnFIFO with the BFRE bit set to 1, set the RCNT bit to 0.	

REW	Buffer pointer rewind
0	Disable (the buffer pointer is not rewound.) Note 1
1	The buffer pointer is rewound.
Specifies whether or not to rewind the buffer pointer. When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed). Do not set REW to 1 simultaneously with modifying the CURPIPE bits. Before setting REW to 1, be sure to check that FRDY is 1. When accessing the DnFIFO with the BFRE bit set to 1, do not set the REW bit to 1 while reading of the short packet data is completed. To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.	

DCLRM	Auto buffer memory clear mode accessed after specified pipe data is read
0	Auto buffer clear mode is disabled.
1	Auto buffer clear mode is enabled.
Enables or disables the buffer memory to be cleared automatically after data has been read out using the selected pipe. With the DCLRM bit set to 1, the USB module sets BCLR to 1 for the FIFO buffer of the selected pipe on receiving a zero-length packet while the FIFO buffer assigned to the selected pipe is empty, or on receiving a short packet and reading the data while BFRE in the PIPECFG register is 1. When using the USB module with BRDYM in the SOFCFG register set to 1, set the DCLRM bit to 0.	

DREQE	DTC transfer request enable
0	DTC transfer request disabled
1	DTC transfer request enabled
Enables or disables the DTC transfer request to be issued. Before setting the DREQE bit to 1 to enable the DTC transfer request to be issued, set the CURPIPE bits. When modifying the setting of the CURPIPE bits, set the DREQE bit to 0 first.	

MBW	DnFIFO (n = 0, 1) port access bit width
0	8-bit width
1	16-bit width
Specifies the bit width for accessing the DnFIFO port. When the selected pipe is in the receiving direction, once reading data is started after setting the MBW bit, this bit should not be modified until all the data has been read. When the selected pipe is in the receiving direction, set the CURPIPE and MBW bits simultaneously. If the size of data to be read is an odd number of bytes when 16-bit width is selected, delete unnecessary bytes after reading the data in words. When the selected pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the buffer memory. An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.	

BIGEND	DnFIFO (n = 0, 1) port endian control
0	Little endian
1	Big endian
Specifies the byte endian for the DnFIFO port.	

CURPIPE3	CURPIPE2	CURPIPE1	CURPIPE0	DnFIFO (n = 0, 1) port access pipe specification Note 2
0	0	0	0	No pipe specified
0	1	0	0	Pipe 4
0	1	0	1	Pipe 5
0	1	1	0	Pipe 6
0	1	1	1	Pipe 7
Other than above				Do not set.
The CURPIPE3 to CURPIPE0 bits specify the pipe number using which data is read or written through the DnFIFO port. After writing to CURPIPE3 to CURPIPE0 bits, then read these bits to check that the written value agrees with the read value before proceeding to the next process. Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective, thus enabling continuous access.				

Note 1. Only 0 can be read.

Note 2. The same pipe number should not be set by the CURPIPE3 to CURPIPE0 bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers.

17.3.8 CFIFO port control register (CFIFOCTR), DnFIFO port control register (DnFIFOCTR) (n = 0, 1)

Figure 17 - 11 Format of CFIFO port control register (CFIFOCTR)

Address: F0622H, F0623H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFIFOCTR	BVAL	BCLR	FRDY	0	0	0	0	DTLN[8:0]								

BVAL	Buffer memory valid flag
0	Invalid
1	Writing ended Note 1

This bit should be set to 1 when data has been completely written to the FIFO buffer on the CPU side for the pipe selected using the CURPIPE3 to CURPIPE0 bits (selected pipe).
When the selected pipe is in the transmitting direction, set the BVAL bit to 1 in the following cases. Then, the USB module switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.
To transmit a short packet, set the BVAL bit to 1 after data has been written.
To transmit a zero-length packet, set the BVAL bit to 1 while the FIFO buffer is empty.
When data of the maximum packet size has been written for the pipe in continuous transfer mode, the USB module sets the BVAL bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.
Writing 1 to the BVAL bit should be done while FRDY is 1 (set by the USB module).
When the selected pipe is in the receiving direction, do not set the BVAL bit to 1.

BCLR	CPU buffer clear Note 2
0	Invalid
1	Clears the buffer memory on the CPU side.

This bit should be set to 1 to clear the FIFO buffer on the CPU side for the selected pipe.
When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USB module clears only one plane of the FIFO buffer even when both planes are read-enabled.
When the selected pipe is the DCP, setting BCLR to 1 allows the USB module to clear the FIFO buffer regardless of whether the FIFO buffer is on the CPU side or SIE side. To clear the buffer on the SIE side, set the PID bits for the DCP control register to NAK before setting BCLR to 1.
When the selected pipe is in the transmitting direction, if 1 is written to the BVAL and BCLR bits simultaneously, the USB module clears the data that has been written before it, enabling transmission of a zero-length packet.
When the selected pipe is not the DCP, writing 1 to the BCLR bit should be done while FRDY in the FIFO port control register is 1 (set by the USB module).

FRDY	FIFO port ready
0	FIFO port access is disabled.
1	FIFO port access is enabled.

Indicates whether the FIFO port can be accessed by the CPU.
In the following cases, the USB module sets FRDY to 1 but data cannot be read via the FIFO port because there is no data to be read. In these cases, set BCLR to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.
- A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty.
- A short packet is received and the data is completely read while BFRE in the PIPECFG register is 1.

DTLN[8:0]	Receive data length

The DTLN[8:0] bits indicate the length of the receive data.
While the FIFO buffer is being read, the DTLN[8:0] bits indicate different values depending on the RCNT bit value as described below.
- RCNT = 0
  The USB module sets the DTLN[8:0] bits to indicate the length of the receive data until the CPU has read all the received data from a single FIFO buffer plane.
  While BFRE in the PIPECFG register is 1, these bits retain the length of the receive data until BCLR is set to 1 even after all the data has been read.
- RCNT = 1
  The USB module decrements the value indicated by these bits each time data is read from the FIFO buffer. (The value is decremented by one when MBW is 0, and by two when MBW is 1.)
  The USB module sets these bits to 0 when all the data has been read from one FIFO buffer plane. However, in double buffer mode, if data has been received in one FIFO buffer plane before all the data has been read from the other plane, the USB module sets the DTLN[8:0] bits to indicate the length of the receive data in the former plane when all the data has been read from the latter plane.

Note 1. Only 1 can be written.
Note 2. Only 0 can be read and 1 can be written.

Figure 17 - 12 Format of DnFIFO Port Control Register (DnFIFOCTR) (n = 0, 1)

Address: F062AH, F062BH (D0FIFOCTR), F062EH, F062FH (D1FIFOCTR) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DnFIFOCTR	BVAL	BCLR	FRDY	0	0	0	0	DTLN[8:0]								

BVAL	Buffer memory valid flag
0	Invalid
1	Writing ended Note 1

This bit should be set to 1 when data has been completely written to the FIFO buffer on the CPU side for the pipe selected using the CURPIPE3 to CURPIPE0 bits (selected pipe).
When the selected pipe is in the transmitting direction, set the BVAL bit to 1 in the following cases. Then, the USB module switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.
• To transmit a short packet, set the BVAL bit to 1 after data has been written.
• To transmit a zero-length packet, set the BVAL bit to 1 while the FIFO buffer is empty.
When data of the maximum packet size has been written for the pipe in continuous transfer mode, the USB module sets the BVAL bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.
Writing 1 to the BVAL bit should be done while FRDY is 1 (set by the USB module).
When the selected pipe is in the receiving direction, do not set the BVAL bit to 1.

BCLR	CPU buffer clear Note 2
0	Invalid
1	Clears the buffer memory on the CPU side.

This bit should be set to 1 to clear the FIFO buffer on the CPU side for the selected pipe.
When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USB module clears only one plane of the FIFO buffer even when both planes are read-enabled.
When the selected pipe is in the transmitting direction, if 1 is written to the BVAL and BCLR bits simultaneously, the USB module clears the data that has been written before it, enabling transmission of a zero-length packet.
When the selected pipe is not the DCP, writing 1 to the BCLR bit should be done while FRDY is 1 (set by the USB module).

FRDY	FIFO port ready
0	FIFO port access is disabled.
1	FIFO port access is enabled.

Indicates whether the FIFO port can be accessed by the CPU or DTC.
In the following cases, the USB module sets FRDY to 1 but data cannot be read via the FIFO port because there is no data to be read. In these cases, set BCLR to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.
• A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty.
• A short packet is received and the data is completely read when BFRE is set to 1.

DTLN[8:0]	Receive data length
The DTLN[8:0] bits indicate the length of the receive data. While the FIFO buffer is being read, the DTLN[8:0] bits indicate different values depending on the RCNT bit value as described below. • RCNT = 0 The USB module sets the DTLN[8:0] bits to indicate the length of the receive data until the CPU has read all the received data from a single FIFO buffer plane. While BFRE in the PIPECFG register is 1, these bits retain the length of the receive data until BCLR is set to 1 even after all the data has been read. • RCNT = 1 The USB module decrements the value indicated by these bits each time data is read from the FIFO buffer. (The value is decremented by one when MBW is 0, and by two when MBW is 1.) The USB module sets DTLN to 0 when all the data has been read from one FIFO buffer plane. However, in double buffer mode, if data has been received in one FIFO buffer plane before all the data has been read from the other plane, the USB module sets these bits to indicate the length of the receive data in the former plane when all the data has been read from the latter plane.	

Note 1. Only 1 can be read.

Note 2. Only 0 can be read and 1 can be written.

17.3.9 Interrupt enable register 0 (INTENB0)

Figure 17 - 13 Format of Interrupt enable register 0 (INTENB0)

Address: F0630H, F0631H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENB0	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	0	0	0	0	0	0	0	0

VBSE	VBUS interrupt enable
0	Interrupt output disabled
1	Interrupt output enabled
Enables or disables the USB interrupt output when the VBINT interrupt is detected.	

RSME	Resume interrupt enable
0	Interrupt output disabled
1	Interrupt output enabled
Enables or disables the USB interrupt output when the RESM interrupt is detected.	

SOFE	Frame number update interrupt enable
0	Interrupt output disabled
1	Interrupt output enabled
Enables or disables the USB interrupt output when the SOFR interrupt is detected.	

DVSE	Device state transition interrupt enable
0	Interrupt output disabled
1	Interrupt output enabled
Enables or disables the USB interrupt output when the DVST interrupt is detected.	

CTRE	Control transfer stage transition interrupt enable
0	Interrupt output disabled
1	Interrupt output enabled
Enables or disables the USB interrupt output when the CTRT interrupt is detected.	

BEMPE	Buffer empty interrupt enable
0	Interrupt output disabled
1	Interrupt output enabled
Enables or disables the USB interrupt output when the BEMP interrupt is detected.	

NRDYE	Buffer Not Ready Response Interrupt Enable
0	Interrupt output disabled
1	Interrupt output enabled
Enables or disables the USB interrupt output when the NRDY interrupt is detected.	

BRDYE	Buffer Ready Interrupt Enable
0	Interrupt output disabled
1	Interrupt output enabled
Enables or disables the USB interrupt output when the BRDY interrupt is detected.	

17.3.10 Interrupt enable register 1 (INTENB1)

Figure 17 - 14 Format of Interrupt enable register 1 (INTENB1)

Address: F0632H, F0633H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENB1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PDDETINTE

PDDETINTE	Portable Device detection interrupt enable
0	Interrupt output disabled
1	Interrupt output enabled
Enables or disables the USB interrupt output when the PDDETINT interrupt is detected.	

17.3.11 BRDY interrupt enable register (BRDYENB)

Figure 17 - 15 Format of BRDY interrupt enable register (BRDYENB)

Address: F0636H, F0637H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRDYENB	0	0	0	0	0	0	0	0	PIPE7BRDYE	PIPE6BRDYE	PIPE5BRDYE	PIPE4BRDYE	0	0	0	PIPE0BRDYE

PIPEnBRDYE	BRDY interrupt enable for PIPEn
0	Interrupt output disabled
1	Interrupt output enabled

Remark n = 7 to 4, 0

17.3.12 NRDY interrupt enable register (NRDYENB)

Figure 17 - 16 Format of NRDY interrupt enable register (NRDYENB)

Address: F0638H, F0639H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NRDYENB	0	0	0	0	0	0	0	0	PIPE7NRDYE	PIPE6NRDYE	PIPE5NRDYE	PIPE4NRDYE	0	0	0	PIPE0NRDYE

PIPEnNRDYE	NRDY interrupt enable for PIPEn
0	Interrupt output disabled
1	Interrupt output enabled

Remark n = 7 to 4, 0

17.3.13 BEMP interrupt enable register (BEMPENB)

Figure 17 - 17 Format of BEMP interrupt enable register (BEMPENB)

Address: F063AH, F063BH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BEMPENB	0	0	0	0	0	0	0	0	PIPE7BEMPE	PIPE6BEMPE	PIPE5BEMPE	PIPE4BEMPE	0	0	0	PIPE0BEMPE

PIPEnBEMPE	BEMP interrupt enable for PIPEn
0	Interrupt output disabled
1	Interrupt output enabled

Remark n = 7 to 4, 0

17.3.14 SOF output configuration register (SOFCFG)

Figure 17 - 18 Format of SOF output configuration register (SOFCFG)

Address: F063CH, F063DH After reset: 0000H R/W

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

BRDYM	BRDY interrupt status clear timing for each pipe
0	Software clears the status.
1	The USB module clears the status when data has been read from the FIFO buffer or data has been written to the FIFO buffer.
Specifies the timing for clearing the BRDY interrupt status for each pipe.	

EDGESTS	Edge interrupt output status monitor Note
0	The edge interrupt output signal is not in the middle of the edge processing.
1	The edge interrupt output signal is in the middle of the edge processing.
Indicates 1 when the edge interrupt output signal is in the middle of the edge processing.	

Note Make sure the EDGESTS bit is 0 when stopping the clock supply to the USB module.

17.3.15 Interrupt status register 0 (INTSTS0)

Figure 17 - 19 Format of Interrupt status register 0 (INTSTS0)

Address: F0640H, F0641H After reset: 00000000 X0000000B R/W Note 1

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTSTS0	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ2	DVSQ1	DVSQ0	VALID	CTSQ2	CTSQ1	CTSQ0

VBINT	VBUS interrupt status Note 2
0	VBUS interrupts are not generated. Note 3
1	VBUS interrupts are generated.
The USB module sets the VBINT bit to 1 on detecting a level change (high to low or low to high) in the UVBUS pin input value. The USB module sets the VBSTS bit to indicate the UVBUS pin input value. When the UVBUS interrupt is generated, use software to repeat reading the VBSTS bit until the same value is read three or more times, and eliminate chattering.	

RESM	Resume interrupt status Notes 2, 4
0	Resume interrupts are not generated. Note 3
1	Resume interrupts are generated.
The USB module sets the RESM bit to 1 on detecting the falling edge of the signal on the DP pin in the suspended state (the DVSQ2 to DVSQ0 bits = 1xxB).	

SOFR	Frame number refresh interrupt status
0	SOF interrupts are not generated. Note 3
1	SOF interrupts are generated.
The USB module sets the SOFR bit to 1 on updating the frame number. (A frame number refresh interrupt is detected every 1 ms.) The USB module can detect an SOFR interrupt through the internal interpolation function even when a damaged SOF packet is received from the USB host.	

DVST	Device state transition interrupt status Note 4
0	Device state transition interrupts are not generated. Note 3
1	Device state transition interrupts are generated.
The USB module updates the DVSQ2 to DVSQ0 value and sets the DVST bit to 1 on detecting a change in the device state. When a device state transition interrupt is generated, clear the status before the USB module detects the next device state transition.	

CTRT	Control transfer stage transition interrupt status Note 4
0	Control transfer stage transition interrupts are not generated. Note 3
1	Control transfer stage transition interrupts are generated.
The USB module updates the CTSQ2 to CTSQ0 value and sets the CTRT bit to 1 on detecting a change in the control transfer stage. When a control transfer stage transition interrupt is generated, clear the status before the USB module detects the next control transfer stage transition.	

BEMP	Buffer empty interrupt status
0	BEMP interrupts are not generated.
1	BEMP interrupts are generated.

Indicates the BEMP interrupt status.
The USB module sets the BEMP bit to 1 when at least one PIPEBEMP bit in the BEMPSTS register is set to 1 among the PIPEBEMP bits corresponding to the PIPEBEMPE bits in the BEMPENB register to which 1 has been set (when the USB module detects the BEMP interrupt status in at least one pipe among the pipes for which software enables the BEMP interrupt output).
For the conditions for PIPEBEMP status assertion, refer to **17.4.3.3 BEMP Interrupt**.
The USB module clears the BEMP bit to 0 when software writes 0 to all the PIPEBEMP bits corresponding to the PIPEBEMPE bits to which 1 has been set.
The BEMP bit cannot be cleared to 0 even if software writes 0 to this bit.

NRDY	Buffer not ready interrupt status
0	NRDY interrupts are not generated.
1	NRDY interrupts are generated.

Indicates the NRDY interrupt status.
The USB module sets the NRDY bit to 1 when at least one PIPENRDY bit in NRDYSTS is set to 1 among the PIPENRDY bits corresponding to the PIPENRDYE bits in NRDYENB to which 1 has been set (when the USB module detects the NRDY interrupt status in at least one pipe among the pipes for which software enables the NRDY interrupt output).
For the conditions for PIPENRDY status assertion, refer to **17.4.3.2 NRDY Interrupt**.
The USB module clears the NRDY bit to 0 when software writes 0 to all the PIPENRDY bits corresponding to the PIPENRDYE bits to which 1 has been set.
The NRDY bit cannot be cleared to 0 even if software writes 0 to this bit.

BRDY	Buffer ready interrupt status
0	BRDY interrupts are not generated.
1	BRDY interrupts are generated.

Indicates the BRDY interrupt status.
The USB module sets the BRDY bit to 1 when at least one PIPEBRDY bit in the BRDYSTS register is set to 1 among the PIPEnBRDY bits corresponding to the PIPEBRDYE bits in the BRDYENB register to which 1 has been set (when the USB module detects the BRDY interrupt status in at least one pipe among the pipes for which software enables the BRDY interrupt output).
For the conditions for PIPEBRDY status assertion, refer to **17.4.3.1 BRDY Interrupt**.
The USB module clears the BRDY bit to 0 when software writes 0 to all the PIPEnBRDY bits corresponding to the PIPEBRDYE bits to which 1 has been set.
The BRDY bit cannot be cleared to 0 even if software writes 0 to this bit.

VBSTS	VBUS interrupt status Note 4
0	UV_{BUS} pin is low.
1	UV_{BUS} pin is high.

DVSQ2	DVSQ1	DVSQ0	Device state
0	0	0	Powered state
0	0	1	Default state
0	1	0	Address state
0	1	1	Configured state
1	x	x	Suspended state
These bits indicate the device status.			

VALID	USB request reception
0	Not detected Note 3
1	Setup packet reception
Indicates the USB request reception status.	

CTSQ2	CTSQ1	CTSQ0	Control transfer stage
0	0	0	Idle or setup stage
0	0	1	Control read data stage
0	1	0	Control read status stage
0	1	1	Control write data stage
1	0	0	Control write status stage
1	0	1	Control write (no data) status stage
1	1	0	Control transfer sequence error
1	1	1	Do not set.
These bits indicate the control transfer stage status.			

Note 1. Bits 12 to 4 and 2 to 0 are read-only.

Note 2. A change in the status indicated by the VBINT and RESM bits can be detected even while the clock supply is stopped (SCKE = 0), and the interrupts are output when the corresponding interrupt enable bits are enabled. Clearing the status through software should be done after enabling the clock supply.

Note 3. To clear the VBINT, RESM, SOFR, DVST, CTRT, or VALID bit, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.

Note 4. The value after reset depends on the value of the UV_{BUS} pin. This bit is 1 when the UV_{BUS} pin input is high level and 0 when the input is low level.

17.3.16 Interrupt status register 1 (INTSTS1)

Figure 17 - 20 Format of Interrupt status register 1 (INTSTS1)

Address: F0642H, F0643H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTSTS1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PDDETINT

PDDETINT	USB port Portable Device detection interrupt status Note
0	Interrupts are not generated.
1	Interrupts are generated.
The USB module detects when a change (high to low or low to high) occurs in the input value to the USB transceiver VDPDET pin, and sets this bit to 1. The USB module indicates the input value to the USB transceiver VDPDET pin to the PDDETSTS bit. When the PDDETINT interrupt is generated, use software to repeat reading the PDDETSTS bit until the same value is read three or more times, and eliminate chattering.	

Note To clear the PDDETINT bit, write 0 only to the bit to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.

17.3.17 BRDY interrupt status register (BRDYSTS)

Figure 17 - 21 Format of BRDY interrupt status register (BRDYSTS)

Address: F0646H, F0647H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRDYSTS	0	0	0	0	0	0	0	0	PIPE7 BRDY	PIPE6 BRDY	PIPE5 BRDY	PIPE4 BRDY	0	0	0	PIPE0 BRDY

PIPEnBRDY	BRDY interrupt status for PIPEn Note 1
0	Interrupts are not generated. Note 2
1	Interrupts are generated.

Note 1. When BRDYM in SOFCGFG is 0, clearing BRDY interrupts should be done before accessing the FIFO.

Note 2. When BRDYM in SOFCGFG is 0, to clear the status indicated by the bits in BRDYSTS, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.

Remark n = 7 to 4, 0

17.3.18 NRDY interrupt status register (NRDYSTS)

Figure 17 - 22 Format of NRDY interrupt status register (NRDYSTS)

Address: F0648H, F0649H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NRDYSTS	0	0	0	0	0	0	0	0	PIPE7 NRDY	PIPE6 NRDY	PIPE5 NRDY	PIPE4 NRDY	0	0	0	PIPE0 NRDY

PIPEnNRDY	NRDY interrupt status for PIPEn
0	Interrupts are not generated. Note
1	Interrupts are generated.

Note To clear the status indicated by the bits in the NRDYSTS register, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.

Remark n = 7 to 4, 0

17.3.19 BEMP interrupt status register (BEMPSTS)

Figure 17 - 23 Format of BEMP interrupt status register (BEMPSTS)

Address: F064AH, F064BH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BEMPSTS	0	0	0	0	0	0	0	0	PIPE7 BEMP	PIPE6 BEMP	PIPE5 BEMP	PIPE4 BEMP	0	0	0	PIPE0 BEMP

PIPEnBEMP	BEMP interrupt status for PIPEn
0	Interrupts are not generated. Note
1	Interrupts are generated.

Note To clear the status indicated by the bits in the BEMPSTS register, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.

Remark n = 7 to 4, 0

17.3.20 Frame number register (FRMNUM)

Figure 17 - 24 Format of Frame number register (FRMNUM)

Address: F064CH, F064DH After reset: 0000H R

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

FRNM[10:0]	Frame number
The USB module sets the FRNM[10:0] bits to indicate the latest frame number, which is updated every time an SOF packet is issued or received (every 1 ms). Repeat reading the FRNM[10:0] bits until the same value is read twice.	

17.3.21 USB address register (USBADDR)

Figure 17 - 25 Format of USB address register (USBADDR)

Address: F0650H, F0651H After reset: 0000H R

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

USBADDR[6:0]	USB address
The current USB address value can be read.	

17.3.22 USB request type register (USBREQ)

Figure 17 - 26 Format of USB request type register (USBREQ)

Address: F0654H, F0655H After reset: 0000H R

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

BREQUEST[7:0]	Request
These bits store the USB request bRequest value. These bits indicate the USB request data value received during the setup transaction. Writing to these bits is invalid.	

BMREQUESTTYPE[7:0]	Request type
These bits store the USB request bmRequestType value. These bits indicate the USB request data value received during the setup transaction. Writing to these bits is invalid.	

17.3.23 USB request value register (USBVAL)

Figure 17 - 27 Format of USB request value register (USBVAL)

Address: F0656H, F0657H After reset: 0000H R

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

WVALUE[15:0]	Value
These bits store the USB request wValue value. These bits indicate the USB request wValue value received during the setup transaction. Writing to these bits is invalid.	

17.3.24 USB request index register (USBINDX)

Figure 17 - 28 Format of USB request index register (USBINDX)

Address: F0658H, F0659H After reset: 0000H R

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

WINDEX[15:0]	Index
These bits store the USB request wIndex value. These bits indicate the USB request wIndex value received during the setup transaction. Writing to these bits is invalid.	

17.3.25 USB request length register (USBLENG)

Figure 17 - 29 Format of USB request length register (USBLENG)

Address: F065AH, F065BH After reset: 0000H R

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

WLENGTH[15:0]	Length
These bits store the USB request wLength value. These bits indicate the USB request wLength value received during the setup transaction. Writing to these bits is invalid.	

17.3.26 DCP configuration register (DCPCFG)

Figure 17 - 30 Format of DCP configuration register (DCPCFG)

Address: F065CH, F065DH After reset: 0000H R/W

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

SHTNAK	Pipe disabled at end of transfer Note
0	Pipe continued at the end of transfer
1	Pipe disabled at the end of transfer
Specifies whether to modify PID to NAK upon the end of transfer when the selected pipe is in the receiving direction. The SHTNAK bit is valid when the selected pipe in the receiving direction. When the SHTNAK bit is set to 1, the USB module modifies the PID bits for the DCP to NAK on determining the end of the transfer. The USB module determines that the transfer has ended on the following condition. • A short packet (including a zero-length packet) is successfully received.	

Note Modify these bits while PID is NAK. Before modifying these bits after modifying the PID bits for the DCP from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

17.3.27 DCP maximum packet size register (DCPMAXP)

Figure 17 - 31 Format of DCP maximum packet size register (DCPMAXP)

Address: F065EH, F065FH After reset: 0040H R/W

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

MXPS[6:0]	Maximum packet size Note
These bits specify the maximum data payload (maximum packet size) for the DCP. The initial value is 40H (64 bytes). These bits should be set to the value based on the USB Specifications. While the MXPS bits are 0, do not write to the FIFO buffer or do not set PID to BUF.	

Note Modify the MXPS[6:0] bits while PID is NAK. Before modifying these bits after modifying the PID bits for the DCP from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary. After modifying the MXPS[6:0] bits and the DCP has been set to the CURPIPE bits, clear the buffer by setting BCLR to 1.

17.3.28 DCP control register (DCPCTR)

Figure 17 - 32 Format of DCP control register (DCPCTR)

Address: F0660H, F0661H After reset: 0040H R/W Note 1

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCPCTR	BSTS	0	0	0	0	0	0	SQCLR	SQSET	SQMON	PBUSY	0	0	CCPL	PID1	PID0

BSTS	Buffer status
0	Buffer access is enabled.
1	Buffer access is disabled.
Indicates whether DCP FIFO buffer access is enabled or disabled. The meaning of the BSTS bit depends on the ISEL bit setting as follows. • When ISEL is 0, the BSTS bit indicates whether the received data can be read from the buffer. • When ISEL is 1, the BSTS bit indicates whether the data to be transmitted can be written to the buffer.	

SQCLR	Toggle bit clear Note 2
0	Invalid
1	Specifies DATA0. Note 3
Specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer. The SQCLR bit always indicates 0 Do not set the SQCLR and SQSET bits to 1 simultaneously.	

SQSET	Toggle bit set Note 2
0	Invalid
1	Specifies DATA1. Note 3
Specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer. Do not set the SQCLR and SQSET bits to 1 simultaneously.	

SQMON	Sequence toggle bit monitor
0	DATA0
1	DATA1
Indicates the expected value of the sequence toggle bit for the next transaction during the DCP transfer. The USB module allows the SQMON bit to toggle upon normal completion of the transaction. However, the SQMON bit is not allowed to toggle when a DATA-PID disagreement occurs during the transfer in the receiving direction. The USB module sets the SQMON bit to 1 (specifies DATA1 as the expected value) upon successful reception of the setup packet. However, the USB module does not reference the SQMON bit during the IN/OUT transaction of the status stage, and does not allow the SQMON bit to toggle upon normal completion.	

PBUSY	Pipe busy
0	DCP is not used for the transaction.
1	DCP is used for the transaction.
Indicates whether DCP is used or not for the transaction when USB changes the PID bits from BUF to NAK. The USB module modifies the PBUSY bit from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY bit from 1 to 0 upon completion of one transaction. Reading the PBUSY bit after software has set PID to NAK allows checking whether modification of the pipe settings is possible. For details, refer to **17.4.4.1 Pipe Control Register Switching Procedures**.	

CCPL	Control transfer end enable
0	Invalid
1	Completion of control transfer is enabled.
Setting the CCPL bit to 1 enables the status stage of the control transfer to be completed. When software sets the CCPL bit to 1 while the corresponding PID bits are set to BUF, the USB module completes the control transfer stage. Specifically, during control read transfer, the USB module transmits the ACK handshake in response to the OUT transaction from the USB host, and transmits the zero-length packet in response to the IN transaction from the USB host during control write or no-data control transfer. However, on detecting the SET_ADDRESS request, the USB module operates in auto response mode from the setup stage up to the status stage completion irrespective of the setting of the CCPL bit. The USB module modifies the CCPL bit from 1 to 0 on receiving a new setup packet. Software cannot write 1 to the CCPL bit while VALID is 1.	

PID1	PID0	Response PID
0	0	NAK response
0	1	BUF response (depending on the buffer state)
1	0	STALL response
1	1	STALL response
The PID1 and PID0 bits control the response type of the USB module during control transfer. The USB module modifies the setting of the PID1 and PID0 bits as follows. • The USB module modifies the PID1 and PID0 bits to NAK on receiving the setup packet. Here, the USB module sets VALID to 1. Software cannot modify the setting of the PID1 and PID0 bits until software sets VALID to 0. • The USB module sets PID to STALL (11B) on receiving the data of a size exceeding the maximum packet size when software has set the PID1 and PID0 bits to BUF. • The USB module sets PID to STALL (1xB) on detecting the control transfer sequence error. • The USB module sets PID to NAK on detecting the USB bus reset. The USB module does not reference to the setting of the PID1 and PID0 bits while the SET_ADDRESS request is processed (auto processing).		

Note 1. Bits 15, 6, and 5 are read-only.

Note 2. Write 1 to the SQSET and SQCLR bits while PID is NAK. Before modifying these bits after modifying the PID bits for the DCP from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

Note 3. This bit is always read as 0. Only 1 can be written.

17.3.29 Pipe window selection register (PIPESEL)

Figure 17 - 33 Format of Pipe window selection register (PIPESEL)

Address: F0664H, F0665H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIPESEL	0	0	0	0	0	0	0	0	0	0	0	0	PIPESEL3	PIPESEL2	PIPESEL1	PIPESEL0

PIPESEL3	PIPESEL2	PIPESEL1	PIPESEL0	Pipe window select
0	0	0	0	No pipe selected
0	1	0	0	PIPE4
0	1	0	1	PIPE5
0	1	1	0	PIPE6
0	1	1	1	PIPE7
Other than above				Do not set.

The PIPESEL3 to PIPESEL0 bits select the pipe number corresponding to the PIPECFG, PIPEBUF, and PIPEMAXP registers which data are written to or read from.
Selecting a pipe number through these bits allows writing to and reading from the PIPECFG and PIPEMAXP registers which correspond to the selected pipe number.
When the PIPESEL3 to PIPESEL0 bits are set to 0000B, 0 is read from all of the bits in the PIPECFG and PIPEMAXP registers. Writing to these bits is invalid.

Caution After selecting the pipe using the PIPESEL register, functions of the pipe should be set using the PIPECFG and PIPEMAXP registers. The PIPEnCTR, PIPEnTRE, and PIPEnTRN registers can be set regardless of the pipe selection in the PIPESEL register.

17.3.30 Pipe configuration register (PIPECFG)

Figure 17 - 34 Format of Pipe configuration register (PIPECFG)

Address: F0668H, F0669H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIPECFG	TYPE1	TYPE0	0	0	0	BFRE	DBLB	0	SHTNAK	0	0	DIR	EPNUM[3:0]			

TYPE1	TYPE0	Transfer type Note 1
PIPE4, PIPE5		
0	0	Pipe not used
0	1	Bulk transfer
1	0	Do not set.
1	1	Do not set.
PIPE6, PIPE7		
0	0	Pipe not used
0	1	Do not set.
1	0	Interrupt transfer
1	1	Do not set.
These bits select the transfer type for the pipe selected by the PIPESEL3 to PIPESEL0 bits (selected pipe). Before setting PID to BUF for the selected pipe (before starting USB communication using the selected pipe), be sure to set the TYPE1 and TYPE0 bits to a value other than 00B.		

BFRE	BRDY interrupt operation specification Notes 2, 3
0	BRDY interrupt upon transmitting or receiving data
1	BRDY interrupt upon completion of reading data
Specifies the BRDY interrupt generation timing from the USB module to the CPU with respect to the selected pipe. When software has set the BFRE bit to 1 and the selected pipe is in the receiving direction, the USB module detects the transfer completion and generates the BRDY interrupt on having read the relevant packet. When the BRDY interrupt is generated with the above conditions, software needs to write 1 to the BCLR bit. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to the BCLR bit. When software has set the BFRE bit to 1 and the selected pipe is in the transmitting direction, the USB module does not generate the BRDY interrupt. For details, refer to **17.4.3.1 BRDY Interrupt**.	

DBLB	Double buffer mode Notes 2, 3
0	Single buffer
1	Double buffer
Selects either single or double buffer mode for the FIFO buffer used by the selected pipe. The DBLB bit is valid when PIPE4 and PIPE5 are selected.	

SHTNAK	Pipe disabled at end of transfer Note 1
0	Pipe continued at the end of transfer
1	Pipe disabled at the end of transfer
Specifies whether to modify PID to NAK upon the end of transfer when the selected pipe is in the receiving direction. The SHTNAK bit is valid when the selected pipe is PIPE4 and PIPE5 in the receiving direction. When software has set the SHTNAK bit to 1 for the selected pipe in the receiving direction, the USB module modifies the PID bits corresponding to the selected pipe to NAK on determining the end of the transfer. The USB module determines that the transfer has ended on any of the following conditions. • A short packet (including a zero-length packet) is successfully received. • The transaction counter is used and the number of packets specified by the counter are successfully received.	

DIR	Transfer direction Notes 2, 3
0	Receiving direction
1	Transmitting direction
Specifies the transfer direction for the selected pipe. When software has set the DIR bit to 0, the USB module uses the selected pipe in the receiving direction, and when software has set the DIR bit to 1, the USB module uses the selected pipe in the transmitting direction.	

EPNUM[3:0]	Endpoint number Note 1
These bits specify the endpoint number for the selected pipe. Setting 0000B means unused pipe. Do not make the settings such that the combination of the settings of the DIR and EPNUM bits should be the same for two or more pipes (EPNUM = 0000B can be set for all of the pipes).	

Note 1. Modify the TYPE1 and TYPE0, SHTNAK, and EPNUM bits while PID is NAK. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

Note 2. Modify the BFRE, DBLB, and DIR bits while PID is NAK and before the pipe is selected by the CURPIPE3 to CURPIPE0 bits. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

Note 3. To modify the BFRE, DBLB, and DIR bits after completing USB communication using the selected pipe, write 1 and then 0 to ACLRM continuously through software to clear the FIFO buffer assigned to the selected pipe while the PID and CURPIPE bits are in the state described in the above Note 2.

17.3.31 Pipe maximum packet size register (PIPEMAXP)

Figure 17 - 35 Format of Pipe maximum packet size register (PIPEMAXP)

Address: F066CH, F066DH After reset: 0000H/0040H R/W

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

MXPS[8:0]	Maximum packet size Note
These bits specify the maximum data payload (maximum packet size). The range of values can be set for each pipe is shown as follows: PIPE4, PIPE5: 8 bytes (008H), 16 bytes (010H), 32 bytes (020H), 64 bytes (040H) (Bits [8:7] and [2:0] are not provided.) PIPE6, PIPE7: 1 byte (001H) to 64 bytes (040H) (Bits [8:7] are not provided.) The MXPS bits should be set to the appropriate value for each transfer type based on the USB Specifications. While the MXPS bits are 0, do not write to the FIFO buffer or set PID to BUF.	

Note Modify the MXPS bits while PID is NAK and before the pipe is selected by the CURPIPE bits. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

Caution The value of the PIPEMAXP register after reset differs depending on when no pipe is selected and when a pipe is selected with the PIPESEL3 to PIPESEL0 bits in the PIPESEL register.
The value after reset is 0000H when no pipe is selected and 0040H when a pipe is selected.

17.3.32 PIPEn Control Registers (PIPEnCTR) (n = 4 to 7)

Figure 17 - 36 Format of PIPEn Control Registers (PIPEnCTR) (n = 4, 5)

Address: F0676H, F0677H (PIPE4CTR), F0678H, F0679H (PIPE5CTR) After reset: 0000H R/W Note 1

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIPEnCTR	BSTS	INBUFM	0	0	0	ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	0	0	0	PID1	PID0

BSTS	Buffer status
0	Buffer access by the CPU is disabled.
1	Buffer access by the CPU is enabled.
Indicates the FIFO buffer status for the relevant pipe. The meaning of the BSTS bit depends on the settings of the DIR, BFRE, and DCLRM bits as shown in Table 17 - 8.	

INBUFM	Transmit buffer monitor
0	There is no data to be transmitted in the buffer memory.
1	There is data to be transmitted in the buffer memory.
Indicates the relevant FIFO buffer status when the relevant pipe is in the transmitting direction. When the relevant pipe is in the transmitting direction (DIR = 1), the USB module sets the INBUFM bit to 1 when software (or DTC) completes writing data to at least one FIFO buffer plane. The USB module sets the INBUFM bit to 0 when the USB module completes transmitting the data from the FIFO buffer plane to which all the data has been written. In double buffer mode (DBLB = 1), the USB module sets the INBUFM bit to 0 when the USB module completes transmitting the data from the two FIFO buffer planes before software (or DTC) completes writing data to one FIFO buffer plane. The INBUFM bit indicates the same value as the BSTS bit when the relevant pipe is in the receiving direction (DIR = 0).	

ATREPM	Auto response mode Note 2
0	Auto response is disabled.
1	Auto response is enabled.
Enables or disables auto response mode for the relevant pipe. When the relevant pipe is for bulk transfer, the ATREPM bit can be set to 1. When the ATREPM bit is set to 1, the USB module responds to the token from the USB host as described below. [When the relevant pipe is for bulk IN transfer (TYPE bits = 01B and DIR = 1)] When ATREPM is 1 and PID is BUF, the USB module transmits a zero-length packet in response to the IN token. The USB module updates (allows toggling of) the sequence toggle bit (DATA-PID) each time the USB module receives ACK from the USB host (in a single transaction, IN token is received, zero-length packet is transmitted, and then ACK is received.). In this case, the USB module does not generate the BRDY or BEMP interrupt. [When the relevant pipe is for bulk OUT transfer (TYPE bits = 01B and DIR = 0)] When ATREPM is 1 and PID is BUF, the USB module returns NAK in response to the OUT token and generates the NRDY interrupt. For USB communication in auto response mode, set the ATREPM bit to 1 while the FIFO buffer is empty.	

ACLRM	Auto buffer clear mode Note 3
0	Disabled
1	Enabled (all buffers are initialized)
Enables or disables auto buffer clear mode for the relevant pipe. To delete the information in the FIFO buffer assigned to the relevant pipe completely, write 1 and then 0 to the ACLRM bit continuously. Table 17 - 9 shows the information cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which clearing the information is necessary.	

SQCLR	Toggle bit clear Note 2
0	Invalid
1	Specifies DATA0. Note 4
The SQCLR bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the relevant pipe. Setting the SQCLR bit to 1 through software allows the USB module to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB module always sets the SQCLR bit to 0.	

SQSET	Toggle bit set Note 2
0	Invalid
1	Specifies DATA1. Note 4
The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe. Setting the SQSET bit to 1 through software allows the USB module to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB module always sets the SQSET bit to 0.	

SQMON	Toggle bit monitor
0	DATA0
1	DATA1
Indicates the expected value of the sequence toggle bit for the next transaction of the relevant pipe. The USB module allows the SQMON bit to toggle upon normal completion of the transaction of the relevant pipe. However, the SQMON bit is not allowed to toggle when a DATA-PID disagreement occurs during the transfer in he receiving direction.	

PBUSY	Pipe busy
0	The relevant pipe is not used for the transaction.
1	The relevant pipe is used for the transaction.
Indicates whether the relevant pipe is being currently used or not for the transaction. The USB module modifies the PBUSY bit from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY bit from 1 to 0 upon completion of one transaction. Reading the PBUSY bit after software has set PID to NAK allows checking whether modification of the pipe settings is possible. For details, refer to **17.4.4.1 Pipe Control Register Switching Procedures**.	

PID1	PID0	Response PID
0	0	NAK response
0	1	BUF response (depending on the buffer state)
1	0	STALL response
1	1	STALL response

The PID1 and PID0 bits specify the response type for the next transaction of the relevant pipe.
The default setting of the PID1 and PID0 bits is NAK. Modify the setting of these bits to BUF to use the relevant pipe for USB transfer. Table 17 - 10 shows the basic operation (operation when there are no errors in the transmitted and received packets) of the USB module depending on the PID bit setting.
After modifying the setting of the PID1 and PID0 bits through software from BUF to NAK during USB communication using the relevant pipe, check that PBUSY is 1 to see if USB transfer using the relevant pipe has actually entered the NAK state.
The USB module modifies the setting of the PID1 and PID0 bits in the following cases.

- The USB module sets PID to NAK on recognizing the completion of the transfer when the relevant pipe is in the receiving direction and software has set the SHTNAK bit for the selected pipe to 1.
- The USB module sets PID to STALL (11B) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB module sets PID to NAK on detecting a USB bus reset.

To specify each response type, set the PID1 and PID0 bits as follows.

- To make a transition from NAK (00B) to STALL, write 10B.
- To make a transition from BUF (01B) to STALL, write 11B.
- To make a transition from STALL (11B) to NAK, write 10B and then 00B.
- To make a transition from STALL to BUF, write 00B (NAK) and then 01B (BUF).

Note 1. Bits 15, 14, 6, and 5 are read-only.

Note 2. Modify the ATREPM bit or write 1 to the SQCLR or SQSET bit while PID is NAK. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

Note 3. Modify the ATREPM bit while PID is NAK and before the pipe is selected by the CURPIPE bits. Before modifying this bit after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

Note 4. Only 0 can be read and 1 can be written.

Table 17 - 8 Operation of BSTS Bit

DIR Bit	BFRE Bit	DCLRM Bit	BSTS Bit Function
0	0	0	1 when the received data can be read from the FIFO buffer; 0 when the received data has been completely read from the FIFO buffer.
		1	Setting prohibited
	1	0	1 when the received data can be read from the FIFO buffer; 0 when software has set BCLR to 1 after the received data has been completely read from the FIFO buffer.
		1	1 when the received data can be read from the FIFO buffer; 0 when the received data has been completely read from the FIFO buffer.
1	0	0	1 when the transmit data can be written to the FIFO buffer; 0 when the transmit data has been completely written to the FIFO buffer.
		1	Setting prohibited
	1	0	Setting prohibited
		1	Setting prohibited

Table 17 - 9 Information Cleared by USB Module by Setting ACLRM = 1

No.	Information Cleared by ACLRM Bit Manipulation	Cases in which Clearing Information is Necessary
1	All the information in the FIFO buffer assigned to the relevant pipe (both FIFO buffer planes are cleared when double buffer mode is selected)	
2	Internal flags concerning the BFRE bit	When the BFRE setting is modified
3	FIFO buffer toggle control	When the DBLB setting is modified
4	Internal flags concerning the transaction count	When the transaction count function is forcibly terminated

Table 17 - 10 Operation of USB Module depending on PID Bit Setting

PID Bits (PID1 and PID0)	Transfer Type	Transfer Direction (DIR Bit)	Operation of USB Module
00 (NAK)	Bulk or interrupt	Operation does not depend on the setting.	Returns NAK in response to the token from the USB host. For the operation when ATREPM is 1, refer to the description of the ATREPM bit.
01 (BUF)	Bulk	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception.
	Interrupt	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception.
	Bulk or interrupt	Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Returns NAK if not ready.
10 (STALL) or 11 (STALL)	Bulk or interrupt	Operation does not depend on the setting.	Returns STALL in response to the token from the USB host.

Figure 17 - 37 Format of PIPEn Control Registers (PIPEnCTR) (n = 6, 7)

Address: F067AH, F067BH (PIPE6CTR), F067CH, F069DH (PIPE7CTR) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIPEnCTR	BSTS	0	0	0	0	0	ACLRM	SQCLR	SQSET	SQMON	PBUSY	0	0	0	PID1	PID0

BSTS	Buffer status
0	Buffer access by the CPU is disabled.
1	Buffer access by the CPU is enabled.
Indicates the FIFO buffer status for the relevant pipe. The meaning of the BSTS bit depends on the settings of the DIR, BFRE, and DCLR bits as shown in Table 17 - 8.	

ACLRM	Auto buffer clear mode Note 1
0	Auto buffer clear mode is disabled.
1	Auto buffer clear mode is enabled (all buffers are initialized)
Enables or disables auto buffer clear mode for the relevant pipe. To delete the information in the FIFO buffer assigned to the relevant pipe completely, write 1 and then 0 to the ACLRM bit continuously. Table 17 - 11 shows the information cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which clearing the information is necessary.	

SQCLR	Toggle bit clear Note 2
0	Invalid
1	Specifies DATA0. Note 3
The SQCLR bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the relevant pipe. Setting the SQCLR bit to 1 through software allows the USB module to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB module always sets the SQCLR bit to 0.	

SQSET	Toggle bit set Note 2
0	Invalid
1	Specifies DATA1. Note 3
The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe. Setting the SQSET bit through software allows the USB module to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB module always sets the SQSET bit to 0.	

SQMON	Toggle bit monitor
0	DATA0
1	DATA1
Indicates the value of the sequence toggle bit for the next transaction of the relevant pipe. When the relevant pipe is not for the isochronous transfer, the USB module allows the SQMON bit to toggle upon normal completion of the transaction of the relevant pipe. However, the SQMON bit is not allowed to toggle when a DATA-PID disagreement occurs during the transfer in the receiving direction.	

PBUSY	Pipe busy
0	The relevant pipe is not used for the transaction.
1	The relevant pipe is used for the transaction.

Indicates whether the relevant pipe is being currently used or not for the transaction.
The USB module modifies the PBUSY bit from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY bit from 1 to 0 upon completion of one transaction.
Reading the PBUSY bit after software has set PID to NAK allows checking whether modification of the pipe settings is possible. For details, refer to **17.4.4.1 Pipe Control Register Switching Procedures**.

PID1	PID0	Response PID
0	0	NAK response
0	1	BUF response (depending on the buffer state)
1	0	STALL response
1	1	STALL response

The PID1 and PID0 bits specify the response type for the next transaction of the relevant pipe.
The default setting of the PID1 and PID0 bits is NAK. Modify the setting of these bits to BUF to use the relevant pipe for USB transfer. Table 17 - 10 shows the basic operation (operation when there are no errors in the transmitted and received packets) of the USB module depending on the PID bit setting.
After modifying the setting of the PID1 and PID0 bits through software from BUF to NAK during USB communication using the relevant pipe, check that PBUSY is 1 to see if USB transfer using the relevant pipe has actually entered the NAK state.
The USB module modifies the setting of the PID1 and PID0 bits in the following cases.

- The USB module sets PID to NAK on recognizing the completion of the transfer when the relevant pipe is in the receiving direction and software has set the SHTNAK bit for the selected pipe to 1.
- The USB module sets PID to STALL (11B) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB module sets PID to NAK on detecting a USB bus reset.

To specify each response type, set the PID1 and PID0 bits as follows.

- To make a transition from NAK (00B) to STALL, write 10B.
- To make a transition from BUF (01B) to STALL, write 11B.
- To make a transition from STALL (11B) to NAK, write 10 and then 00B.
- To make a transition from STALL to BUF, write 00B (NAK) and then 01B (BUF).

Note 1. Modify the ACLRM bit while PID is NAK and before the pipe is selected by the CURPIPE bits. Before modifying this bit after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

Note 2. Write 1 to the SQCLR or SQSET bit while PID is NAK. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

Note 3. Only 0 can be read and 1 can be written.

Table 17 - 11 Information Cleared by USB Module by Setting ACLRM = 1

No.	Information Cleared by ACLRM Bit Manipulation	Cases in which Clearing Information is Necessary
1	All the information in the FIFO buffer assigned to the selected pipe	
2	Internal flags concerning the BFRE bit	When the BFRE setting is modified
3	Internal flags concerning the transaction count	When the transaction count function is forcibly terminated

17.3.33 PIPEn Transaction Counter Enable Registers (PIPEnTRE) (n = 4, 5)

Figure 17 - 38 Format of PIPEn Transaction Counter Enable Registers (PIPEnTRE) (n = 4, 5)

Address: F069CH, F069DH (PIPE4TRE), F06A0H, F06A1H (PIPE5TRE) After reset: 0000H R/W

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

TRENB	Transaction counter enable
0	Transaction counter is disabled.
1	Transaction counter is enabled.

Enables or disables the transaction counter.
For the pipe in the receiving direction, setting the TRENB bit to 1 after setting the total number of the packets to be received in the TRNCNT bits through software allows the USB module to control hardware as described below on having received the number of packets equal to the setting of the TRNCNT bits.
- While SHTNAK is 1, the USB module modifies the PID bits to NAK for the corresponding pipe on having received the number of packets equal to the setting of the TRNCNT bits.
- While BFRE is 1, the USB module asserts the BRDY interrupt on having received the number of packets equal to the setting of the TRNCNT bits and then reading out the last received data.

For the pipe in the transmitting direction, set the TRENB bit to 0.
When the transaction counter is not used, set the TRENB bit to 0.
When the transaction counter is used, set the TRNCNT bits before setting the TRENB bit to 1. Set the TRENB bit to 1 before receiving the first packet to be counted by the transaction counter.

TRCLR	Transaction counter clear
0	Invalid
1	The current counter value is cleared.

Clears the current value of the transaction counter corresponding to the relevant pipe and then sets the TRCLR bit to 0.

Caution Modify each bit in the PIPEnTRE register while CSST is 1 and PID is NAK.
Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

17.3.34 PIPEn Transaction Counter Registers (PIPEnTRN) (n = 4, 5)

Figure 17 - 39 Format of PIPEn Transaction Counter Registers (PIPEnTRN) (4, 5)

Address: F069EH, F069FH (PIPE4TRN), F06A2H, F06A3H (PIPE5TRN) After reset: 0000H R/W

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

TRNCNT[15:0]	Transaction counter
[When written to] Specifies the number of transactions during a DTC transfer. [When read from] Indicates the specified number of transactions if TRENB is 0. Indicates the number of currently counted transactions if TRENB is 1. The USB module increments the value of the TRNCNT bits by one when all of the following conditions are satisfied on receiving the packet. • TRENB = 1 • (TRNCNT set value ≠ current counter value + 1) on receiving the packet. • The payload of the received packet agrees with the setting of the MXPS bits. The USB module clears the value of the TRNCNT bits to 0 when any of the following conditions are satisfied. (1) All of the following conditions are satisfied. • TRENB = 1 • (TRNCNT set value = current counter value + 1) on receiving the packet. • The payload of the received packet agrees with the setting of the MXPS bits. (2) All of the following conditions are satisfied. • TRENB = 1 • The USB module has received a short packet. (3) All of the following conditions are satisfied. • TRENB = 1 • Software has set the TRCLR bit to 1. For the pipe in the transmitting direction, set the TRNCNT bits to 0. When the transaction counter is not used, set the TRNCNT bits to 0. Setting the number of transactions to be transferred to the TRNCNT bits is only enabled when the TRENB bit in the PIPEnTRE register is 0. To modify the number of transactions to be transferred, set the TRCLR bit in the PIPEnTRE register to 1 (to clear the current counter value) before setting TRENB to 1.	

17.3.35 BC control register 0 (USBBCCTRL0)

Figure 17 - 40 Format of BC control register 0 (USBBCCTRL0)

Address: F06B0H, F06B1H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBBCCTRL0	0	0	0	0	0	0	PDDETSTS0	CHGDETSTS0	BATCHGE0	0	VDMSRCE0	IDPSINKE0	VDPSRCE0	IDMSINKE0	IDPSRCE0	RPDME0

PDDETSTS0	VDP_SRC (0.6 V) input detection flag for UDP pin
0	Not detected
1	Detected
Detects VDP_SRC (0.6 V) is applied to the UDP pin from the connected device. (Detects that the voltage applied to UDP is in the range of VDAT_REF to VIH (UDP).) To use this bit for detection, set the CNEN bit (bit 8) in the SYSCFG register to 1, and enable the single end receiver of USB port. Valid when IDPSINKE0 is 1.	

CHGDETSTS0	VDM_SRC (0.6 V) input detection flag for UDM pin
0	Not detected
1	Detected
Detects VDM_SRC (0.6 V) is applied to the UDM pin from the connected device. (Detects that the voltage applied to UDM is in the range of VDAT_REF to VIH (UDM).) To use this bit for detection, set the CNEN bit (bit 8) in the SYSCFG register to 1, and enable the single end receiver of USB port. Valid when IDMSINKE0 is 1.	

BATCHGE0	USB port BC connection detection operation enable
0	Operation disabled
1	Operation enabled
When this bit is set to enabled, each bit setting of VDPSRCE0, VDMSRCE0, IDPSINKE0, IDMSINKE0, and IDPSRCE0 is valid BC connection detection can be operated via USB port.	

VDMSRCE0	UDM pin VDM_SRC (0.6 V) output control
0	VDM_SRC output disabled
1	VDM_SRC output enabled (0.6 V output)
Controls the VDM_SRC output.	

IDPSINKE0	UDP pin VDP_SRC (0.6 V) input detection (comparator and sink) control
0	UDP pin (0.6 V) input detection disabled
1	UDP pin (0.6 V) input detection enabled
Controls the IDP_SINK (sink current) used for the 0.6 V input detection circuit (comparator) and detection for the UDP pin.	

VDPSRCE0	UDP pin V_{DP_SRC} (0.6 V) output control
0	V_{DP_SRC} output disabled
1	V_{DP_SRC} output enabled (0.6 V output)
Controls the V_{DP_SRC} output.	

IDMSINKE0	UDM pin V_{DM_SRC} (0.6 V) input detection (comparator and sink) control
0	UDM pin (0.6 V) input detection disabled
1	UDM pin (0.6 V) input detection enabled
Controls the I_{DM_SINK} (sink current) used for the 0.6 V input detection circuit (comparator) and detection for the UDM pin.	

IDPSRCE0	UDP pin IDP_SRC (10 μA) output control
0	I_{DP_SRC} output disabled
1	I_{DP_SRC} output enabled (10 μA output)
Controls the I_{DP_SRC} output.	

RPDME0	UDM pull-down control
0	Pulling down disabled
1	Pulling down enabled
Only the UDM pin can be pulled down (R_{PD}) by setting this bit.	

17.3.36 BC option control register 0 (USBBCOPT0)

Figure 17 - 41 Format of BC option control register 0 (USBBCOPT0)

Address: F06B8H, F06B9H After reset: 0000H R/W Note 1

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBBCOPT0	0	0	0	0	0	0	DMCUSDET0	DPCUSDET0	0	CUSDETE0	0	0	VDSEL03	VDSEL02	VDSEL01	VDSEL00

DMCUSDET0 Note 2	UDM voltage detection (Option BC)
0	UDM pin voltage is lower than the comparison voltage selected by the VDSEL0x bit.
1	UDM pin voltage exceeds the comparison voltage selected by the VDSEL0x bit.

DPCUSDET0 Note 2	UDP voltage detection (Option BC)
0	UDP pin voltage is lower than the comparison voltage selected by the VDSEL0x bit.
1	UDP pin voltage exceeds the comparison voltage selected by the VDSEL0x bit.

CUSDETE0	Option voltage detection circuit control (Option BC)
0	Disabled
1	Enabled

VDSEL03 to VDSEL00	UDP/UDM pin option comparison voltage selection (Option BC)
Select UDP/UDM pin comparison voltage value (Refer to **Remark**).	

Note 1. Bits 9 and 8 are read-only.

Note 2. Valid when CUSDETE0 = 1.

Remark The comparison voltage values of the option BC detection circuit of the UDP and UDM pins when UV_{BUS} = 5.0 V are shown below. The following voltages vary in proportion to the UV_{BUS} input voltage.

VDSEL03	VDSEL02	VDSEL01	VDSEL00	Comparison voltage (V) (Valid when CUSDETE0 = 1)	
				UDP	UDM
0	0	0	0	1.60	
0	0	0	1	1.70	
0	0	1	0	1.85	
0	0	1	1	2.00	
0	1	0	0	2.15	
0	1	0	1	2.30	
0	1	1	0	2.45	
0	1	1	1	2.60	
1	0	0	0	2.80	
1	0	0	1	3.00	
1	0	1	0	3.20	
1	0	1	1	3.40	
1	1	0	0	3.60	
1	1	0	1	3.80	
1	1	1	0	4.00	
1	1	1	1	4.20	

17.3.37 USB clock selection register (UCKSEL)

Figure 17 - 42 Format of USB clock selection register (UCKSEL)

Address: F06C4H, F06C5H After reset: 0000H R/W

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

UCKSELC	USB clock selection
0	High-speed on-chip oscillator clock (f_{HOCO}) is not selected as USB clock
1	High-speed on-chip oscillator clock (f_{HOCO}) is selected as USB clock

Caution 1. When selecting the high-speed on-chip oscillator clock (f_{HOCO}) as the USB clock, set UCKSELC = 1 and the CKSELR bit in the MCKC register to 0 at the same time.

Caution 2. The USB clock select register can be rewritten only when the USB is disconnected.

Caution 3. The high-speed on-chip oscillator clock can be selected only when T_A = -20 to +85°C.

Caution 4. If the high-speed on-chip oscillator clock is selected as the USB clock, when the USB is suspended, perform USB suspended processing while the high-speed on-chip oscillator clock is selected (UCKSELC = 1).

Caution 5. If the high-speed on-chip oscillator clock is selected as the USB clock, when the USB is disconnected, perform USB stopped processing (including setting DPRPU = 0) before setting UCKSELC = 0.

17.3.38 USB module control register (USBMC)

Figure 17 - 43 Format of USB module control register (USBMC)

Address: F06CCH, F06CDH After reset: 0002H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBMC	0	0	0	0	0	0	0	0	VBRPDCUT	0	0	0	0	0	PXXCON	VDDUSBE

VBRPDCUT Note 1	UVBUS pin pull-down resistor control
0	Pull-down resistor is valid.
1	Pull-down resistor is invalid.

PXXCON Note 2	USB power supply control bit 1
0	VDDUSBE bit is invalid.
1	VDDUSBE bit is valid.

VDDUSBE Note 3	USB power supply control bit 0
0	The USB power supply is stopped.
1	The USB power supply 3.3 V is turned on.

Note 1. While the positive voltage is applied to the UVBUS pin, the input leakage current can be reduced by setting VBRPDCUT = 1 (pull-down resistor is invalid).

Note 2. When the USB function is used, set PXXCON = 1.

Note 3. Valid when PXXCON is 1 (VDDUSBE bit is valid).
When VDDUSBE = 1, UDP/UDM pin operates using the level of USB power supply 3.3 V as high level and the UREGC pin outputs 3.3 V.

17.4 Operation

17.4.1 System Control

This section describes the register settings that are necessary for initialization of this module and power consumption control.

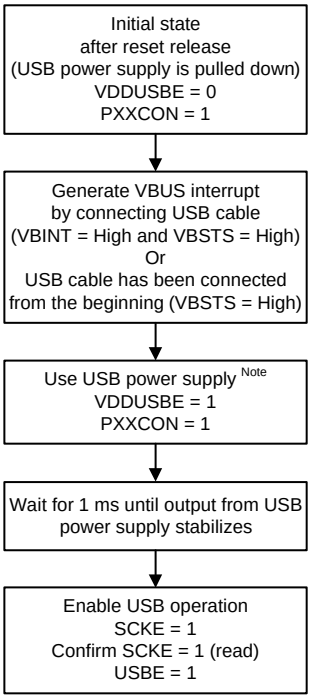
17.4.1.1 Starting Operation

The USB module starts operating when the USBE bit in the SYSCFG register is set to 1 while clock supply to the USB module is started (SCKE bit in SYSCFG register = 1).

To use the on-chip USB power supply, it is necessary to connect an external 0.33 μF stabilization capacitance (for VSS) to the UREGC pin. While the power supply for the USB is used, it is not possible to simultaneously perform A/D conversion that uses the temperature sensor or internal reference voltage.

Figure 17 - 44 shows the Flow for Turning on USB Power.

Figure 17 - 44 Flow for Turning on USB Power



Note When the USB power supply is used, A/D conversion cannot be performed using the temperature sensor or internal reference voltage.

17.4.1.2 Controlling USB Data Bus Resistors

The USB module contains pull-up resistor of the D+ and D- lines. Pull up these lines by setting the DPRPU bit in the SYSCFG register.

Confirm that connection to the USB host is made, then set the DPRPU bit in the SYSCFG register to 1 and pull up the D+ (full speed)/D- (low speed) line.

When the DPRPU bit in the SYSCFG register is set to 1 during communication with the PC, the USB module disables the pull-up resistor of the USB data line, thus notifying the USB host of connection.

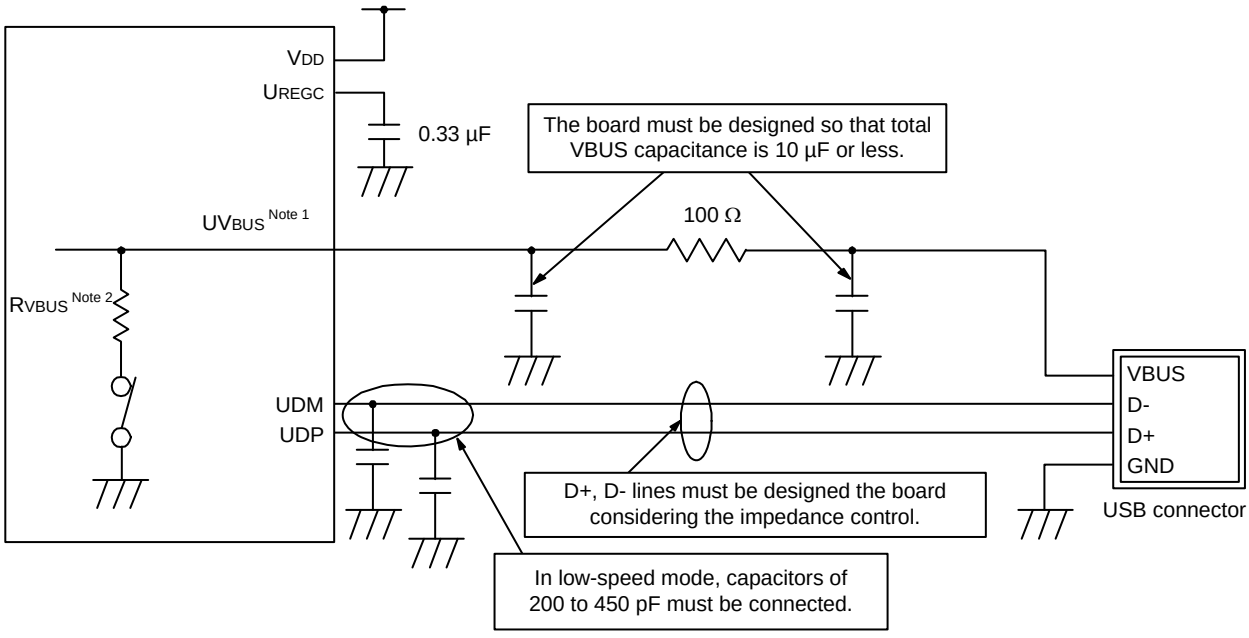
Table 17 - 12 lists the settings for Controlling USB Data Bus Resistors of USB port.

Table 17 - 12 Controlling USB Data Bus Resistors of USB port

Settings		USB Data Bus Resistor Control		
DPRPU	DMRPU	D- Line	D+ Line	Remarks
0	0	Open	Open	When USB port is not used
1	0	Open	Pull-up	Set to this state when operating as the function controller (full speed).
0	1	Pull-up	Open	Set to this state when operating as the function controller (low speed).
1	1	—	—	Setting prohibited

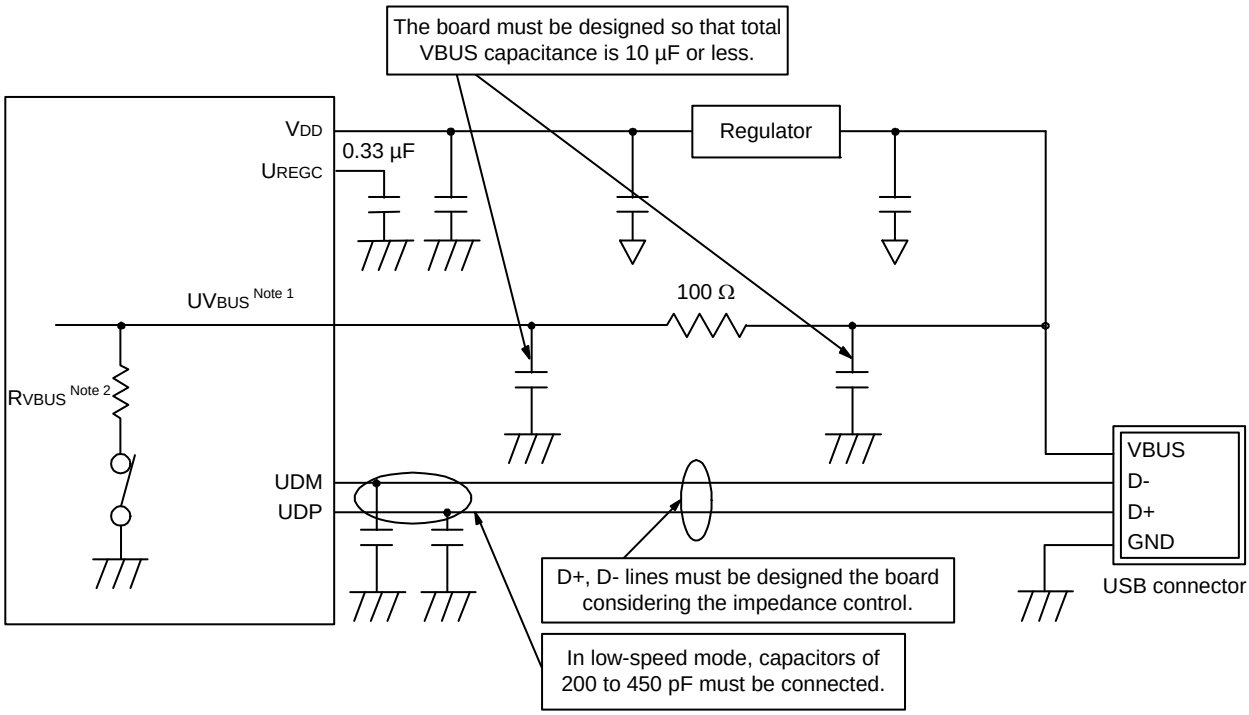
Figure 17 - 45 shows the USB Connector Connection Example in Self-powered (3.3 V). Figure 17 - 46 shows the USB Connector Connection Example in Bus-powered (3.3 V).

Figure 17 - 45 USB Connector Connection Example in Self-powered (3.3 V)



(**Notes** are listed on the next page.)

Figure 17 - 46 USB Connector Connection Example in Bus-powered (3.3 V)



Note 1. 5 V tolerant

Note 2. Set the VBRPDCUT bit (bit 7 in the USBMC register) to 0 and connect the on-chip pull-down resistor of the UVBUS pin.

17.4.2 Interrupt Sources

Table 17 - 13 lists the Interrupt Sources in the USB module.

When an interrupt generation condition is satisfied and the interrupt output is enabled using the corresponding interrupt enable register, the USB issues a USB interrupt request to the interrupt controller and an USB interrupt will be generated.

Table 17 - 13 Interrupt Sources

Status Bit	Name	Interrupt Source	Status Flag
VBINT	VBUS interrupt	When a change in the state of the VBUS input pin has been detected (low to high or high to low)	VBSTS
RESM	Resume interrupt	When a change in the state of the USB bus has been detected in the suspended state (J-state to K-state or J-state to SE0)	—
SOFR	Frame number update interrupt	• When an SOF packet with a different frame number has been received	—
DVST	Device state transition interrupt	• When a device state transition has been detected A USB bus reset detected Suspend state detected SET_ADDRESS request received SET_CONFIGURATION request received	DVSQ2 to DVSQ0
CTRT	Control transfer stage transition interrupt	• When a stage transition has been detected in control transfer Setup stage completed Control write transfer status stage transition Control read transfer status stage transition Control transfer completed A control transfer sequence error occurred	CTSQ2 to CTSQ0
BEMP	Buffer empty interrupt	• When transmission of all data in the buffer memory has been completed and the buffer has become empty • When a packet larger than the maximum packet size has been received	PIPEnBEMP (n = 0, 4 to 7) in BEMPSTS register
NRDY	Buffer not ready interrupt	• When NAK has been returned for an IN or OUT token	PIPEnNRDY (n = 0, 4 to 7) in NRDYSTS register
BRDY	Buffer ready interrupt	• When the buffer has become ready for read access or write access.	PIPEnBRDY (n = 0, 4 to 7) in BRDYSTS register
PDDETINT	Portable Device detection interrupt	• When connection of the Portable Device has been detected	PDDETSTS0
DREQE0/ DTCEN31	DTC transfer request of D0FIFO	• When D0FIFO transfer has been completed	—
DREQE1/ DTCEN30	DTC transfer request of D1FIFO	• When D1FIFO transfer has been completed	—

Figure 17 - 47 shows the USB Interrupts Relationship. Table 17 - 14 lists the USB Interrupts.

Figure 17 - 47 USB Interrupts Relationship

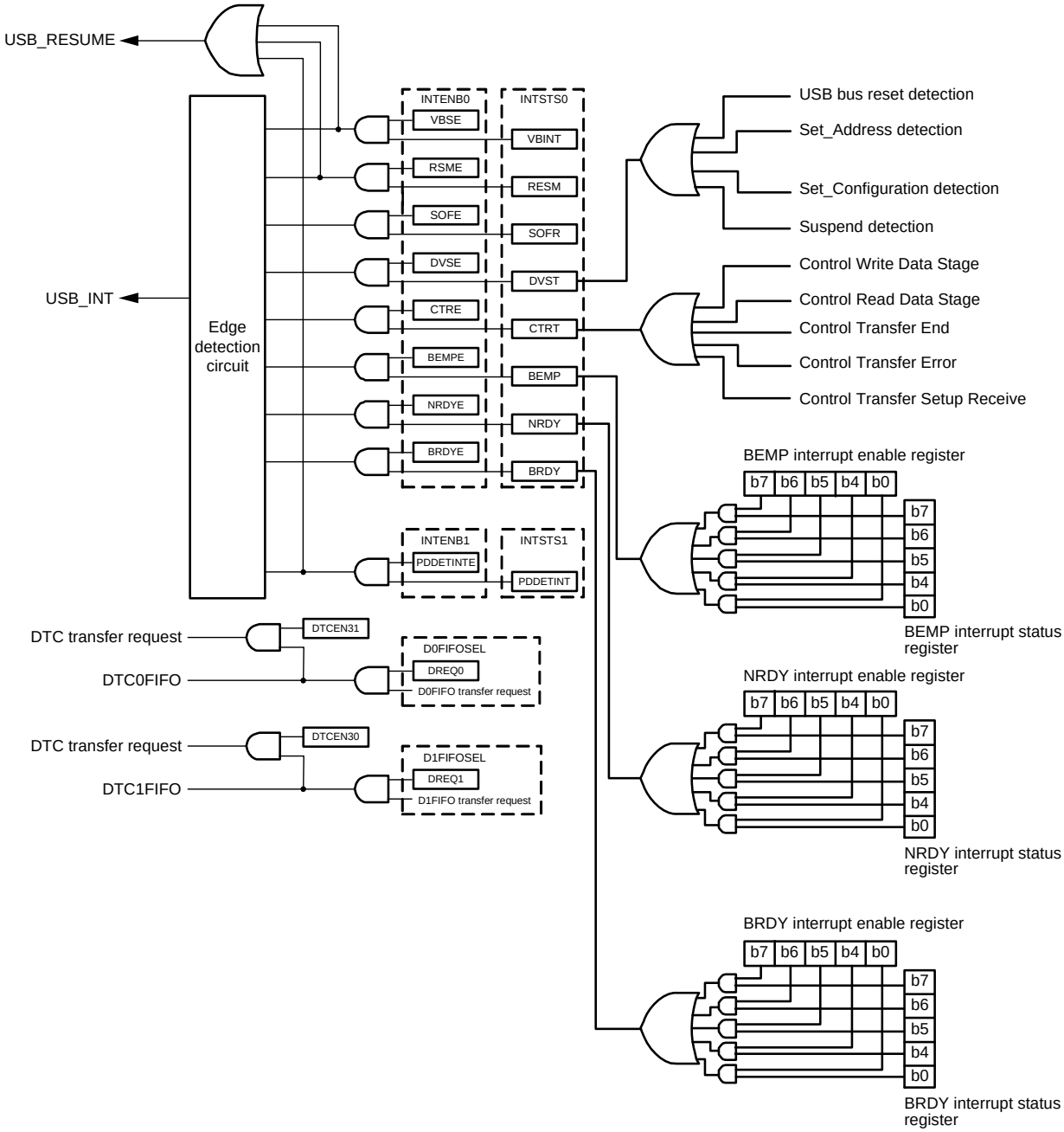


Table 17 - 14 USB Interrupts

Interrupt Name	Interrupt Flag	Priority
USB_INT	VBUS interrupt Resume interrupt Frame number update interrupt Device state transition interrupt Control transfer stage transition interrupt Buffer empty interrupt Buffer not ready interrupt Buffer ready interrupt Portable device detection interrupt	High
USB_RESUME	VBUS interrupt Resume interrupt Portable device detection interrupt	
DTC0FIFO	Transfer end interrupts for D0FIFO	
DTC1FIFO	Transfer end interrupts for D1FIFO	Low

17.4.3 Interrupts

17.4.3.1 BRDY Interrupt

The following shows the conditions under which the USB module sets 1 to a corresponding bit in the BRDYSTS register. Under this condition, the USB module generates a BRDY interrupt if software has set 1 to the PIPEBRDYE bit in the BRDYENB register that corresponds to the pipe and 1 to the BRDYE bit in the INTENB0 register.
The conditions for generating and clearing the BRDY interrupt depend on the settings of the BRDYM bit and BFRE bit for each pipe as described below.

(1) When BRDYM = 0 and BFRE = 0
With these settings, the BRDY interrupt indicates that the FIFO port is accessible.
On any of the following conditions, the USB module generates an internal BRDY interrupt request trigger and sets 1 to the PIPEBRDY bit corresponding to the pertinent pipe.

[For the pipe set to the transmitting direction]
- When software changes the DIR bit from 0 to 1.
- When packet transmission is completed using the pertinent pipe while write-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS bit is read as 0).
- When one FIFO buffer is empty on completion of writing data to the other FIFO buffer in double buffer mode.
- No request trigger is generated until completion of writing data to the currently-written FIFO buffer even if transmission to the other FIFO buffer is completed.
- When 1 is written to the ACLRM bit, which causes the FIFO buffer to make transition from the write-disabled to write-enabled state.

No request trigger is generated for the DCP (that is, during data transmission for control transfers).

[For the pipe set to the receiving direction]
- When packet reception is completed successfully thus enabling the FIFO buffer to be read while read-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS bit is read as 0).
  No request trigger is generated for the transaction in which DATA-PID disagreement has occurred.
- When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode.
  No request trigger is generated until completion of reading data from the currently-read FIFO buffer even if reception by the other FIFO buffer is completed.

The BRDY interrupt is not generated in the status stage of control transfers.
The PIPEBRDY interrupt status of the pertinent pipe can be cleared to 0 by writing 0 to the corresponding PIPEBRDY bit in BRDYSTS through software. In this case, 1s should be written to the PIPEBRDY bits for the other pipes.
Be sure to clear the BRDY status before accessing the FIFO buffer.

(2) When BRDYM = 0 and BFRE = 1

With these settings, the USB module generates a BRDY interrupt on completion of reading all data for a single transfer using the pipe in the receiving direction, and sets 1 to the bit in BRDYSTS corresponding to the pertinent pipe.
On any of the following conditions, the USB module determines that the last data for a single transfer has been received.

- When a short packet including a zero-length packet is received.
- When the transaction counter register (TRNCNT bits) is used and the number of packets specified by the TRNCNT bits are completely received.

When the pertinent data is completely read out after any of the above conditions has been satisfied, the USB module determines that all data for a single transfer has been completely read out.
When a zero-length packet is received while the FIFO buffer is empty, the USB module determines that all data for a single transfer has been completely read out upon passing the zero-length packet data to the CPU. In this case, to start the next transfer, write 1 to the BCLR bit in the corresponding FIFOCTR register through software.
With these settings, the USB module does not detect a BRDY interrupt for the pipe in the transmitting direction.
The PIPEBRDY interrupt status of the pertinent pipe can be cleared to 0 by writing 0 to the corresponding PIPEBRDY bit through software. In this case, 1s should be written to the PIPEBRDY bits for the other pipes.
In this mode, the BFRE bit setting should not be modified until all data for a single transfer has been processed. When it is necessary to modify the BFRE bit before completion of processing, all FIFO buffers for the pertinent pipe should be cleared using the ACLRM bit.

(3) When BRDYM = 1 and BFRE = 0

With these settings, the PIPEBRDY values are linked to the BSTS bit setting for each pipe. In other words, the BRDY interrupt status bits (PIPEBRDY) are set to 1 or 0 by the USB module depending on the FIFO buffer status.

[For the pipe set to the transmitting direction]
The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for write access, and are set to 0 when it is not ready.
However, the BRDY interrupt is not generated even if the DCP in the transmitting direction is ready for write access.

[For the pipe set to the receiving direction]

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for read access, and are set to 0 when all data have been read (not ready for read access).

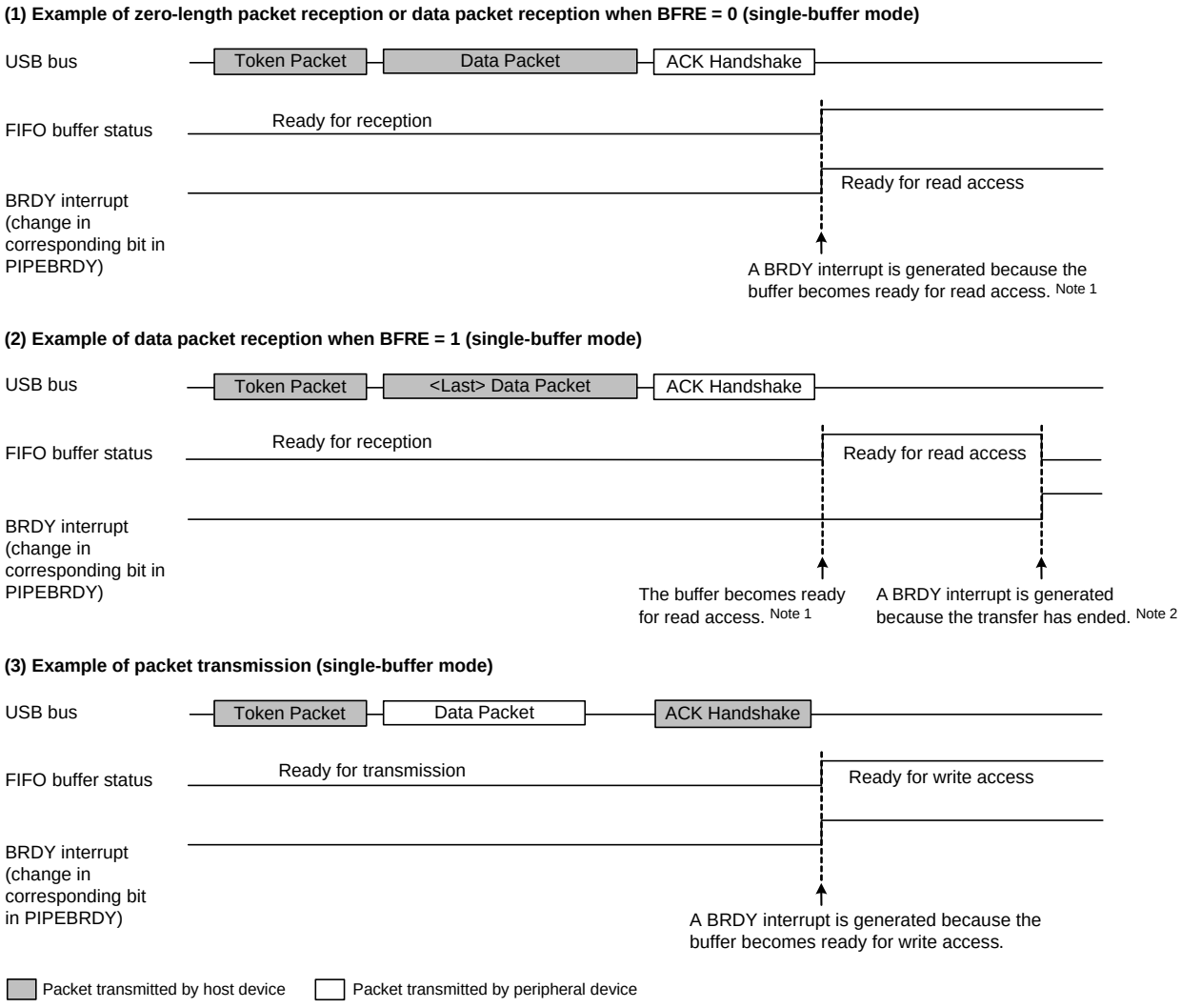
When a zero-length packet is received while the FIFO buffer is empty, the pertinent bit is set to 1 and the BRDY interrupt is continuously generated until BCLR = 1 is written through software.

With this setting, the PIPEBRDY bit cannot be cleared to 0 through software.

When the BRDYM bit is set to 1, all BFRE bits (for all pipes) should be cleared to 0.

Figure 17 - 48 shows the Timing of BRDY Interrupt Generation.

Figure 17 - 48 Timing of BRDY Interrupt Generation



Note 1. The FIFO buffer becomes ready for read access under the following condition:
When a packet is received while no data remains unread in the buffer in the CPU.

Note 2. A transfer ends under either of the following conditions:
(1) When a short packet including a zero-length packet is received
(2) When the number of packets specified in the transaction counter are received

The condition that USB module clears the BRDY bit in INTSTS0 depends on the setting of the BRDYM bit in the SOFCFG register. Table 17 - 15 shows the Condition for Clearing BRDY Bit.

Table 17 - 15 Condition for Clearing BRDY Bit

BRDYM	Condition for Clearing BRDY Bit
0	The USB module clears the BRDY bit in the INTSTS0 register when software has cleared all bits in the BRDYSTS register.
1	The USB module clears the BRDY bit in the INTSTS0 register when the BSTS bits for all piles have become 0.

17.4.3.2 NRDY Interrupt

On generating an internal NRDY interrupt request for the pipe whose PID bits are set to BUF by software, the USB module sets the corresponding PIPENRDY bit in the NRDYSTS register to 1. If the corresponding bit in the NRDYENB register has been set to 1 by software, the USB module sets the NRDY bit in the INTSTS0 register to 1 and generates a USB interrupt.
The following describes the conditions on which the USB module generates the internal NRDY interrupt request for a given pipe.
The internal NRDY interrupt request is not generated during status stage execution of the control transfer.

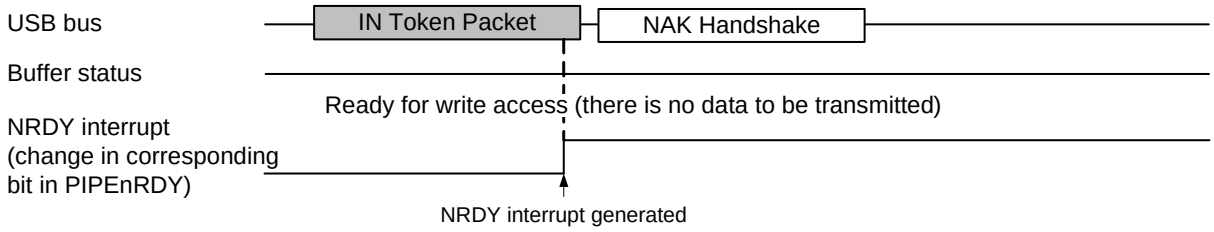
On any of the following conditions, the USB module detects an NRDY interrupt.
[For the pipe set to the transmitting direction]
- When an IN token is received while there is no data to be transmitted in the FIFO buffer.
  In this case, the USB module generates a NRDY interrupt request at the reception of the IN token and sets the PIPENRDY bit to 1.

[For the pipe set to the receiving direction]
- When an OUT token is received while there is no space available in the FIFO buffer.
  The USB module generates a NRDY interrupt request when a NAK handshake is transferred after the data following the OUT token is received, and sets the PIPENRDY bit to 1.
  However, during re-transmission (due to DATA-PID disagreement), the NRDY interrupt request is not generated. In addition, if an error occurs in the DATA packet, the NRDY interrupt request is not generated.

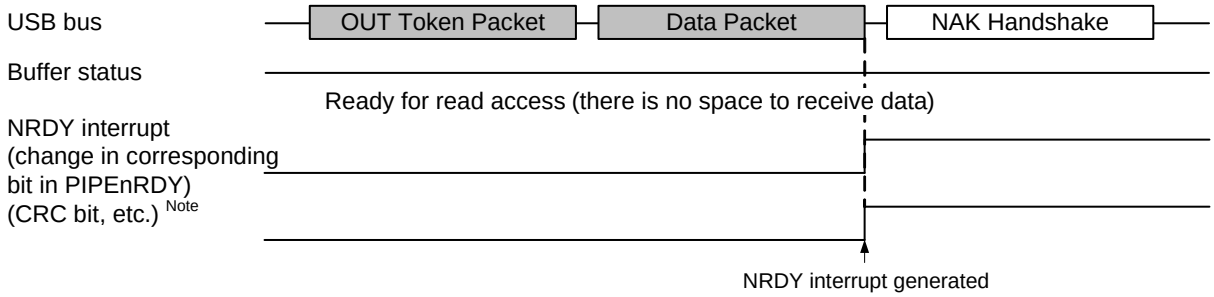
Figure 17 - 49 shows the Timing of NRDY Interrupt Generation.

Figure 17 - 49 Timing of NRDY Interrupt Generation

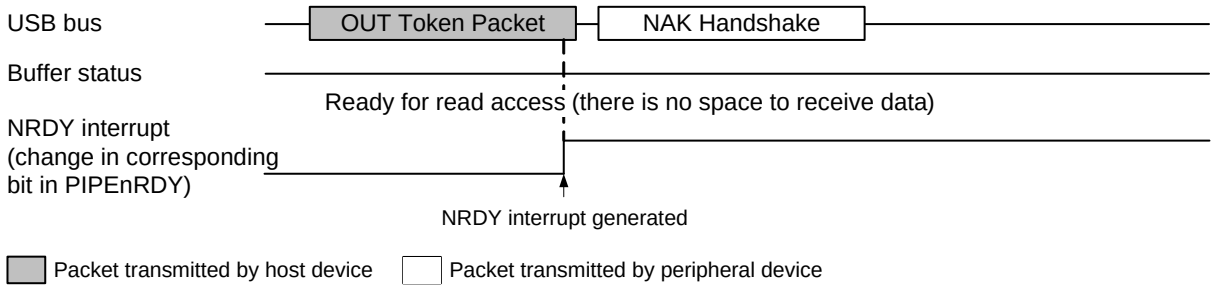
(1) Example of data transmission (single-buffer mode)



(2) Example of data reception: OUT token reception (single-buffer mode)



(3) Example of data reception: PING token reception (single-buffer mode)



Note The CRC and OVRN bits change only while the target pipe is set to isochronous transfers.

17.4.3.3 BEMP Interrupt

On detecting a BEMP interrupt for the pipe whose PID bits are set to BUF by software, the USB module sets the corresponding PIPEBEMP bit in the BEMPSTS register to 1. If the corresponding bit in the BEMPENB register has been set to 1 by software, the USB module sets the BEMP bit in the INTSTS0 register to 1 and generates a USB interrupt.

The following describes the conditions on which the USB module generates an internal BEMP interrupt request.

[For the pipe set to the transmitting direction]

- When the FIFO buffer of the corresponding pipe is empty on completion of transmission (including zero-length packet transmission).
  In single buffer mode, an internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for the pipe other than DCP.
  However, the internal BEMP interrupt request is not generated on any of the following conditions.
- When software (DTC) has already started writing data to the FIFO buffer of the CPU on completion of transmitting data from one FIFO buffer in double buffer mode.
- When the buffer is cleared (emptied) by setting the ACLRM or BCLR bit to 1.
- When IN transfer (zero-length packet transmission) is performed during the control transfer status stage.

[For the pipe set to the receiving direction]

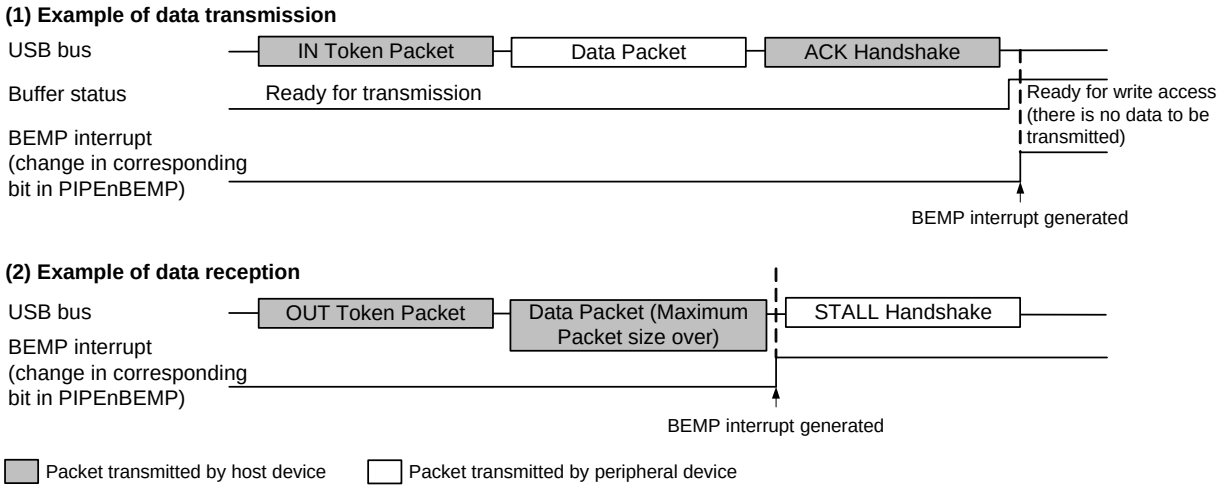
When data whose size is greater than the setting value of MaxPacketSize is successfully received.
In this case, the USB module generates a BEMP interrupt request, sets the corresponding PIPEnBEMP bit in the BEMPSTS register to 1, discards the received data, and modifies the setting of the PID bits of the corresponding pipe to STALL (11B).
Here, the USB module returns STALL response.
However, the internal BEMP interrupt request is not generated on any of the following conditions.

- When a CRC error or a bit stuffing error is detected in the received data.
- When a setup transaction is being performed.

Writing 0 to the PIPEnBEMP bit in the BEMPSTS register clears the status.
Writing 1 to the PIPEnBEMP bit in the BEMPSTS register has no effect.

Figure 17 - 50 shows the Timing of BEMP Interrupt Generation.

Figure 17 - 50 Timing of BEMP Interrupt Generation

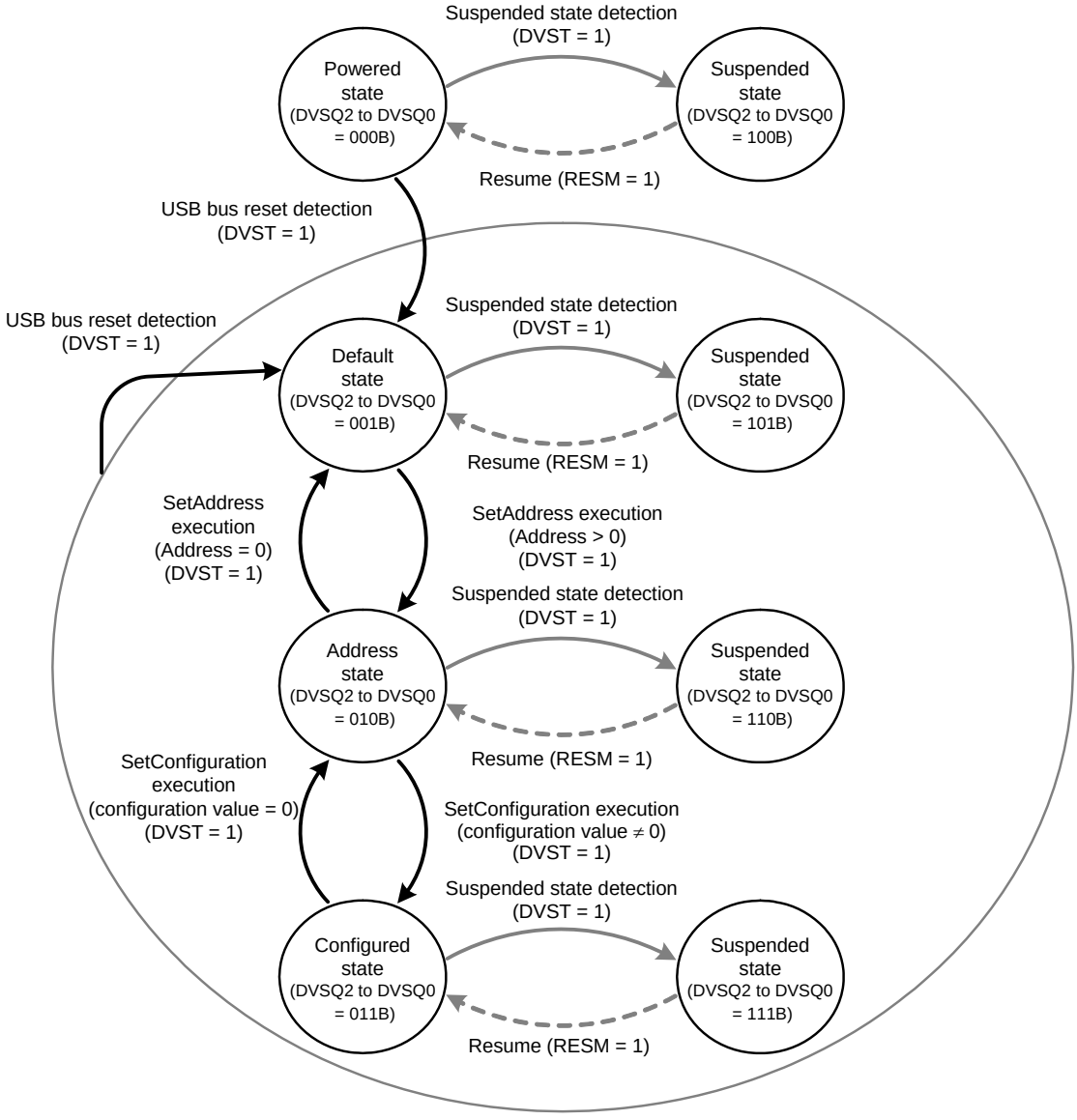


17.4.3.4 Device State Transition Interrupt

Figure 17 - 51 is a diagram of Device State Transitions in the USB module. The USB module controls device state and generates device state transition interrupts. However, recovery from the suspended state (resume signal detection) is detected by means of the resume interrupt. The device state transition interrupts can be enabled or disabled individually using the INTENB0 register. The device state to which a transition was made can be confirmed using the DVSQ2 to DVSQ0 bits in the INTSTS0 register.

When a transition is made to the default state, a device state transition interrupt is generated after a USB bus reset is detected.

Figure 17 - 51 Device State Transitions



Caution For the transition indicated in solid line, the DVST bit is set to 1.
For the transition indicated in dashed line, the RESM bit is set to 1.

17.4.3.5 Control Transfer Stage Transition Interrupt

Figure 17 - 52 is a diagram of Control Transfer Stage Transitions in the USB module. The USB module controls the control transfer sequence and generates control transfer stage transition interrupts. The control transfer stage transition interrupts can be enabled or disabled individually using the INTENB0 register. The transfer stage to which a transition was made can be confirmed using the CTSQ2 to CTSQ0 bits in the INTSTS0 register.
The control transfer sequence errors are listed below. If an error occurs, the PID bits in the DCPCTR register are set to 1xB (STALL response).

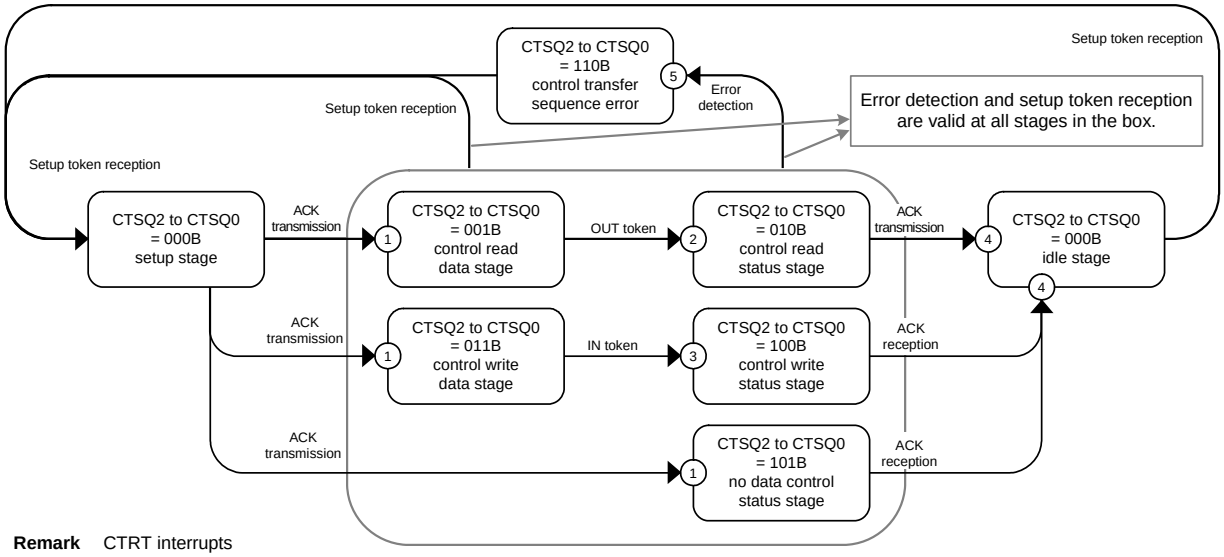
(1) During control read transfer
• An OUT token is received while no data has been transferred for the IN token at the data stage.
• An IN token is received at the status stage.
• A data packet with DATAPID = DATA0 is received at the status stage.

(2) During control write transfer
• An IN token is received while no ACK response has been returned for the OUT token at the data stage.
• A data packet with DATAPID = DATA0 is received for the first data packet at the data stage.
• An OUT token is received at the status stage.

(3) During no-data control transfers
• An OUT token is received at the status stage.

At the control write transfer data stage, if the number of receive data exceeds the wLength value of the USB request, it cannot be recognized as a control transfer sequence error. At the control read transfer status stage, packets other than zero-length packets are received by an ACK response and the transfer ends normally.
When a CTRT interrupt occurs in response to a sequence error (SERR = 1), the CTSQ2 to CTSQ0 bits = 110B value is retained until CTRT = 0 is written from the system (the interrupt status is cleared). Therefore, while the CTSQ2 to CTSQ0 bits = 110B is being held, the CTRT interrupt that ends the setup stage will not be generated even if a new USB request is received. (The USB module retains the setup stage end, and after the interrupt status has been cleared by software, a setup stage end interrupt is generated.)

Figure 17 - 52 Control Transfer Stage Transitions



Remark CTRT interrupts
(1) Setup stage completed
(2) Control read transfer status stage transition
(3) Control write transfer status stage transition
(4) Control transfer completed
(5) Control transfer sequence error

17.4.3.6 Frame Update Interrupt

The USB module updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation.

17.4.3.7 VBUS Interrupt

When the VBUS pin level changes, a VBUS interrupt is generated. The level of the VBUS pin can be checked with the VBSTS bit in the INTSTS0 register. Whether the host controller is connected or disconnected can be confirmed using the VBUS interrupt. However, if the system is activated with the host controller connected, the first VBUS interrupt is not generated because there is no change in the VBUS pin level.

17.4.3.8 Resume Interrupt

A resume interrupt is generated when the device state is the suspended state and the USB bus state has changed (from J-state to K-state, or from J-state to SE0). Recovery from the suspended state is detected by means of the resume interrupt.

17.4.3.9 Portable Device Detection Interrupt

A portable device detection interrupt is generated when the USB module detects a level change (high from low or low from high) in the PDDET output from the USB-PHY. When a portable device detection interrupt is generated, use software to repeat reading the PDDETSTS bit until the same value is read three or more times, and eliminate chattering.

17.4.3.10 DTC Transfer End Interrupts for D0FIFO/D1FIFO

For PIPE4 to PIEP7, the FIFO port can be accessed using the DTC. When accessing the buffer for the pipe set for the DTC is enabled, a DTC transfer request is output. When DTC transfer is completed, a D0FIFO/D1FIFO interrupt request is generated.

17.4.4 Pipe Control

Table 17 - 16 lists the Pipe Settings in the USB module. With USB data transfer, data transfer has to be carried out using the logic pipe called the endpoint. The USB module has five pipes that are used for data transfer. Appropriate settings should be made for each of the pipes according to the specifications of the system.

Table 17 - 16 Pipe Settings

Register Name	Bit Name	Setting	Remarks
DCPCFG PIPECFG	TYPE	Specifies the transfer type	PIPE4, PIPE5: Can be set
	BFRE	Selects the BRDY interrupt mode	PIPE4, PIPE5: Can be set
	DBLB	Selects double buffer mode	PIPE4, PIPE5: Can be set
	DIR	Selects transfer direction	IN or OUT can be set
	EPNUM	Endpoint number	PIPE4, PIPE5: Can be set A value other than 0000B should be set when the pipe is used.
	SHTNAK	Selects disabled state for pipe when transfer ends	PIPE4, PIPE5: Can be set
DCPMAXP PIPEMAXP	MXPS	Maximum packet size	Compliant with the USB standard.
DCPCTR PIPEnCTR	BSTS	Buffer status	For the DCP, receive buffer status and transmit buffer status are switched with the ISEL bit.
	INBUFM	IN buffer monitor	Available only for PIPE4, PIPE5.
	ATREPM	Auto response mode	PIPE4, PIPE5: Can be set
	ACLRM	Auto buffer clear	PIPE4 to PIPE7: Can be set
	SQCLR	Sequence clear	Clears the data toggle bit.
	SQSET	Sequence set	Sets the data toggle bit.
	SQMON	Sequence monitor	Monitors the data toggle bit.
	PBUSY	Pipe busy status	Monitors if the pipe is busy.
	PID	Response PID	Refer to **17.4.4.6 Response PID**.
PIPEnTRE	TRENB	Transaction counter enable	PIPE4, PIPE5: Can be set
	TRCLR	Current transaction counter clear	PIPE4, PIPE5: Can be set
PIPEnTRN	TRNCNT	Transaction counter	PIPE4, PIPE5: Can be set

17.4.4.1 Pipe Control Register Switching Procedures

The following bits in the pipe control registers can be modified only when USB communication is disabled (PID = NAK).

Registers that Should Not be Set in the USB Communication Enabled (PID = BUF) State:
- Bits in the DCPCFG and DCPMAXP registers
- The SQCLR and SQSET bits in the DCPCTR registers
- Bits in the PIPECFG and PIPEMAXP registers
- The ATREPM, ACLRM, SQCLR, and SQSET bits in the PIPEnCTR register
- Bits in the PIPEnTRE and PIPEnTRN registers

In order to modify the above bits in the USB communication enabled (PID = BUF) state, follow the procedure shown below:

(1) A request to modify bits in the pipe control register occurs.
(2) Modify the PID corresponding to the pipe to NAK.
(3) Wait until the corresponding PBUSY bit is cleared to 0.
(4) Modify the bits in the pipe control register.

The following bits in the pipe control registers can be modified only when the pertinent pipe information has not been set by the CURPIPE bit in the CFIFOSEL, D0FIFOSEL, and D1FIFOSE registers.

Registers that Should Not be Set When CURPIPE in FIFO-PORT is set:
- Bits in the DCPCFG and DCPMAXP registers
- Bits in the PIPECFG and PIPEMAXP registers

In order to modify pipe information, the CURPIPE bits in the port select registers should be set to a pipe other than the pipe to be modified. For the DCP, the buffer should be cleared using BCLR after the pipe information is modified.

17.4.4.2 Transfer Types

The TYPE bits in the PIPEPCFG register are used to specify the transfer type for each pipe. The transfer types that can be set for the pipes are as follows.
- DCP: No setting is necessary (fixed at control transfer).
- PIPE4, PIPE5: These should be set to bulk transfer.
- PIPE6, PIPE7: These should be set to interrupt transfer.

17.4.4.3 Endpoint Number

The EPNUM bits in the PIPEPCFG register are used to set the endpoint number for each pipe. The DCP is fixed at endpoint 0. The other pipes can be set from endpoint 1 to endpoint 15.

- DCP: No setting is necessary (fixed at end point 0).
- PIPE4 to PIPE7: The endpoint numbers from 1 to 15 should be selected and set.
  These should be set so that the combination of the DIR bit and EPNUM bits is unique.

17.4.4.4 Maximum Packet Size Setting

The MXPS bits in the DCPMAXP and PIPEMAXP registers are used to specify the maximum packet size for each pipe. DCP and PIPE4, PIPE5 can be set to any of the maximum pipe sizes defined by the USB specification. For PIPE6, PIPE7, 64 bytes are the upper limit of the maximum packet size. The maximum packet size should be set before beginning the transfer (PID = BUF).

- DCP: Set 8, 16, 32, or 64.
- PIPE4, PIPE5: Set 8, 16, 32, or 64 when using bulk transfer.
- PIPE6, PIPE7: Set a value between 1 and 64.

17.4.4.5 Transaction Counter (For PIPE4, PIPE5 in Reading Direction)

When the specified number of transactions has been completed in the data packet receiving direction, the USB module recognizes that the transfer has ended. Two transaction counters are provided: one is the TRNCNT register that specifies the number of transactions to be executed and the other is the current counter that internally counts the number of executed transactions. With the SHTNAK bit set to 1, when the current counter value matches the specified number of transactions, the PID of the corresponding PIPE is set to NAK and the subsequent transfer is disabled. The transactions can be counted again from the beginning by initializing the current counter of the transaction counter function through the TRCLR bit. The information read from TRNCNT differs depending on the setting of the TRENB bit.

- TRENB = 0: The specified transaction counter value can be read.
- TRENB = 1: The current counter value indicating the internally counted number of executed transactions can be read.

When operating the TRCLR bit, the following should be noted.

- If the transactions are being counted and PID = BUF, the current counter cannot be cleared.
- If there is any data left in the buffer, the current counter cannot be cleared.

17.4.4.6 Response PID

The PID bits in DCPCTR and PIPEnCTR are used to set the response PID for each pipe.
The following shows the USB module operation with various response PID settings:

(1) Response PID settings

The response PID is used to specify the response to transactions from the host.

- NAK setting: The NAK response is always returned in response to the generated transaction.
- BUF setting: Responses are made to transactions according to the status of the buffer memory.
- STALL setting: The STALL response is always returned in response to the generated transaction.

Caution For setup transactions, an ACK response is always returned regardless of the PID setting, and the USB request is stored in the register.

The USB module may write to the PID bits, depending on the results of the transaction as described below.

(2) When the response PID is set by hardware

NAK setting: In the following cases, PID = NAK is set and NAK is always returned in response to transactions:

- When the SETUP token is received normally (DCP only).
- If the transaction counting ends or a short packet is received when the SHTNAK bit in PIPECFG has been set to 1 for bulk transfer.

BUF setting: There is no BUF writing by the USB module.

STALL setting: In the following cases, PID = STALL is set and STALL is always returned in response to transactions:

- When a maximum packet size exceeded error is detected in the received data packet.
- When a control transfer sequence error has been detected (DCP only).

17.4.4.7 Data PID Sequence Bit

The USB module automatically toggles the sequence bit in the data PID when data is transferred successfully in the control transfer data stage, bulk transfer, and interrupt transfer. The sequence bit of the next data PID to be transmitted can be confirmed with the SQMON bit in the DCPCTR and PIPEnCTR registers. When data is transmitted, the sequence bit switches at the timing of ACK handshake reception. When data is received, the sequence bit switches at the timing of ACK handshake transmission. The SQCLR bit in the DCPCTR register and the SQSET bit in the PIPEnCTR register can be used to change the data PID sequence bit.
The USB module automatically sets the sequence bit when a stage transition is made. DATA0 is returned when the setup stage is ended and DATA1 is returned in a status stage. Therefore, software settings are not required.
For the ClearFeature request transmission or reception, the data PID sequence bit should be set by software.

17.4.4.8 Response PID = NAK Function

The USB module has a function that disables pipe operation (response PID = NAK) at the timing at which the final data packet of a transaction is received (the USB module automatically distinguishes this based on reception of a short packet or the transaction counter) by setting the SHTNAK bit in the PIPECFG register to 1.
When the double buffer mode is being used for the buffer memory, using this function enables reception of data packets in transfer units. If pipe operation has been disabled, software should set the pipe to the enabled state again (response PID = BUF).
The response PID = NAK function can be used only when bulk transfers are used.

17.4.4.9 Auto Response Mode

With the pipes for bulk transfer (PIPE4, PIPE5), when the ATREPM bit in the PIPEnCTR register is set to 1, a transition is made to auto response mode. During an OUT transfer (DIR = 0), OUT-NAK mode is entered, and during an IN transfer (DIR = 1), null auto response mode is entered.

17.4.4.10 OUT-NAK Mode

With the pipes for bulk OUT transfer, NAK is returned in response to an OUT token and an NRDY interrupt is output when the ATREPM bit is set to 1. To make a transition from normal mode to OUT-NAK mode, OUT-NAK mode should be specified in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, OUT-NAK mode becomes valid. However, if an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is retuned to the host.
To make a transition from OUT-NAK mode to normal mode, OUT-NAK mode should be canceled in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). In normal mode, reception of OUT data is enabled.

17.4.4.11 Null Auto Response Mode

With the pipes for bulk IN transfer, zero-length packets are continuously transmitted when the ATREPM bit is set to 1.
To make a transition from normal mode to null auto response mode, null auto response mode should be set in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, null auto response mode becomes valid. Before setting null auto response mode, the INBUFM bit = 0 should be confirmed because the mode can be set only when the buffer is empty. If the INBUFM bit is 1, the buffer should be emptied with the ACLRM bit. While a transition to null auto response mode is being made, data should not be written from the FIFO port.
To make a transition from null auto response mode to normal mode, pipe operation disabled state (response PID = NAK) should be retained for the period of zero-length packet transmission (about 10 µs) before canceling null auto response mode. In normal mode, data can be written from the FIFO port; therefore, packet transmission to the host is enabled by enabling pipe operation (response PID = BUF).

17.4.5 FIFO Buffer Memory

17.4.5.1 FIFO Buffer Memory

The USB module has FIFO buffer memory for data transfer. The memory area used for each pipe is managed by the USB module. The FIFO buffer memory has two states depending on whether the access right is assigned to the system (CPU side) or the USB module (SIE side).

(1) Buffer Status

Tables 17 - 17 and 17 - 18 show the buffer status in the USB module. The buffer memory status can be confirmed using the BSTS bit in the DCPCTR register and the INBUFM bit in the PIPEnCTR register. The access direction for the buffer memory can be specified using either the DIR bit in the PIPEnCFG register or the ISEL bit in the CFIFOSEL register (when DCP is selected).

The INBUFM bit is valid for PIPE4, PIPE5 in the transmitting direction.

When a transmitting pipe uses the double buffer configuration, software can read the BSTS bit to monitor the buffer memory status on the CPU side and the INBUFM bit to monitor the buffer memory status on the SIE side. When the BEMP interrupt may not show the buffer empty status because the write access to the FIFO port by the CPU (DTC) is slow, software can use the INBUFM bit to confirm the end of transmission.

Table 17 - 17 Buffer Status Indicated by BSTS Bit

ISEL or DIR	BSTS	Buffer Memory Status
0 (receiving direction)	0	There is no received data, or data is being received. Reading from the FIFO port is disabled.
0 (receiving direction)	1	There is received data, or a zero-length packet has been received. Reading from the FIFO port is allowed. Note that when a zero-length packet is received, reading is not possible and the buffer must be cleared.
1 (transmitting direction)	0	The transmission has not been completed. Writing to the FIFO port is disabled.
1 (transmitting direction)	1	The transmission has been completed. CPU write is allowed.

Table 17 - 18 Buffer Status Indicated by INBUFM Bit

DIR	INBUFM	Buffer Memory Status
0 (receiving direction)	Invalid	Invalid
1 (transmitting direction)	0	The transmission has been completed. There is no waiting data to be transmitted.
1 (transmitting direction)	1	The FIFO port has written data to the buffer. There is data to be transmitted.

(2) FIFO Buffer Clearing

Table 17 - 19 shows the clearing of the FIFO buffer memory by the USB module. The buffer memory can be cleared using the BCLR, DCLRM, and ACLRM bits.

Table 17 - 19 Buffer Clearing Methods

FIFO Buffer Clearing Mode	Clearing Buffer Memory on CPU Side	Mode for Automatically Clearing Buffer Memory after Reading Specified Pipe Data	Auto Buffer Clear Mode for Discarding All Received Packets
Register used	CFIFOCTR DnFIFOCTR	DnFIFOSEL	PIPEnCTR
Bit used	BCLR	DCLRM	ACLRM
Clearing condition	Cleared by writing 1	1: Mode valid 0: Mode invalid	1: Mode valid 0: Mode invalid

(3) Auto Buffer Clear Mode Function

With the USB module, all received data packets are discarded if the ACLRM bit in the PIPEnCTR register is set to 1. If a correct data packet has been received, the ACK response is returned to the host controller. The auto buffer clear mode function can be set only in the buffer memory reading direction.

If the ACLRM bit is set to 1 and then to 0, the buffer memory of the selected pipe can be cleared regardless of the access direction.

However, an access cycle of at least 100 ns is required for the internal hardware sequence processing time between the ACLRM bit = 1 and the ACLRM bit = 0.

(4) Buffer Memory Specifications (Single or Double Setting)

Either a single or double buffer configuration can be selected for PIPE4 and PIPE5, using the DBLB bit in the PIPEnCFG register.

17.4.5.2 FIFO Port Functions

Table 17 - 20 shows the FIFO Port Function Settings for the USB module. In write access, writing data until the maximum packet size is reached automatically enables transmission of the data. To enable transmission before the maximum packet size is reached, the BVAL bit in the CFIFOCTR or DnFIFOCTR register should be set to end writing. To send a zero-length packet, the BCLR bit in the register should be used to clear the buffer and then the BVAL bit set in order to end writing.

In read access, reception of new packets is automatically enabled when all data has been read. Data cannot be read when a zero-length packet has been received (DTLN = 0), so the BCLR bit in the register should be used to clear the buffer. The length of the receive data can be confirmed using the DTLN bits in the CFIFOCTR or DnFIFOCTR register.

Table 17 - 20 FIFO Port Function Settings

Register Name	Bit Name	Function	Remark
CFIFOSEL DnFIFOSEL	RCNT	Selects DTLN read mode.	
	REW	Buffer memory rewind (re-read, rewrite).	
	DCLRM	Automatically clears receive data for a specified pipe after the data has been read.	Only for DnFIFO.
	DREQE	Enables DTC transfers.	Only for DnFIFO.
	MBW	Selects the FIFO port access bit width.	
	BIGEND	Selects FIFO port endian.	
	ISEL	FIFO port access direction.	Only for DCP.
	CURPIPE	Selects the current pipe.	
CFIFOCTR DnFIFOCTR	BVAL	Ends writing to the buffer memory.	
	BCLR	Clears the buffer memory on the CPU side.	
	DTLN	Checks the length of receive data.	

(1) FIFO Port Selection

Table 17 - 21 shows the pipes that can be selected with the various FIFO ports. The pipe to be accessed should be selected using the CURPIPE bits in the CFIFOSEL or DnFIFOCTR register. After the pipe is selected, whether the written value can be correctly read from the CURPIPE bits should be checked. (If the previous pipe number is read, it indicates that the pipe modification is being executed by the USB controller.) Then, the FIFO port can be accessed after FRDY = 1 is checked.

In addition, the bus width to be accessed should be selected using the MBW bit. The buffer memory access direction conforms to the DIR bit in the PIPEnCFG register. Only for the DCP, the ISEL bit determines the direction.

Table 17 - 21 FIFO Port Access Categorized by Pipe

Pipe	Access Method	Port that can be Used
DCP	CPU access	CFIFO port register (CFIFOM)
PIPE4 to PIPE7	CPU access	CFIFO port register (CFIFOM) D0FIFO/D1FIFO port register (D0FIFOM/D1FIFOM)
	DTC access	DTC transfer D0FIFO/D1FIFO port register (D0FIFOD00/D1FIFOD00)

(2) REW Bit

It is possible to temporarily stop access to the pipe currently being accessed, access a different pipe, and then continue processing for the current pipe again. The REW bit in the CFIFOSEL or DnFIFOSEL register is used for this processing.

If a pipe is selected through the CURPIPE bits in the CFIFOSEL or DnFIFOSEL register with the REW bit set to 1, the pointer used for reading from and writing to the buffer memory is reset, and reading or writing can be carried out from the first byte. If a pipe is selected with 0 set for the REW bit, data can be read and written in continuation from the previous selection, without the pointer being reset.

To access the FIFO port, FRDY = 1 should be checked after selecting a pipe.

17.4.5.3 DTC Transfers (D0FIFO and D1FIFO Ports)

(1) Overview of DTC Transfers

For PIPE4 to PIEP7, the FIFO port can be accessed using the DTC. When accessing the buffer for the pipe targeted for DTC transfer is enabled, a DTC transfer request is issued.

The unit of transfer to the FIFO port should be selected using the MBW bit in the DnFIFOSEL register and the pipe targeted for the DTC transfer should be selected using the CURPIPE bits. The selected pipe should not be changed during the DTC transfer.

Cycle steal transfer and block transfer are available as DTC transfers, and can be set by the D0DBLK/D1DBLK bit.

A DTC transfer interrupt is generated for each bus access in cycle steal transfer. One block of data is transferred by one DTC interrupt in block transfer.

Controlling input of the DTC transfer end signal allows the USB module to end writing of FIFO data using a DTC transfer. The buffer memory goes to transmit-enabled (same status when BVAL = 1) when the transfer end signal is sampled.

(2) DTC Setting

Figure 17 - 53 shows the procedure for setting the DTC and the USB. Table 17 - 22 lists the setting values of the DTC.

Figure 17 - 53 Procedure for Setting DTC Transfer

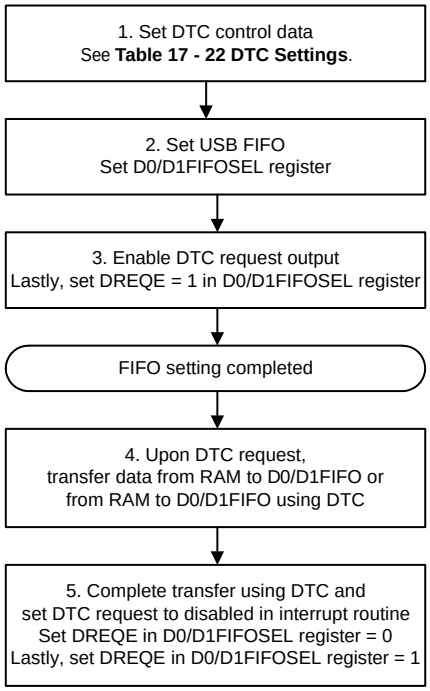


Table 17 - 22 DTC Settings

	Cycle steal transfer	Block transfer
DTCCRj	MODE = 0 (Use this setting in normal mode.) SAMOD = FIFO read direction: 0, FIFO write direction: 1 DAMOD = FIFO read direction: 1, FIFO write direction: 0 (Fix the address of the FIFO side.) CHNE = 0 (Disable chain transfers.) Specify the setting according to the setting of Sz = MBW. Setting other bits is invalid due to normal mode.	
DTBLSj (DTC block size)	01H (Sz = 0: 1 byte/Sz = 1: 2 bytes)	Sz = 0: Max. Packet Size Sz = 1: Max. Packet Size/2
DTCCTj	Any value (Max. 256 times)	Any value (Max. 256 times)
DTDARj (Destination address)	FIFO Read direction: Data transfer destination address FIFO Write direction: D0FIFOD00/D1FIFOD00	
DTSARj (Source address)	FIFO Read direction: D0FIFOD00/D1FIFOD00 FIFO Write direction: Data transfer source address	

Caution j = D0FIFO/D1FIFO are assigned to activation source (0 to 23)
For details of DTC setting, see CHAPTER 19 DATA TRANSFER CONTROLLER (DTC).

(3) DnFIFO Auto Clear Mode (D0FIFO and D1FIFO Port Reading Direction)

If 1 is set in the DCLRM bit in the DnFIFOSEL register, the USB module automatically clears the buffer memory of the selected pipe when reading of data from the buffer memory has been completed.

Table 17 - 23 shows the Packet Reception and Buffer Memory Clearing Processing by Software for each of the various settings. As shown in Table 17 - 23, the buffer clearing conditions depend on the value set in the BFRE bit. Using the DCLRM bit eliminates the need for the buffer to be cleared by software in any situation that requires buffer clearing. This enables DTC transfers without involving software.

The DnFIFO auto clear mode can be set only in the buffer memory reading direction.

Table 17 - 23 Packet Reception and Buffer Memory Clearing Processing by Software

Register Setting / Buffer Status When Packet is Received	DCLRM = 0		DCLRM = 1	
	BFRE = 0	BFRE = 1	BFRE = 0	BFRE = 1
Buffer full	Automatically cleared	Automatically cleared	Automatically cleared	Automatically cleared
Zero-length packet reception	Cleared by software	Cleared by software	Automatically cleared	Automatically cleared
Normal short packet reception	Automatically cleared	Cleared by software	Automatically cleared	Automatically cleared
Transaction count end	Automatically cleared	Cleared by software	Automatically cleared	Automatically cleared

17.4.6 Control Transfers (DCP)

In the data stage of control transfers, data is transferred using the default control pipe (DCP).
The DCP buffer memory is a 64-byte single buffer and is a fixed area that is shared for both control reading and control writing. The buffer memory can be accessed only through the CFIFO port.

(1) Setup Stage

The USB module always sends an ACK response for a correct setup packet targeted to the USB module. The operation of the USB module in the setup stage is described below.

(a) When receiving a new setup packet, the USB module sets the following bits.
- Set the VALID bit in the INTSTS0 register to 1.
- Set the PID bits in the DCPCTR register to NAK.
- Set the CCPL bit in the DCPCTR register to 0.

(b) When receiving a data packet right after the setup packet, the USB module stores the USB request parameters in the USBREQ, USBVAL, USBINDX, and USBLENG registers.

Response processing with respect to the control transfer should always be carried out after setting VALID = 0. In VALID = 1 state, PID = BUF cannot be set, and the data stage cannot be terminated.
Using the function of the VALID bit, the USB module can suspend the current request processing when receiving a new USB request during a control transfer, and can send a response to the newest request.
In addition, the USB module automatically detects the direction bit (bit 8 of bmRequestType) and the request data length (wLength) of the received USB request, distinguishes between control read transfer, control write transfer, and no-data control transfer, and controls stage transitions. For a wrong sequence, the sequence error of the control transfer stage transition interrupt is generated, and the software is notified of occurrence of the error. For control on the stages of the USB module, refer to **Figure 17 - 52**.

(2) Data Stage

Data transfers corresponding to received USB requests should be done using the DCP. Before accessing the DCP buffer memory, the access direction should be specified using the ISEL bit in the CFIFOSEL register. If the transfer data is larger than the size of the DCP buffer memory, the data transfer should be carried out using the BRDY interrupt for control write transfers and the BEMP interrupt for control read transfers.

(3) Status Stage

Control transfers are terminated by setting the CCPL bit to 1 while the PID bits in the DCPCTR register are set to BUF.
After the above settings have been made, the USB module automatically executes the status stage in accordance with the data transfer direction determined at the setup stage. The specific procedure is as follows.

[For control read transfers]
The USB module receives a zero-length packet and transmits an ACK response.

[For control write transfers and no-data control transfers]
The USB module transmits a zero-length packet and receives an ACK response from the USB host.

(4) Control Transfer Auto Response Function

The USB module automatically responds to a correct SET_ADDRESS request. If any of the following errors occurs in the SET_ADDRESS request, a response from the software is necessary.

- Any transfer other than a control read transfer: bmRequestType ≠ 00H
- Request error: wIndex ≠ 00H
- Any transfer other than a no-data control transfer: wLength ≠ 00H
- Request error: wValue > 7FH
- Control transfer of a device state error: DVSQ2 to DVSQ2 bits = 011B (Configured)

For all requests other than the SET_ADDRESS request, a response is required from the corresponding software.

17.4.7 Bulk Transfers (PIPE4, PIPE5)

The buffer memory usage (single/double buffer setting) can be selected for bulk transfers. The USB provides the following functions for bulk transfers.

- BRDY interrupt function (BFRE bit: refer to **17.4.3.1 BRDY Interrupt**)
- Transaction count function
  (TRENB, TRCLR, and TRNCNT bits: refer to **17.4.4.5 Transaction Counter (For PIPE4, PIPE5 in Reading Direction)**)
- Response PID = NAK function (SHTNAK bit: refer to **17.4.4.8 Response PID = NAK Function**)
- Auto response mode (ATREPM bit: refer to **17.4.4.9 Auto Response Mode**)

17.4.8 Interrupt Transfers (PIPE6, PIPE7)

The USB module carries out interrupt transfers in accordance with the timing controlled by the host controller.

17.4.9 SOF Interpolation Function

If data could not be received at intervals of 1 ms because an SOF packet was corrupted or missing, the USB module interpolates the SOF. The SOF interpolation operation begins when the USBE and SCKE bits in the SYSCFG register have been set to 1 and an SOF packet is received. The interpolation function is initialized under the following conditions.

- Power-on reset
- USB bus reset
- Suspended state detected

The SOF interpolation operates as follows.

- The interpolation function is not activated until an SOF packet is received.
- After the first SOF packet is received, interpolation is carried out by counting 1 ms with an internal clock of 48 MHz.
- After the second and subsequent SOF packets are received, interpolation is carried out at the previous reception interval.
- Interpolation is not carried out in the suspended state or while a USB bus reset is being received.

The USB module supports the following functions based on the SOF packet reception. These functions also operate normally with SOF interpolation, if the SOF packet was missing.

- Updating of the frame number
- SOFR interrupt timing

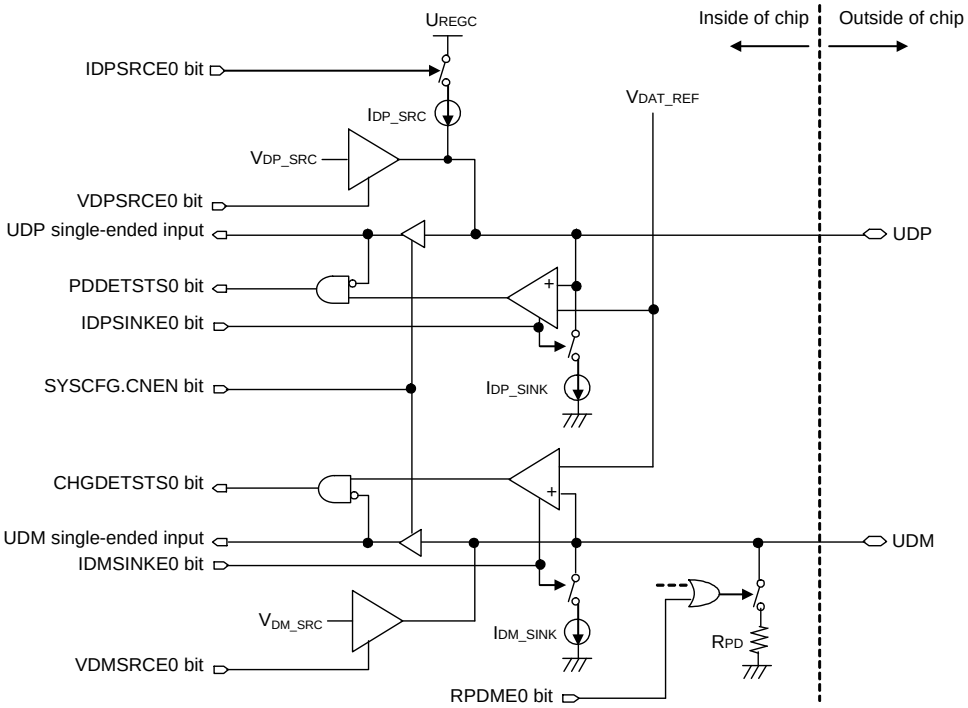
If an SOF packet is missing, the FRNM bit in the FRMNUM register is not updated.

17.4.10 Controlling battery charging detection

It is possible to control the processing for data contact detection (D+ line contact check), primary detection (charger detection), and secondary detection (charger verification) in compliance with Battery Charging Specification Revision 1.2.
By executing these processes according to charger detection algorithms, it is possible to determine whether the connected device is a standard downstream port, charging downstream port, or dedicated charging port (BC connection detection function; USB port) as a portable device.
For details on the BC connection detection processing (flow), refer to the related “application technology document (application note)”.

The above data contact detection, primary detection, and secondary detection are each detected by the interface detection circuit for BC connection detection that is provided along with the USB transceiver. This circuit has a function to detect the voltage sources (V_{DP_SRC}, V_{DM_SRC}), current source (I_{DP_SRC}), and voltage (V_{DAT_REF}) necessary to perform connection detection compliant to Battery Charging Specification Revision 1.2. These can be controlled and monitored with the bits in BC control register 0 (USBBCCTRL0).
Figures 17 - 54 shows the interface circuit for BC connection detection.

Figure 17 - 54 BC Connection Detection Interface Circuit (USB port) with BC Connection Detection Function



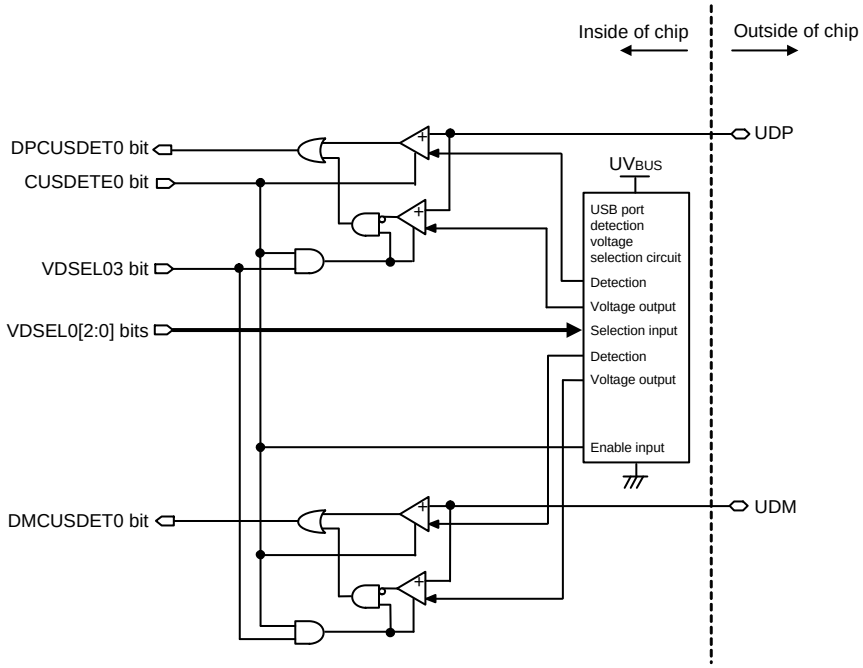
17.4.11 Battery charging connection detection optional functions

For extensibility of the battery charging specifications, the following optional functions are added to control connection detection.

• USB port voltage detection function (16 stages)
As an optional function of the BC connection detection function, this function can detect the voltage level to be input to the USB port, using the 16-stage reference voltage that is obtained by dividing 5 V to be applied to the UV_{BUS} pin.

After voltage is applied to the UV_{BUS} pin, these functions can control various functions and detection results by setting BC option control register 0 (USBBCOPT0).
For details on the BC connection detection optional functions, refer to the related “application technology document (application note)”.
Figure 17 - 55 shows the BC Connection Detection Optional Function Interface Circuit (USB Port) with BC Connection Detection Function.

Figure 17 - 55 BC Connection Detection Optional Function Interface Circuit (USB Port) with BC Connection Detection Function



17.4.12 Battery Charging Detection Processing

It is possible to control to the processing for Data Contact Detection (D+ line contact check), Primary Detection (Charger detection), and Secondary Detection (Charger verification), which are defined by the Battery Charging Specification.
The following shows required operations for a Peripheral Device.

17.4.12.1 Processing when Function Controller is Selected

The following processing is required when operating the USB module as a Portable Device for Battery Charging.

(1) Detect when the data lines (D+/D-) have made contact and start the processing for Primary Detection.
(2) After Primary Detection starts, wait 40 ms for masking, and then check the D- voltage level to confirm the Primary Detection result.
(3) If the Charger is detected during Primary Detection, also start Secondary Detection.
(4) After Secondary Detection starts, wait 40 ms for masking, and then check the D+ voltage level to confirm the Secondary Detection result.

For the above step (1), after VBUS is detected using the VBINT interrupt and the VBSTS bit, wait for 300 ms to 900 ms by software, and then set the VDPSRCE and IDMSINKE bits in the USBBCCTRL register. Or set the IDPSRCE bit, and after a change from high to low on the D+ line is detected using the LNST bits, clear the IDPSRCE bit and set the VDPSRCE and IDMSINKE bits. The VDPSRCE and IDMSINKE bits must be set at the same time. Note 1

For the above step (2), set the VDPSRCE and IDMSINKE bits and wait for 40 ms by software, and then use the CHGDETSTS bit to verify the Primary Detection result. Note 2

For the above step (3), verify that the Charger is detected if the CHGDETSTS bit is set in the above step (2), and then clear the VDPSRCE and IDMSINKE bits and set the VDMSRCE and IDPSINKE bits.

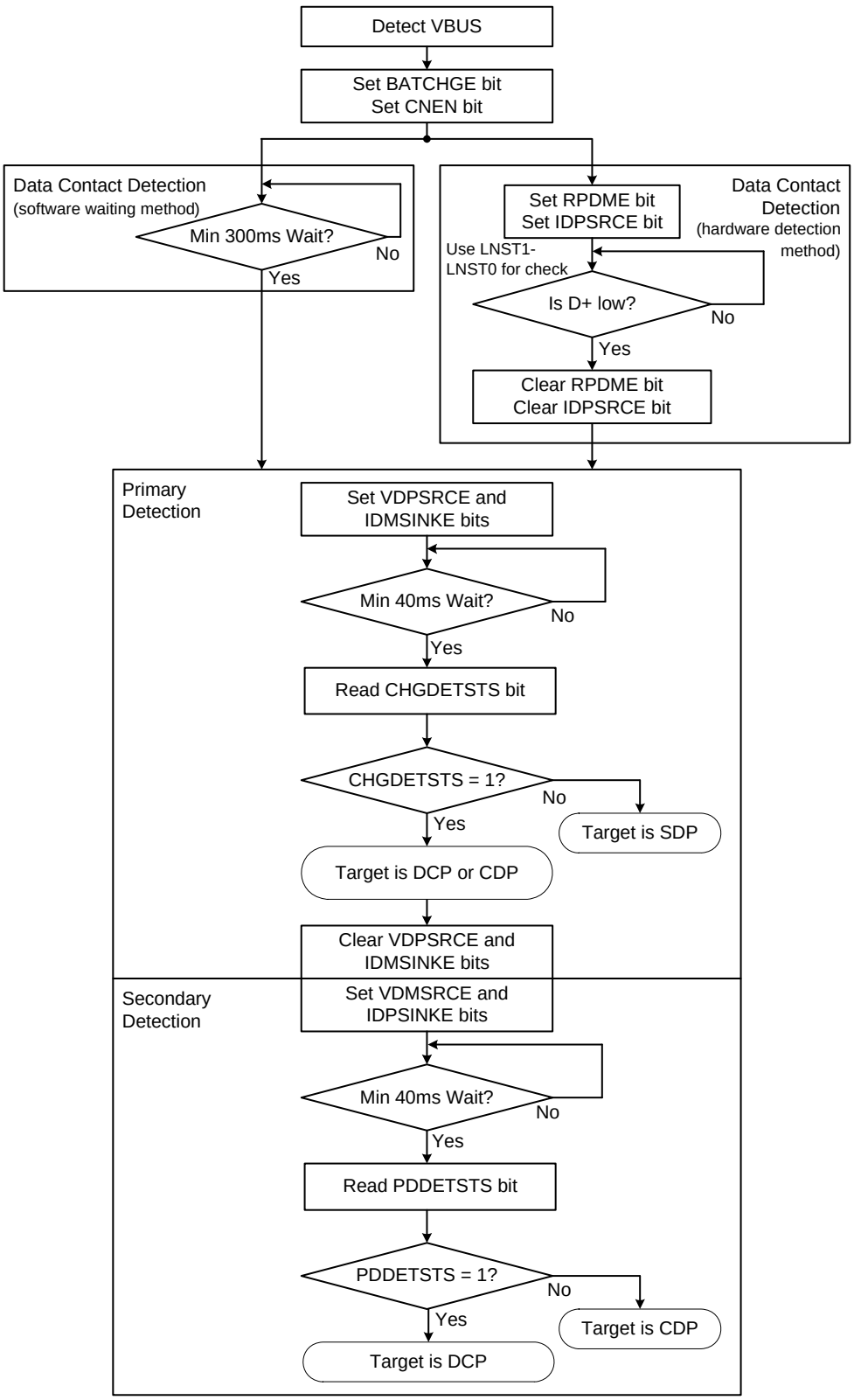
For the above step (4), set the VDMSRCE and IDPSINKE bits and wait for 40 ms by software, and then use the PDDETSTS bit to verify the Secondary Detection result.

Figure 17 - 56 shows the Process Flow for Operating as Portable Device.

Note 1. The Battery Charging Specification describes two implementation methods of the process flow for Data Contact Detection (D+/D- line contact check). One of the methods is to detect a change to logic low due to the pull-down resistor of the Host Device when the D+/D- lines have made contact with the target while the D+ line is held at logic high by applying a current of 7 to 13 µA on the D+ line.
The other method is to wait for 300 ms to 900 ms after VBUS is detected.

Note 2. During Primary Detection, when the voltage on the D- line is detected to be 0.25 V to 0.4 V or above and 0.8 V to 2.0 V or below, the target device is recognized as the Host Device for Battery Charging (Charging Downstream Port).
When using a PHY in which the 0CHGDETSTS bit only indicates that the voltage on the D- line is 0.25 V to 0.4 V or above, add the processing to check that the voltage on D- line is 0.8 V to 2.0 V or below using the LNST bits, as necessary.

Figure 17 - 56 Process Flow for Operating as Portable Device



CHAPTER 18 LCD CONTROLLER/DRIVER

The number of LCD display function pins of the RL78/L1C differs depending on the product. The following table shows the number of pins of each product.

Table 18 - 1 Number of LCD Display Function Pins of Each Product

Item		RL78/L1C															
		80/85-pin (R5F11xM/R5F11xN (x = 0, 1))								100-pin (R5F11xP (x = 0, 1))							
Number of LCD output pins		Segment signal outputs: 44 (40) Note Common signal outputs: 8								Segment signal outputs: 56 (52) Note Common signal outputs: 8							
Shared I/O port		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Segment	P0	SEG 55	SEG 54	SEG 53	SEG 52	SEG 51	SEG 50	SEG 49	SEG 48	SEG 55	SEG 54	SEG 53	SEG 52	SEG 51	SEG 50	SEG 49	SEG 48
	P1	—	—	—	—	—	SEG 42	SEG 41	SEG 40	SEG 47	SEG 46	SEG 45	SEG 44	SEG 43	SEG 42	SEG 41	SEG 40
	P2	SEG 39	SEG 38	SEG 37	SEG 36	SEG 35	SEG 34	SEG 33	SEG 32	SEG 39	SEG 38	SEG 37	SEG 36	SEG 35	SEG 34	SEG 33	SEG 32
	P3	—	—	SEG 25	SEG 24	SEG 23	SEG 22	SEG 21	SEG 20	SEG 27	SEG 26	SEG 25	SEG 24	SEG 23	SEG 22	SEG 21	SEG 20
	P5	—	—	—	—	—	SEG 6	SEG 5	SEG 4	SEG 11	SEG 10	SEG 9	SEG 8	SEG 7	SEG 6	SEG 5	SEG 4
	P7	SEG 19	SEG 18	SEG 17	SEG 16	SEG 15	SEG 14	SEG 13	SEG 12	SEG 19	SEG 18	SEG 17	SEG 16	SEG 15	SEG 14	SEG 13	SEG 12
	P14	—	—	—	—	SEG 31	SEG 30	SEG 29	SEG 28	—	—	—	—	SEG 31	SEG 30	SEG 29	SEG 28
Shared functions between COM signal output pins and I/O ports		—								—							
Shared functions between COM signal output pins and other LCD display function pins	COM4	SEG0								SEG0							
	COM5	SEG1								SEG1							
	COM6	SEG2								SEG2							
	COM7	SEG3								SEG3							

Note ( ) indicates the number of signal output pins when 8 com is used.

18.1 Functions of LCD Controller/Driver

The functions of the LCD controller/driver in the RL78/L1C microcontrollers are as follows.

(1) Waveform A or B is selectable
(2) The LCD driver voltage generator can switch internal voltage boosting method, capacitor split method, and external resistance division method.
(3) Automatic output of segment and common signals based on automatic display data register read
(4) The reference voltage to be generated when operating the voltage boost circuit can be selected from 16 steps (contrast adjustment).
(5) LCD blinking is available

Tables 18 - 2 and 18 - 3 list the maximum number of pixels that can be displayed in each display mode.

Table 18 - 2 Maximum Number of Pixels (1/2)

(a) 80/85-pin products

Drive Waveform for LCD Driver	LCD Driver Voltage Generator	Bias Mode	Number of Time Slices	Maximum Number of Pixels
Waveform A	External resistance division	—	Static	44 (44 segment signals, 1 common signal)
		1/2	2	88 (44 segment signals, 2 common signals)
			3	132 (44 segment signals, 3 common signals)
		1/3	3	
			4	176 (44 segment signals, 4 common signals)
		1/4	8	320 (40 segment signals, 8 common signals)
	Internal voltage boosting	1/3	3	132 (44 segment signals, 3 common signals)
			4	176 (44 segment signals, 4 common signals)
		1/4	8	320 (40 segment signals, 8 common signals)
	Capacitor split	1/3	3	132 (44 segment signals, 3 common signals)
			4	176 (44 segment signals, 4 common signals)
Waveform B	External resistance division, internal voltage boosting	1/3	4	
		1/4	8	320 (40 segment signals, 8 common signals)
	Capacitor split	1/3	4	176 (44 segment signals, 4 common signals)

Table 18 - 3 Maximum Number of Pixels (2/2)

(b) 100-pin products

Drive Waveform for LCD Driver	LCD Driver Voltage Generator	Bias Mode	Number of Time Slices	Maximum Number of Pixels
Waveform A	External resistance division	—	Static	56 (56 segment signals, 1 common signal)
		1/2	2	112 (56 segment signals, 2 common signals)
			3	168 (56 segment signals, 3 common signals)
		1/3	3	
			4	224 (56 segment signals, 4 common signals)
		1/4	8	416 (52 segment signals, 8 common signals)
	Internal voltage boosting	1/3	3	168 (56 segment signals, 3 common signals)
			4	224 (56 segment signals, 4 common signals)
		1/4	8	416 (52 segment signals, 8 common signals)
	Capacitor split	1/3	3	168 (56 segment signals, 3 common signals)
			4	224 (56 segment signals, 4 common signals)
Waveform B	External resistance division, internal voltage boosting	1/3	4	
		1/4	8	416 (52 segment signals, 8 common signals)
	Capacitor split	1/3	4	224 (56 segment signals, 4 common signals)

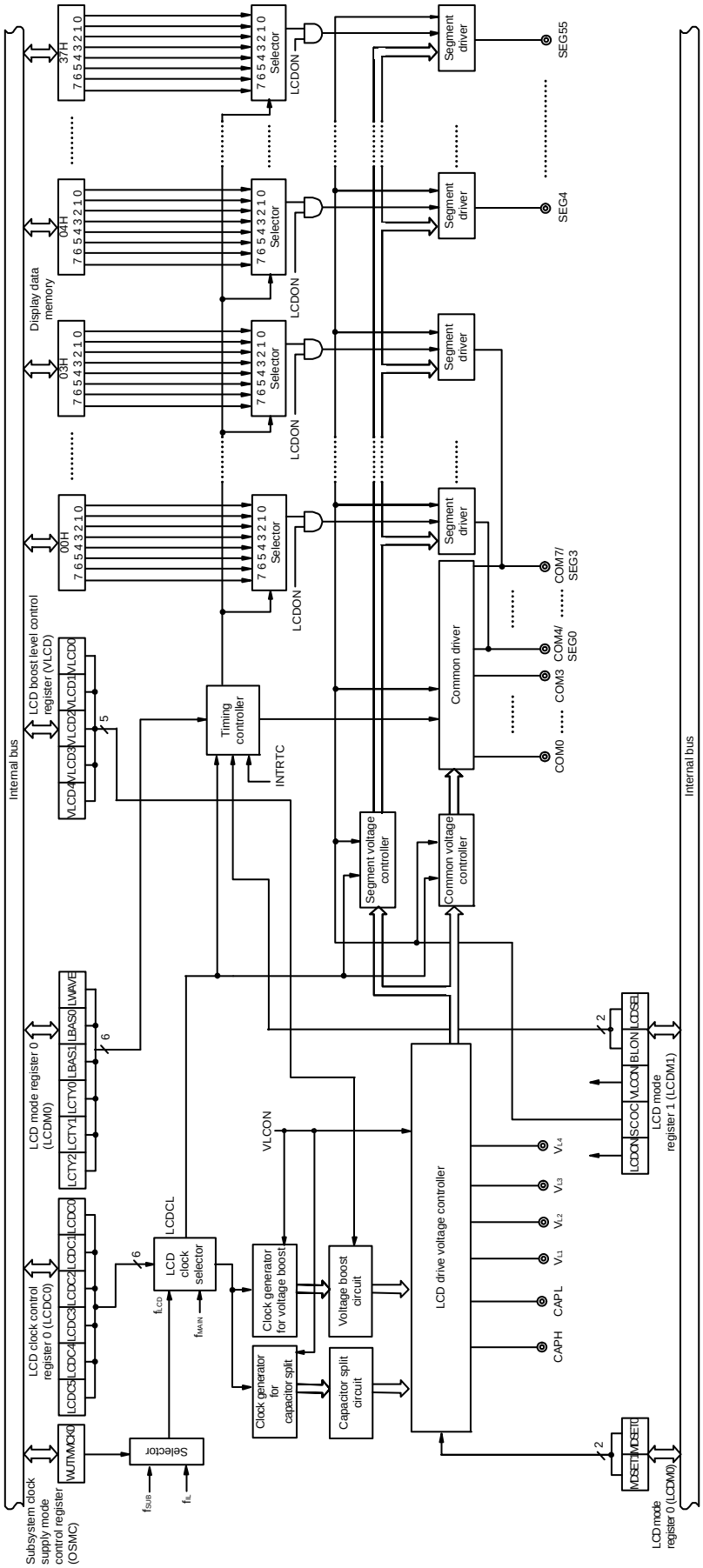
18.2 Configuration of LCD Controller/Driver

The LCD controller/driver consists of the following hardware.

Table 18 - 4 Configuration of LCD Controller/Driver

Item	Configuration
Control registers	LCD mode register 0 (LCDM0) LCD mode register 1 (LCDM1) Subsystem clock supply mode control register (OSMC) LCD clock control register 0 (LCDC0) LCD boost level control register (VLCD) LCD input switch control register (ISCLCD) LCD port function registers 0 to 6 (PFSEG0 to PFSEG6) Port mode registers 0 to 3, 5, 7, 14 (PM0 to PM3, PM5, PM7, PM14)

Figure 18 - 1 Block Diagram of LCD Controller/Driver



18.3 Registers Controlling LCD Controller/Driver

The following nine registers are used to control the LCD controller/driver.

- LCD mode register 0 (LCDM0)
- LCD mode register 1 (LCDM1)
- Subsystem clock supply mode control register (OSMC)
- LCD clock control register 0 (LCDC0)
- LCD boost level control register (VLCD)
- LCD input switch control register (ISCLCD)
- LCD port function registers 0 to 6 (PFSEG0 to PFSEG6)
- Port mode registers 0 to 3, 5, 7, 14 (PM0 to PM3, PM5, PM7, PM14)

18.3.1 LCD mode register 0 (LCDM0)

LCDM0 specifies the LCD operation.

This register is set by using an 8-bit memory manipulation instruction.

Reset signal generation sets LCDM0 to 00H.

Figure 18 - 2 Format of LCD mode register 0 (LCDM0)

Address: FFF40H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDM0	MDSET1	MDSET0	LWAVE	LDTY2	LDTY1	LDTY0	LBAS1	LBAS0

MDSET1	MDSET0	LCD drive voltage generator selection
0	0	External resistance division method
0	1	Internal voltage boosting method
1	0	Capacitor split method
1	1	Setting prohibited

LWAVE	LCD display waveform selection
0	Waveform A
1	Waveform B

LDTY2	LDTY1	LDTY0	Selection of time slice of LCD display
0	0	0	Static
0	0	1	2-time slice
0	1	0	3-time slice
0	1	1	4-time slice
1	0	1	8-time slice
Other than above			Setting prohibited

LBAS1	LBAS0	LCD display bias mode selection
0	0	1/2 bias method
0	1	1/3 bias method
1	0	1/4 bias method
1	1	Setting prohibited

Caution 1. Do not rewrite the LCDM0 value while the SCOC bit of the LCDM1 register = 1.

Caution 2. When "Static" is selected (LDTY2 to LDTY0 bits = 000B), be sure to set the LBAS1 and LBAS0 bits to the default value (00B). Otherwise, the operation will not be guaranteed.

Caution 3. Only the combinations of display waveform, number of time slices, and bias method shown in Table 18 - 5 are supported.
Combinations of settings not shown in Table 18 - 5 are prohibited.

Table 18 - 5 Combinations of Display Waveform, Time Slices, Bias Method, and Frame Frequency Settings

Display Mode			Set Value							Driving Voltage Generation Method		
Display Waveform	Number of Time Slices	Bias Mode	LCTZS	LWAVE	LDTY2	LDTY1	LDTY0	LBAS1	LBAS0	External Resistance Division	Internal Voltage Boosting	Capacitor Split
Waveform A	8	1/4	0	0	1	0	1	1	0	√ (24 to 128 Hz)	√ (24 to 64 Hz)	×
Waveform A	4	1/3	0	0	0	1	1	0	1	√ (24 to 128 Hz)	√ (24 to 128 Hz)	√ (24 to 128 Hz)
Waveform A	3	1/3	0	0	0	1	0	0	1	√ (32 to 128 Hz)	√ (32 to 128 Hz)	√ (32 to 128 Hz)
Waveform A	3	1/2	0	0	0	1	0	0	0	√ (32 to 128 Hz)	×	×
Waveform A	2	1/2	0	0	0	0	1	0	0	√ (24 to 128 Hz)	×	×
Waveform A	Static		0	0	0	0	0	0	0	√ (24 to 128 Hz)	×	×
Waveform B	8	1/4	0	1	1	0	1	1	0	√ (24 to 128 Hz)	√ (24 to 64 Hz)	×
Waveform B	4	1/3	0	1	0	1	1	0	1	√ (24 to 128 Hz)	√ (24 to 128 Hz)	√ (24 to 128 Hz)

Remark √: Supported
×: Not supported

18.3.2 LCD mode register 1 (LCDM1)

LCDM1 enables or disables display operation, voltage boost circuit operation, and capacitor split circuit operation, and specifies the display data area and the low voltage mode.

LCDM1 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets LCDM1 to 00H.

Figure 18 - 3 Format of LCD mode register 1 (LCDM1)

Address: FFF41H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	<0>
LCDM1	LCDON	SCOC	VLCON	BLON	LCDSEL	0	0	LCDVLM

SCOC	LCDON	LCD display enable/disable
0	0	Output ground level to segment/common pin
0	1	
1	0	Display off (all segment outputs are deselected.)
1	1	Display on

VLCON	Voltage boost circuit or capacitor split circuit operation enable/disable
0	Stops voltage boost circuit or capacitor split circuit operation
1 Note 1	Enables voltage boost circuit or capacitor split circuit operation

BLON Note 2	LCDSEL	Display data area control
0	0	Displaying an A-pattern area data (lower four bits of LCD display data register)
0	1	Displaying a B-pattern area data (higher four bits of LCD display data register)
1	0	Alternately displaying A-pattern and B-pattern area data (blinking display corresponding to the constant-period interrupt (INTRTC) timing of the real-time clock 2 (RTC2))
1	1	

LCDVLM Note 3	Initialization control of voltage boosting pin
0	Initialization of voltage boosting pin is not controlled
1	Initialization of voltage boosting pin is controlled

Note 1. Setting is prohibited when External resistance division method.

Note 2. When f_{IL} is selected as the LCD source clock (f_{LCD}), be sure to set the BLON bit to “0”.

Note 3. This bit is used to boost the voltage efficiently when using the voltage boost circuit by setting the initial V_{LX} pin status.
When using the internal boosting method, set this bit to 1.
When using the resistance division or capacitance division method, set this bit to 0.

Caution 1. To reduce power consumption when the LCD display is off while the boosting circuit is used, set SCOC = 0 and VLCON = 0, and MDSET1 and MDSET0 = 00B.
When MDSET1 and MDSET0 = 01B, power is consumed because the internal reference voltage generator operates.

Caution 2. When the external resistance division method has been set (MDSET1 and MDSET0 of LCDM0 = 00B) or capacitor split method has been set (MDSET1 and MDSET0 = 10B), set the LCDVLM bit to 0.

Caution 3. Do not rewrite the VLCON and LCDVLM bits while SCOC = 1.

Caution 4. Set the BLON and LCDSEL bits to 0 when 8 has been selected as the number of time slices for the display mode.

Caution 5. To use the internal voltage boosting method, specify the reference voltage by using the VLCD register (select the internal boosting method (by setting the MDSET1 and MDSET0 bits of the LCDM0 register to 01B) if the default reference voltage is used), wait for the reference voltage setup time (5 ms (min.)), and then set the VLCON bit to 1.

18.3.3 Subsystem clock supply mode control register (OSMC)

OSMC is used to reduce power consumption by stopping as many unnecessary clock functions as possible.

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 2, 12-bit interval timer, clock output/buzzer output, and LCD controller/driver, is stopped in STOP mode or HALT mode while the subsystem clock is selected as the CPU clock.

In addition, the OSMC register can be used to select the operation clock of the real-time clock 2, 12-bit interval timer, clock output/buzzer output, and LCD controller/driver.

This register is set by using an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Figure 18 - 4 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 subsystem clock supply to peripheral functions. (See **Table 23 - 1** for the peripheral functions whose operations are enabled.)
1	Stops subsystem clock supply to peripheral functions except real-time clock 2, 12-bit interval timer, clock output/buzzer output, and LCD controller/driver.

WUTMMCK0 Note	Selection of operation clock for real-time clock 2, 12-bit interval timer, and LCD driver/controller	Selection of clock output from PCLBUZn pin of clock output/buzzer output
0	Subsystem clock (f_{SUB})	Selecting the subsystem clock (f_{SUB}) is enabled.
1	Low-speed on-chip oscillator clock (f_{IL})	Selecting the subsystem clock (f_{SUB}) is disabled.

Note Be sure to select the subsystem clock (WUTMMCK0 bit = 0) if the subsystem clock is oscillating.

Caution **The subsystem clock and low-speed on-chip oscillator clock can only be switched by using the WUTMMCK0 bit if the real-time clock 2, 12-bit interval timer, and LCD driver/controller are all stopped. These are stopped as follows:**

Real-time clock 2: **Set the RTCE bit to 0.**
12-bit interval timer: **Set the RINTE bit to 0.**
LCD driver/controller: **Set the SCOC and VLCON bits to 0.**

Remark RTCE: Bit 7 of real-time clock control register 0 (RTCC0)
RINTE: Bit 15 of interval timer control register (ITMC)
SCOC: Bit 6 of LCD mode register 1 (LCDM1)
VLCON: Bit 5 of LCD mode register 1 (LCDM1)

18.3.4 LCD clock control register 0 (LCDC0)

LCDC0 specifies the LCD clock.

The frame frequency is determined according to the LCD clock and the number of time slices.

This register is set by using an 8-bit memory manipulation instruction.

Reset signal generation sets LCDC0 to 00H.

Figure 18 - 5 Format of LCD clock control register 0 (LCDC0)

Address: FFF42H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDC0	0	0	LCDC05	LCDC04	LCDC03	LCDC02	LCDC01	LCDC00

LCDC05	LCDC04	LCDC03	LCDC02	LCDC01	LCDC00	LCD clock (LCDCL)
0	0	0	1	0	0	$f_{SUB}/2^5$ or $f_{IL}/2^5$
0	0	0	1	0	1	$f_{SUB}/2^6$ or $f_{IL}/2^6$
0	0	0	1	1	0	$f_{SUB}/2^7$ or $f_{IL}/2^7$
0	0	0	1	1	1	$f_{SUB}/2^8$ or $f_{IL}/2^8$
0	0	1	0	0	0	$f_{SUB}/2^9$ or $f_{IL}/2^9$
0	0	1	0	0	1	$f_{SUB}/2^{10}$
0	1	0	0	1	1	$f_{MAIN}/2^{10}$
0	1	0	1	0	0	$f_{MAIN}/2^{11}$
0	1	0	1	0	1	$f_{MAIN}/2^{12}$
0	1	0	1	1	0	$f_{MAIN}/2^{13}$
0	1	0	1	1	1	$f_{MAIN}/2^{14}$
0	1	1	0	0	0	$f_{MAIN}/2^{15}$
0	1	1	0	0	1	$f_{MAIN}/2^{16}$
0	1	1	0	1	0	$f_{MAIN}/2^{17}$
0	1	1	0	1	1	$f_{MAIN}/2^{18}$
1	0	1	0	1	1	$f_{MAIN}/2^{19}$
Other than above						Setting prohibited

Caution 1. Do not set LCDC0 when the SCOC bit of the LCDM1 register is 1.

Caution 2. Be sure to set bits 6 and 7 to “0”.

Caution 3. When the internal boosting method or capacitance division method is set, set the LCD clock (LCDCL) as follows:

• 512 Hz or less when f_{SUB} is selected.

• 235 Hz or less when f_{IL} is selected.

For details, see Table 18 - 5 Combinations of Display Waveform, Time Slices, Bias Method, and Frame Frequency Settings.

Remark f_{MAIN}: Main system clock frequency

f_{SUB}: Subsystem clock frequency

f_{IL}: Low-speed on-chip oscillator clock frequency

18.3.5 LCD boost level control register (VLCD)

VLCD selects the reference voltage that is to be generated when operating the voltage boost circuit (contrast adjustment). The reference voltage can be selected from 16 steps.

This register is set by using an 8-bit memory manipulation instruction.

Reset signal generation sets VLCD to 04H.

Figure 18 - 6 Format of LCD boost level control register (VLCD)

Address: FFF43H After reset: 04H R/W

Symbol	7	6	5	4	3	2	1	0
VLCD	0	0	0	VLCD4	VLCD3	VLCD2	VLCD1	VLCD0

VLCD4	VLCD3	VLCD2	VLCD1	VLCD0	Reference voltage selection (contrast adjustment)	V_{L4} voltage	
						1/3 bias method	1/4 bias method
0	0	1	0	0	1.00 V (default)	3.00 V	4.00 V
0	0	1	0	1	1.05 V	3.15 V	4.20 V
0	0	1	1	0	1.10 V	3.30 V	4.40 V
0	0	1	1	1	1.15 V	3.45 V	4.60 V
0	1	0	0	0	1.20 V	3.60 V	4.80 V
0	1	0	0	1	1.25 V	3.75 V	5.00 V
0	1	0	1	0	1.30 V	3.90 V	5.20 V
0	1	0	1	1	1.35 V	4.05 V	Setting prohibited
0	1	1	0	0	1.40 V	4.20 V	Setting prohibited
0	1	1	0	1	1.45 V	4.35 V	Setting prohibited
0	1	1	1	0	1.50 V	4.50 V	Setting prohibited
0	1	1	1	1	1.55 V	4.65 V	Setting prohibited
1	0	0	0	0	1.60 V	4.80 V	Setting prohibited
1	0	0	0	1	1.65 V	4.95 V	Setting prohibited
1	0	0	1	0	1.70 V	5.10 V	Setting prohibited
1	0	0	1	1	1.75 V	5.25 V	Setting prohibited
Other than above					Setting prohibited		

Caution 1. The VLCD setting is valid only when the voltage boost circuit is operating.

Caution 2. Be sure to set bits 5 to 7 to "0".

Caution 3. Be sure to change the VLCD value after having stopped the operation of the voltage boost circuit (VLCON = 0).

Caution 4. To use the internal voltage boosting method, specify the reference voltage by using the VLCD register (select the internal boosting method (by setting the MDSET1 and MDSET0 bits of the LCDM0 register to 01B) if the default reference voltage is used), wait for the reference voltage setup time (5 ms (min.)), and then set VLCON to 1.

Caution 5. With the external resistance division method and the capacitor split method, use the default value (04H) for the VLCD resistor.

18.3.6 LCD input switch control register (ISCLCD)

Input to the Schmitt trigger buffer must be disabled until the CAPL/P126, CAPH/P127, and V_{L3}/P125 pins are set to operate as LCD function pins in order to prevent through-current from entering.

This register is set by using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets ISCLCD to 00H.

Figure 18 - 7 Format of LCD input switch control register (ISCLCD)

Address: F0308H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISCLCD	0	0	0	0	0	0	ISCVL3	ISCCAP

ISCVL3	V_{L3}/P125 pin Schmitt trigger buffer control
0	Input invalid
1	Input valid

ISCCAP	CAPL/P126, CAPH/P127 pins Schmitt trigger buffer control
0	Input invalid
1	Input valid

Caution 1. If ISCVL3 = 0, set the corresponding port control registers as follows:
PU125 bit of PU12 register = 0, P125 bit of P12 register = 0

Caution 2. If ISCCAP = 0, set the corresponding port control registers as follows:
PU127 bit of PU12 register = 0, P127 bit of P12 register = 0

(1) Operation of ports that alternately function as V_{L3}, CAPL, and CAPH pins

The functions of the V_{L3}/P125, CAPL/P126, and CAPH/P127 pins can be selected by using the LCD input switch control register (ISCLCD), LCD mode register 0 (LCDM0), and port mode register 12 (PM12).

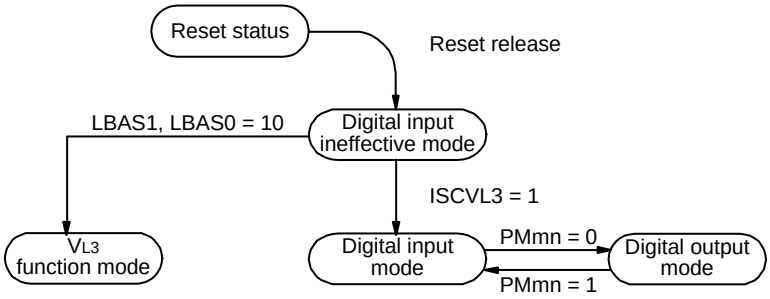
- **V_{L3}/P125**

Table 18 - 6 Settings of V_{L3}/P125 Pin Function

Bias Setting (LBAS1 and LBAS0 Bits of LCDM0 Register)	ISCVL3 Bit of ISCLCD Register	PM125 Bit of PM12 Register	Pin Function	Initial Status
Other than 1/4 bias method (LBAS1, LBAS0 = 00 or 01)	0	1	Digital input ineffective mode	√
	1	0	Digital output mode	—
	1	1	Digital input mode	—
1/4 bias method (LBAS1, LBAS0 = 10)	0	1	V_{L3} function mode	—
Other than above			Setting prohibited	

The following shows the V_{L3}/P125 pin function status transitions.

Figure 18 - 8 V_{L3}/P125 Pin Function Status Transitions



Caution Be sure to set the V_{L3} function mode before segment output starts (while SCOC bit of LCD mode register 1 (LCDM1) is 0).

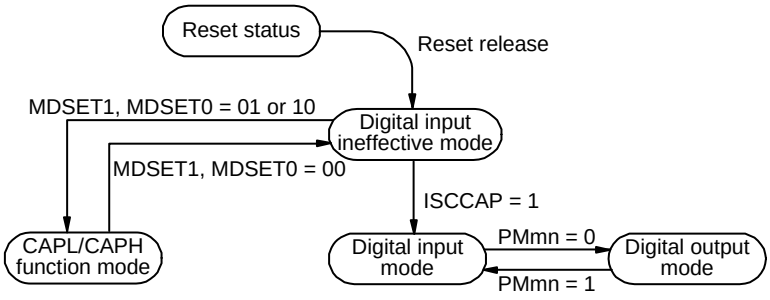
• CAPL/P126, CAPH/P127

Table 18 - 7 Settings of CAPL/P126 and CAPH/P127 Pin Functions

LCD Drive Voltage Generator (MDSET1 and MDSET0 Bits of LCDM0 Register)	ISCCAP Bit of ISCLCD Register	PM126 and PM127 Bits of PM12 Register	Pin Function	Initial Status
External resistance division (MDSET1, MDSET0 = 00)	0	1	Digital input ineffective mode	√
	1	0	Digital output mode	—
	1	1	Digital input mode	—
Internal voltage boosting or capacitor split (MDSET1, MDSET0 = 01 or 10)	0	1	CAPL/CAPH function mode	—
Other than above			Setting prohibited	

The following shows the CAPL/P126 and CAPH/P127 pin function status transitions.

Figure 18 - 9 CAPL/P126 and CAPH/P127 Pin Function Status Transitions



Caution Be sure to set the CAPL/CAPH function mode before segment output starts (while SCOC bit of LCD mode register 1 (LCDM1) is 0).

18.3.7 LCD port function registers 0 to 6 (PFSEG0 to PFSEG6)

These registers specify whether to use pins P00 to P07, P10 to P17, P20 to P27, P30 to P37, P50 to P57, P70 to P77, P140 to P143 as port pins (other than segment output pins) or segment output pins.
These registers are set by using a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets these registers to FFH (PFSEG0 is F0H).

Remark The correspondence between the segment output pins (SEGxx) and the PFSEG register (PFSEGxx bits) and the existence of SEGxx pins in each product are shown in Table 18 - 8 Segment Output Pins in Each Product and Correspondence with PFSEG Register (PFSEG Bits).

Figure 18 - 10 Format of LCD Port Function Registers 0 to 6 (100-pin Products)

Address: F0300H After reset: F0H R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG0	PFSEG07	PFSEG06	PFSEG05	PFSEG04	0	0	0	0

Address: F0301H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG1	PFSEG15	PFSEG14	PFSEG13	PFSEG12	PFSEG11	PFSEG10	PFSEG09	PFSEG08

Address: F0302H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG2	PFSEG23	PFSEG22	PFSEG21	PFSEG20	PFSEG19	PFSEG18	PFSEG17	PFSEG16

Address: F0303H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG3	PFSEG31	PFSEG30	PFSEG29	PFSEG28	PFSEG27	PFSEG26	PFSEG25	PFSEG24

Address: F0304H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG4	PFSEG39	PFSEG38	PFSEG37	PFSEG36	PFSEG35	PFSEG34	PFSEG33	PFSEG32

Address: F0305H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG5	PFSEG47	PFSEG46	PFSEG45	PFSEG44	PFSEG43	PFSEG42	PFSEG41	PFSEG40

Address: F0306H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG6	PFSEG55	PFSEG54	PFSEG53	PFSEG52	PFSEG51	PFSEG50	PFSEG49	PFSEG48

PFSEGxx (xx = 04-55)	Port (other than segment output)/segment outputs specification of Pmn pins (mn = 00 to 07, 10 to 17, 20 to 27, 30 to 37, 50 to 57, 70 to 77, 140 to 143)
0	Used as port (other than segment output)
1	Used as segment output

Caution To use the Pmn pins as segment output pins (PFSEGxx = 1), be sure to set the PUmn bit of the PUm register, POMmn bit of the POMm register, and PIMmn bit of the PIMm register to "0".

Table 18 - 8 Segment Output Pins in Each Product and Correspondence with PFSEG Register (PFSEG Bits)

Bit name of PFSEG register	Corresponding SEGxx pins	Alternate port	100-pin	80/85-pin
PFSEG04	SEG4	P50	√	√
PFSEG05	SEG5	P51	√	√
PFSEG06	SEG6	P52	√	√
PFSEG07	SEG7	P53	√	—
PFSEG08	SEG8	P54	√	—
PFSEG09	SEG9	P55	√	—
PFSEG10	SEG10	P56	√	—
PFSEG11	SEG11	P57	√	—
PFSEG12	SEG12	P70	√	√
PFSEG13	SEG13	P71	√	√
PFSEG14	SEG14	P72	√	√
PFSEG15	SEG15	P73	√	√
PFSEG16	SEG16	P74	√	√
PFSEG17	SEG17	P75	√	√
PFSEG18	SEG18	P76	√	√
PFSEG19	SEG19	P77	√	√
PFSEG20	SEG20	P30	√	√
PFSEG21	SEG21	P31	√	√
PFSEG22	SEG22	P32	√	√
PFSEG23	SEG23	P33	√	√
PFSEG24	SEG24	P34	√	√
PFSEG25	SEG25	P35	√	√
PFSEG26	SEG26	P36	√	—
PFSEG27	SEG27	P37	√	—
PFSEG28	SEG28	P140	√	√
PFSEG29	SEG29	P141	√	√
PFSEG30	SEG30	P142	√	√
PFSEG31	SEG31	P143	√	√
PFSEG32	SEG32	P20	√	√
PFSEG33	SEG33	P21	√	√
PFSEG34	SEG34	P22	√	√
PFSEG35	SEG35	P23	√	√
PFSEG36	SEG36	P24	√	√
PFSEG37	SEG37	P25	√	√
PFSEG38	SEG38	P26	√	√
PFSEG39	SEG39	P27	√	√
PFSEG40	SEG40	P10	√	√
PFSEG41	SEG41	P11	√	√
PFSEG42	SEG42	P12	√	√
PFSEG43	SEG43	P13	√	—
PFSEG44	SEG44	P14	√	—
PFSEG45	SEG45	P15	√	—
PFSEG46	SEG46	P16	√	—
PFSEG47	SEG47	P17	√	—
PFSEG48	SEG48	P00	√	√
PFSEG49	SEG49	P01	√	√
PFSEG50	SEG50	P02	√	√
PFSEG51	SEG51	P03	√	√
PFSEG52	SEG52	P04	√	√
PFSEG53	SEG53	P05	√	√
PFSEG54	SEG54	P06	√	√
PFSEG55	SEG55	P07	√	√

(1) Operation of ports that alternately function as SEGxx pins

The functions of ports that also serve as segment output pins (SEGxx) can be selected by using the port mode control register (PMCxx), port mode register (PMxx), and LCD port function registers 0 to 6 (PFSEG0 to PFSEG6).

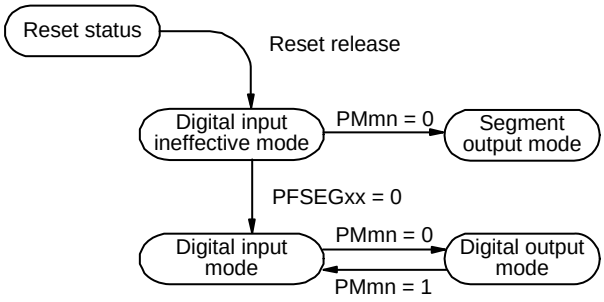
• P00 to P07, P10 to P17, P22 to P27, P30 to P37, P50 to P57, P70 to P77 (ports that do not serve as analog input pins (ANIxx))

Table 18 - 9 Settings of SEGxx/Port Pin Function

PFSEGxx Bit of PFSEG0 to PFSEG6 Registers	PMxx Bit of PMxx Register	Pin Function	Initial Status
1	1	Digital input ineffective mode	√
0	0	Digital output mode	—
0	1	Digital input mode	—
1	0	Segment output mode	—

The following shows the SEGxx/port pin function status transitions.

Figure 18 - 11 SEGxx/Port Pin Function Status Transitions



Caution Be sure to set the segment output mode before segment output starts (while SCOC bit of LCD mode register 1 (LCDM1) is 0).

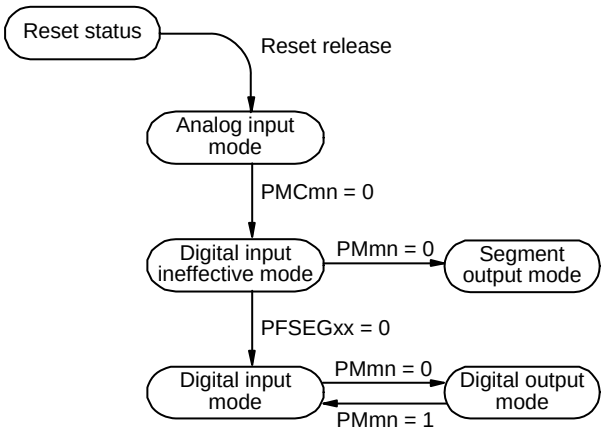
• P20, P21, P140 to P143 (ports that serve as analog input pins (ANIxx))

Table 18 - 10 Settings of ANIxx/SEGxx/Port Pin Function

PMCxx Bit of PMCxx Register	PFSEGxx Bit of PFSEG2 and PFSEG3 Registers	PMxx Bit of PMxx Register	Pin Function	Initial Status
1	1	1	Analog input mode	√
0	0	0	Digital output mode	—
0	0	1	Digital input mode	—
0	1	0	Segment output mode	—
0	1	1	Digital input ineffective mode	—
Other than above			Setting prohibited	

The following shows the ANIxx/SEGxx/port pin function status transitions.

Figure 18 - 12 ANIxx/SEGxx/Port Pin Function Status Transitions



Caution Be sure to set the segment output mode before segment output starts (while SCOC bit of LCD mode register 1 (LCDM1) is 0).

18.3.8 Port mode registers 0 to 3, 5, 7, 14 (PM0 to PM3, PM5, PM7, PM14)

These registers specify input/output of ports 0 to 3, 5, 7, and 14 in 1-bit units.

When using the ports (such as P00/SEG48) to be shared with the segment output pin for segment output, set the port mode register (PMxx) bit and port register (Pxx) bit corresponding to each port to 0.

Example: When using P00/SEG48 for segment output
Set the PM00 bit of port mode register 0 to “0”.
Set the P00 bit of port register 0 to “0”.

These registers are set by using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 18 - 13 Format of Port mode registers 0 to 3, 5, 7, 14 (PM0 to PM3, PM5, PM7, PM14) (100-pin Products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FFF23H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FFF25H	FFH	R/W
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
PM14	1	1	1	1	PM143	PM142	PM141	PM140	FFF2EH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 3, 5, 7, 14; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Remark The figure shown above presents the format of port mode registers 0 to 3, 5, 7, and 14 of the 100-pin products. The format of the port mode register of other products, see **Tables 4 - 2 to 4 - 5 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and the bits mounted on each product**.

18.4 LCD Display Data Registers

The LCD display data registers are mapped as shown in Tables 18 - 11 to 18 - 14. The contents displayed on the LCD can be changed by changing the contents of the LCD display data registers.

Table 18 - 11 Relationship Between LCD Display Data Register Contents and Segment/Common Outputs (1/4)

(a) Other than 8-time-slice (static, 2-time slice, 3-time slice, and 4-time slice) (1/2)

Register Name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	100-pin	80/85-pin
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0		
SEG0	F0400H	SEG0 (B-pattern area)				SEG0 (A-pattern area)				√	√
SEG1	F0401H	SEG1 (B-pattern area)				SEG1 (A-pattern area)				√	√
SEG2	F0402H	SEG2 (B-pattern area)				SEG2 (A-pattern area)				√	√
SEG3	F0403H	SEG3 (B-pattern area)				SEG3 (A-pattern area)				√	√
SEG4	F0404H	SEG4 (B-pattern area)				SEG4 (A-pattern area)				√	√
SEG5	F0405H	SEG5 (B-pattern area)				SEG5 (A-pattern area)				√	√
SEG6	F0406H	SEG6 (B-pattern area)				SEG6 (A-pattern area)				√	√
SEG7	F0407H	SEG7 (B-pattern area)				SEG7 (A-pattern area)				√	—
SEG8	F0408H	SEG8 (B-pattern area)				SEG8 (A-pattern area)				√	—
SEG9	F0409H	SEG9 (B-pattern area)				SEG9 (A-pattern area)				√	—
SEG10	F040AH	SEG10 (B-pattern area)				SEG10 (A-pattern area)				√	—
SEG11	F040BH	SEG11 (B-pattern area)				SEG11 (A-pattern area)				√	—
SEG12	F040CH	SEG12 (B-pattern area)				SEG12 (A-pattern area)				√	√
SEG13	F040DH	SEG13 (B-pattern area)				SEG13 (A-pattern area)				√	√
SEG14	F040EH	SEG14 (B-pattern area)				SEG14 (A-pattern area)				√	√
SEG15	F040FH	SEG15 (B-pattern area)				SEG15 (A-pattern area)				√	√
SEG16	F0410H	SEG16 (B-pattern area)				SEG16 (A-pattern area)				√	√
SEG17	F0411H	SEG17 (B-pattern area)				SEG17 (A-pattern area)				√	√
SEG18	F0412H	SEG18 (B-pattern area)				SEG18 (A-pattern area)				√	√
SEG19	F0413H	SEG19 (B-pattern area)				SEG19 (A-pattern area)				√	√
SEG20	F0414H	SEG20 (B-pattern area)				SEG20 (A-pattern area)				√	√
SEG21	F0415H	SEG21 (B-pattern area)				SEG21 (A-pattern area)				√	√
SEG22	F0416H	SEG22 (B-pattern area)				SEG22 (A-pattern area)				√	√
SEG23	F0417H	SEG23 (B-pattern area)				SEG23 (A-pattern area)				√	√
SEG24	F0418H	SEG24 (B-pattern area)				SEG24 (A-pattern area)				√	√
SEG25	F0419H	SEG25 (B-pattern area)				SEG25 (A-pattern area)				√	√
SEG26	F041AH	SEG26 (B-pattern area)				SEG26 (A-pattern area)				√	—
SEG27	F041BH	SEG27 (B-pattern area)				SEG27 (A-pattern area)				√	—
SEG28	F041CH	SEG28 (B-pattern area)				SEG28 (A-pattern area)				√	√
SEG29	F041DH	SEG29 (B-pattern area)				SEG29 (A-pattern area)				√	√
SEG30	F041EH	SEG30 (B-pattern area)				SEG30 (A-pattern area)				√	√
SEG31	F041FH	SEG31 (B-pattern area)				SEG31 (A-pattern area)				√	√
SEG32	F0420H	SEG32 (B-pattern area)				SEG32 (A-pattern area)				√	√
SEG33	F0421H	SEG33 (B-pattern area)				SEG33 (A-pattern area)				√	√

Table 18 - 12 Relationship Between LCD Display Data Register Contents and Segment/Common Outputs (2/4)

(a) Other than 8-time-slice (static, 2-time slice, 3-time slice, and 4-time slice) (2/2)

Register Name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	100-pin	80/85-pin
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0		
SEG34	F0422H	SEG34 (B-pattern area)				SEG34 (A-pattern area)				√	√
SEG35	F0423H	SEG35 (B-pattern area)				SEG35 (A-pattern area)				√	√
SEG36	F0424H	SEG36 (B-pattern area)				SEG36 (A-pattern area)				√	√
SEG37	F0425H	SEG37 (B-pattern area)				SEG37 (A-pattern area)				√	√
SEG38	F0426H	SEG38 (B-pattern area)				SEG38 (A-pattern area)				√	√
SEG39	F0427H	SEG39 (B-pattern area)				SEG39 (A-pattern area)				√	√
SEG40	F0428H	SEG40 (B-pattern area)				SEG40 (A-pattern area)				√	√
SEG41	F0429H	SEG41 (B-pattern area)				SEG41 (A-pattern area)				√	√
SEG42	F042AH	SEG42 (B-pattern area)				SEG42 (A-pattern area)				√	√
SEG43	F042BH	SEG43 (B-pattern area)				SEG43 (A-pattern area)				√	—
SEG44	F042CH	SEG44 (B-pattern area)				SEG44 (A-pattern area)				√	—
SEG45	F042DH	SEG45 (B-pattern area)				SEG45 (A-pattern area)				√	—
SEG46	F042EH	SEG46 (B-pattern area)				SEG46 (A-pattern area)				√	—
SEG47	F042FH	SEG47 (B-pattern area)				SEG47 (A-pattern area)				√	—
SEG48	F0430H	SEG48 (B-pattern area)				SEG48 (A-pattern area)				√	√
SEG49	F0431H	SEG49 (B-pattern area)				SEG49 (A-pattern area)				√	√
SEG50	F0432H	SEG50 (B-pattern area)				SEG50 (A-pattern area)				√	√
SEG51	F0433H	SEG51 (B-pattern area)				SEG51 (A-pattern area)				√	√
SEG52	F0434H	SEG52 (B-pattern area)				SEG52 (A-pattern area)				√	√
SEG53	F0435H	SEG53 (B-pattern area)				SEG53 (A-pattern area)				√	√
SEG54	F0436H	SEG54 (B-pattern area)				SEG54 (A-pattern area)				√	√
SEG55	F0437H	SEG55 (B-pattern area)				SEG55 (A-pattern area)				√	√

Remark √: Supported, —: Not supported

Table 18 - 13 Relationship Between LCD Display Data Register Contents and Segment/Common Outputs (3/4)

(b) 8-time slice (1/2)

Register Name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	100-pin	80/85-pin
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0		
SEG0	F0400H	SEG0 Note								√	√
SEG1	F0401H	SEG1 Note								√	√
SEG2	F0402H	SEG2 Note								√	√
SEG3	F0403H	SEG3 Note								√	√
SEG4	F0404H	SEG4								√	√
SEG5	F0405H	SEG5								√	√
SEG6	F0406H	SEG6								√	√
SEG7	F0407H	SEG7								√	—
SEG8	F0408H	SEG8								√	—
SEG9	F0409H	SEG9								√	—
SEG10	F040AH	SEG10								√	—
SEG11	F040BH	SEG11								√	—
SEG12	F040CH	SEG12								√	√
SEG13	F040DH	SEG13								√	√
SEG14	F040EH	SEG14								√	√
SEG15	F040FH	SEG15								√	√
SEG16	F0410H	SEG16								√	√
SEG17	F0411H	SEG17								√	√
SEG18	F0412H	SEG18								√	√
SEG19	F0413H	SEG19								√	√
SEG20	F0414H	SEG20								√	√
SEG21	F0415H	SEG21								√	√
SEG22	F0416H	SEG22								√	√
SEG23	F0417H	SEG23								√	√
SEG24	F0418H	SEG24								√	√
SEG25	F0419H	SEG25								√	√
SEG26	F041AH	SEG26								√	—
SEG27	F041BH	SEG27								√	—
SEG28	F041CH	SEG28								√	√
SEG29	F041DH	SEG29								√	√
SEG30	F041EH	SEG30								√	√
SEG31	F041FH	SEG31								√	√
SEG32	F0420H	SEG32								√	√
SEG33	F0421H	SEG33								√	√
SEG34	F0422H	SEG34								√	√
SEG35	F0423H	SEG35								√	√
SEG36	F0424H	SEG36								√	√
SEG37	F0425H	SEG37								√	√
SEG38	F0426H	SEG38								√	√
SEG39	F0427H	SEG39								√	√
SEG40	F0428H	SEG40								√	√

Table 18 - 14 Relationship Between LCD Display Data Register Contents and Segment/Common Outputs (4/4)

(b) 8-time slice (2/2)

Register Name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	100-pin	80/85-pin
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0		
SEG41	F0429H	SEG41								√	√
SEG42	F042AH	SEG42								√	√
SEG43	F042BH	SEG43								√	—
SEG44	F042CH	SEG44								√	—
SEG45	F042DH	SEG45								√	—
SEG46	F042EH	SEG46								√	—
SEG47	F042FH	SEG47								√	—
SEG48	F0430H	SEG48								√	√
SEG49	F0431H	SEG49								√	√
SEG50	F0432H	SEG50								√	√
SEG51	F0433H	SEG51								√	√
SEG52	F0434H	SEG52								√	√
SEG53	F0435H	SEG53								√	√
SEG54	F0436H	SEG54								√	√
SEG55	F0437H	SEG55								√	√

Note The COM4 to COM7 pins and SEG0 to SEG3 pins are used alternatively.

Remark √: Supported, —: Not supported

To use the LCD display data register when the number of time slices is static, two, three, or four, the lower four bits and higher four bits of each address of the LCD display data register become an A-pattern area and a B-pattern area, respectively.
The correspondences between A-pattern area data and COM signals are as follows: bit 0 ⇔ COM0, bit 1 ⇔ COM1, bit 2 ⇔ COM2, and bit 3 ⇔ COM3.
The correspondences between B-pattern area data and COM signals are as follows: bit 4 ⇔ COM0, bit 5 ⇔ COM1, bit 6 ⇔ COM2, and bit 7 ⇔ COM3.
A-pattern area data will be displayed on the LCD panel when BLON = LCDSEL = 0 has been selected, and B-pattern area data will be displayed on the LCD panel when BLON = 0 and LCDSEL = 1 have been selected.

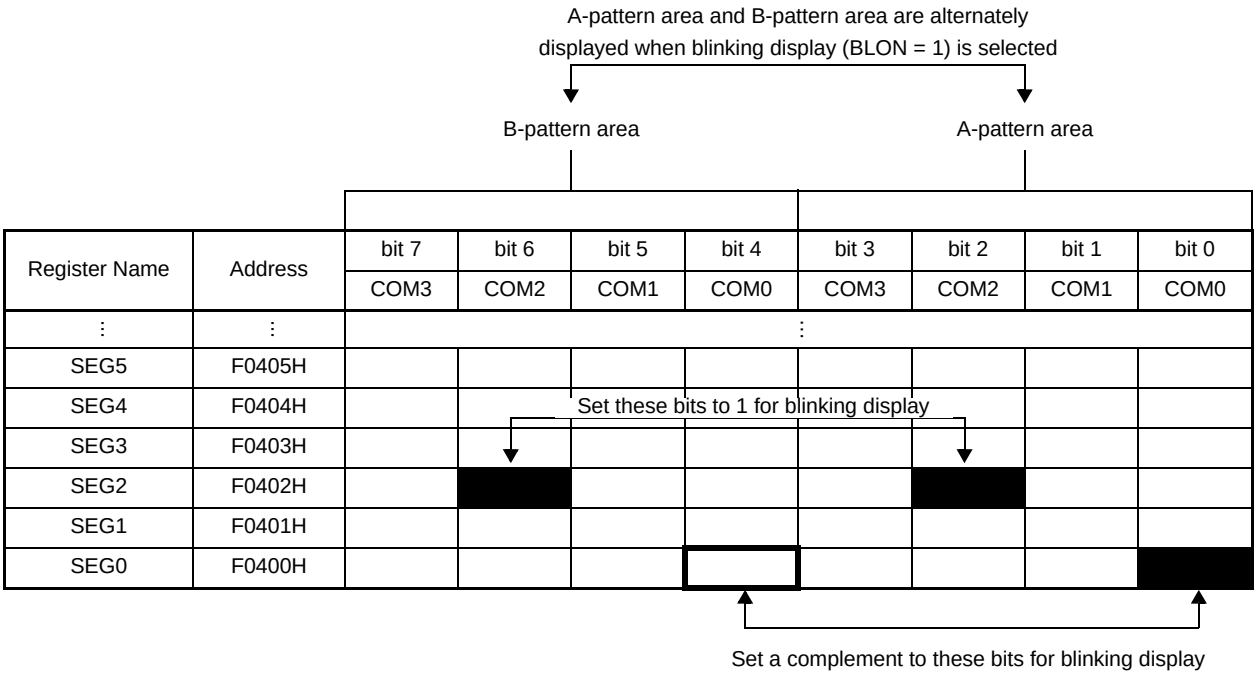
18.5 Selection of LCD Display Register

With RL78/L1C, to use the LCD display data registers when the number of time slices is static, two, three, or four, the LCD display data register can be selected from the following three types, according to the BLON and LCDSEL bit settings.

- Displaying an A-pattern area data (lower four bits of LCD display data register)
- Displaying a B-pattern area data (higher four bits of LCD display data register)
- Alternately displaying A-pattern and B-pattern area data (blinking display corresponding to the constant-period interrupt timing of the real-time clock 2 (RTC2))

Caution When 8-time slice is used, LCD display data registers (A-pattern, B-pattern, or blinking display) cannot be selected.

Figure 18 - 14 Example of Setting LCD Display Registers When Pattern Is Changed



18.5.1 A-pattern area and B-pattern area data display

When BLON = LCDSEL = 0, A-pattern area (lower four bits of the LCD display data register) data will be output as the LCD display register.

When BLON = 0, and LCDSEL = 1, B-pattern area (higher four bits of the LCD display data register) data will be output as the LCD display register.

See **18.4 LCD Display Data Registers** about the display area.

18.5.2 Blinking display (Alternately displaying A-pattern and B-pattern area data)

When BLON = 1 has been set, A-pattern and B-pattern area data will be alternately displayed, according to the constant-period interrupt (INTRTC) timing of the real-time clock 2 (RTC2). See **CHAPTER 8 REAL-TIME CLOCK 2** about the setting of the RTC constant-period interrupt (INTRTC, 0.5 s setting only) timing.

For blinking display of the LCD, set inverted values to the B-pattern area bits corresponding to the A-pattern area bits. (Example: Set 1 to bit 0 of F0400H, and set 0 to bit 4 of F0400H for blinking display.) When not setting blinking display of the LCD, set the same values. (Example: Set 1 to bit 2 of F0402H, and set 1 to bit 6 of F0402H for lighting display.)

See **18.4 LCD Display Data Registers** about the display area.

Next, the timing operation of display switching is shown.

Figure 18 - 15 Switching Operation from A-Pattern Display to Blinking Display

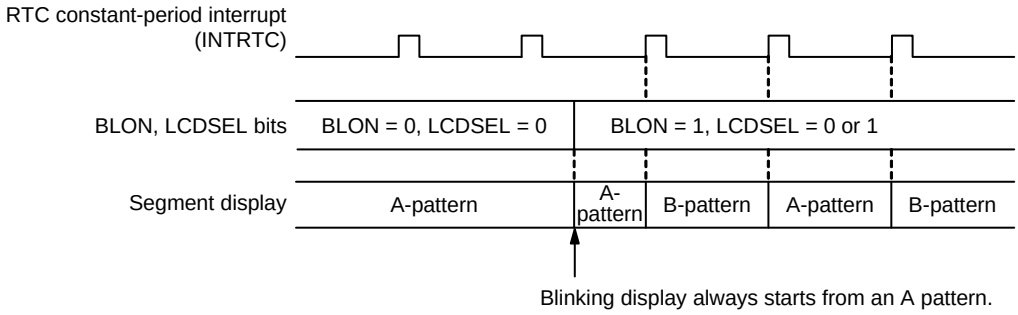
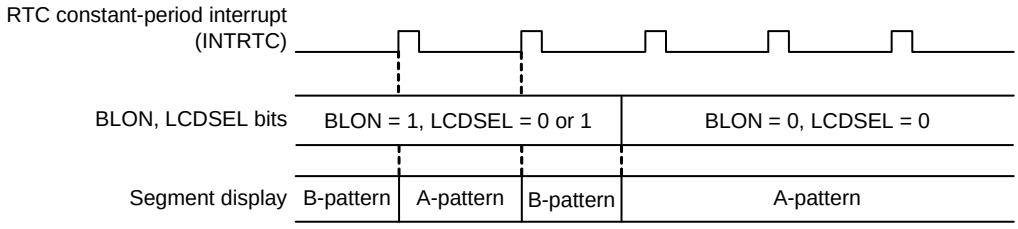


Figure 18 - 16 Switching Operation from Blinking Display to A-Pattern Display



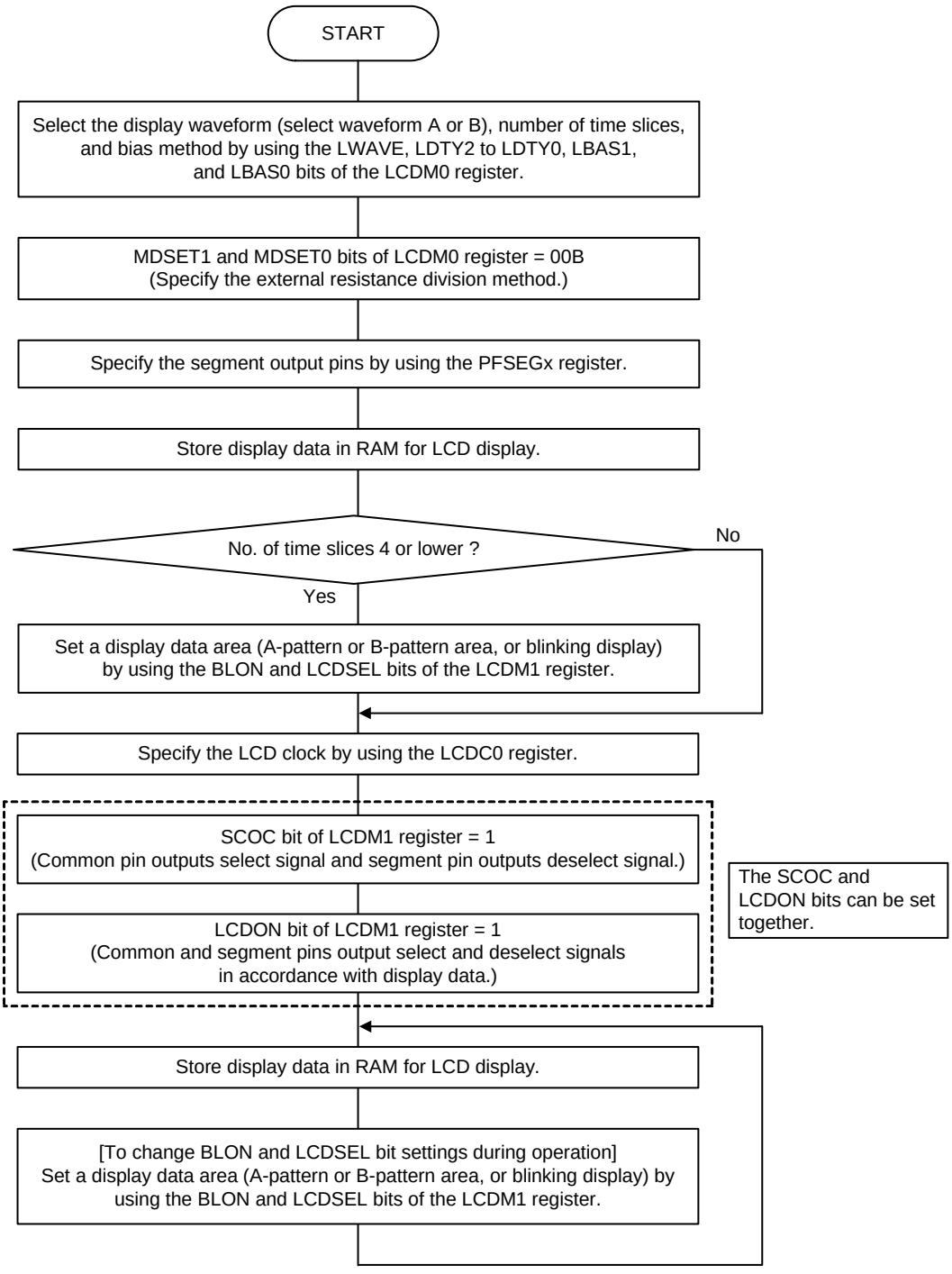
18.6 Setting the LCD Controller/Driver

Set the LCD controller/driver using the following procedure.

Caution To operate the LCD controller/driver, be sure to follow procedures (1) to (4). Unless these procedures are observed, the operation will not be guaranteed.

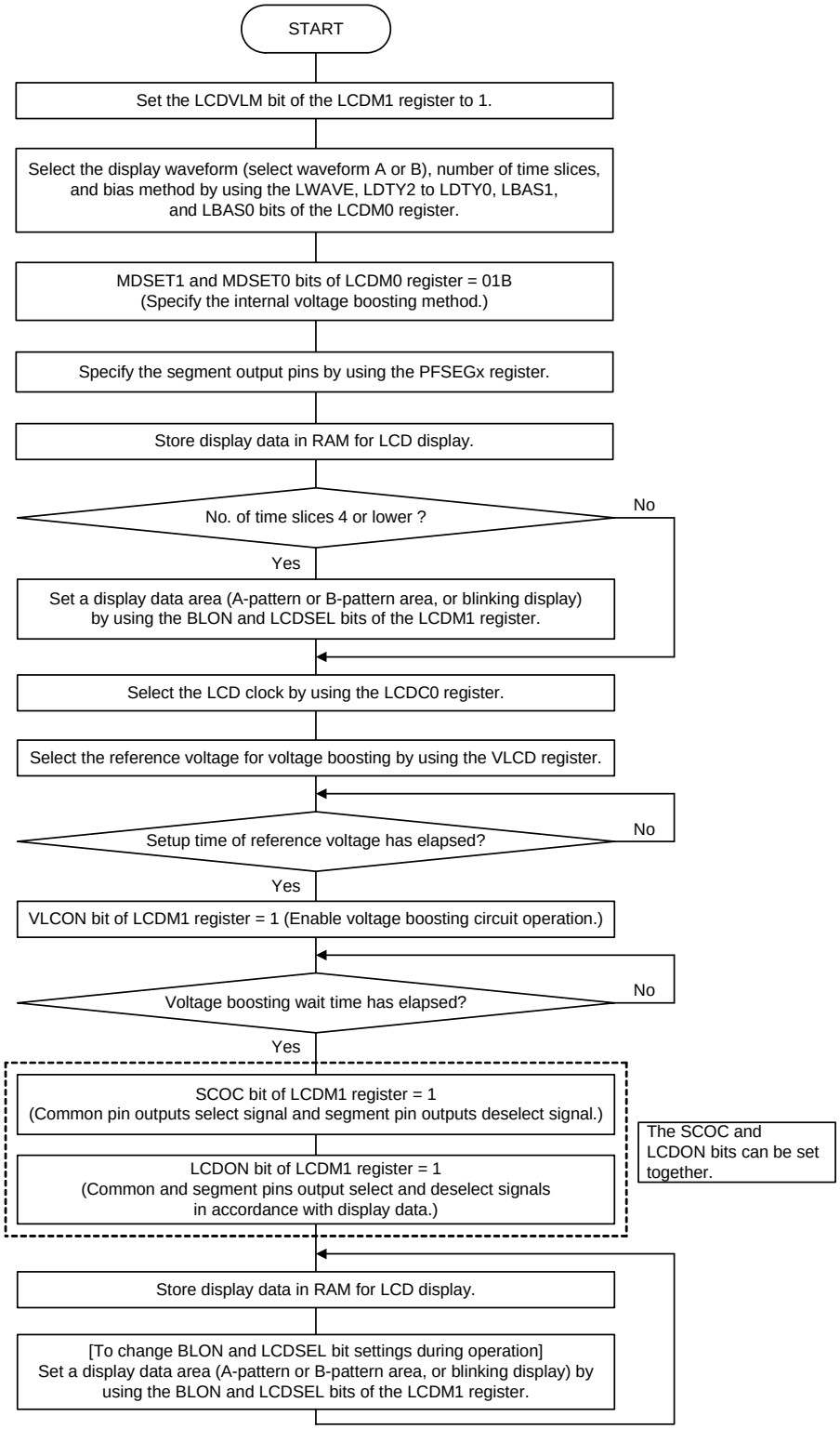
(1) External resistance division method

Figure 18 - 17 External Resistance Division Method Setting Procedure



(2) Internal voltage boosting method

Figure 18 - 18 Internal Voltage Boosting Method Setting Procedure

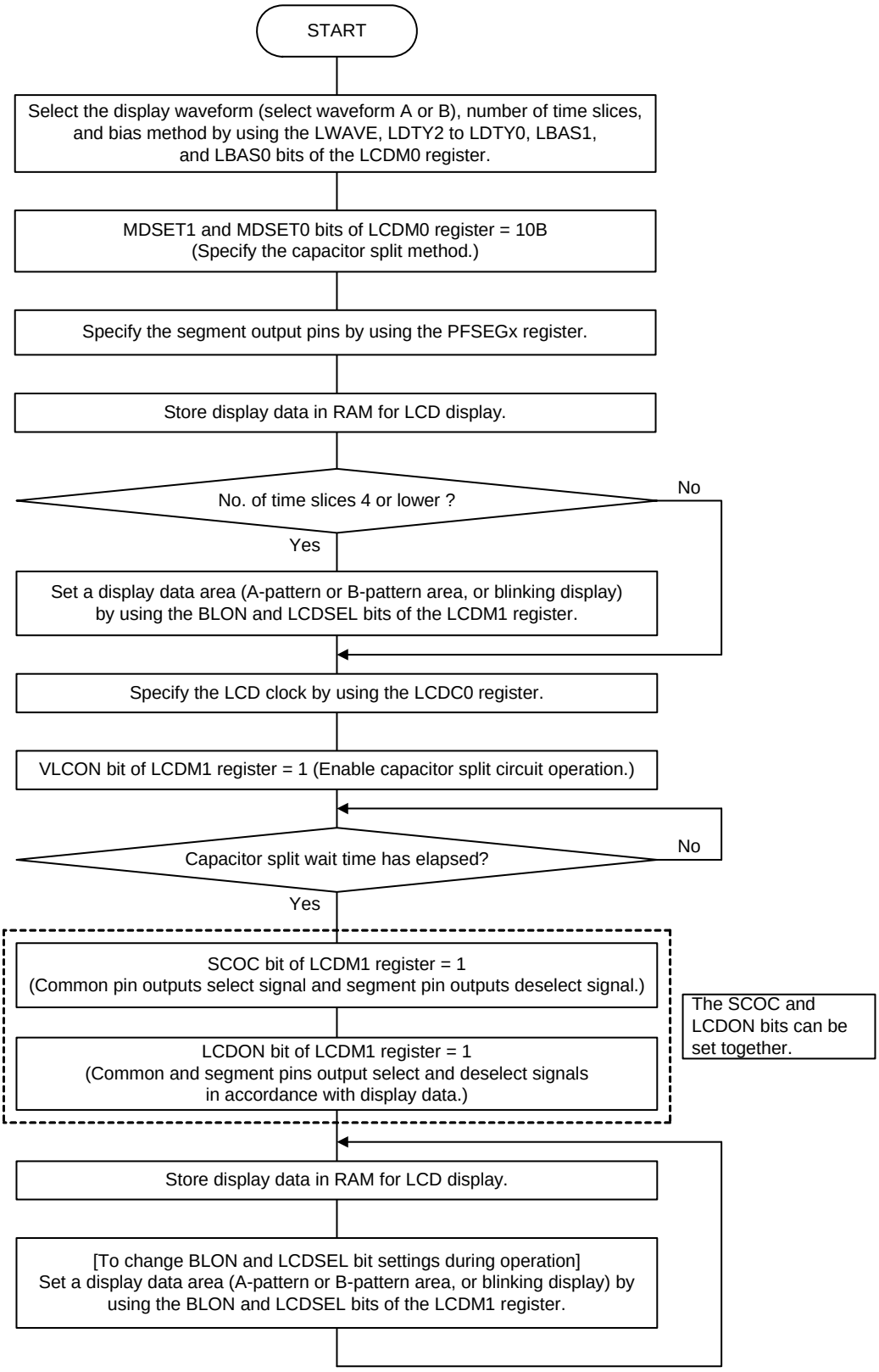


Caution 1. Wait until the setup time has elapsed even if not changing the setting of the VLCD register.

Caution 2. For the specifications of the reference voltage setup time and voltage boosting wait time, see CHAPTER 34 or CHAPTER 35 ELECTRICAL SPECIFICATIONS.

(3) Capacitor split method

Figure 18 - 19 Capacitor Split Method Setting Procedure



Caution For the specifications of the capacitor split wait time, see CHAPTER 34 or CHAPTER 35 ELECTRICAL SPECIFICATIONS.

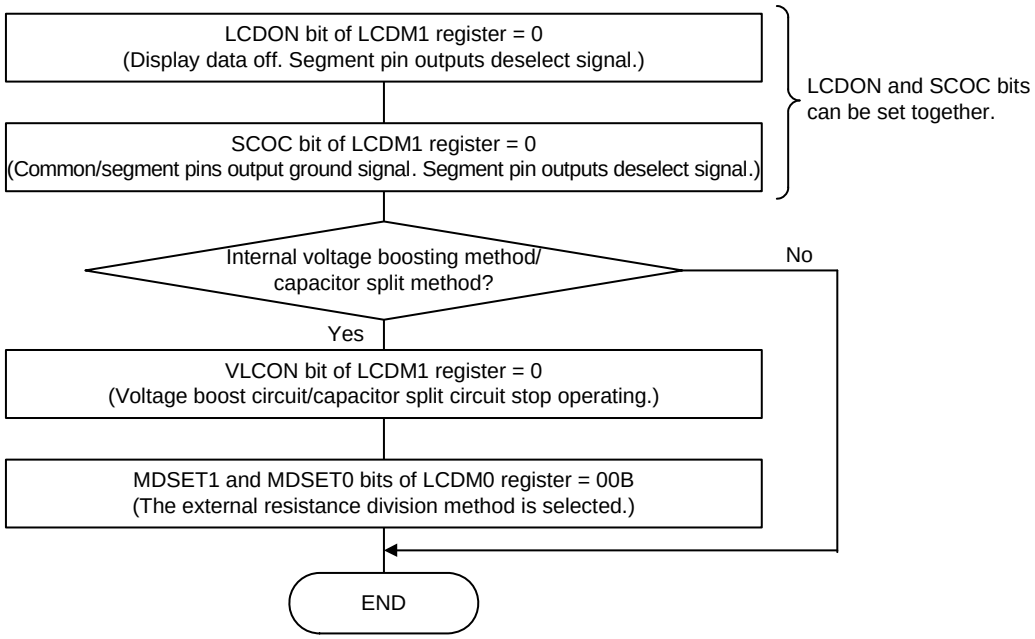
18.7 Operation stop procedure

To stop the operation of the LCD while it is displaying waveforms, follow the steps shown in the flowchart below.

The LCD stops operating when the LCDON bit of LCDM1 register and SCOC bit of the LCDM1 register are set to 0.

Figure 18 - 20 Operation Stop Procedure

(a) During waveform A or waveform B



Caution Stopping the voltage boost/capacitor split circuits is prohibited while the display is on (SCOC and LCDON bits of LCDM1 register = 11B). Otherwise, the operation will not be guaranteed. Be sure to turn off display (SCOC and LCDON bits of LCDM1 register = 10B) before stopping the voltage boost/capacitor split circuits (VLCON bit of LCDM1 register = 0).

18.8 Supplying LCD Drive Voltages V_{L1}, V_{L2}, V_{L3}, and V_{L4}

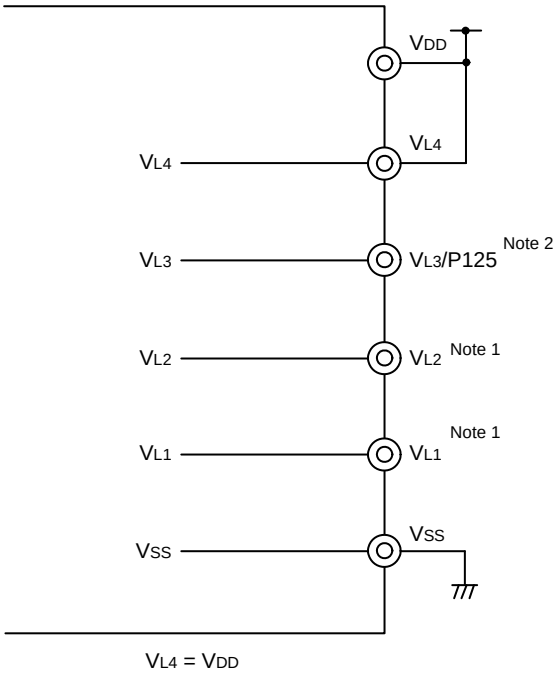
An LCD drive power supply can be generated using either of three methods: the resistance division method, internal voltage boosting method, or capacitor split method.

18.8.1 Resistance division method

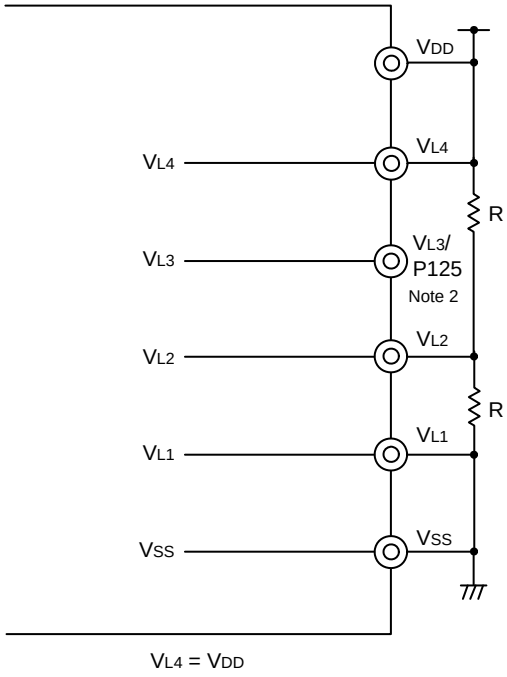
Figures 18 - 21 and 18 - 22 show examples of LCD drive voltage connection, corresponding to each bias method.

Figure 18 - 21 Examples of LCD Drive Power Connections (External Resistance Division Method) (1/2)

(a) Static display mode



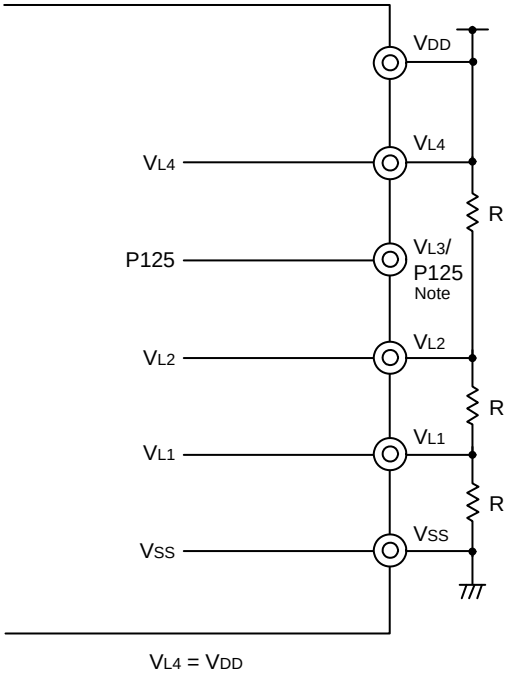
(b) 1/2 bias method



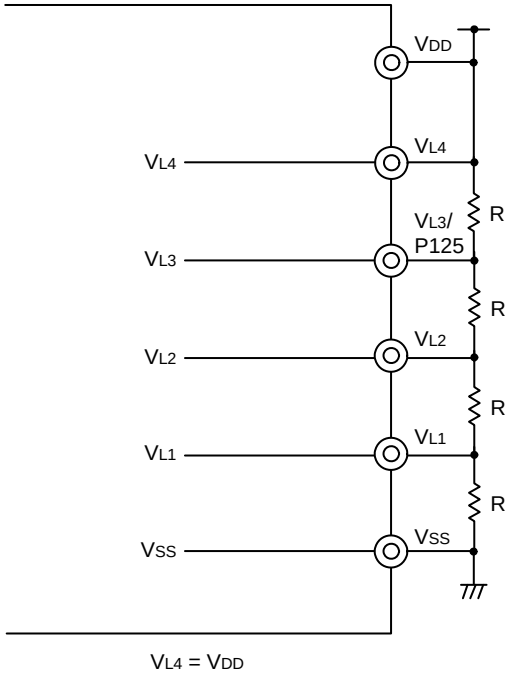
Note 1. Connect V_{L1} and V_{L2} to GND or leave open.
Note 2. V_{L3} can be used as port (P125).

Figure 18 - 22 Examples of LCD Drive Power Connections (External Resistance Division Method) (2/2)

(c) 1/3 bias method



(d) 1/4 bias method



Note V_{L3} can be used as port (P125).

Caution The reference resistance "R" value for external resistance division is 10 kΩ to 1 MΩ. In addition, to stabilize the voltage of the V_{L1} to V_{L4} pins, connect a capacitor between each of pins V_{L1} to V_{L4} and the GND pin as needed. The reference capacitance is about 0.47 μF but it depends on the LCD panel used, the number of segment pins, the number of common pins, the frame frequency, and the operating environment. Thoroughly evaluate these values in accordance with your system and adjust and determine the capacitance.

18.8.2 Internal voltage boosting method

RL78/L1C contains an internal voltage boost circuit for generating LCD drive power supplies. The internal voltage boost circuit and external capacitors (0.47 μF ±30%) are used to generate an LCD drive voltage. Only 1/3 bias mode or 1/4 bias mode can be set for the internal voltage boosting method.

The LCD drive voltage of the internal voltage boosting method can supply a constant voltage, regardless of changes in V_{DD}, because it is a power supply separate from the main unit.

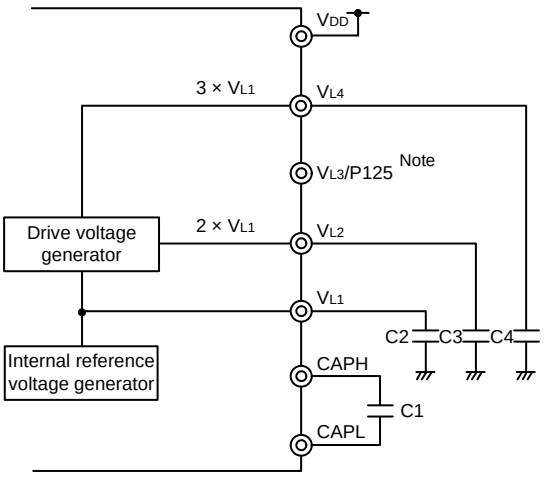
In addition, a contrast can be adjusted by using the LCD boost level control register (VLCD).

Table 18 - 15 LCD Drive Voltages (Internal Voltage Boosting Method)

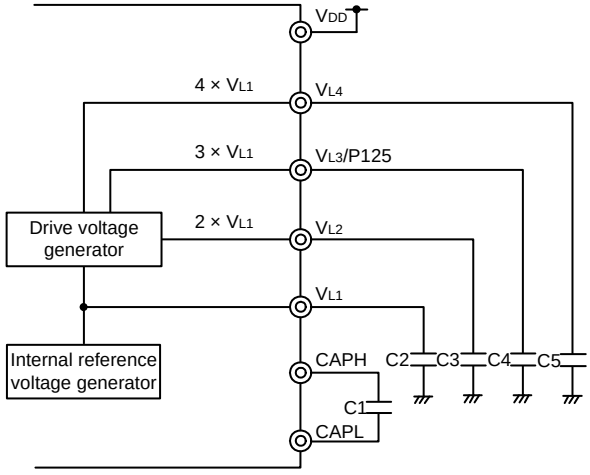
Bias Method / LCD Drive Voltage Pin	1/3 Bias Method	1/4 Bias Method
V_{L4}	3 × V_{L1}	4 × V_{L1}
V_{L3}	—	3 × V_{L1}
V^{L2}	2 × V_{L1}	2 × V_{L1}
V_{L1}	LCD reference voltage	LCD reference voltage

Figure 18 - 23 Examples of LCD Drive Power Connections (Internal Voltage Boosting Method)

(a) 1/3 bias method



(b) 1/4 bias method



Note V_{L3} can be used as port (P125).

Remark Use a capacitor with as little leakage as possible.
In addition, make C1 a nonpolar capacitor.

18.8.3 Capacitor split method

RL78/L1C contains an internal voltage reduction circuit for generating LCD drive power supplies. The internal voltage reduction circuit and external capacitors (0.47 µF±30%) are used to generate an LCD drive voltage. Only 1/3 bias mode can be set for the capacitor split method.

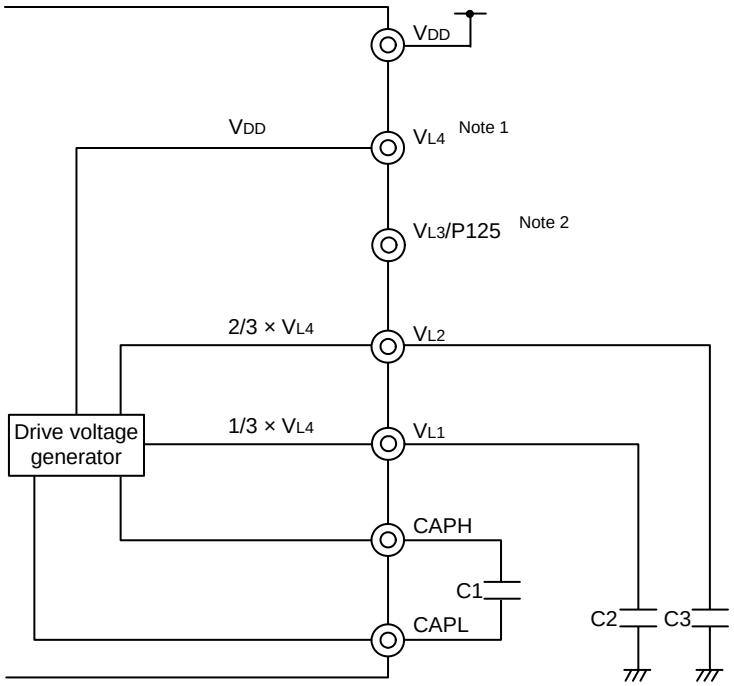
Different from the external resistance division method, there is always no current flowing with the capacitor split method, so current consumption can be reduced.

Table 18 - 16 LCD Drive Voltages (Capacitor Split Method)

LCD Drive Voltage Pin \ Bias Method	1/3 Bias Method
V_{L4}	V_{DD}
V_{L3}	—
V_{L2}	$2/3 \times V_{L4}$
V_{L1}	$1/3 \times V_{L4}$

Figure 18 - 24 Examples of LCD Drive Power Connections (Capacitor Split Method)

• 1/3 bias method



Note 1. When using the internal voltage boosting method, connect the C4 capacitor to this pin as shown in Figure 18 - 23 Examples of LCD Drive Power Connections (Internal Voltage Boosting Method).

Note 2. V_{L3} can be used as port (P125).

Remark Use a capacitor with as little leakage as possible.
In addition, make C1 a nonpolar capacitor.

18.9 Common and Segment Signals

Each pixel of the LCD panel turns on when the potential difference between the corresponding common and segment signals becomes higher than a specific voltage (LCD drive voltage, V_{LCD}). The pixels turn off when the potential difference becomes lower than V_{LCD}.

Applying DC voltage to the common and segment signals of an LCD panel causes deterioration. To avoid this problem, this LCD panel is driven by AC voltage.

(1) Common signals

Each common signal is selected sequentially according to a specified number of time slices at the timing listed in Table 18 - 17. In the static display mode, the same signal is output to COM0 to COM3.

In the two-time-slice mode, leave the COM2 and COM3 pins open. In the three-time-slice mode, leave the COM3 pin open.

Use the COM4 to COM7 pins other than in the eight-time-slice mode as open or segment pins.

Table 18 - 17 COM Signals

COM Signal / Number of Time Slices	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7
Static display mode					Note	Note	Note	Note
Two-time-slice mode			Open	Open	Note	Note	Note	Note
Three-time-slice mode				Open	Note	Note	Note	Note
Four-time-slice mode					Note	Note	Note	Note
Eight-time-slice mode								

Note Use the pins as open or segment pins.

(2) Segment signals

The segment signals correspond to the LCD display data register (see **18.4 LCD Display Data Registers**).

When the number of time slices is eight, bits 0 to 7 of each display data register are read in synchronization with COM0 to COM7, respectively. If a bit is 1, it is converted to the select voltage, and if it is 0, it is converted to the deselect voltage. The conversion results are output to the segment pins (SEG4 to SEG55).

When the number of time slices is number other than eight, bits 0 to 3 of each byte in A-pattern area are read in synchronization with COM0 to COM3, and bits 4 to 7 of each byte in B-pattern area are read in synchronization with COM0 to COM3, respectively. If a bit is 1, it is converted to the select voltage, and if it is 0, it is converted to the deselect voltage. The conversion results are output to the segment pins (SEG0 to SEG55).

Check, with the information given above, what combination of front-surface electrodes (corresponding to the segment signals) and rear-surface electrodes (corresponding to the common signals) forms display patterns in the LCD display data register, and write the bit data that corresponds to the desired display pattern on a one-to-one basis.

Remark The mounted segment output pins vary depending on the product.

- 80/85-pin products: SEG0 to SEG6, SEG12 to SEG25, SEG28 to SEG42, SEG48 to SEG55
- 100-pin products: SEG0 to SEG55

(3) Output waveforms of common and segment signals

The voltages listed in Table 18 - 18 are output as common and segment signals.

When both common and segment signals are at the select voltage, a display on-voltage of ±V_{LCD} is obtained. The other combinations of the signals correspond to the display off-voltage.

Table 18 - 18 LCD Drive Voltage

(a) Static display mode

Common Signal \ Segment Signal	Select Signal Level	Deselect Signal Level
	V_{SS}/V_{L4}	V_{L4}/V_{SS}
V_{L4}/V_{SS}	$-V_{LCD}/+V_{LCD}$	0 V/0 V

(b) 1/2 bias method

Common Signal \ Segment Signal		Select Signal Level	Deselect Signal Level
		V_{SS}/V_{L4}	V_{L4}/V_{SS}
Select signal level	V_{L4}/V_{SS}	$-V_{LCD}/+V_{LCD}$	0 V/0 V
Deselect signal level	V_{L2}	$-\frac{1}{2}V_{LCD}/+\frac{1}{2}V_{LCD}$	$+\frac{1}{2}V_{LCD}/-\frac{1}{2}V_{LCD}$

(c) 1/3 bias method (waveform A or B)

Common Signal \ Segment Signal		Select Signal Level	Deselect Signal Level
		V_{SS}/V_{L4}	V_{L2}/V_{L1}
Select signal level	V_{L4}/V_{SS}	$-V_{LCD}/+V_{LCD}$	$-\frac{1}{3}V_{LCD}/+\frac{1}{3}V_{LCD}$
Deselect signal level	V_{L1}/V_{L2}	$-\frac{1}{3}V_{LCD}/+\frac{1}{3}V_{LCD}$	$+\frac{1}{3}V_{LCD}/-\frac{1}{3}V_{LCD}$

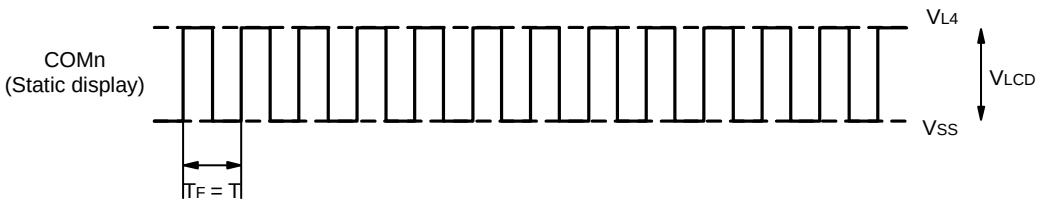
(d) 1/4 bias method (waveform A or B)

Common Signal \ Segment Signal		Select Signal Level	Deselect Signal Level
		V_{SS}/V_{L4}	V_{L2}
Select signal level	V_{L4}/V_{SS}	$-V_{LCD}/+V_{LCD}$	$-\frac{1}{2}V_{LCD}/+\frac{1}{2}V_{LCD}$
Deselect signal level	V_{L1}/V_{L3}	$-\frac{1}{4}V_{LCD}/+\frac{1}{4}V_{LCD}$	$+\frac{1}{4}V_{LCD}/-\frac{1}{4}V_{LCD}$

Figures 18 - 25 and 18 - 26 show the common signal waveforms, and Figures 18 - 27 to Figure 18 - 29 show the voltages and phases of the common and segment signals.

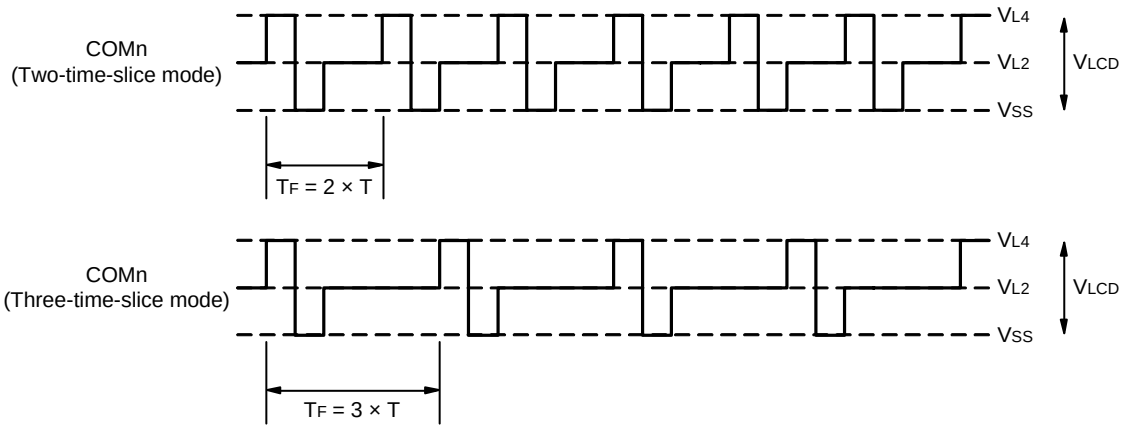
Figure 18 - 25 Common Signal Waveforms (1/2)

(a) Static display mode



T: One LCD clock period TF: Frame frequency

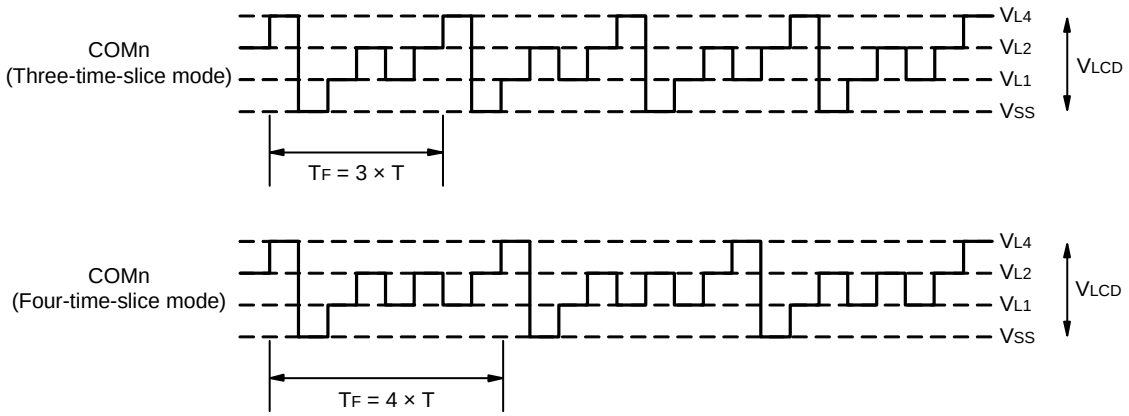
(b) 1/2 bias method



T: One LCD clock period TF: Frame frequency

Figure 18 - 26 Common Signal Waveforms (2/2)

(c) 1/3 bias method



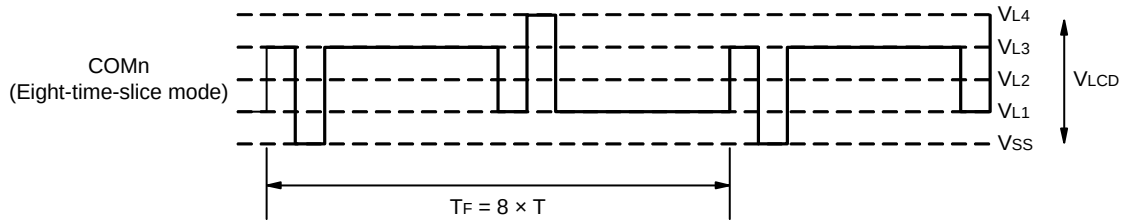
T: One LCD clock period TF: Frame frequency

< Example of calculation of LCD frame frequency (When four-time slot mode is used) >

LCD clock: $32768/2^8$ = 256 Hz (When setting to LCDC0 = 07H)

LCD frame frequency: 64 Hz

(d) 1/4 bias method



T: One LCD clock period TF: Frame frequency

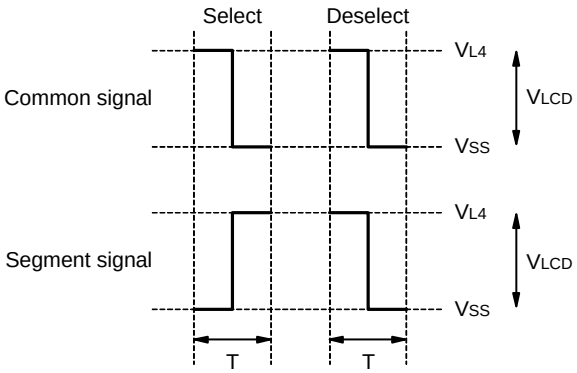
< Example of calculation of LCD frame frequency (When eight-time slot mode is used) >

LCD clock: $32768/2^8$ = 256 Hz (When setting to LCDC0 = 07H)

LCD frame frequency: 32 Hz

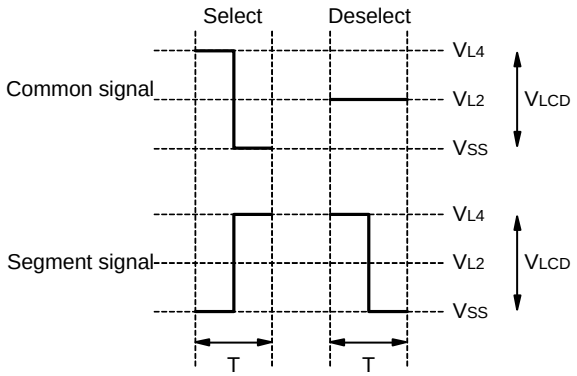
Figure 18 - 27 Voltages and Phases of Common and Segment Signals (1/3)

(a) Static display mode (waveform A)



T: One LCD clock period

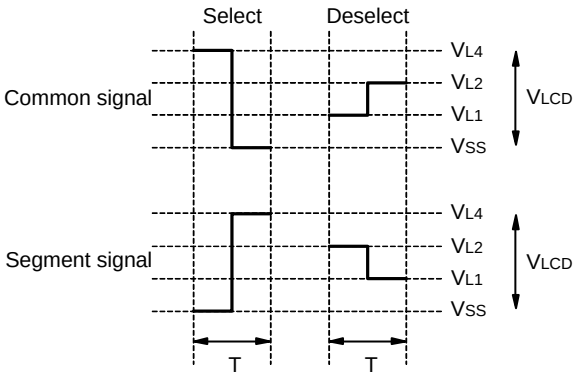
(b) 1/2 bias method (waveform A)



T: One LCD clock period

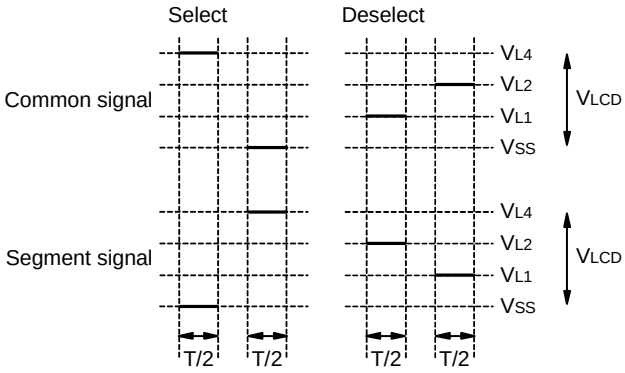
Figure 18 - 28 Voltages and Phases of Common and Segment Signals (2/3)

(c) 1/3 bias method (waveform A)



T: One LCD clock period

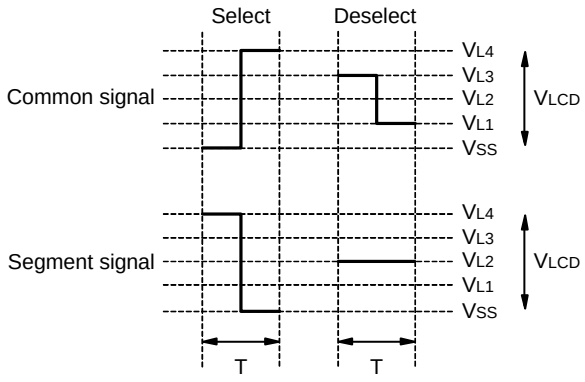
(d) 1/3 bias method (waveform B)



T: One LCD clock period

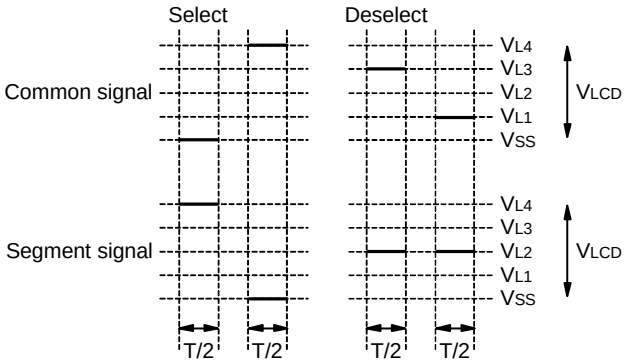
Figure 18 - 29 Voltages and Phases of Common and Segment Signals (3/3)

(e) 1/4 bias method (waveform A)



T: One LCD clock period

(f) 1/4 bias method (waveform B)



T: One LCD clock period

18.10 Display Modes

18.10.1 Static display example

Figure 18 - 31 shows how the three-digit LCD panel having the display pattern shown in Figure 18 - 30 is connected to the segment signals (SEG0 to SEG23) and the common signal (COM0). This example displays data “12.3” in the LCD panel.

The contents of the display data register (F0400H to F0417H) correspond to this display.

The following description focuses on numeral “2.” (2.) displayed in the second digit. To display “2.” in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG8 to SEG15 pins according to Table 18 - 19 at the timing of the common signal COM0; see **Figure 18 - 30** for the relationship between the segment signals and LCD segments.

Table 18 - 19 Select and Deselect Voltages (COM0)

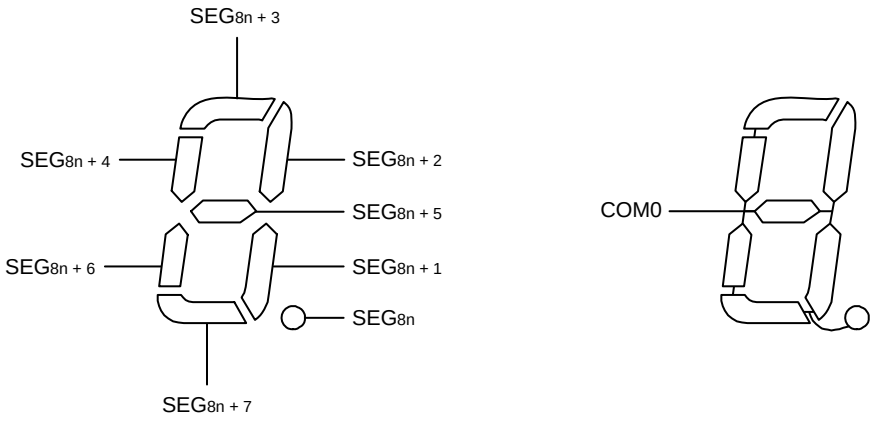
Segment / Common	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15
COM0	Select	Deselect	Select	Select	Deselect	Select	Select	Select

According to Table 18 - 19, it is determined that the bit-0 pattern of the display data register locations (F0408H to F040FH) must be 10110111.

Figure 18 - 32 shows the LCD drive waveforms of SEG11 and SEG12, and COM0. When the select voltage is applied to SEG11 at the timing of COM0, an alternate rectangle waveform, $+V_{LCD}/-V_{LCD}$, is generated to turn on the corresponding LCD segment.

COM1 to COM3 are supplied with the same waveform as for COM0. So, COM0 to COM3 may be connected together to increase the driving capacity.

Figure 18 - 30 Static LCD Display Pattern and Electrode Connections



Remark 80/85-pin products: n = 0 to 5

Figure 18 - 31 Example of Connecting Static LCD Panel

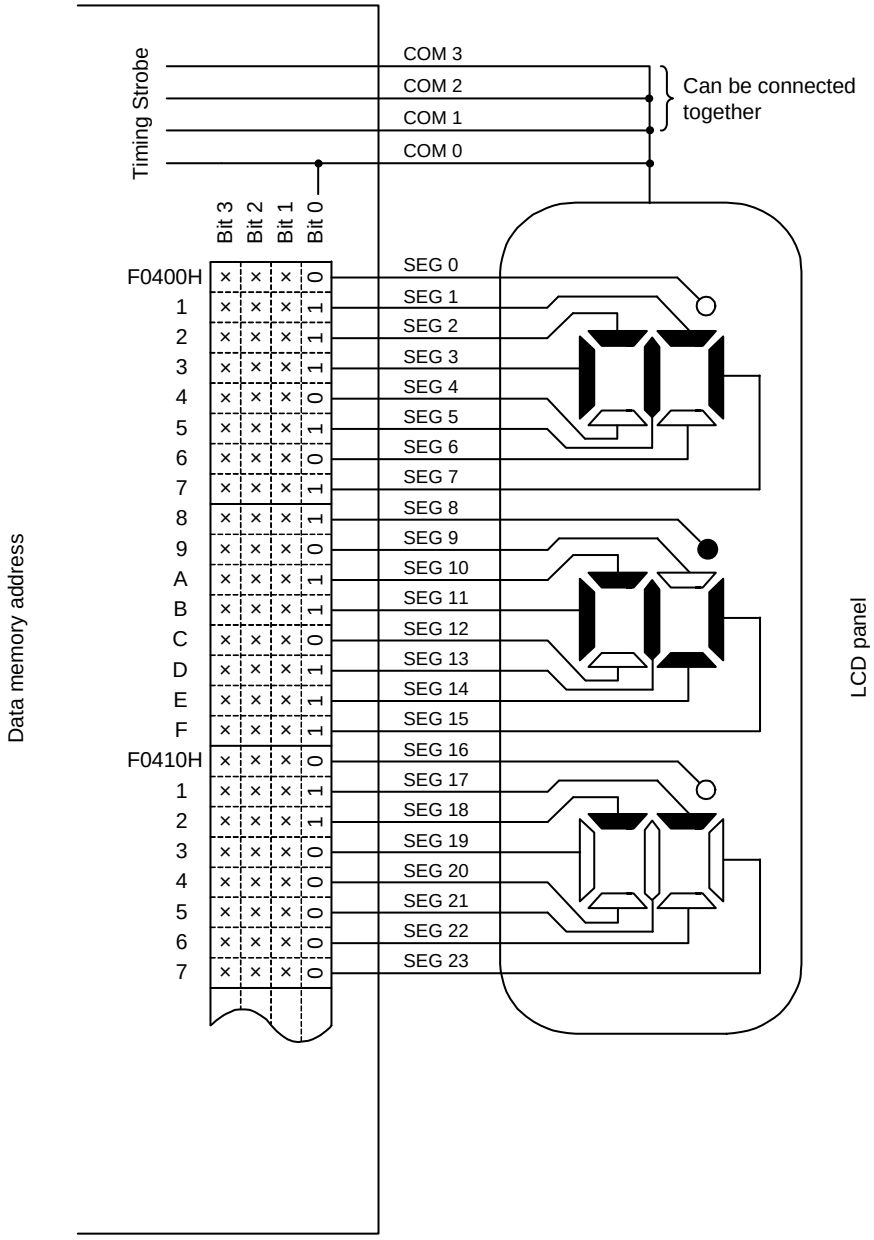
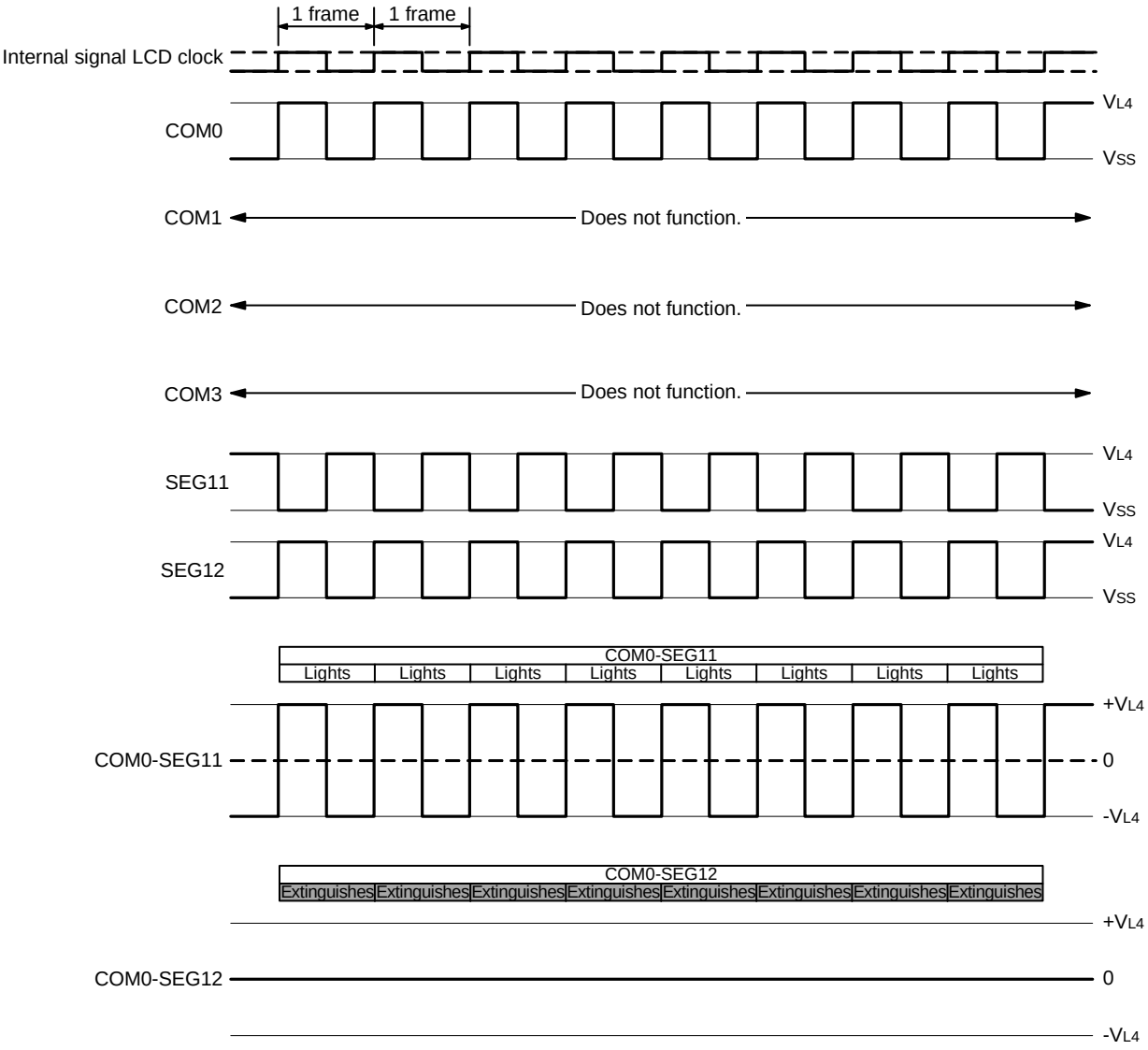


Figure 18 - 32 Static LCD Drive Waveform Examples for SEG11, SEG12, and COM0



18.10.2 Two-time-slice display example

Figure 18 - 34 shows how the 6-digit LCD panel having the display pattern shown in Figure 18 - 33 is connected to the segment signals (SEG0 to SEG23) and the common signals (COM0 and COM1). This example displays data “12345.6” in the LCD panel. The contents of the display data register (F0400H to F0417H) correspond to this display.

The following description focuses on numeral “3” (3) displayed in the fourth digit. To display “3” in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG12 to SEG15 pins according to Table 18 - 20 at the timing of the common signals COM0 and COM1; see **Figure 18 - 33** for the relationship between the segment signals and LCD segments.

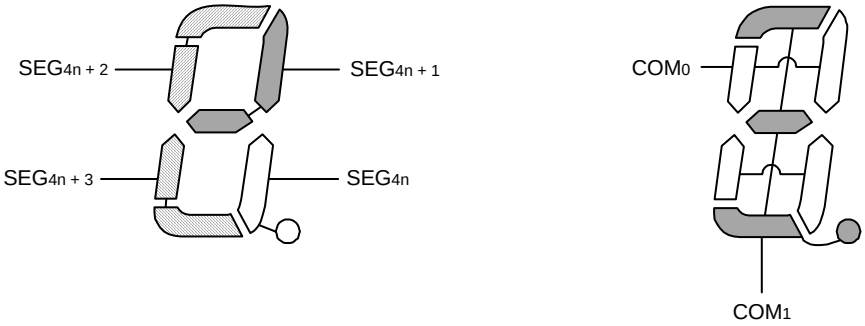
Table 18 - 20 Select and Deselect Voltages (COM0 and COM1)

Segment / Common	SEG12	SEG13	SEG14	SEG15
COM0	Select	Select	Deselect	Deselect
COM1	Deselect	Select	Select	Select

According to Table 18 - 20, it is determined that the display data register location (F040FH) that corresponds to SEG15 must contain xx10.

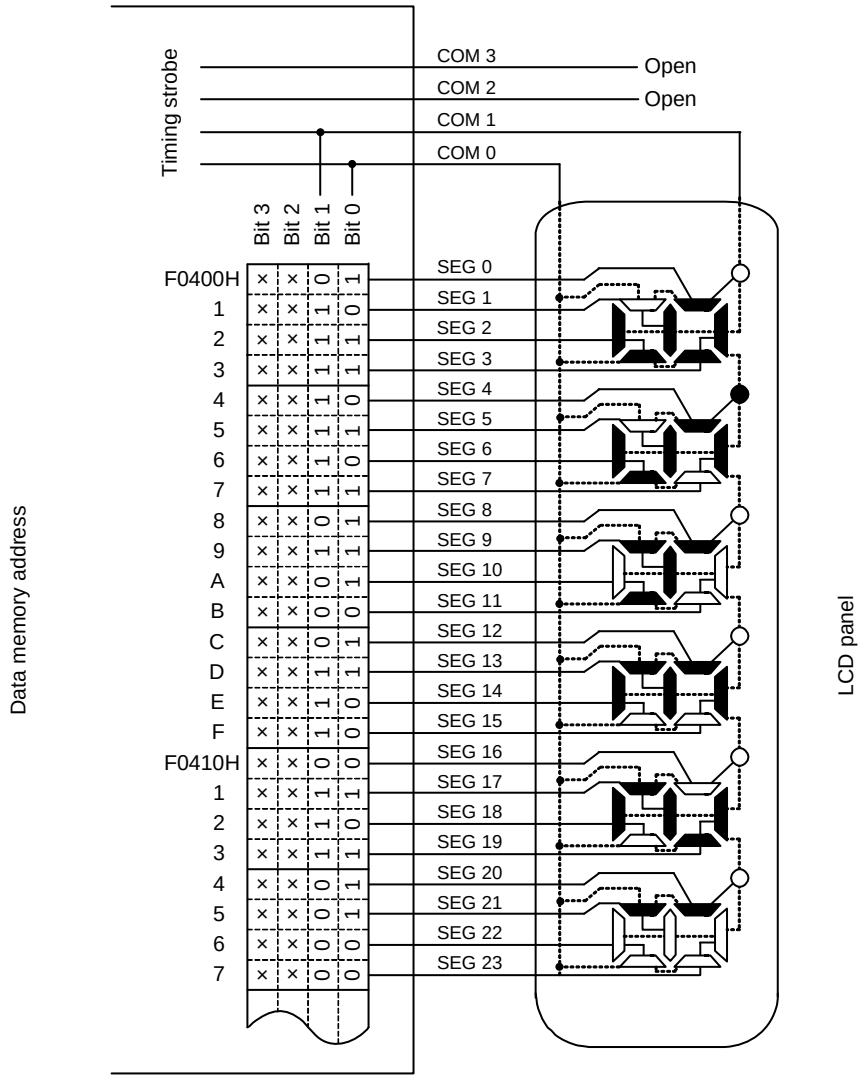
Figure 18 - 35 shows examples of LCD drive waveforms between the SEG15 signal and each common signal. When the select voltage is applied to SEG15 at the timing of COM1, an alternate rectangle waveform, +V_{LCD}/-V_{LCD}, is generated to turn on the corresponding LCD segment.

Figure 18 - 33 Two-Time-Slice LCD Display Pattern and Electrode Connections



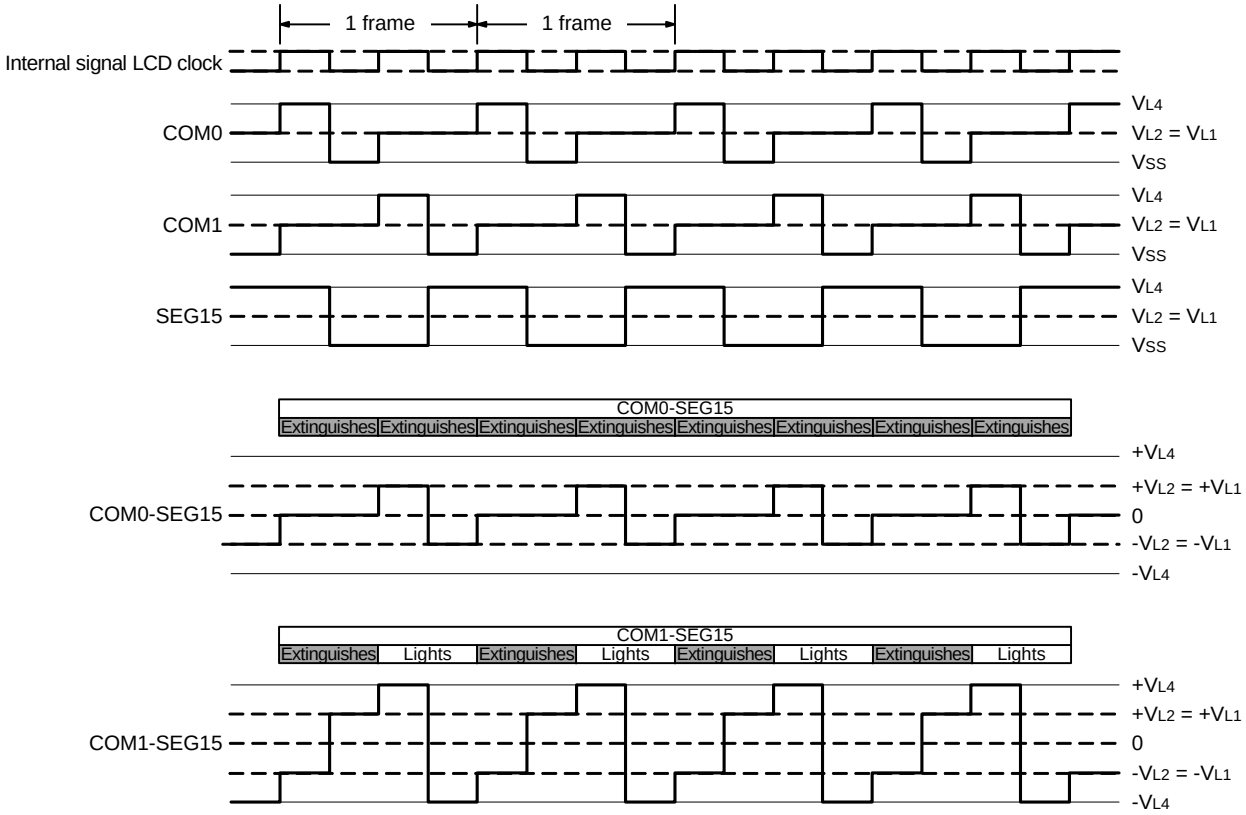
Remark 80/85-pin products: n = 0 to 12

Figure 18 - 34 Example of Connecting Two-Time-Slice LCD Panel



×: Can always be used to store any data because the two-time-slice mode is being used.

Figure 18 - 35 Two-Time-Slice LCD Drive Waveform Examples Between SEG15 and Each Common Signals (1/2 Bias Method)



18.10.3 Three-time-slice display example

Figure 18 - 37 shows how the 8-digit LCD panel having the display pattern shown in Figure 18 - 36 is connected to the segment signals (SEG0 to SEG23) and the common signals (COM0 to COM2). This example displays data “123456.78” in the LCD panel. The contents of the display data register (addresses F0400H to F0417H) correspond to this display.

The following description focuses on numeral “6.” (6.) displayed in the third digit. To display “6.” in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG6 to SEG8 pins according to Table 18 - 21 at the timing of the common signals COM0 to COM2; see **Figure 18 - 36** for the relationship between the segment signals and LCD segments.

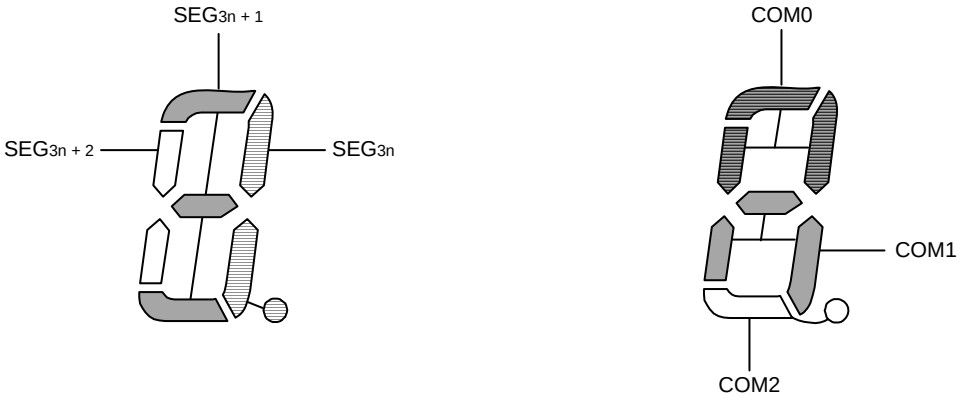
Table 18 - 21 Select and Deselect Voltages (COM0 to COM2)

Segment / Common	SEG6	SEG7	SEG8
COM0	Deselect	Select	Select
COM1	Select	Select	Select
COM2	Select	Select	—

According to Table 18 - 21, it is determined that the display data register location (F0406H) that corresponds to SEG6 must contain x110.

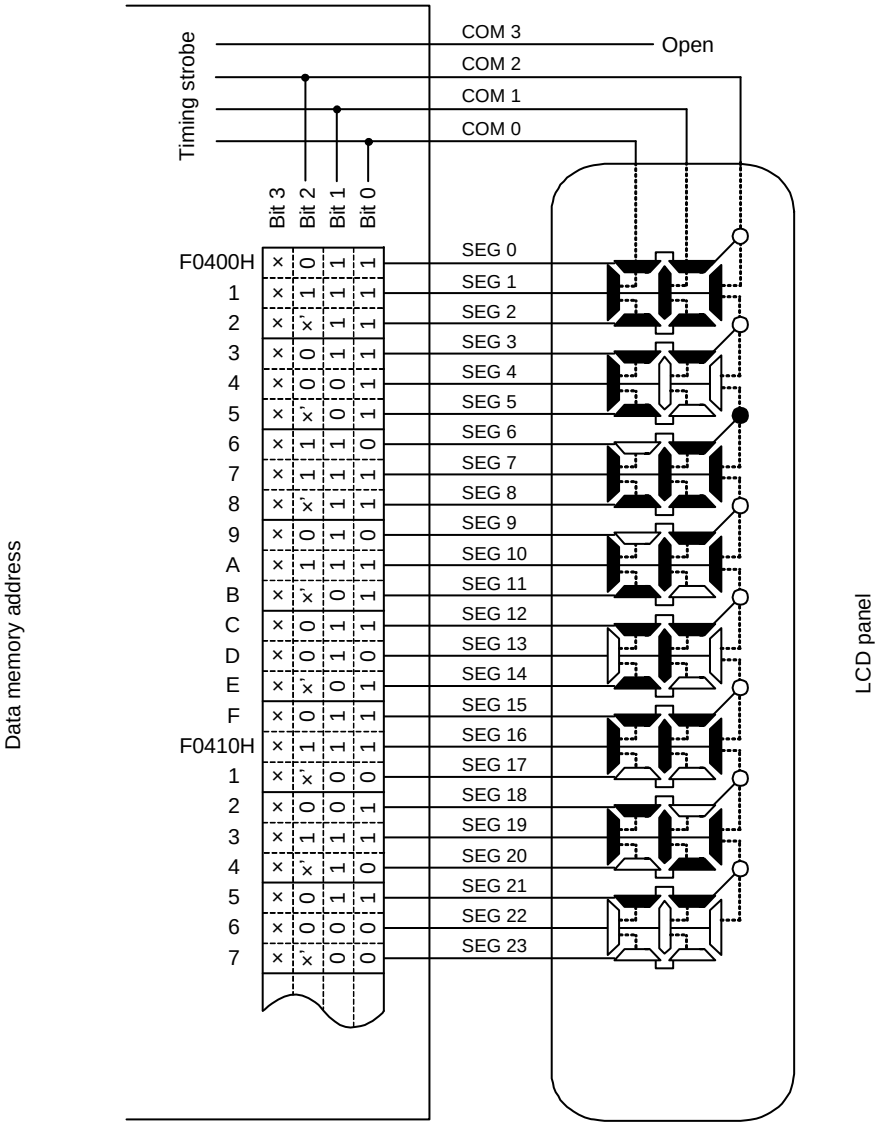
Figures 18 - 38 and 18 - 39 show examples of LCD drive waveforms between the SEG6 signal and each common signal in the 1/2 and 1/3 bias methods, respectively. When the select voltage is applied to SEG6 at the timing of COM1 or COM2, an alternate rectangle waveform, +V_{LCD}/-V_{LCD}, is generated to turn on the corresponding LCD segment.

Figure 18 - 36 Three-Time-Slice LCD Display Pattern and Electrode Connections



Remark 80/85-pin products: n = 0 to 16

Figure 18 - 37 Example of Connecting Three-Time-Slice LCD Panel



×': Can be used to store any data because there is no corresponding segment in the LCD panel.

×: Can always be used to store any data because the three-time-slice mode is being used.

Figure 18 - 38 Three-Time-Slice LCD Drive Waveform Examples Between SEG6 and Each Common Signals (1/2 Bias Method)

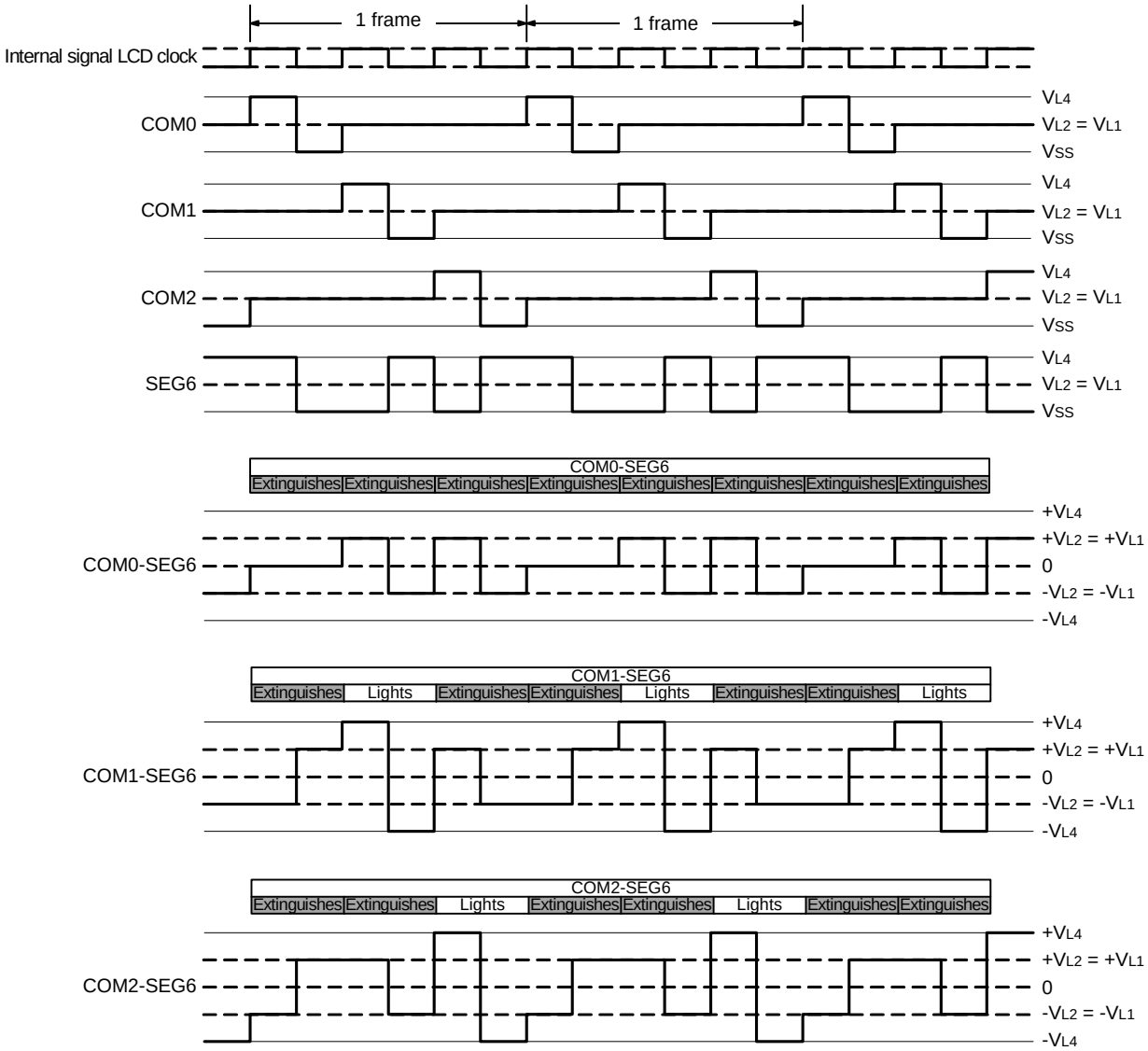
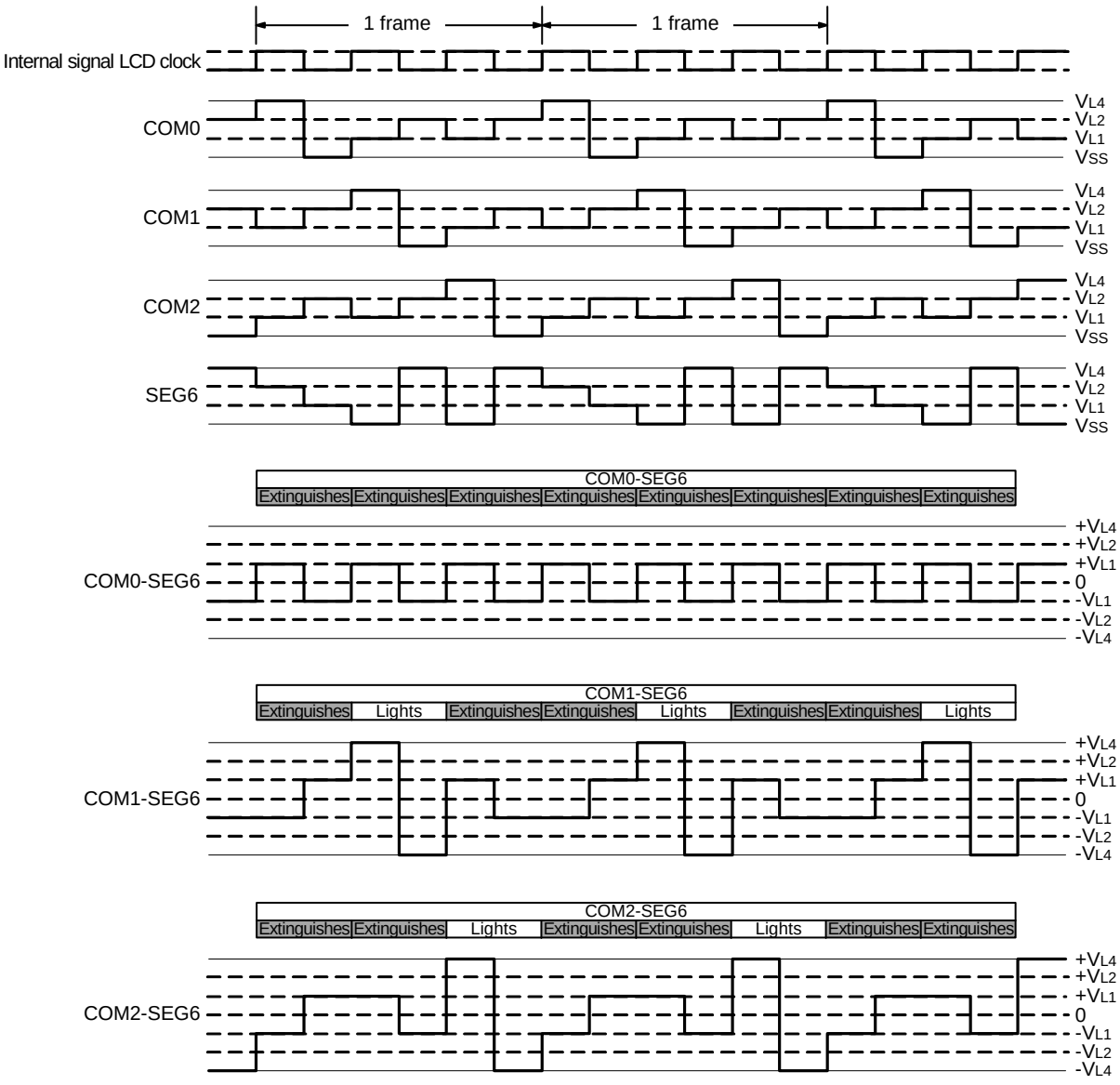


Figure 18 - 39 Three-Time-Slice LCD Drive Waveform Examples Between SEG6 and Each Common Signals (1/3 Bias Method)



18.10.4 Four-time-slice display example

Figure 18 - 41 shows how the 12-digit LCD panel having the display pattern shown in Figure 18 - 40 is connected to the segment signals (SEG0 to SEG23) and the common signals (COM0 to COM3). This example displays data “123456.789012” in the LCD panel. The contents of the display data register (addresses F0400H to F0417H) correspond to this display.

The following description focuses on numeral “6.” (6.) displayed in the seventh digit. To display “6.” in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG12 and SEG13 pins according to Table 18 - 22 at the timing of the common signals COM0 to COM3; see **Figure 18 - 40** for the relationship between the segment signals and LCD segments.

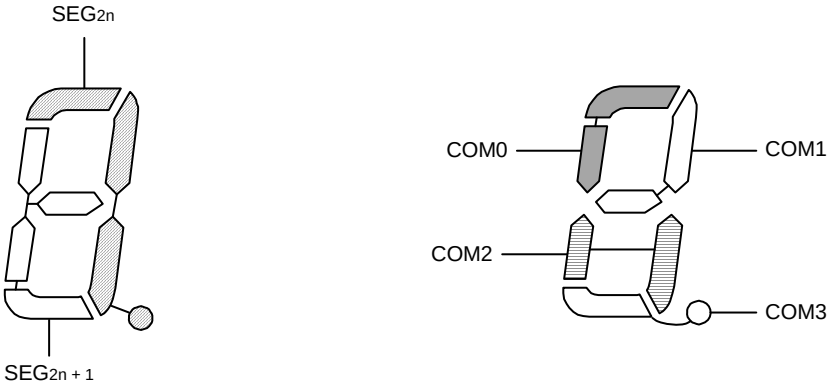
Table 18 - 22 Select and Deselect Voltages (COM0 to COM3)

Segment / Common	SEG12	SEG13
COM0	Select	Select
COM1	Deselect	Select
COM2	Select	Select
COM3	Select	Select

According to Table 18 - 22, it is determined that the display data register location (F040CH) that corresponds to SEG12 must contain 1101.

Figures 18 - 42 and 18 - 43 show examples of LCD drive waveforms between the SEG12 signal and each common signal. When the select voltage is applied to SEG12 at the timing of COM0, an alternate rectangle waveform, $+V_{LCD}/-V_{LCD}$, is generated to turn on the corresponding LCD segment.

Figure 18 - 40 Four-Time-Slice LCD Display Pattern and Electrode Connections



Remark 80/85-pin products: n = 0 to 25

Figure 18 - 41 Example of Connecting Four-Time-Slice LCD Panel

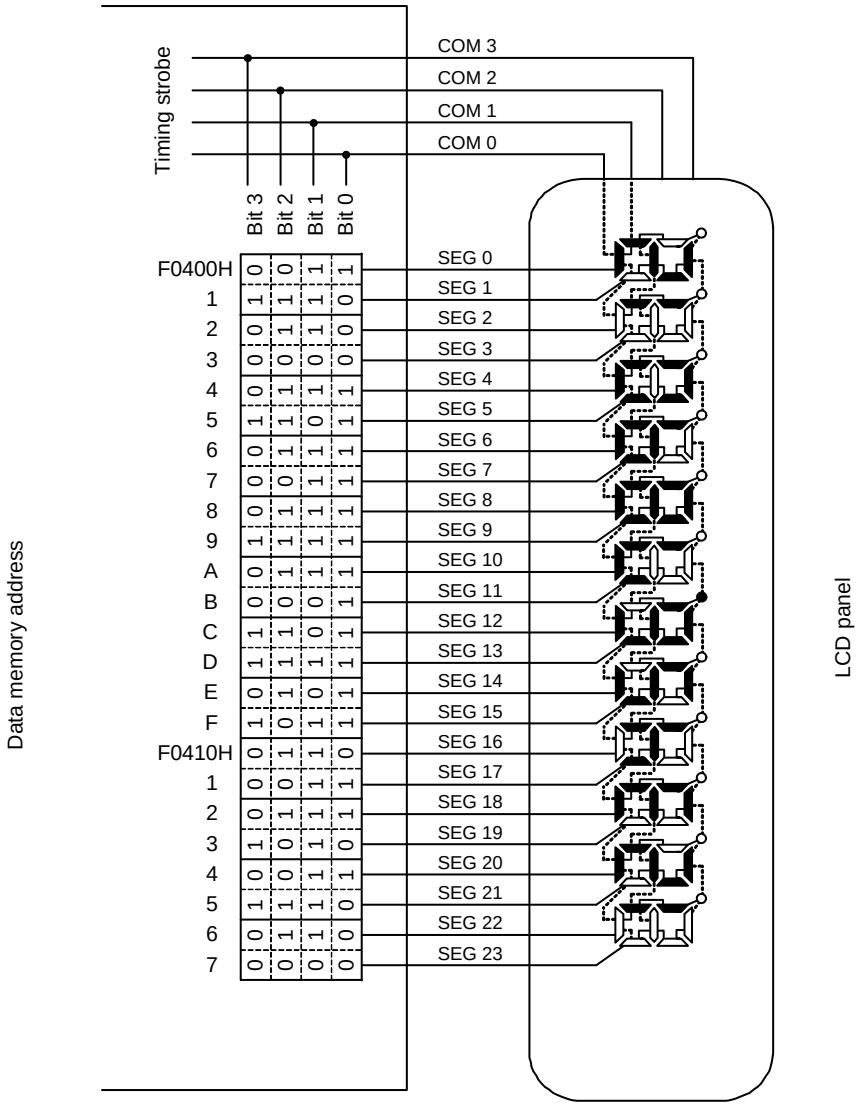


Figure 18 - 42 Four-Time-Slice LCD Drive Waveform Examples Between SEG12 and Each Common Signals (1/3 Bias Method) (1/2)

(a) Waveform A

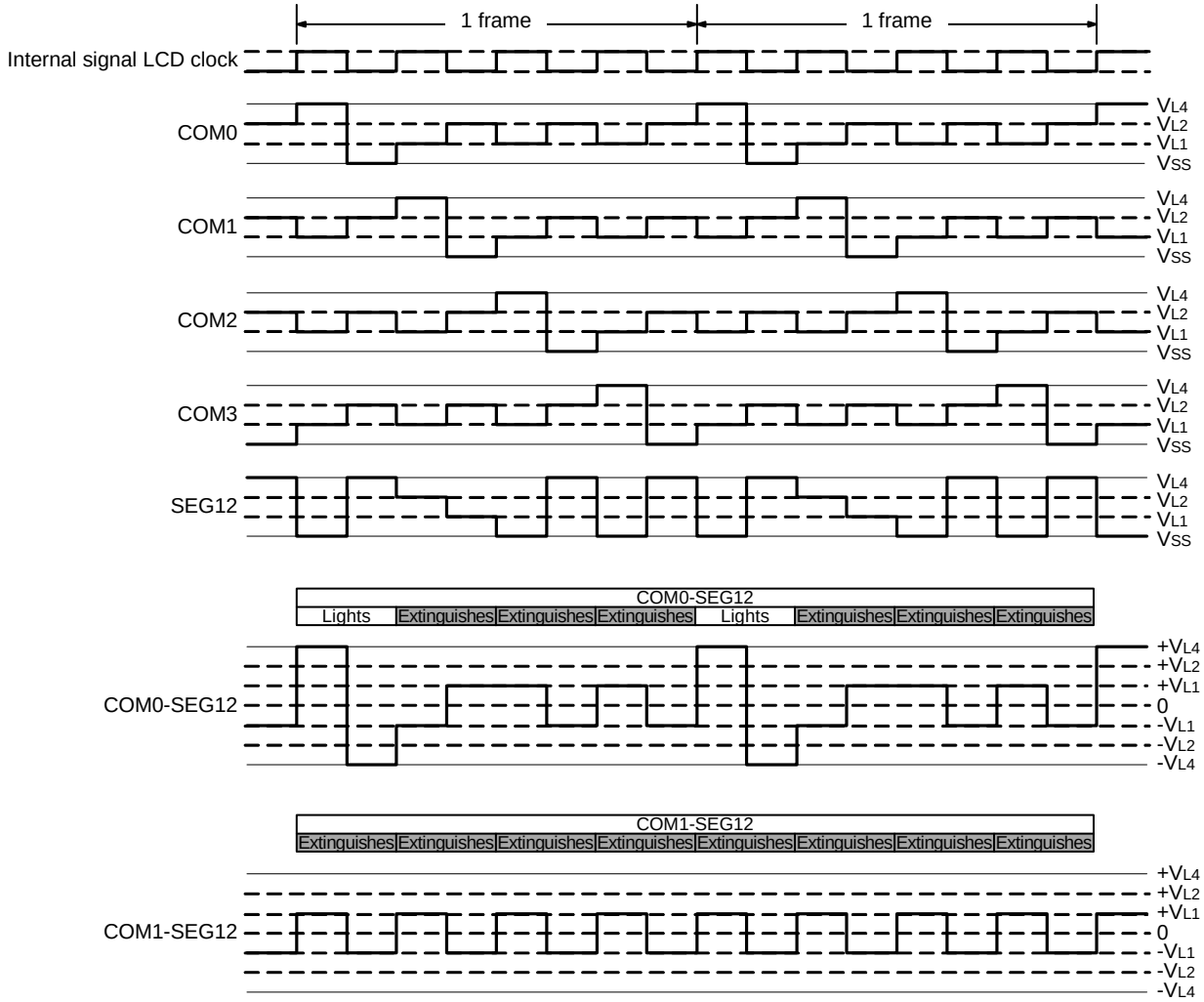
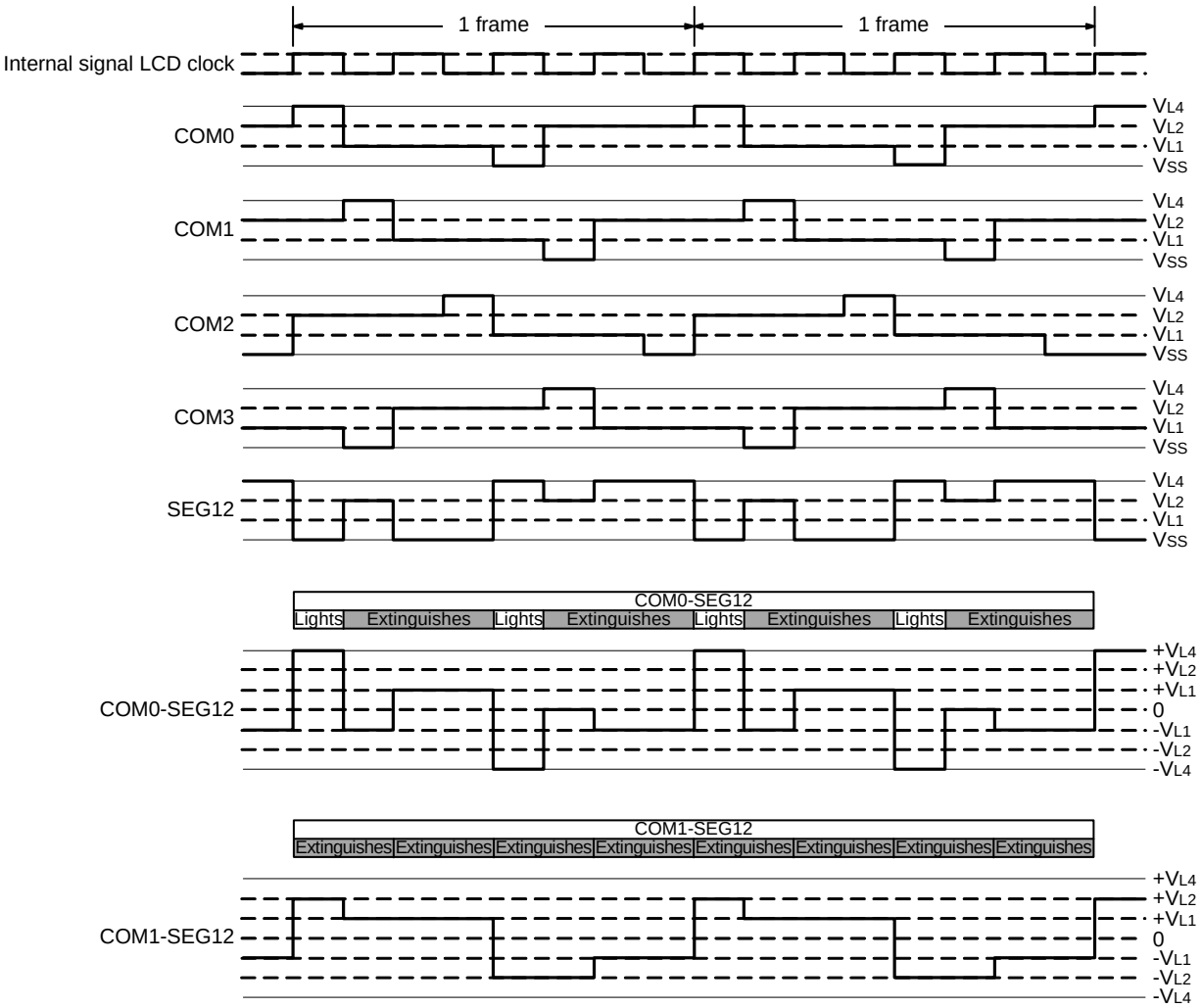


Figure 18 - 43 Four-Time-Slice LCD Drive Waveform Examples Between SEG12 and Each Common Signals (1/3 Bias Method) (2/2)

(b) Waveform B



18.10.5 Eight-time-slice display example

Figure 18 - 45 shows how the 15 × 8 dot LCD panel having the display pattern shown in Figure 18 - 44 is connected to the segment signals (SEG4 to SEG18) and the common signals (COM0 to COM7). This example displays data “123” in the LCD panel. The contents of the display data register (addresses F0404H to F0412H) correspond to this display.

The following description focuses on numeral “3” (3) displayed in the first digit. To display “3” in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG4 to SEG8 pins according to Table 18 - 23 at the timing of the common signals COM0 to COM7; see **Figure 18 - 44** for the relationship between the segment signals and LCD segments.

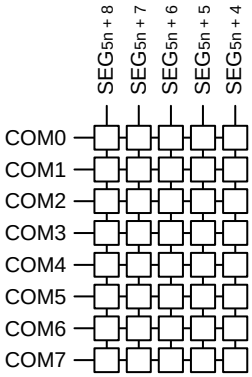
Table 18 - 23 Select and Deselect Voltages (COM0 to COM7)

Segment / Common	SEG4	SEG5	SEG6	SEG7	SEG8
COM0	Select	Select	Select	Select	Select
COM1	Deselect	Select	Deselect	Deselect	Deselect
COM2	Deselect	Deselect	Select	Deselect	Deselect
COM3	Deselect	Select	Deselect	Deselect	Deselect
COM4	Select	Deselect	Deselect	Deselect	Deselect
COM5	Select	Deselect	Deselect	Deselect	Select
COM6	Deselect	Select	Select	Select	Deselect
COM7	Deselect	Deselect	Deselect	Deselect	Deselect

According to Table 18 - 23, it is determined that the display data register location (F0404H) that corresponds to SEG4 must contain 00110001.

Figures 18 - 46 and 18 - 47 show examples of LCD drive waveforms between the SEG4 signal and each common signal. When the select voltage is applied to SEG4 at the timing of COM0, a waveform is generated to turn on the corresponding LCD segment.

Figure 18 - 44 Eight-Time-Slice LCD Display Pattern and Electrode Connections



Remark 80/85-pin products: n = 0 to 8

Figure 18 - 45 Example of Connecting Eight-Time-Slice LCD Panel

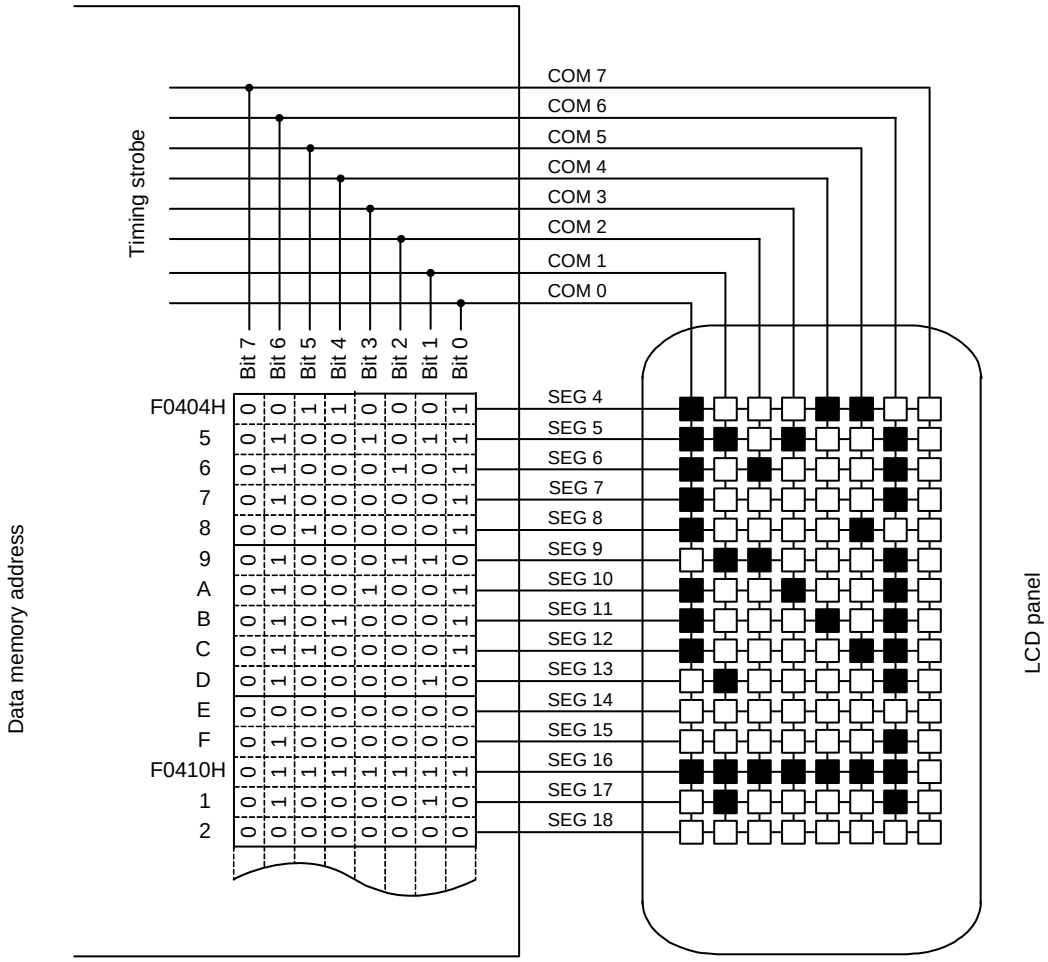


Figure 18 - 46 Eight-Time-Slice LCD Drive Waveform Examples Between SEG4 and Each Common Signals (1/4 Bias Method) (1/2)

(a) Waveform A

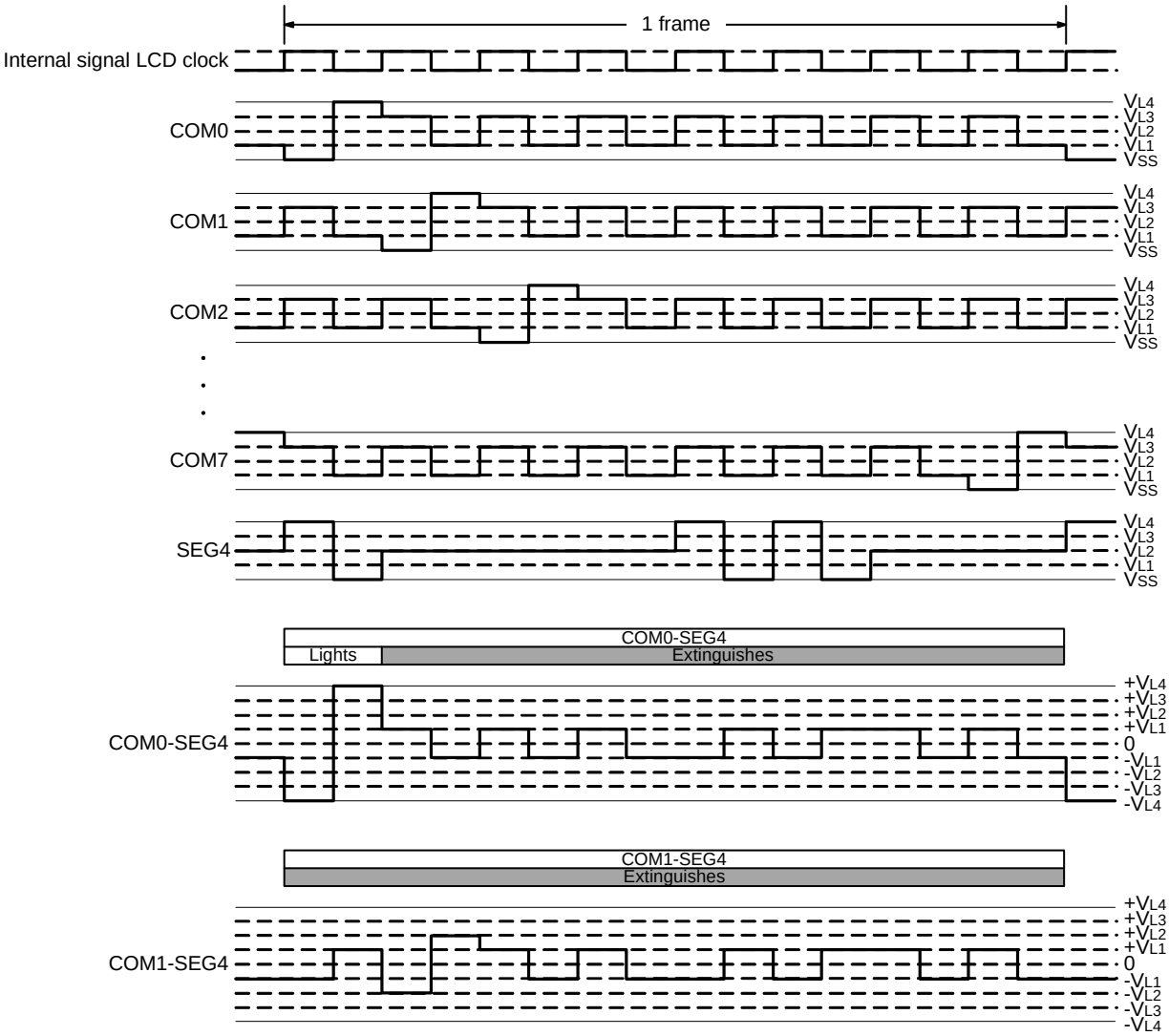
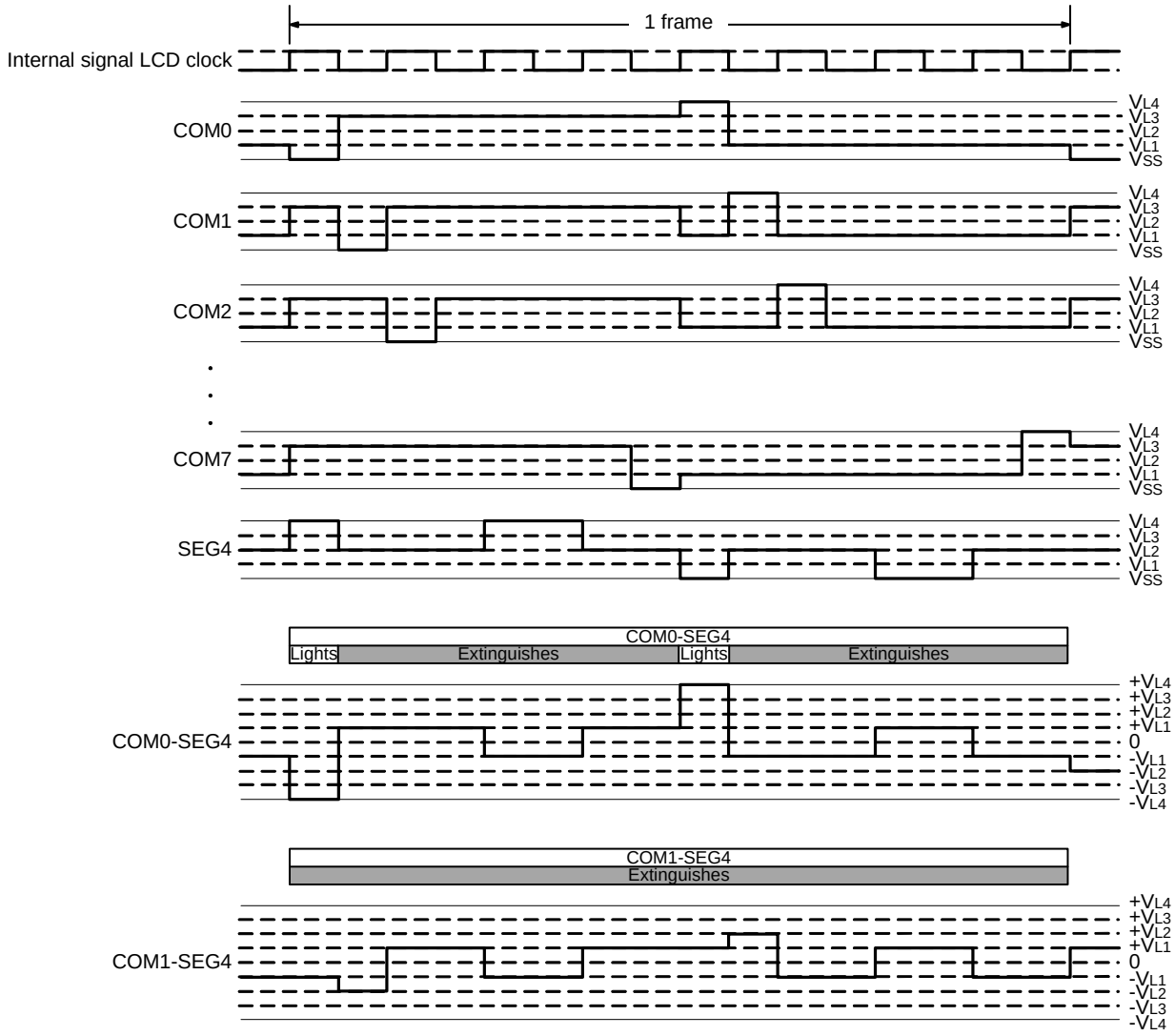


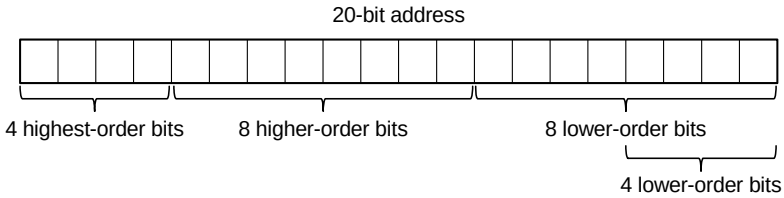
Figure 18 - 47 Eight-Time-Slice LCD Drive Waveform Examples Between SEG4 and Each Common Signals (1/4 Bias Method) (2/2)

(b) Waveform B



CHAPTER 19 DATA TRANSFER CONTROLLER (DTC)

The term “8 higher-order bits of the address” in this chapter indicates bits 15 to 8 of 20-bit address as shown below.



Unless otherwise specified, the 4 highest-order address bits all become 1 (values are of the form FxxxxH).

19.1 Functions of DTC

The data transfer controller (DTC) is a function that transfers data between memories without using the CPU. The DTC is activated by a peripheral function interrupt to perform data transfers. The DTC and CPU use the same bus, and the DTC takes priority over the CPU in using the bus.

Table 19 - 1 lists the DTC Specifications.

Table 19 - 1 DTC Specifications

Item		Specification
Activation sources		30 sources (80/85-pin products without USB)/31 sources (100-pin products without USB) 32 sources (80/85-pin products with USB)/33 sources (100-pin products with USB)
Allocatable control data		24 sets
Address space which can be transferred	Address space	64 Kbytes (F0000H to FFFFFH), excluding general-purpose registers
	Sources	1st SFR area, RAM area (excluding general-purpose registers), mirror area Note, data flash memory area Note, 2nd SFR area
	Destinations	1st SFR area, RAM area (excluding general-purpose registers), 2nd SFR area
Maximum number of transfers	Normal mode	256 times
	Repeat mode	255 times
Maximum size of block to be transferred	Normal mode (8-bit transfer)	256 bytes
	Normal mode (16-bit transfer)	512 bytes
	Repeat mode	255 bytes
Unit of transfers		8 bits/16 bits
Transfer mode	Normal mode	Transfers end on completion of the transfer causing the DTCCTj register value to change from 1 to 0.
	Repeat mode	On completion of the transfer causing the DTCCTj register value to change from 1 to 0, the repeat area address is initialized and the DTRLDj register value is reloaded to the DTCCTj register to continue transfers.
Address control	Normal mode	Fixed or incremented
	Repeat mode	Addresses of the area not selected as the repeat area are fixed or incremented.
Priority of activation sources		Refer to **Table 19 - 4 DTC Activation Sources and Vector Addresses**.
Interrupt request	Normal mode	When the data transfer causing the DTCCTj register value to change from 1 to 0 is performed, the activation source interrupt request is generated for the CPU, and interrupt handling is performed on completion of the data transfer.
	Repeat mode	When the data transfer causing the DTCCTj register value to change from 1 to 0 is performed while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled), the activation source interrupt request is generated for the CPU, and interrupt handling is performed on completion of the transfer.
Transfer start		When bits DTCENi0 to DTCENi7 in the DTCENi registers are 1 (activation enabled), data transfer is started each time the corresponding DTC activation sources are generated.

Table 19 - 1 DTC Specifications

Item		Specification
Transfer stop	Normal mode	• When bits DTCENi0 to DTCENi7 are set to 0 (activation disabled). • When the data transfer causing the DTCCTj register value to change from 1 to 0 is completed.
	Repeat mode	• When bits DTCENi0 to DTCENi7 are set to 0 (activation disabled). • When the data transfer causing the DTCCTj register value to change from 1 to 0 is completed while the RPTINT bit is 1 (interrupt generation enabled).
Operation in standby mode	HALT state	DTC operates
	SNOOZE state	DTC operates
	STOP state	DTC stops

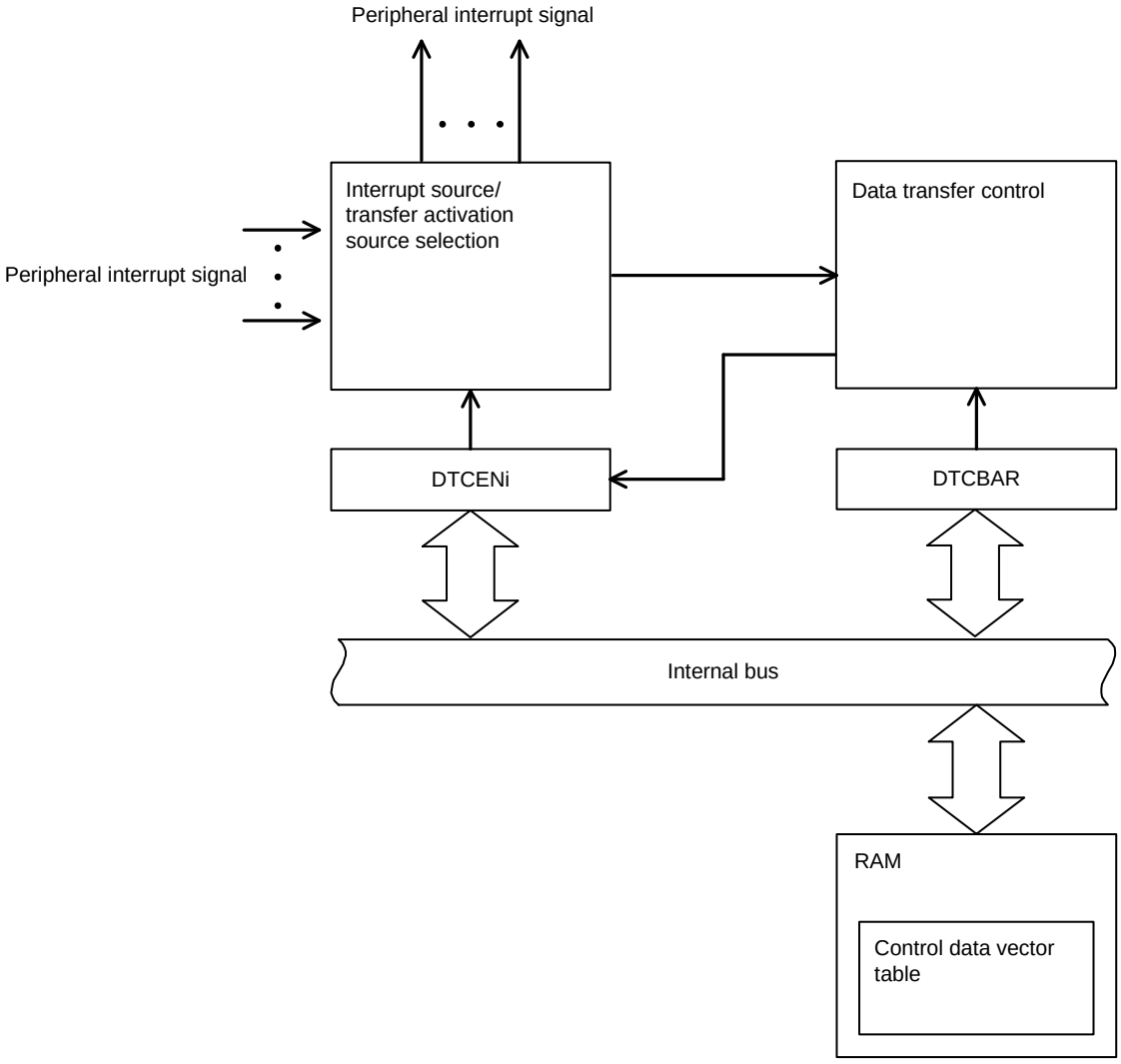
Note In the HALT mode or SNOOZE mode, these areas cannot be set as the sources for DTC transfer since the flash memory is stopped.

Remark i = 0 to 4, j = 0 to 23

19.2 Configuration of DTC

Figure 19 - 1 shows the DTC Block Diagram.

Figure 19 - 1 DTC Block Diagram



19.3 Registers Controlling DTC

Table 19 - 2 lists the Registers Controlling DTC.

Table 19 - 2 Registers Controlling DTC

Register Name	Symbol
Peripheral enable register 1	PER1
DTC activation enable register 0	DTCEN0
DTC activation enable register 1	DTCEN1
DTC activation enable register 2	DTCEN2
DTC activation enable register 3	DTCEN3
DTC activation enable register 4	DTCEN4
DTC base address register	DTCBAR

Table 19 - 3 lists DTC Control Data.

DTC control data is allocated in the DTC control data area in RAM.

The DTCBAR register is used to set the 256-byte area, including the DTC control data area and the DTC vector table area where the start address for control data is stored.

Table 19 - 3 DTC Control Data

Register Name	Symbol
DTC Control Register j	DTCCRj
DTC Block Size Register j	DTBLSj
DTC Transfer Count Register j	DTCCTj
DTC Transfer Count Reload Register j	DTRLDj
DTC Source Address Register j	DTSARj
DTC Destination Address Register j	DTDARj

Remark j = 0 to 23

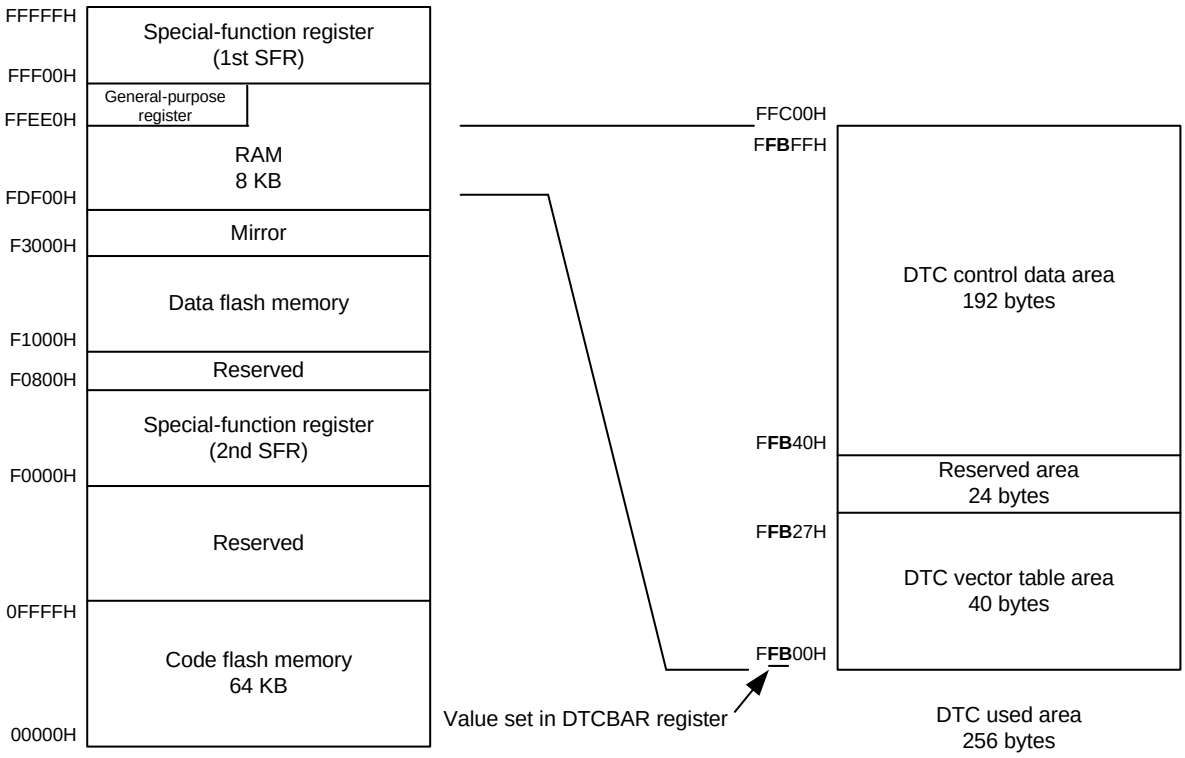
19.3.1 Allocation of DTC Control Data Area and DTC Vector Table Area

The DTCBAR register is used to set the 256-byte area where DTC control data and the vector table within the RAM area.

Figure 19 - 2 shows a Memory Map Example when DTCBAR Register is Set to FBH.

In the 192-byte DTC control data area, the space not used by the DTC can be used as RAM.

Figure 19 - 2 Memory Map Example when DTCBAR Register is Set to FBH



The areas where the DTC control data and vector table can be allocated differ depending on the product.

Caution 1. It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space as the DTC control data area or DTC vector table area.

Caution 2. Make sure the stack area, the DTC control data area, and the DTC vector table area do not overlap.

Caution 3. The internal RAM area in the following products cannot be used as the DTC control data area or DTC vector table area when using the self-programming and data-flash functions.
R5F110xH, R5F111xH (x = M, N, P): FBF00H to FC309H
R5F110xJ, R5F111xJ (x = M, N, P): FBF00H to FC309H

Caution 4. The internal RAM area in the following products cannot be used as the DTC control data area or DTC vector table area when using the on-chip debugging trace function.
R5F110xH, R5F111xH (x = M, N, P): FC300H to FC6FFH
R5F110xJ, R5F111xJ (x = M, N, P): FC300H to FC6FFH

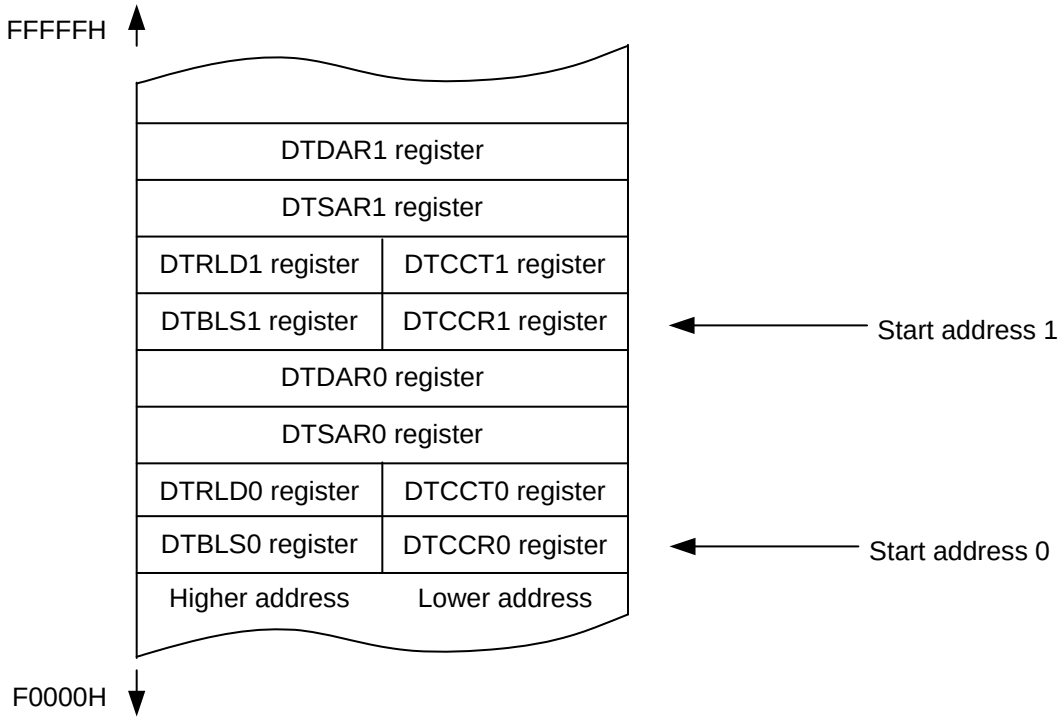
19.3.2 Control Data Allocation

Control data is allocated beginning with each start address in the order: Registers DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, and DTDARj (j = 0 to 23).
The higher 8 bits for start addresses 0 to 23 are set by the DTCBAR register, and the lower 8 bits are separately set according to the vector table assigned to each activation source.
Figure 19 - 3 shows Control Data Allocation.

Note 1. Change the data in registers DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, and DTDARj when the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 4) in the DTCENi register is set to 0 (DTC activation disabled).

Note 2. Do not access DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, or DTDARj using a DTC transfer.

Figure 19 - 3 Control Data Allocation



19.3.3 Vector Table

When the DTC is activated, one control data is selected according to the data read from the vector table which has been assigned to each activation source, and the selected control data is read from the DTC control data area.
Table 19 - 4 lists the DTC Activation Sources and Vector Addresses. A one byte of the DTC vector table is assigned to each activation source, and data from 40H to F8H is stored in each area to select one of the 24 control data sets. The higher 8 bits for the DTC vector address are set by the DTCBAR register, and 00H to 21H are allocated to the lower 8 bits corresponding to the DTC activation source.

Note Change the start address of the DTC control data area to be set in the vector table when the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 4) in the DTCENi register is set to 0 (activation disabled).

Figure 19 - 4 Start Address of Control Data and Vector Table

Example: When DTCBAR is set to FBH.

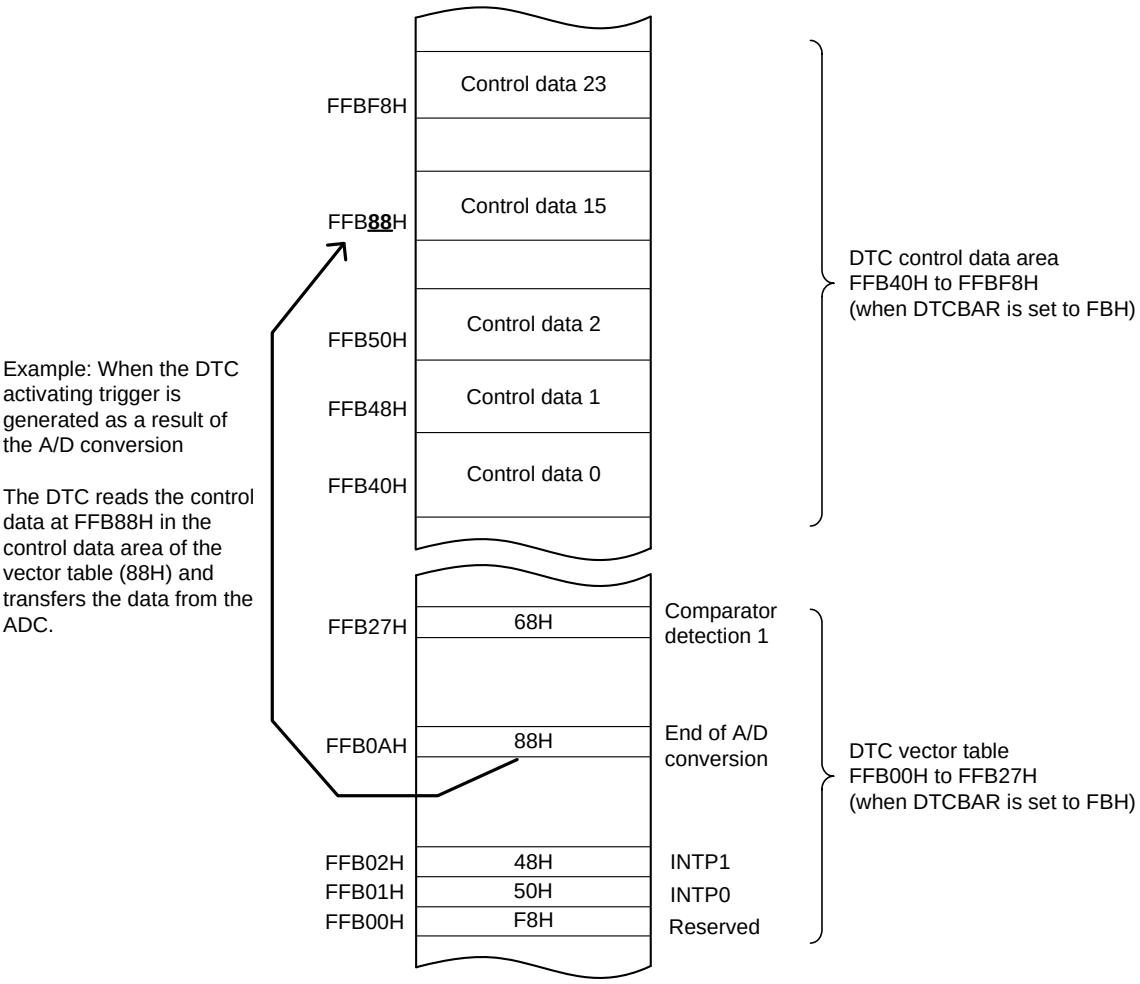


Table 19 - 4 DTC Activation Sources and Vector Addresses

Interrupt Request Source	Source No.	DTC Vector Address	Priority
Reserved	0	Address set in DTCBAR register +00H	Highest
INTP0	1	Address set in DTCBAR register +01H	
INTP1	2	Address set in DTCBAR register +02H	
INTP2	3	Address set in DTCBAR register +03H	
INTP3	4	Address set in DTCBAR register +04H	
INTP4	5	Address set in DTCBAR register +05H	
INTP5	6	Address set in DTCBAR register +06H	
INTP6	7	Address set in DTCBAR register +07H	
INTP7	8	Address set in DTCBAR register +08H	
Key input	9	Address set in DTCBAR register +09H	
A/D conversion end	10	Address set in DTCBAR register +0AH	
UART0 reception transfer end	11	Address set in DTCBAR register +0BH	
UART0 transmission transfer end/CSI00 transfer end or buffer empty/IIC00 transfer end	12	Address set in DTCBAR register +0CH	
UART1 reception transfer end	13	Address set in DTCBAR register +0DH	
UART1 transmission transfer end/CSI10 transfer end or buffer empty/IIC10 transfer end	14	Address set in DTCBAR register +0EH	
UART2 reception transfer end	15	Address set in DTCBAR register +0FH	
UART2 transmission transfer end/CSI20 transfer end or buffer empty/IIC20 transfer end	16	Address set in DTCBAR register +10H	
UART3 reception transfer end	17	Address set in DTCBAR register +11H	
UART3 transmission transfer end/CSI30 transfer end or buffer empty/IIC30 transfer end	18	Address set in DTCBAR register +12H	
End of channel 0 of timer array unit 0 count or capture	19	Address set in DTCBAR register +13H	
End of channel 1 of timer array unit 0 count or capture	20	Address set in DTCBAR register +14H	
End of channel 2 of timer array unit 0 count or capture	21	Address set in DTCBAR register +15H	
End of channel 3 of timer array unit 0 count or capture	22	Address set in DTCBAR register +16H	
End of channel 4 of timer array unit 0 count or capture	23	Address set in DTCBAR register +17H	
End of channel 5 of timer array unit 0 count or capture	24	Address set in DTCBAR register +18H	
End of channel 6 of timer array unit 0 count or capture	25	Address set in DTCBAR register +19H	
End of channel 7 of timer array unit 0 count or capture	26	Address set in DTCBAR register +1AH	
End of TMKB2_0 count	27	Address set in DTCBAR register +1BH	
End of TMKB2_1 count	28	Address set in DTCBAR register +1CH	
End of TMKB2_2 count	29	Address set in DTCBAR register +1DH	
USB D0FIFO transfer end using DTC Note 1	30	Address set in DTCBAR register +1EH	
USB D1FIFO transfer end using DTC Note 1	31	Address set in DTCBAR register +1FH	
Comparator detection 0	32	Address set in DTCBAR register +20H	
Comparator detection 1 Note 2	33	Address set in DTCBAR register +21H	Lowest

Note 1. Products with USB only.

Note 2. 100-pin products only.

19.3.4 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.
When using the DTC, be sure to set bit 3 (DTCEN) to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 19 - 5 Format of Peripheral enable register 1 (PER1)

Address: F007AH After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	2	1	<0>
PER1	TMKAEN	0	CMPEN	TKB20EN	DTCEN	0	0	DACEN

DTCEN	Control of DTC input clock supply
0	Stops input clock supply. • DTC cannot run.
1	Enables input clock supply. • DTC can run.

Caution Be sure to clear bits 6, 2, and 1 to 0.

19.3.5 DTC control register j (DTCCRj) (j = 0 to 23)

The DTCCRj register is used to control the DTC operating mode.

Figure 19 - 6 Format of DTC control register j (DTCCRj)

Address: Refer to **19.3.2 Control Data Allocation**. After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
DTCCRj	0	SZ	RPTINT	CHNE	DAMOD	SAMOD	RPTSEL	MODE

SZ	Data size selection
0	8 bits
1	16 bits

RPTINT	Enabling/disabling repeat mode interrupts
0	Interrupt generation disabled
1	Interrupt generation enabled
The setting of the RPTINT bit is invalid when the MODE bit is 0 (normal mode).	

CHNE	Enabling/disabling chain transfers
0	Chain transfers disabled
1	Chain transfers enabled
Set the CHNE bit in the DTCCR23 register to 0 (chain transfers disabled).	

DAMOD	Transfer destination address control
0	Fixed
1	Incremented
The setting of the DAMOD bit is invalid when the MODE bit is 1 (repeat mode) and the RPTSEL bit is 0 (transfer destination is the repeat area).	

SAMOD	Transfer source address control
0	Fixed
1	Incremented
The setting of the SAMOD bit is invalid when the MODE bit is 1 (repeat mode) and the RPTSEL bit is 1 (transfer source is the repeat area).	

RPTSEL	Repeat area selection
0	Transfer destination is the repeat area
1	Transfer source is the repeat area
The setting of the RPTSEL bit is invalid when the MODE bit is 0 (normal mode).	

MODE	Transfer mode selection
0	Normal mode
1	Repeat mode

Caution Do not access the DTCCRj register using a DTC transfer.

19.3.6 DTC block size register j (DTBLSj) (j = 0 to 23)

This register is used to set the block size of the data to be transferred by one activation.

Figure 19 - 7 Format of DTC block size register j (DTBLSj)

Address: Refer to **19.3.2 Control Data Allocation**. After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
DTBLSj	DTBLSj7	DTBLSj6	DTBLSj5	DTBLSj4	DTBLSj3	DTBLSj2	DTBLSj1	DTBLSj0

DTBLSj	Transfer Block Size	
	8-Bit Transfer	16-Bit Transfer
00H	256 bytes	512 bytes
01H	1 byte	2 bytes
02H	2 bytes	4 bytes
03H	3 bytes	6 bytes
• • •	• • •	• • •
FDH	253 bytes	506 bytes
FEH	254 bytes	508 bytes
FFH	255 bytes	510 bytes

Caution Do not access the DTBLSj register using a DTC transfer.

19.3.7 DTC transfer count register j (DTCCTj) (j = 0 to 23)

This register is used to set the number of DTC data transfers. The value is decremented by 1 each time DTC transfer is activated once.

Figure 19 - 8 Format of DTC transfer count register j (DTCCTj)

Address: Refer to **19.3.2 Control Data Allocation**. After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
DTCCTj	DTCCTj7	DTCCTj6	DTCCTj5	DTCCTj4	DTCCTj3	DTCCTj2	DTCCTj1	DTCCTj0

DTCCTj	Number of Transfers
00H	256 times
01H	Once
02H	2 times
03H	3 times
• • •	• • •
FDH	253 times
FEH	254 times
FFH	255 times

Caution Do not access the DTCCTj register using a DTC transfer.

19.3.8 DTC transfer count reload register j (DTRLDj) (j = 0 to 23)

This register is used to set the initial value of the transfer count register in repeat mode. Since the value of this register is reloaded to the DTCCT register in repeat mode, set the same value as the initial value of the DTCCT register.

Figure 19 - 9 Format of DTC transfer count reload register j (DTRLDj)

Address: Refer to **19.3.2 Control Data Allocation**. After reset: Undefined R/W

	7	6	5	4	3	2	1	0
DTRLDj	DTRLDj7	DTRLDj6	DTRLDj5	DTRLDj4	DTRLDj3	DTRLDj2	DTRLDj1	DTRLDj0

Caution Do not access the DTRLDj register using a DTC transfer.

19.3.9 DTC source address register j (DTSARj) (j = 0 to 23)

This register is used to specify the transfer source address for data transfer.

When the SZ bit in the DTCCRj register is set to 1 (16-bit transfer), the lowest bit is ignored and the address is handled as an even address.

Figure 19 - 10 Format of DTC source address register j (DTSARj)

Address: Refer to **19.3.2 Control Data Allocation**. After reset: Undefined R/W

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTSARj	DTS ARj15	DTS ARj14	DTS ARj13	DTS ARj12	DTS ARj11	DTS ARj10	DTS ARj9	DTS ARj8	DTS ARj7	DTS ARj6	DTS ARj5	DTS ARj4	DTS ARj3	DTS ARj2	DTS ARj1	DTS ARj0

Caution 1. Do not set the general-purpose register (FFEE0H to FFEFFH) space to the transfer source address.
Caution 2. Do not access the DTSARj register using a DTC transfer.

19.3.10 DTC destination address register j (DTDARj) (j = 0 to 23)

This register is used to specify the transfer destination address for data transfer.

When the SZ bit in the DTCCRj register is set to 1 (16-bit transfer), the lowest bit is ignored and the address is handled as an even address.

Figure 19 - 11 Format of DTC destination address register j (DTDARj)

Address: Refer to **19.3.2 Control Data Allocation**. After reset: Undefined R/W

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTDARj	DTD ARj15	DTD ARj14	DTD ARj13	DTD ARj12	DTD ARj11	DTD ARj10	DTD ARj9	DTD ARj8	DTD ARj7	DTD ARj6	DTD ARj5	DTD ARj4	DTD ARj3	DTD ARj2	DTD ARj1	DTD ARj0

Caution 1. Do not set the general-purpose register (FFEE0H to FFEFFH) space to the transfer source address.
Caution 2. Do not access the DTDARj register using a DTC transfer.

19.3.11 DTC activation enable register i (DTCENi) (i = 0 to 4)

This is an 8-bit register which enables or disables DTC activation by interrupt sources. Table 19 - 5 lists the Correspondences between Interrupt Sources and Bits DTCENi0 to DTCENi7.

The DTCENi register can be set by an 8-bit memory manipulation instruction and a 1-bit memory manipulation instruction.

Note 1. Modify bits DTCENi0 to DTCENi7 if an activation source corresponding to the bit has not been generated.

Note 2. Do not access the DTCENi register using a DTC transfer.

Figure 19 - 12 Format of DTC activation enable register i (DTCENi) (i = 0 to 4)

Address: F02E8H (DTCEN0), F02E9H (DTCEN1), F02EAH (DTCEN2), F02EBH (DTCEN3), F02ECH (DTCEN4) | After reset: 00H | R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
DTCENi	DTCENi7	DTCENi6	DTCENi5	DTCENi4	DTCENi3	DTCENi2	DTCENi1	DTCENi0

DTCENi7	DTC activation enable i7
0	Activation disabled
1	Activation enabled
The DTCENi7 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi6	DTC activation enable i6
0	Activation disabled
1	Activation enabled
The DTCENi6 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi5	DTC activation enable i5
0	Activation disabled
1	Activation enabled
The DTCENi5 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi4	DTC activation enable i4
0	Activation disabled
1	Activation enabled
The DTCENi4 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi3	DTC activation enable i3
0	Activation disabled
1	Activation enabled
The DTCENi3 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi2	DTC activation enable i2
0	Activation disabled
1	Activation enabled
The DTCENi2 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi1	DTC activation enable i1
0	Activation disabled
1	Activation enabled
The DTCENi1 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi0	DTC activation enable i0
0	Activation disabled
1	Activation enabled
The DTCENi0 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

Table 19 - 5 Correspondences between Interrupt Sources and Bits DTCENi0 to DTCENi7

Register	DTCENi7 Bit	DTCENi6 Bit	DTCENi5 Bit	DTCENi4 Bit	DTCENi3 Bit	DTCENi2 Bit	DTCENi1 Bit	DTCENi0 Bit
DTCEN0	Reserved	INTP0	INTP1	INTP2	INTP3	INTP4	INTP5	INTP6
DTCEN1	INTP7	Key input	A/D conversion end	UART0 reception transfer end	UART0 transmission transfer end/CSI00 transfer end or buffer empty/IIC00 transfer end	UART1 reception transfer end	UART1 transmission transfer end/CSI10 transfer end or buffer empty/IIC10 transfer end	UART2 reception transfer end
DTCEN2	UART2 transmission transfer end/CSI20 transfer end or buffer empty/IIC20 transfer end	UART3 reception transfer end	UART3 transmission transfer end/CSI30 transfer end or buffer empty/IIC30 transfer end Note 1	End of channel 0 of timer array unit 0 count or capture	End of channel 1 of timer array unit 0 count or capture	End of channel 2 of timer array unit 0 count or capture	End of channel 3 of timer array unit 0 count or capture	End of channel 4 of timer array unit 0 count or capture
DTCEN3	End of channel 5 of timer array unit 0 count or capture	End of channel 6 of timer array unit 0 count or capture	End of channel 7 of timer array unit 0 count or capture	End of TMKB2_0 count	End of TMKB2_1 count	End of TMKB2_2 count	USB DTCD0FIFO Note 1	USB DTCD1FIFO Note 1
DTCEN4	Comparator detection 0	Comparator detection 1 Note 2	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Remark i = 0 to 4

Note 1. Products with USB only.

Note 2. 100-pin products only.

19.3.12 DTC base address register (DTCBAR)

This is an 8-bit register used to set the following addresses: the vector address where the start address of the DTC control data area is stored and the address of the DTC control data area. The value of the DTCBAR register is handled as the higher 8 bits to generate a 16-bit address.

Caution 1. Change the DTCBAR register value with all DTC activation sources set to activation disabled.
Caution 2. Do not rewrite the DTCBAR register more than once.
Caution 3. Do not access the DTCBAR register using a DTC transfer.
Caution 4. For the allocation of the DTC control data area and the DTC vector table area, refer to the notes on 19.3.1 Allocation of DTC Control Data Area and DTC Vector Table Area.

Figure 19 - 13 Format of DTC base address register (DTCBAR)

Address: F02E0H After reset: FDH R/W

Symbol	7	6	5	4	3	2	1	0
DTCBAR	DTCBAR7	DTCBAR6	DTCBAR5	DTCBAR4	DTCBAR3	DTCBAR2	DTCBAR1	DTCBAR0

19.4 DTC Operation

When the DTC is activated, control data is read from the DTC control data area to perform data transfers and control data after data transfer is written back to the DTC control data area. Twenty-four sets of control data can be stored in the DTC control data area, which allows 24 types of data transfers to be performed.
There are two transfer modes (normal mode and repeat mode) and two transfer sizes (8-bit transfer and 16-bit transfer). When the CHNE bit in the DTCCRj (j = 0 to 23) register is set to 1 (chain transfers enabled), multiple control data is read and data transfers are continuously performed by one activation source (chain transfers).
A transfer source address is specified by the 16-bit register DTSARj, and a transfer destination address is specified by the 16-bit register DTDARj.
The values in registers DTSARj and DTDARj are separately incremented or fixed according to the control data after the data transfer.

19.4.1 Activation Sources

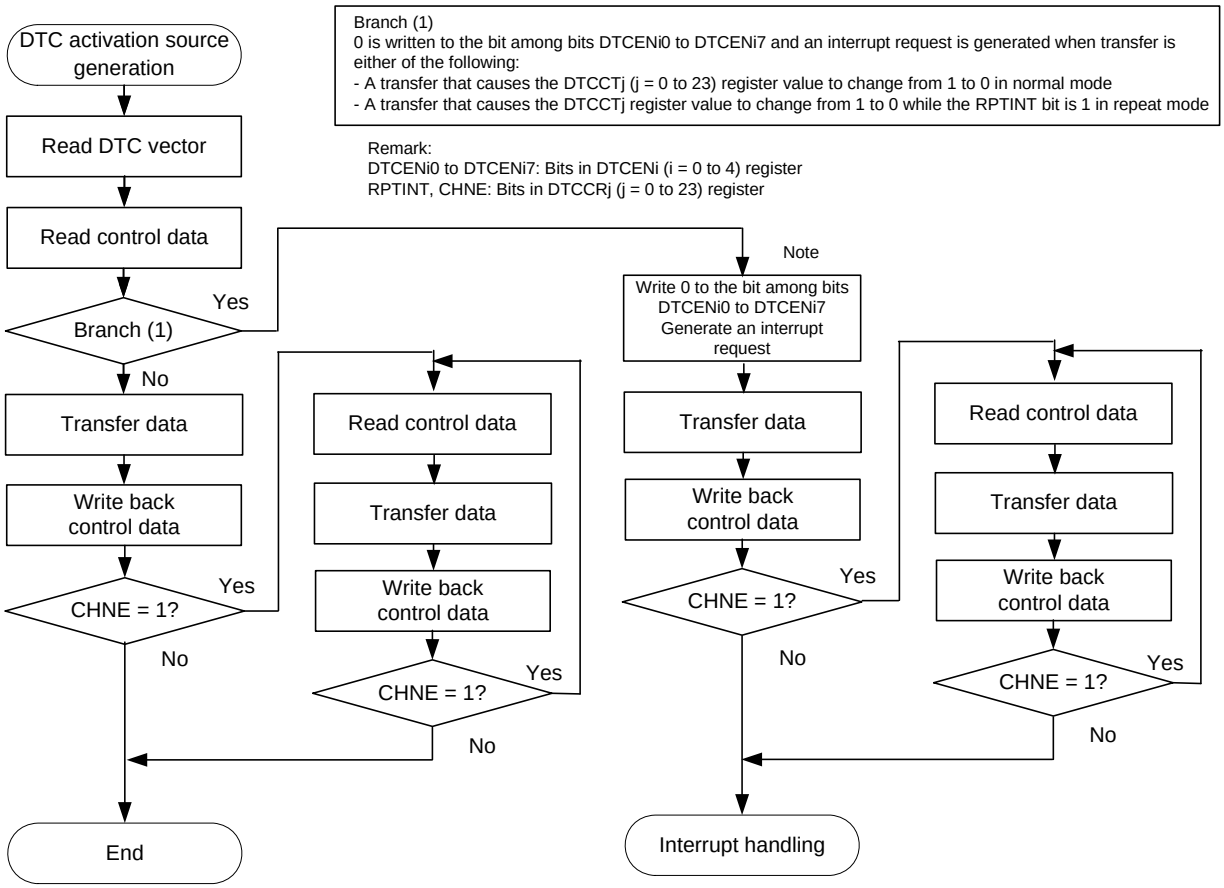
The DTC is activated by an interrupt signal from the peripheral functions. The interrupt signals to activate the DTC are selected with the DTCENi (i = 0 to 4) register.

The DTC sets the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi register to 0 (activation disabled) during operation when the setting of data transfer (the first transfer in chain transfers) is either of the following:

- A transfer that causes the DTCCTj (j = 0 to 23) register value to change to 0 in normal mode
- A transfer that causes the DTCCTj register value to change to 0 while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled) in repeat mode

Figure 19 - 14 shows the DTC Internal Operation Flowchart.

Figure 19 - 14 DTC Internal Operation Flowchart



Note 0 is not written to the bit among bits DTCENi0 to DTCENi7 for data transfers activated by the setting to enable chain transfers (the CHNE bit is 1). Also, no interrupt request is generated.

19.4.2 Normal Mode

One to 256 bytes of data are transferred by one activation during 8-bit transfer and 2 to 512 bytes during 16-bit transfer. The number of transfers can be 1 to 256 times. When the data transfer causing the DTCCTj (j = 0 to 23) register value to change to 0 is performed, the DTC generates an interrupt request corresponding to the activation source to the interrupt controller during DTC operation, and sets the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 4) in the DTCENi register to 0 (activation disabled).
Table 19 - 6 lists Register Functions in Normal Mode. Figure 19 - 15 shows Data Transfers in Normal Mode.

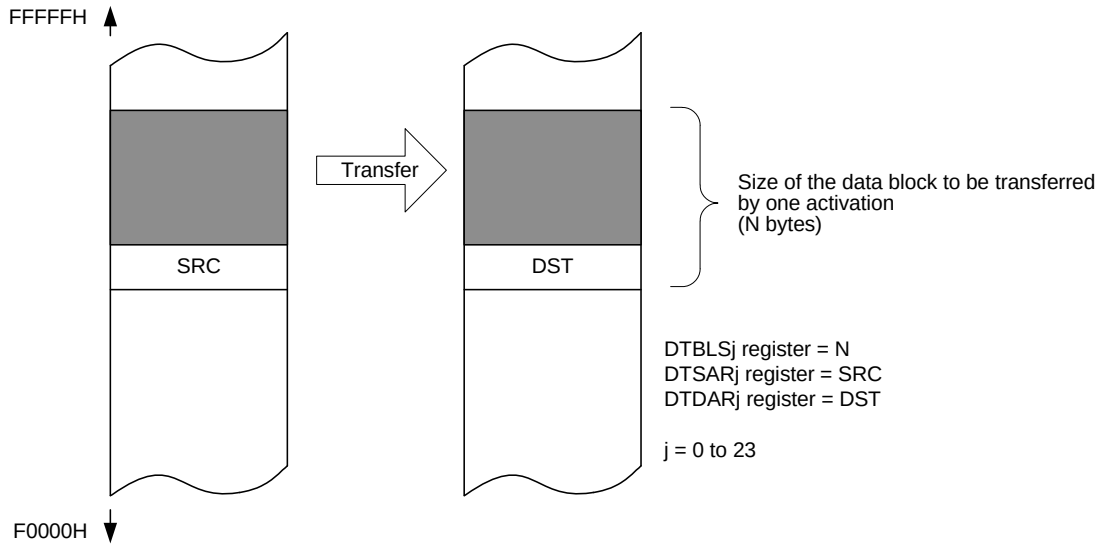
Table 19 - 6 Register Functions in Normal Mode

Register Name	Symbol	Function
DTC block size register j	DTBLSj	Size of the data block to be transferred by one activation
DTC transfer count register j	DTCCTj	Number of data transfers
DTC transfer count reload register j	DTRLDj	Not used Note
DTC source address register j	DTSARj	Data transfer source address
DTC destination address register j	DTDARj	Data transfer destination address

Note Initialize this register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

Remark j = 0 to 23

Figure 19 - 15 Data Transfers in Normal Mode



DTCCR Register Setting				Source Address Control	Destination Address Control	Source Address after Transfer	Destination Address after Transfer
DAMOD	SAMOD	RPTSEL	MODE				
0	0	X	0	Fixed	Fixed	SRC	DST
0	1	X	0	Incremented	Fixed	SRC + N	DST
1	0	X	0	Fixed	Incremented	SRC	DST + N
1	1	X	0	Incremented	Incremented	SRC + N	DST + N

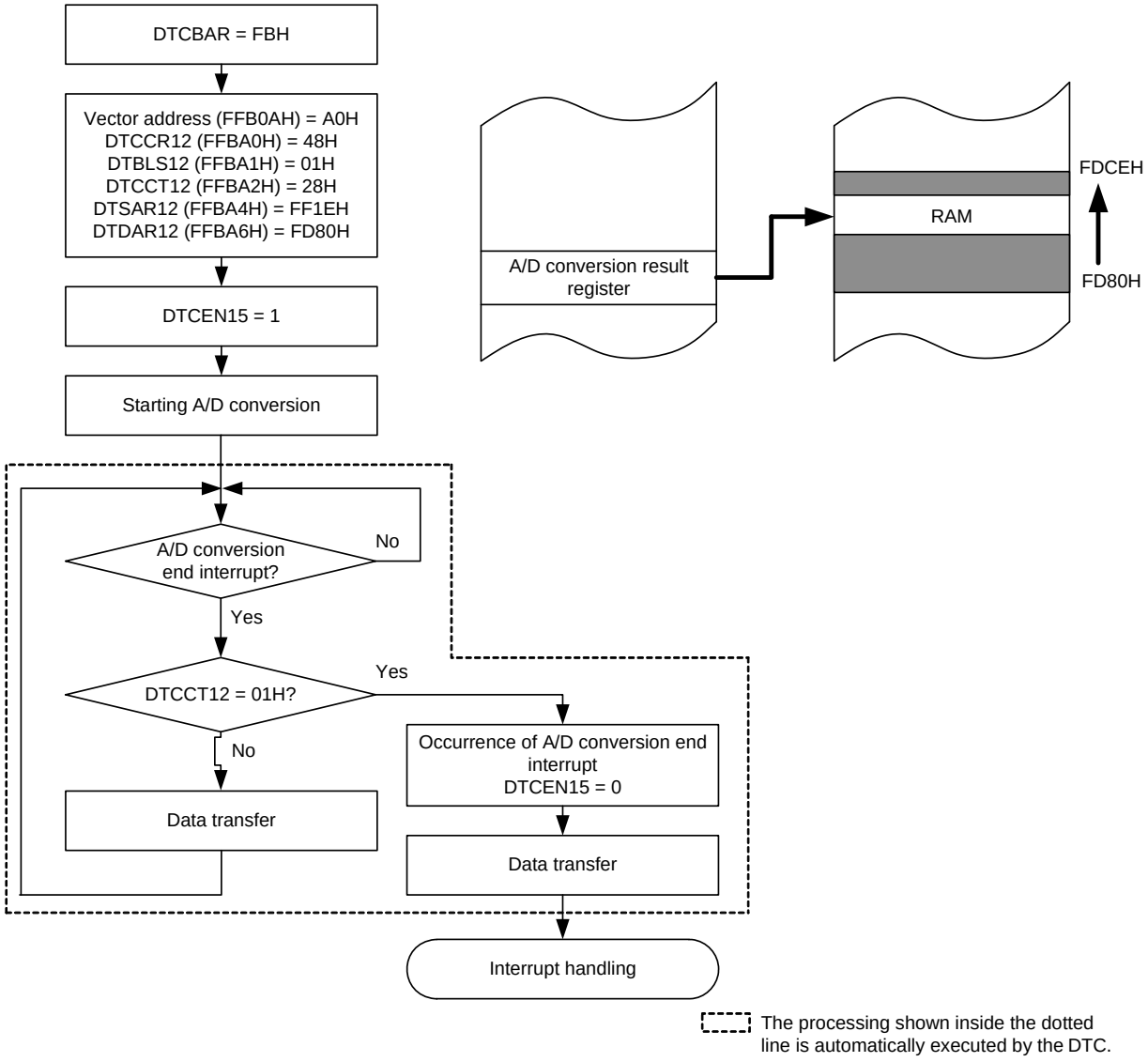
X: 0 or 1

(1) Example 1 of using normal mode: Consecutively capturing A/D conversion results

The DTC is activated by an A/D conversion end interrupt and the value of the A/D conversion result register is transferred to RAM.

- The vector address is FFB0AH and control data is allocated at FFBA0H to FFBA7H
- Transfers 2-byte data of the A/D conversion result register (FFF1EH, FFF1FH) to 80 bytes of FFD80H to FFDCFH of RAM 40 times

Figure 19 - 16 Example 1 of using normal mode: Consecutively capturing A/D conversion results



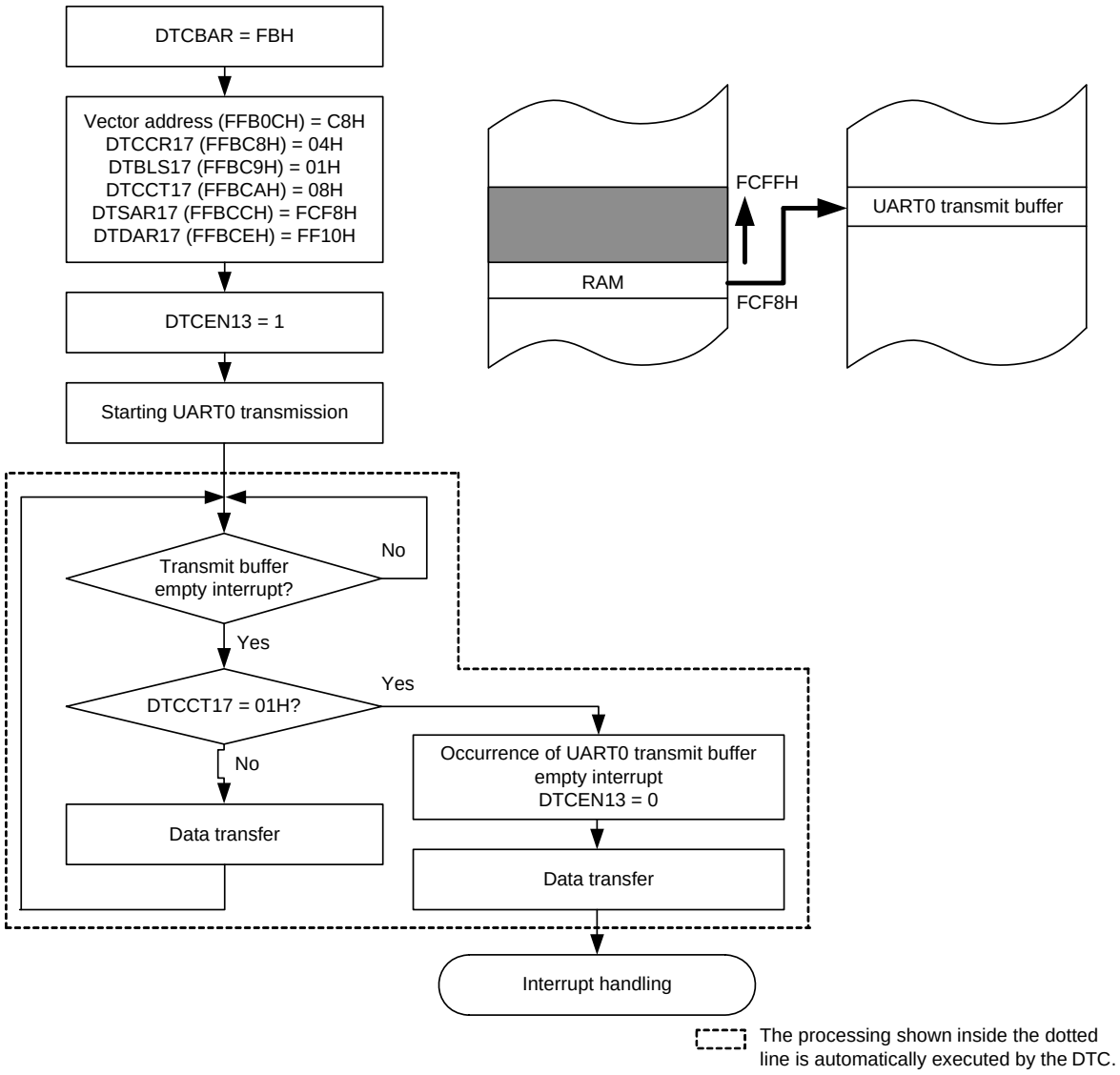
The value of the DTRLD12 register is not used because of normal mode, but initialize the register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

(2) Example 2 of using normal mode: UART consecutive transmission

The DTC is activated by a UART0 transmit buffer empty interrupt and the value of RAM is transferred to the UART0 transmit buffer.

- The vector address is FFB0CH and control data is allocated at FFBC8H to FFBCFH
- Transfers 8 bytes of FFCF8H to FFCFFH of RAM to the UART0 transmit buffer (FFF10H)

Figure 19 - 17 Example 2 of using normal mode: UART0 consecutive transmission



The value of the DTRLD17 register is not used because of normal mode, but initialize the register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

Start the first UART0 transmission by software. The second and subsequent transmissions are automatically sent when the DTC is activated by a transmit buffer empty interrupt.

19.4.3 Repeat Mode

One to 255 bytes of data are transferred by one activation. Either of the transfer source or destination should be specified as the repeat area. The number of transfers can be 1 to 255 times. On completion of the specified number of transfers, the DTCCTj (j = 0 to 23) register and the address specified for the repeat area are initialized to continue transfers. When the data transfer causing the DTCCTj register value to change to 0 is performed while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled), the DTC generates an interrupt request corresponding to the activation source to the interrupt controller during DTC operation, and sets the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi (i = 0 to 4) register to 0 (activation disabled). When the RPTINT bit in the DTCCRj register is 0 (interrupt generation disabled), no interrupt request is generated even if the data transfer causing the DTCCTj register value to change to 0 is performed. Also, bits DTCENi0 to DTCENi7 are not set to 0.

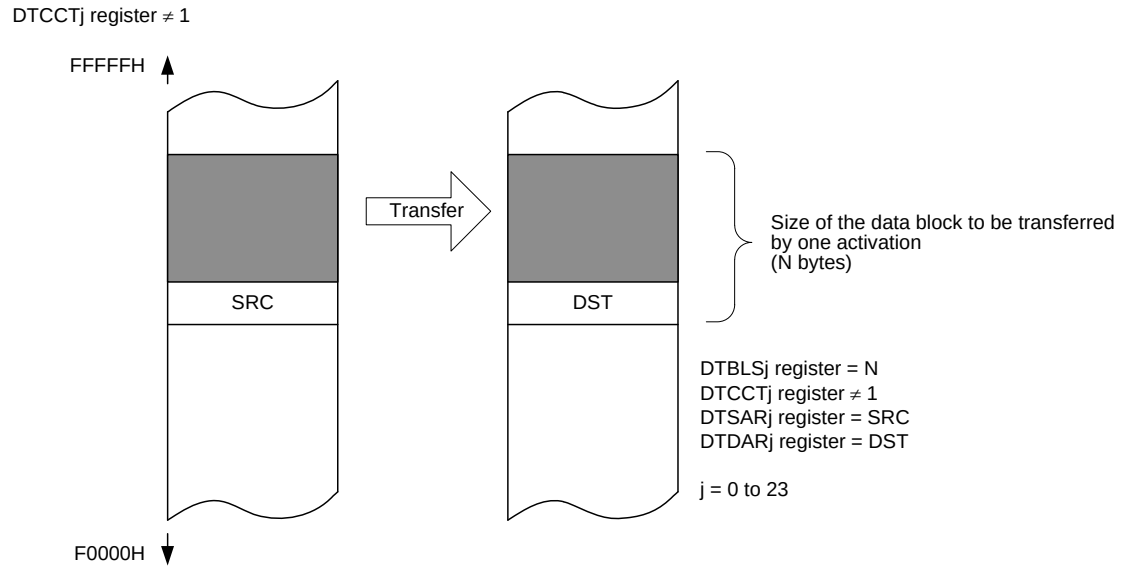
Table 19 - 7 lists Register Functions in Repeat Mode. Figure 19 - 18 shows Data Transfers in Repeat Mode.

Table 19 - 7 Register Functions in Repeat Mode

Register Name	Symbol	Function
DTC block size register j	DTBLSj	Size of the data block to be transferred by one activation
DTC transfer count register j	DTCCTj	Number of data transfers
DTC transfer count reload register j	DTRLDj	This register value is reloaded to the DTCCTj register (the number of transfers is initialized).
DTC source address register j	DTSARj	Data transfer source address
DTC destination address register j	DTDARj	Data transfer destination address

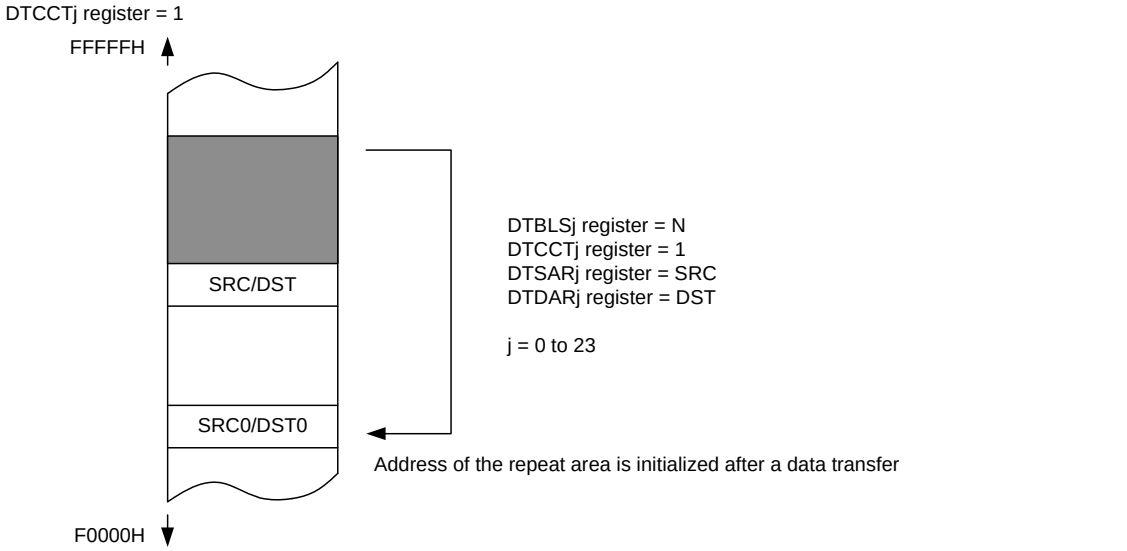
Remark j = 0 to 23

Figure 19 - 18 Data Transfers in Repeat Mode



DTCCR Register Setting				Source Address Control	Destination Address Control	Source Address after Transfer	Destination Address after Transfer
DAMOD	SAMOD	RPTSEL	MODE				
0	X	1	1	Repeat area	Fixed	SRC + N	DST
1	X	1	1	Repeat area	Incremented	SRC + N	DST + N
X	0	0	1	Fixed	Repeat area	SRC	DST + N
X	1	0	1	Incremented	Repeat area	SRC + N	DST + N

X: 0 or 1



DTCCR Register Setting				Source Address Control	Destination Address Control	Source Address after Transfer	Destination Address after Transfer
DAMOD	SAMOD	RPTSEL	MODE				
0	X	1	1	Repeat area	Fixed	SRC0	DST
1	X	1	1	Repeat area	Incremented	SRC0	DST + N
X	0	0	1	Fixed	Repeat area	SRC	DST0
X	1	0	1	Incremented	Repeat area	SRC + N	DST0

SRC0: Initial source address value
DST0: Initial destination address value
X: 0 or 1

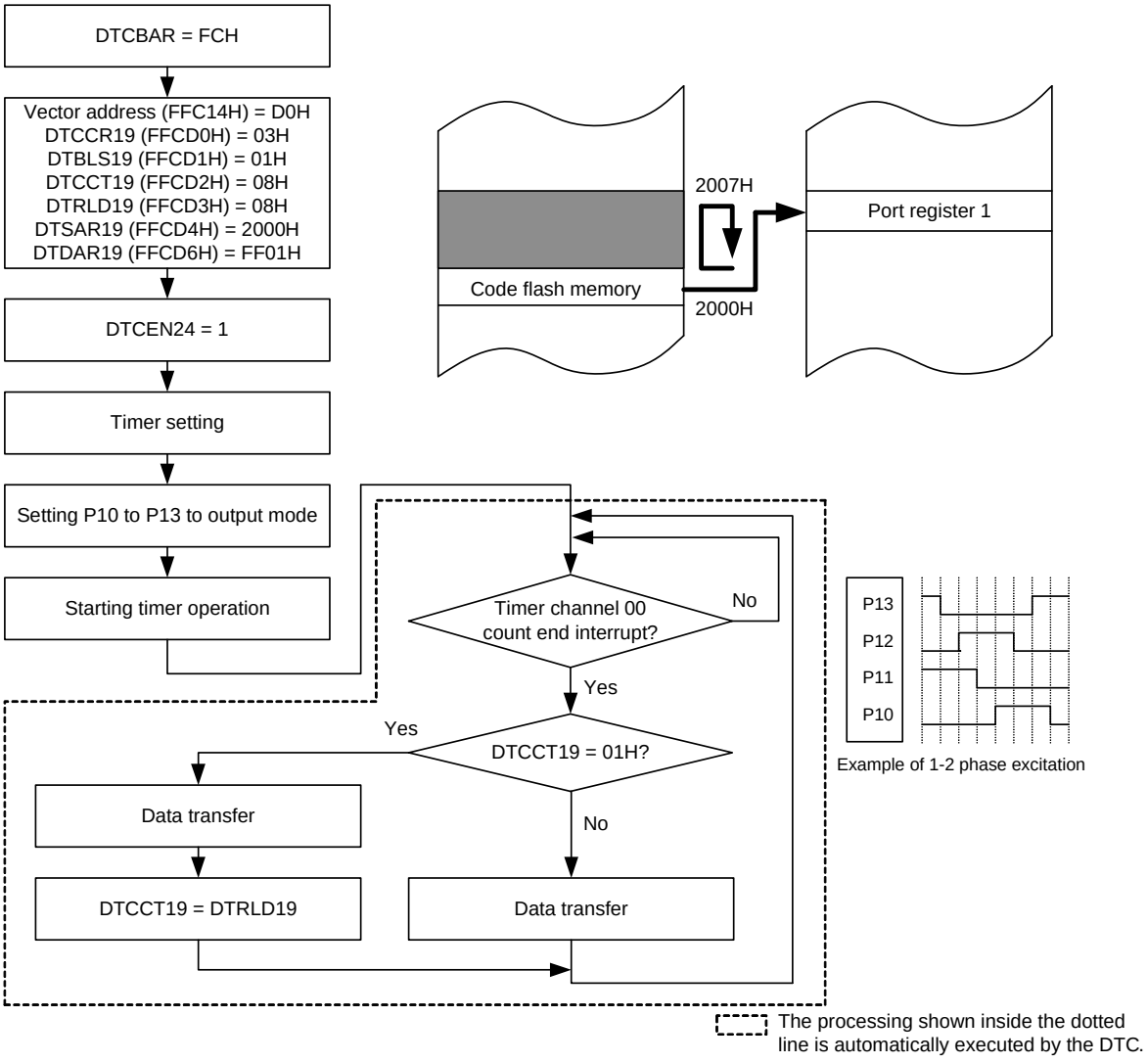
Caution 1. When repeat mode is used, the lower 8 bits of the initial value for the repeat area address must be 00H.
Caution 2. When repeat mode is used, the data size of the repeat area must be set to 255 bytes or less.

(1) Example 1 of using repeat mode: Outputting a stepping motor control pulse using ports

The DTC is activated using the interval timer function of channel 0 of timer array unit 0, and the pattern of the motor control pulse stored in the code flash memory is transferred to the general-purpose port.

- The vector address is FFC14H and control data is allocated at FFCD0H to FFCD7H
- Transfers 8-byte data of 02000H to 02007H of the code flash memory from the mirror space (F2000H to F2007H) to port register 1 (FFF01H)
- A repeat mode interrupt is disabled

Figure 19 - 19 Example 1 of using repeat mode: Outputting a stepping motor control pulse using ports



To stop the output, stop the timer first and then clear DTCEN24.

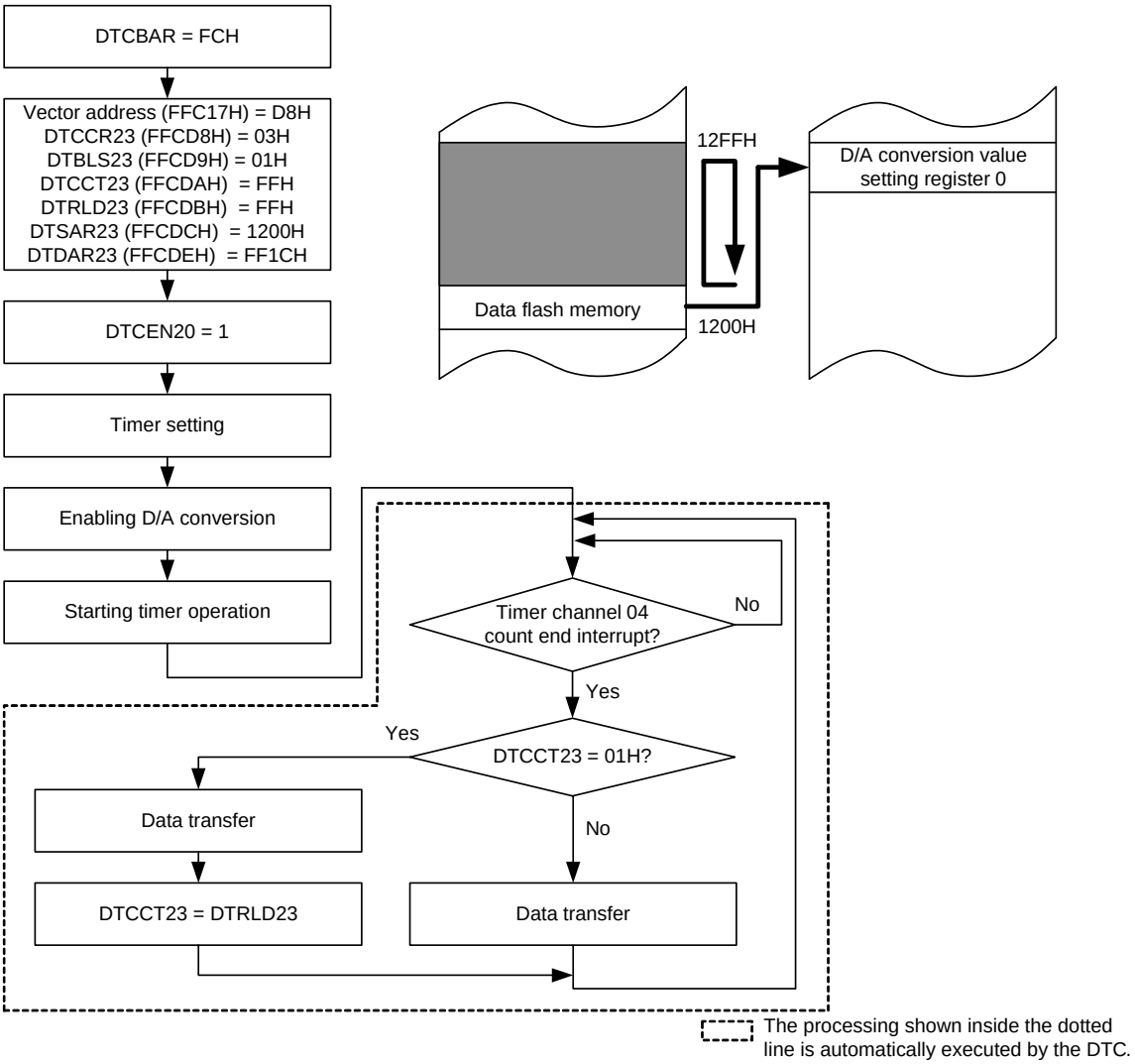
(2) Example 2 of using repeat mode: Outputting a sine wave using the 8-bit D/A converter

The DTC is activated using the interval timer function of channel 4 of timer array unit 0, and the table of the sine wave stored in the data flash memory is transferred to the 8-bit D/A conversion value setting register 0 (FFF74H).

The timer interval time is set to the D/A output setup time.

- The vector address is FFC17H and control data is allocated at FFCD8H to FFCDFH
- Transfers 255-byte data of F1200H to F12FEH of the data flash memory to the D/A conversion value setting register (FFF74H)
- A repeat mode interrupt is disabled

Figure 19 - 20 Example 2 of using repeat mode: Outputting a sine wave using the 8-bit D/A converter



To stop the output, stop the timer first and then clear DTCEN20.

19.4.4 Chain Transfers

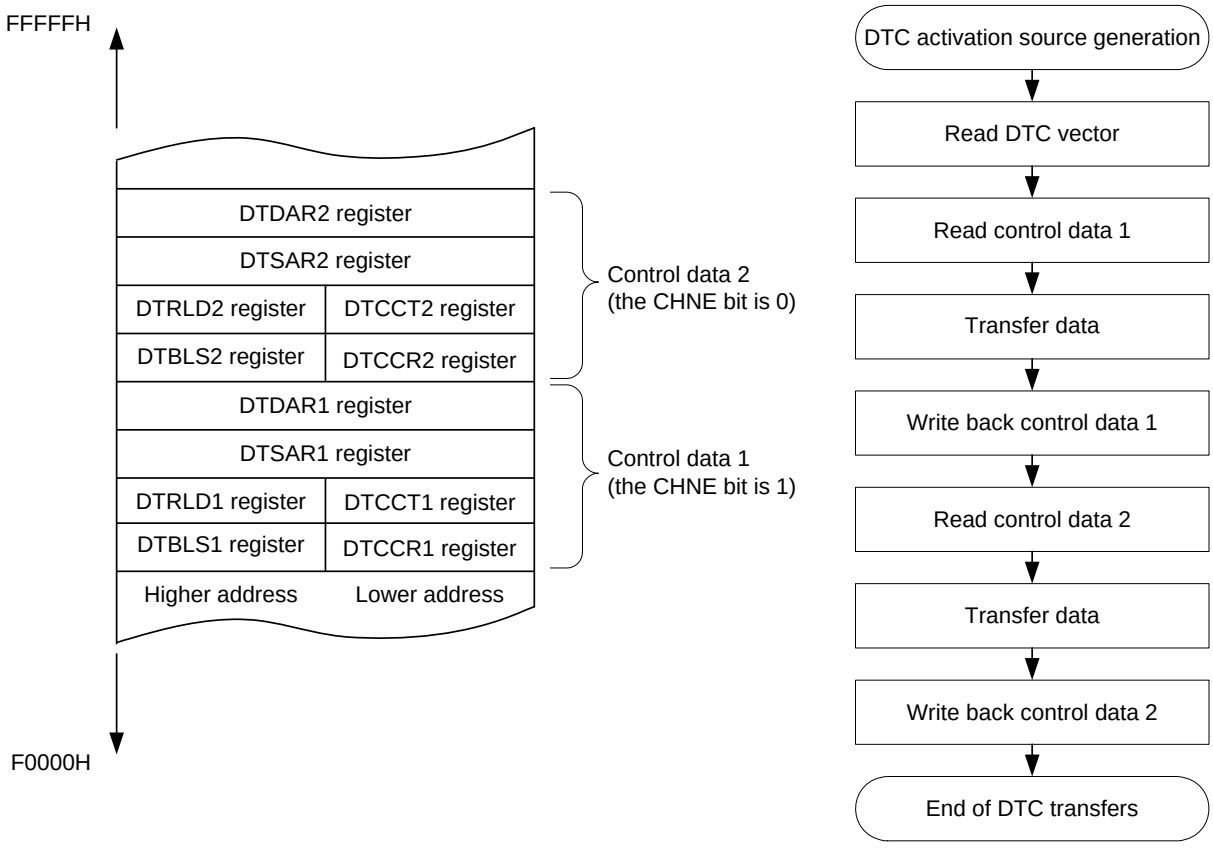
When the CHNE bit in the DTCCRj (j = 0 to 22) register is 1 (chain transfers enabled), multiple data transfers can be continuously performed by one activation source.

When the DTC is activated, one control data is selected according to the data read from the vector address corresponding to the activation source, and the selected control data is read from the DTC control data area. When the CHNE bit for the control data is 1 (chain transfers enabled), the next control data immediately following the current control data is read and transferred after the current transfer is completed. This operation is repeated until the data transfer with the control data for which the CHNE bit is 0 (chain transfers disabled) is completed.

When chain transfers are performed using multiple control data, the number of transfers set for the first control data is enabled, and the number of transfers set for the second and subsequent control data to be processed will be invalid.

Figure 19 - 21 shows Data Transfers during Chain Transfers.

Figure 19 - 21 Data Transfers during Chain Transfers



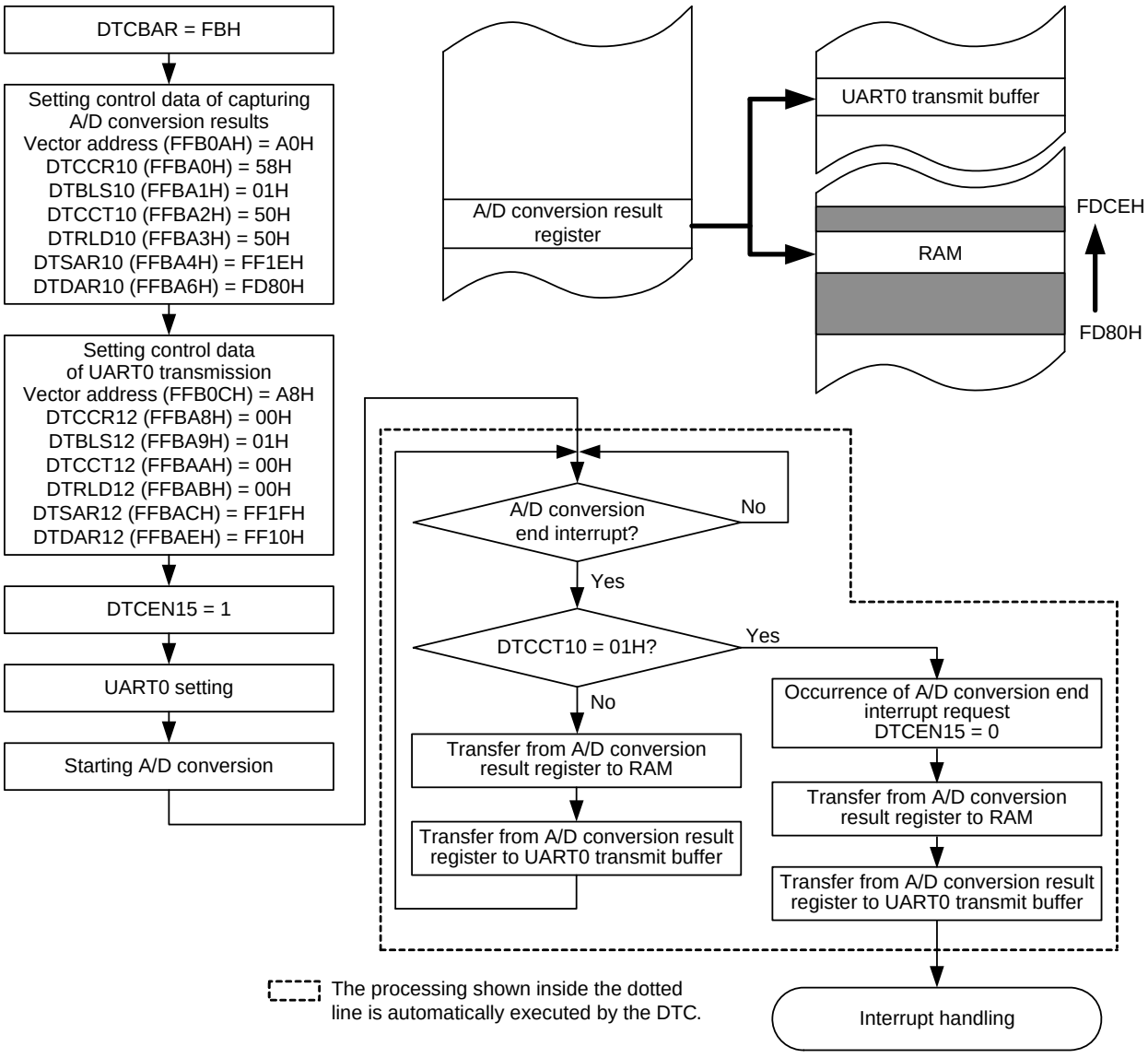
Caution 1. Set the CHNE bit in the DTCCR23 register to 0 (chain transfers disabled).

Caution 2. During chain transfers, bits DTCENi0 to DTCENi7 (i = 0 to 4) in the DTCENi register are not set to 0 (activation disabled) for the second and subsequent transfers. Also, no interrupt request is generated.

(1) Example of using chain transfers: Consecutively capturing A/D conversion results and UART0 transmission
The DTC is activated by an A/D conversion end interrupt and A/D conversion results are transferred to RAM, and then transmitted using the UART0.

- The vector address is FFB0AH
- Control data of capturing A/D conversion results is allocated at FFBA0H to FFBA7H
- Control data of UART0 transmission is allocated at FFBA8H at FFBAFH
- Transfers 2-byte data of the A/D conversion result register (FFF1FH, FFF1EH) to FFD80H to FFDCFH of RAM, and transfers the upper 1 byte (FFF1FH) of the A/D conversion result register to the UART0 transmit buffer (FFF10H)

Figure 19 - 22 Example of using chain transfers: Consecutively capturing A/D conversion results and UART0 transmission



19.5 Notes on DTC

19.5.1 Setting DTC Registers and Vector Table

- Do not access the DTC extended special function register (2nd SFR), the DTC control data area, the DTC vector table area, or the general-register (FFEE0H to FFEFFH) space using a DTC transfer.
- Modify the DTC base address register (DTCBAR) while all DTC activation sources are set to activation disabled.
- Do not rewrite the DTC base address register (DTCBAR) twice or more.
- Modify the data of the DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, or DTDARj (j = 0 to 23) register when the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi (i = 0 to 4) register is 0 (activation disabled).
- Modify the start address of the DTC control data area to be set in the vector table when the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi (i = 0 to 4) register is 0 (activation disabled).
- Do not allocate RAM addresses which are used as a DTC transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

19.5.2 Allocation of DTC Control Data Area and DTC Vector Table Area

The areas where the DTC control data and vector table can be allocated differ, depending on the usage conditions.

- It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space as the DTC control data area or DTC vector table area.
- Make sure the stack area, the DTC control data area, and the DTC vector table area do not overlap.
- The internal RAM area in the following products cannot be used as the DTC control data area or DTC vector table area when using the self-programming and data-flash functions.
  R5F110xH, R5F111xH (x = M, N, P): FBF00H to FC309H
  R5F110xJ, R5F111xJ (x = M, N, P): FBF00H to FC309H
- The internal RAM area in the following products cannot be used as the DTC control data area or DTC vector table area when using the on-chip debugging trace function.
  R5F110xH, R5F111xH (x = M, N, P): FC300H to FC6FFH
  R5F110xJ, R5F111xJ (x = M, N, P): FC300H to FC6FFH
- Initialize the DTRLD register to 00H even in normal mode when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

19.5.3 DTC Pending Instruction

Even if a DTC transfer request is generated, DTC transfer is held pending immediately after the following instructions. Also, the DTC is not activated between PREFIX instruction code and the instruction immediately after that code.

- Call/return instruction
- Unconditional branch instruction
- Conditional branch instruction
- Read access instruction for code flash memory
- Bit manipulation instructions for IFxx, MKxx, PRxx, and PSW, and an 8-bit manipulation instruction that has the ES register as operand
- Instruction for accessing the data flash memory
- Instruction of Multiply, Divide, Multiply & Accumulate (excluding MULU)

Caution 1. When a DTC transfer request is acknowledged, all interrupt requests are held pending until DTC transfer is completed.

Caution 2. While the DTC is held pending by the DTC pending instruction, all interrupt requests are held pending.

19.5.4 Operation when Accessing Data Flash Memory Space

Because DTC data transfer is suspended to access the data flash space, be sure to add the DTC pending instruction.

If the data flash space is accessed after an instruction execution from start of DTC data transfer, a 3-clock wait will be inserted to the next instruction.

```
Instruction 1
DTC data transfer
Instruction ←  The wait of three clock cycles occurs.
MOV A,         ! Data Flash space
```

19.5.5 Number of DTC Execution Clock Cycles

Table 19 - 8 lists the Operations Following DTC Activation and Required Number of Cycles for each operation.

Table 19 - 8 Operations Following DTC Activation and Required Number of Cycles

Vector Read	Control Data		Data Read	Data Write
	Read	Write-back		
1	4	Note 1	Note 2	Note 2

Note 1. For the number of clock cycles required for control data write-back, refer to **Table 19 - 9 Number of Clock Cycles Required for Control Data Write-Back Operation**.

Note 2. For the number of clock cycles required for data read/write, refer to **Table 19 - 9 Number of Clock Cycles Required for Control Data Write-Back Operation**.

Table 19 - 9 Number of Clock Cycles Required for Control Data Write-Back Operation

DTCCR Register Setting				Address Setting		Control Register to be Written Back				Number of Clock Cycles
DAMOD	SAMOD	RPTSEL	MODE	Source	Destination	DTCCTj Register	DTRLDj Register	DTSARj Register	DTDARj Register	
0	0	X	0	Fixed	Fixed	Written back	Written back	Not written back	Not written back	1
0	1	X	0	Incremented	Fixed	Written back	Written back	Written back	Not written back	2
1	0	X	0	Fixed	Incremented	Written back	Written back	Not written back	Written back	2
1	1	X	0	Incremented	Incremented	Written back	Written back	Written back	Written back	3
0	X	1	1	Repeat area	Fixed	Written back	Written back	Written back	Not written back	2
1	X	1	1		Incremented	Written back	Written back	Written back	Written back	3
X	0	0	1	Fixed	Repeat area	Written back	Written back	Not written back	Written back	2
X	1	0	1	Incremented		Written back	Written back	Written back	Written back	3

Remark j = 0 to 23; X: 0 or 1

Table 19 - 10 Number of Clock Cycles Required for One Data Read/Write Operation

Operation	RAM	Code Flash Memory	Data Flash Memory	Special function register (SFR)	Extended special function register (2nd SFR)	
					No Wait State	Wait States
Data read	1	2	4	1	1	1 + number of wait states Note
Data write	1	—	—	1	1	1 + number of wait states Note

Note The number of wait states differs depending on the specifications of the register allocated to the extended special function register (2nd SFR) to be accessed.

19.5.6 DTC Response Time

Table 19 - 11 lists the DTC Response Time. The DTC response time is the time from when the DTC activation source is detected until DTC transfer starts. It does not include the number of DTC execution clocks.

Table 19 - 11 DTC Response Time

	Minimum Time	Maximum Time
Response Time	3 clocks	19 clocks

Note that the response from the DTC may be further delayed under the following cases. The number of delayed clock cycles differs depending on the conditions.

- When executing an instruction from the internal RAM
  Maximum response time: 20 clocks
- When executing a DTC pending instruction (refer to **19.5.3 DTC Pending Instruction**)
- Maximum response time: Maximum response time for each condition + execution clock cycles for the instruction to be held pending under the condition.
- When accessing the USB register that a wait occurs
  Maximum response time: Maximum response time for each condition + 1 clock

Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU/peripheral hardware clock)

19.5.7 DTC Activation Sources

- After inputting a DTC activation source, do not input the same activation source again until DTC transfer is completed.
- While a DTC activation source is generated, do not manipulate the DTC activation enable bit corresponding to the source.
- If DTC activation sources conflict, their priority levels are determined in order to select the source for activation when the CPU acknowledges the DTC transfer. For details on the priority levels of activation sources, refer to **19.3.3 Vector Table**.
- When DTC activation is enabled under either of the following conditions, a DTC transfer is started and an interrupt is generated after completion of the transfer. Therefore, enable DTC activation after confirming the comparator monitor flag (CnMON) as necessary. (n = 0, 1)
  - The comparator Note is set to an interrupt request on one-edge detection (CnEDG = 0), an interrupt request at the rising edge for the comparator, and IVCMP > IVREF (or internal reference voltage: 1.45 V)
  - The comparator is set to an interrupt request on one-edge detection (CnEDG = 0), an interrupt request at the falling edge for the comparator, and IVCMP < IVREF (or internal reference voltage: 1.45 V)

Note Comparator 1 is provided in 100-pin products.

19.5.8 Operation in Standby Mode Status

Status	DTC Operation
HALT mode	Operable (Operation is disabled while in the low power consumption RTC mode)
STOP mode	DTC activation sources can be accepted Note 1
SNOOZE mode	Operable Notes 2, 3, 4, 5

Note 1. In the STOP mode, detecting a DTC activation source enables transition to SNOOZE mode and DTC transfer. After completion of transfer, the system returns to the STOP mode. However, since the code flash memory and the data flash memory are stopped during the SNOOZE mode, the flash memory cannot be set as the transfer source.

Note 2. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected as f_{CLK}.

Note 3. When a transfer end interrupt is set as a DTC activation source from the CSIp SNOOZE mode function, release the SNOOZE mode by the transfer end interrupt to start CPU processing after completion of DTC transfer, or use a chained transfer to set CSIp reception again (writing 1 to the STm0 bit, writing 0 to the SWCm bit, setting of the SSCm register, and writing 1 to the SSm0 bit).

Note 4. When a transfer end interrupt is set as a DTC activation source from the UARTq SNOOZE mode function, release the SNOOZE mode by the transfer end interrupt to start CPU processing after completion of DTC transfer, or use a chained transfer to set UARTq reception again (writing 1 to the STm1 bit, writing 0 to the SWCm bit, setting of the SSCm register, and writing 1 to the SSm1 bit).

Note 5. When an A/D conversion end interrupt is set as a DTC activation source from the A/D converter SNOOZE mode function, release the SNOOZE mode by the A/D conversion end interrupt to start CPU processing after completion of DTC transfer, or use a chained transfer to set the A/D converter SNOOZE mode function again after clear the AWC bit.

Remark p = 00, 20; q = 0, 2; m = 0, 1

CHAPTER 20 EVENT LINK CONTROLLER (ELC)

20.1 Functions of ELC

The event link controller (ELC) mutually connects (links) events output from each peripheral function. By linking events, it becomes possible to coordinate operation between peripheral functions directly without going through the CPU.

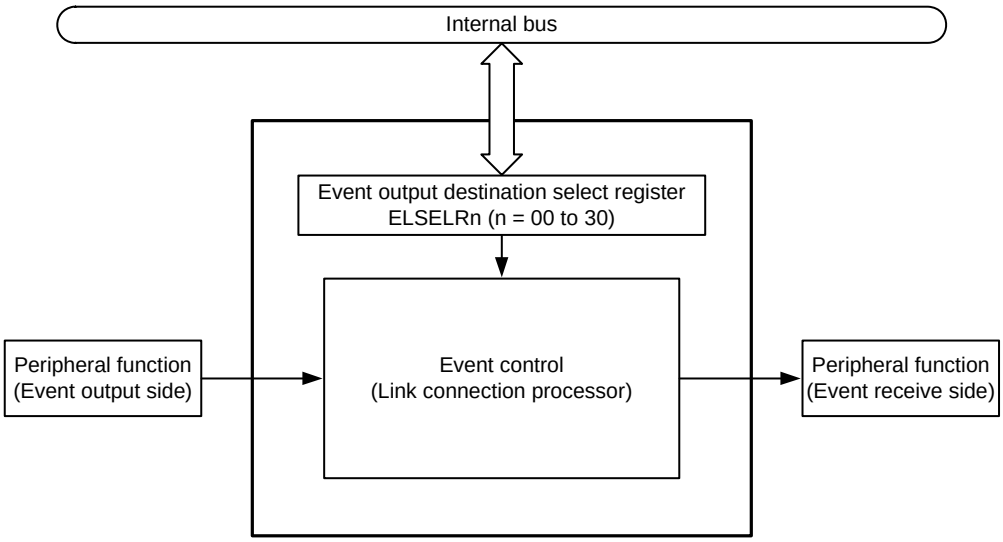
The ELC has the following functions.

- Capable of directly linking event signals from 30 types (80/85-pin products) or 31 types (100-pin products) of peripheral functions to specified peripheral functions
- Event signals can be used as activation sources for operating any one of 23 types (80/85- and 100-pin products) of peripheral functions

20.2 Configuration of ELC

Figure 20 - 1 shows the ELC Block Diagram.

Figure 20 - 1 ELC Block Diagram



20.3 Registers Controlling ELC

Table 20 - 1 lists the Registers Controlling ELC.

Table 20 - 1 Registers Controlling ELC

Register name	Symbol
Event link setting register 00	ELSELR00
Event link setting register 01	ELSELR01
Event link setting register 02	ELSELR02
Event link setting register 03	ELSELR03
Event link setting register 04	ELSELR04
Event link setting register 05	ELSELR05
Event link setting register 06	ELSELR06
Event link setting register 07	ELSELR07
Event link setting register 08	ELSELR08
Event link setting register 09	ELSELR09
Event link setting register 10	ELSELR10
Event link setting register 11	ELSELR11
Event link setting register 12	ELSELR12
Event link setting register 13	ELSELR13
Event link setting register 14	ELSELR14
Event link setting register 15	ELSELR15
Event link setting register 16	ELSELR16
Event link setting register 17	ELSELR17
Event link setting register 18	ELSELR18
Event link setting register 19 Note 1	ELSELR19
Event link setting register 20	ELSELR20
Event link setting register 21	ELSELR21
Event link setting register 22	ELSELR22
Event link setting register 23 Note 2	ELSELR23
Event link setting register 24 Note 2	ELSELR24
Event link setting register 25 Note 2	ELSELR25
Event link setting register 26 Note 2	ELSELR26
Event link setting register 27 Note 2	ELSELR27
Event link setting register 28 Note 2	ELSELR28
Event link setting register 29 Note 2	ELSELR29
Event link setting register 30 Note 2	ELSELR30

Note 1. For 100-pin products only.

Note 2. Event output destination select registers for 16-bit timers KB20, KB21, and KB22

20.3.1 Event output destination select register n (ELSELRn) (n = 00 to 30)

An ELSELRn register links each event signal to an operation of an event-receiving peripheral function (link destination peripheral function) after reception.

Do not set multiple event inputs to the same event output destination (event receive side). The operation of the event-receiving peripheral function will become undefined, and event signals may not be received correctly. In addition, do not set the event link generation source and the event link output destination to the same function.

Set an ELSELRn register during a period when no event output peripheral functions are generating event signals.

Table 20 - 2 lists the Correspondence Between ELSELRn (n = 00 to 30) Registers and Peripheral Functions.

Figure 20 - 2 Format of Event output destination select register n (ELSELRn)

Address: F01C0H (ELSELR00) to F01DEH (ELSELR30) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ELSELRn	0	0	0	ELSELn4	ELSELn3	ELSELn2	ELSELn1	ELSELn0

ELSELn4	ELSELn3	ELSELn2	ELSELn1	ELSELn0	Event Link Selection
0	0	0	0	0	Event link disabled
0	0	0	0	1	Select operation of peripheral function 1 to link Note
0	0	0	1	0	Select operation of peripheral function 2 to link Note
0	0	0	1	1	Select operation of peripheral function 3 to link Note
0	0	1	0	0	Select operation of peripheral function 4 to link Note
0	0	1	0	1	Select operation of peripheral function 5 to link Note
0	0	1	1	0	Select operation of peripheral function 6 to link Note
0	0	1	1	1	Select operation of peripheral function 7 to link Note
0	1	0	0	0	Select operation of peripheral function 8 to link Note
0	1	0	0	1	Select operation of peripheral function 9 to link Note
0	1	0	1	0	Select operation of peripheral function 10 to link Note
0	1	0	1	1	Select operation of peripheral function 11 to link Note
0	1	1	0	0	Select operation of peripheral function 12 to link Note
0	1	1	0	1	Select operation of peripheral function 13 to link Note
0	1	1	1	0	Select operation of peripheral function 14 to link Note
0	1	1	1	1	Select operation of peripheral function 15 to link Note
1	0	0	0	0	Select operation of peripheral function 16 to link Note
1	0	0	0	1	Select operation of peripheral function 17 to link Note
Other than above					Setting prohibited

Note See **Tables 20 - 3** to **20 - 6 Correspondence Between Values Set to ELSELRn (n = 00 to 30) Registers and Operation of Link Destination Peripheral Functions at Reception**.

Table 20 - 2 Correspondence Between ELSELRn (n = 00 to 30) Registers and Peripheral Functions

Register Name	Event Generator (Output Origin of Event Input n)	Event Description
ELSELR00	External interrupt edge detection 0	INTP0 Note 1
ELSELR01	External interrupt edge detection 1	INTP1 Note 1
ELSELR02	External interrupt edge detection 2	INTP2 Note 1
ELSELR03	External interrupt edge detection 3	INTP3 Note 1
ELSELR04	External interrupt edge detection 4	INTP4 Note 1
ELSELR05	External interrupt edge detection 5	INTP5 Note 1
ELSELR06	External interrupt edge detection 6	INTP6 Note 1
ELSELR07	External interrupt edge detection 7	INTP7 Note 1
ELSELR08	Key return signal detection	INTKR
ELSELR09	RTC fixed-cycle signal/Alarm match detection	INTRTC
ELSELR10	TAU channel 00 Count end/Capture end	INTTM00
ELSELR11	TAU channel 01 Count end/Capture end	INTTM01
ELSELR12	TAU channel 02 Count end/Capture end	INTTM02
ELSELR13	TAU channel 03 Count end/Capture end	INTTM03
ELSELR14	TAU channel 04 Count end/Capture end	INTTM04
ELSELR15	TAU channel 05 Count end/Capture end	INTTM05
ELSELR16	TAU channel 06 Count end/Capture end	INTTM06
ELSELR17	TAU channel 07 Count end/Capture end	INTTM07
ELSELR18	Comparator detection 0	COMP_C0EVT (comparator 0 detection)
ELSELR19	Comparator detection 1	COMP_C1EVT Note 3 (comparator 1 detection)
ELSELR20	Timer KB20 trigger output	Timer KB20 compare match signal
ELSELR21	Timer KB21 trigger output	Timer KB21 compare match signal
ELSELR22	Timer KB22 trigger output	Timer KB22 compare match signal
ELSELR23	External interrupt detection 0	INTP0NF Note 2
ELSELR24	External interrupt detection 1	INTP1NF Note 2
ELSELR25	External interrupt detection 2	INTP2NF Note 2
ELSELR26	External interrupt detection 3	INTP3NF Note 2
ELSELR27	External interrupt detection 4	INTP4NF Note 2
ELSELR28	External interrupt detection 5	INTP5NF Note 2
ELSELR29	External interrupt detection 6	INTP6NF Note 2
ELSELR30	External interrupt detection 7	INTP7NF Note 2

Note 1. INTPm (m = 0 to 7) is affected by the setting of the external interrupt rising edge enable register (EGP0) and the external interrupt falling edge enable register (EGN0) that can be used for edge detection.

Note 2. INTPmNF (m = 0 to 7) is not affected by the setting of the external interrupt rising edge enable register (EGP0) and the external interrupt falling edge enable register (EGN0). INTPmNF is an event generator only for 16-bit timers KB20, KB21, and KB22 link destinations.

Note 3. For 100-pin products only.

Table 20 - 3 Correspondence Between Values Set to ELSELRn (n = 00 to 17) Registers and Operation of Link Destination Peripheral Functions at Reception

Bits ELSELn4 to ELSELn0 in ELSELRn Register	Link Destination Number	Link Destination Peripheral Function	Link Destination Peripheral Function
00000B	—	Event link disabled	—
00001B	1	A/D converter	A/D conversion starts
00010B	2	Timer input of timer array unit 0 channel 0 Note 1	Delay counter, input pulse interval measurement, external event counter
00011B	3	Timer input of timer array unit 0 channel 1 Note 2	Delay counter, input pulse interval measurement, external event counter
00100B	4	16-bit timer KB20 count restart trigger source 0	Timer output restarts Compare register batch overwrite
00101B	5	16-bit timer KB20 count restart trigger source 1	Timer output restarts Compare register batch overwrite
00110B	6	16-bit timer KB20 count restart trigger source 2	Timer output restarts Compare register batch overwrite
01000B	8	16-bit timer KB21 count restart trigger source 0	Timer output restarts Compare register batch overwrite
01001B	9	16-bit timer KB21 count restart trigger source 1	Timer output restarts Compare register batch overwrite
01010B	10	16-bit timer KB21 count restart trigger source 2	Timer output restarts Compare register batch overwrite
01100B	12	16-bit timer KB22 count restart trigger source 0	Timer output restarts Compare register batch overwrite
01101B	13	16-bit timer KB22 count restart trigger source 1	Timer output restarts Compare register batch overwrite
01110B	14	16-bit timer KB22 count restart trigger source 2	Timer output restarts Compare register batch overwrite
10000B	16	DA0 Note 3	Real-time output
10001B	17	DA1 Note 3	Real-time output
Other than above	—	Setting prohibited	—

Note 1. To select the timer input of timer array unit 0 channel 0 as the link destination peripheral function, set the operating clock for channel 0 to f_{CLK} using timer clock select register 0 (TPS0), set the noise filter of the TI00 pin to OFF (TNFEN00 = 0) using noise filter enable register 1 (NFEN1), and then set the timer output used for channel 0 to an event input signal from the ELC using timer input select register 0 (TIS0).

Note 2. To select the timer input of timer array unit 0 channel 1 as the link destination peripheral function, set the operating clock for channel 1 to f_{CLK} using timer clock select register 0 (TPS0), set the noise filter of the TI01 pin to OFF (TNFEN01 = 0) using noise filter enable register 1 (NFEN1), and then set the timer output used for channel 1 to an event input signal from the ELC using timer input select register 0 (TIS0).

Note 3. When entering the STOP mode while the real-time output event mode for D/A conversion is enabled, disable linking of ELC events before entering STOP mode.

Table 20 - 4 Correspondence Between Values Set to ELSELRn (n = 18, 19) Registers and Operation of Link Destination Peripheral Functions at Reception

Bits ELSELn4 to ELSELn0 in ELSELRn Register	Link Destination Number	Link Destination Peripheral Function	Link Destination Peripheral Function
00000B	—	Event link disabled	—
00001B	1	A/D converter	A/D conversion starts
00010B	2	Timer input of timer array unit 0 channel 0 Note 1	Delay counter, input pulse interval measurement, external event counter
00011B	3	Timer input of timer array unit 0 channel 1 Note 2	Delay counter, input pulse interval measurement, external event counter
00100B	4	16-bit timer KB20 count restart trigger source 0	Timer output restarts Compare register batch overwrite
00101B	5	16-bit timer KB20 count restart trigger source 1	Timer output restarts Compare register batch overwrite
00110B	6	16-bit timer KB20 count restart trigger source 2	Timer output restarts Compare register batch overwrite
01000B	8	16-bit timer KB21 count restart trigger source 0	Timer output restarts Compare register batch overwrite
01001B	9	16-bit timer KB21 count restart trigger source 1	Timer output restarts Compare register batch overwrite
01010B	10	16-bit timer KB21 count restart trigger source 2	Timer output restarts Compare register batch overwrite
01100B	12	16-bit timer KB22 count restart trigger source 0	Timer output restarts Compare register batch overwrite
01101B	13	16-bit timer KB22 count restart trigger source 1	Timer output restarts Compare register batch overwrite
01110B	14	16-bit timer KB22 count restart trigger source 2	Timer output restarts Compare register batch overwrite
10000B	16	DA0 Note 3	Real-time output
10001B	17	DA1 Note 3	Real-time output
10010B	18	16-bit timers KB20,KB21,KB22 forced output stop source 0 Note 4	16-bit timers KB20,KB21,KB22 forced output stop function 1/2
10011B	19	16-bit timers KB20,KB21,KB22 forced output stop source 1 Note 4	16-bit timers KB20,KB21,KB22 forced output stop function 1/2
10100B	20	16-bit timer KB20 forced output stop source Note 4	16-bit timer KB20 forced output stop function 1/2
10101B	21	16-bit timer KB21 forced output stop source Note 4	16-bit timer KB21 forced output stop function 1/2
10110B	22	16-bit timer KB22 forced output stop source Note 4	16-bit timer KB22 forced output stop function 1/2
10111B	23	16-bit timers KB20,KB21,KB22 forced output stop source 3 Note 4	16-bit timers KB20,KB21,KB22 forced output stop function 2
Other than above	—	Setting prohibited	—

(**Notes** are listed on the next page.)

Note 1. To select the timer input of timer array unit 0 channel 0 as the link destination peripheral function, set the operating clock for channel 0 to f_{CLK} using timer clock select register 0 (TPS0), set the noise filter of the TI00 pin to OFF (TNFEN00 = 0) using noise filter enable register 1 (NFEN1), and then set the timer output used for channel 0 to an event input signal from the ELC using timer input select register 0 (TIS0).

Note 2. To select the timer input of timer array unit 0 channel 1 as the link destination peripheral function, set the operating clock for channel 1 to f_{CLK} using timer clock select register 0 (TPS0), set the noise filter of the TI01 pin to OFF (TNFEN01 = 0) using noise filter enable register 1 (NFEN1), and then set the timer output used for channel 1 to an event input signal from the ELC using timer input select register 0 (TIS0).

Note 3. When entering the STOP mode while the real-time output event mode for D/A conversion is enabled, disable linking of ELC events before entering STOP mode.

Note 4. For details on the forced output stop sources of 16-bit timers KB20, KB21, and KB22, see the forced output stop function control registers TKBPACTL0p, TKBPACTL1p, and TKBPACTL2p.

Table 20 - 5 Correspondence Between Values Set to ELSELRn (n = 20 to 22) Registers and Operation of Link Destination Peripheral Functions at Reception

Bits ELSELn4 to ELSELn0 in ELSELRn Register	Link Destination Number	Link Destination Peripheral Function	Link Destination Peripheral Function
00000B	—	Event link disabled	—
00001B	1	A/D converter	A/D conversion starts
00010B	2	Timer input of timer array unit 0 channel 0 Note 1	Delay counter, input pulse interval measurement, external event counter
00011B	3	Timer input of timer array unit 0 channel 1 Note 2	Delay counter, input pulse interval measurement, external event counter
10000B	16	DA0 Note 3	Real-time output
10001B	17	DA1 Note 3	Real-time output
Other than above	—	Setting prohibited	—

Note 1. To select the timer input of timer array unit 0 channel 0 as the link destination peripheral function, set the operating clock for channel 0 to f_{CLK} using timer clock select register 0 (TPS0), set the noise filter of the TI00 pin to OFF (TNFEN00 = 0) using noise filter enable register 1 (NFEN1), and then set the timer output used for channel 0 to an event input signal from the ELC using timer input select register 0 (TIS0).

Note 2. To select the timer input of timer array unit 0 channel 1 as the link destination peripheral function, set the operating clock for channel 1 to f_{CLK} using timer clock select register 0 (TPS0), set the noise filter of the TI01 pin to OFF (TNFEN01 = 0) using noise filter enable register 1 (NFEN1), and then set the timer output used for channel 1 to an event input signal from the ELC using timer input select register 0 (TIS0).

Note 3. When entering the STOP mode while the real-time output event mode for D/A conversion is enabled, disable linking of ELC events before entering STOP mode.

Table 20 - 6 Correspondence Between Values Set to ELSELRn (n = 23 to 30) Registers and Operation of Link Destination Peripheral Functions at Reception

Bits ELSELn4 to ELSELn0 in ELSELRn Register	Link Destination Number	Link Destination Peripheral Function	Link Destination Peripheral Function
00000B	—	Event link disabled	—
10010B	18	16-bit timers KB20,KB21,KB22 forced output stop source 0 Note	16-bit timers KB20,KB21,KB22 forced output stop function 1/2
10011B	19	16-bit timers KB20,KB21,KB22 forced output stop source 1 Note	16-bit timers KB20,KB21,KB22 forced output stop function 1/2
10100B	20	16-bit timer KB20 forced output stop source Note	16-bit timer KB20 forced output stop function 1/2
10101B	21	16-bit timer KB21 forced output stop source Note	16-bit timer KB21 forced output stop function 1/2
10110B	22	16-bit timer KB22 forced output stop source Note	16-bit timer KB22 forced output stop function 1/2
10111B	23	16-bit timers KB20,KB21,KB22 forced output stop source 3 Note	16-bit timers KB20,KB21,KB22 forced output stop function 2
Other than above	—	Setting prohibited	—

Note For details on the forced output stop sources of 16-bit timers KB20, KB21, and KB22, see the forced output stop function control registers TKBPACTL0p, TKBPACTL1p, and TKBPACTL2p.

20.4 ELC Operation

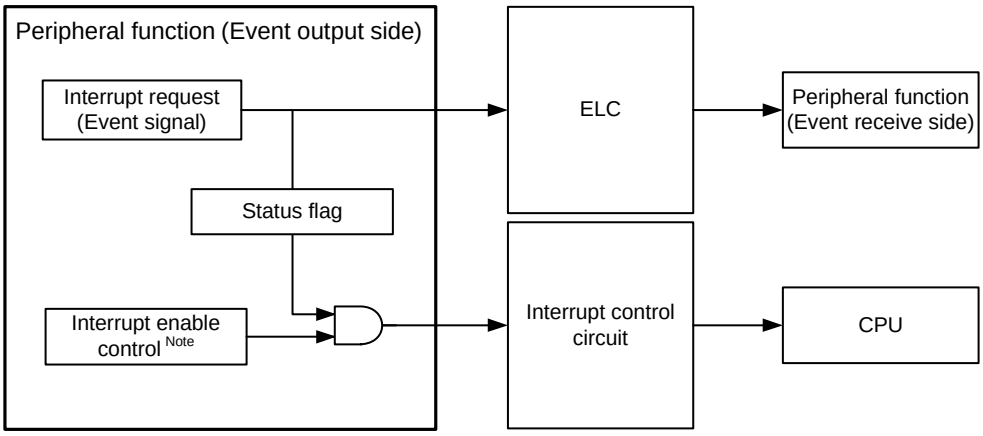
The path for using an event signal generated by a peripheral function as an interrupt request to the interrupt control circuit is independent from the path for using it as an ELC event. Therefore, each event signal can be used as an event signal for operation of an event-receiving peripheral function, regardless of interrupt control.

In addition, event link operation can be performed without being influenced by the presence or absence of a CPU clock supply. However, the operating clock of a peripheral function needs to be supplied and be in an operational state.

Figure 20 - 3 shows the Relationship Between Interrupt Handling and ELC. The figure show an example of an interrupt request status flag and a peripheral function possessing the enable bits that control enabling/disabling of such interrupts.

A peripheral function which receives an event from the ELC will perform the operation corresponding to the event-receiving peripheral function after reception of an event.

Figure 20 - 3 Relationship Between Interrupt Handling and ELC



Note Not available depending on the peripheral function.

Table 20 - 7 lists the Response of Peripheral Functions That Receive Events.

Table 20 - 7 Response of Peripheral Functions That Receive Events

Event Receiver No.	Event Link Destination Function	Operation after Event Reception	Response
1	A/D converter	A/D conversion	A hardware trigger of A/D conversion is generated after 2 or 3 cycles of fCLK after an ELC event is generated.
2	Timer array unit 0 Timer input of channel 0	Delay counter Input pulse width measurement External event counter	The edge is detected 3 or 4 cycles of fCLK after an ELC event is generated.
3	Timer array unit 0 Timer input of channel 1		
4	16-bit timer KB20 count restart trigger source 0	16-bit timer KB20 output Restart compare register batch overwrite	The edge is detected 2 or 3 cycles of fCLK after an ELC event is generated.
5	16-bit timer KB20 count restart trigger source 1		
6	16-bit timer KB20 count restart trigger source 2		
7	Reserved	—	—
8	16-bit timer KB21 count restart trigger source 0	16-bit timer KB21 output Restart compare register batch overwrite	The edge is detected 2 or 3 cycles of fCLK after an ELC event is generated.
9	16-bit timer KB21 count restart trigger source 1		
10	16-bit timer KB21 count restart trigger source 2		
11	Reserved	—	—
12	16-bit timer KB22 count restart trigger source 0	16-bit timer KB22 output Restart compare register batch overwrite	The edge is detected 2 or 3 cycles of fCLK after an ELC event is generated.
13	16-bit timer KB22 count restart trigger source 1		
14	16-bit timer KB22 count restart trigger source 2		
15	Reserved	—	—
16	Channel 0 of D/A converter	Real-time output (channel 0)	An event from the ELC is directly used as a trigger of D/A conversion of channel 0.
17	Channel 1 of D/A converter	Real-time output (channel 1)	An event from the ELC is directly used as a trigger of D/A conversion of channel 1.
18	16-bit timers KB20,KB21,KB22 forced output stop source 0	16-bit timers KB20,KB21,KB22 forced output stop function 1/2	An event from the ELC is directly used as a trigger of the output stop function.
19	16-bit timers KB20,KB21,KB22 forced output stop source 1		
20	16-bit timer KB20 forced output stop source	16-bit timer KB20 forced output stop function 1/2	
21	16-bit timer KB21 forced output stop source	16-bit timer KB21 forced output stop function 1/2	
22	16-bit timer KB22 forced output stop source	16-bit timer KB22 forced output stop function 1/2	
23	16-bit timers KB20,KB21,KB22 forced output stop source 2	16-bit timers KB20,KB21,KB22 forced output stop function 2	

CHAPTER 21 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.

		Products with USB		Products without USB	
		80/85-pin	100-pin	80/85-pin	100-pin
Maskable interrupts	External	9	9	9	9
	Internal	36	37	32	33

21.1 Interrupt Function Types

The following two types of interrupt functions are used.

(1) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into four priority groups by setting the priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L).

Multiple interrupt servicing can be applied to low-priority interrupts when high-priority interrupts are generated. If two or more interrupt requests, each having the same priority, are simultaneously generated, then they are processed according to the default priority of vectored interrupt servicing. For the default priority, see **Tables 21 - 1** to **21 - 4**.

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.

21.2 Interrupt Sources and Configuration

Interrupt sources include maskable interrupts and software interrupts. In addition, they also have up to five reset sources (see **Tables 21 - 1** to **21 - 4**). 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 21 - 1 Interrupt Source List (1/4)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type Note 2	Products with USB		Products without USB	
		Name	Trigger				100-pin	80/85-pin	100-pin	80/85-pin
Maskable	0	INTWDTI	Watchdog timer interval Note 3 (75% of overflow time + 1/2fIL)	Internal	00004H	(A)	√	√	√	√
	1	INTLVI	Voltage detection Note 4		00006H		√	√	√	√
	2	INTP0	Pin input edge detection	External	00008H	(B)	√	√	√	√
	3	INTP1			0000AH		√	√	√	√
	4	INTP2			0000CH		√	√	√	√
	5	INTP3			0000EH		√	√	√	√
	6	INTP4			00010H		√	√	√	√
	7	INTP5			00012H		√	√	√	√
	8	INTST2/ INTCSI20/ INTIIC20	UART2 transmission transfer end or buffer empty interrupt/CSI20 transfer end or buffer empty interrupt/IIC20 transfer end	Internal	00014H	(A)	√	√	√	√
	9	INTSR2/ INTCSI21/ INTIIC21	UART2 reception transfer end/CSI21 transfer end or buffer empty interrupt/IIC21 transfer end		00016H		√	√	√	√
	10	INTSRE2	UART2 reception communication error occurrence		00018H		√	√	√	√
	11	INTST0/ INTCSI00/ INTIIC00	UART0 transmission transfer end or buffer empty interrupt/CSI00 transfer end or buffer empty interrupt/IIC00 transfer end		0001EH		√	√	√	√
	12	INTTM00	End of timer channel 0 count or capture		00020H		√	√	√	√
	13	INTSR0	UART0 reception transfer end		00022H		√	√	√	√

Note 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 45 indicates the lowest priority.

Note 2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 21 - 1.

Note 3. When bit 7 (WDTINT) of the option byte (000C0H) is set to 1.

Note 4. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is cleared to 0.

Table 21 - 2 Interrupt Source List (2/4)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type Note 2	Products with USB		Products without USB	
		Name	Trigger				100-pin	80/85-pin	100-pin	80/85-pin
Maskable	14	INTSRE0	UART0 reception communication error occurrence	Internal	00024H	(A)	√	√	√	√
		INTTM01H	End of timer channel 1 count or capture (at higher 8-bit timer operation)				√	√	√	√
	15	INTST1/ INTCSI10/ INTIIC10	UART1 transmission transfer end or buffer empty interrupt/CSI10 transfer end or buffer empty interrupt/IIC10 transfer end		00026H		√	√	√	√
	16	INTSR1	UART1 reception transfer end		00028H		√	√	√	√
	17	INTSRE1	UART1 reception communication error occurrence		0002AH		√	√	√	√
		INTTM03H	End of timer channel 03 count or capture (at higher 8-bit timer operation)				√	√	√	√
	18	INTIICA0	End of IICA0 communication		0002CH		√	√	√	√
	19	INTRTIT	RTC correction timing		0002EH		√	√	√	√
	20	INTTM01	End of timer channel 01 count or capture (at 16-bit/lower 8-bit timer operation)		00032H		√	√	√	√
	21	INTTM02	End of timer channel 02 count or capture		00034H		√	√	√	√
	22	INTTM03	End of timer channel 03 count or capture (at 16-bit/lower 8-bit timer operation)		00036H		√	√	√	√
	23	INTAD	End of A/D conversion		00038H		√	√	√	√
	24	INTRTC	Fixed-cycle signal of real-time clock/alarm match detection		0003AH		√	√	√	√
	25	INTIT	Interval signal of 12-bit interval timer detection		0003CH		√	√	√	√
	26	INTKR	Key return signal detection	External	0003EH	(C)	√	√	√	√
	27	INTST3/ INTCSI30/ INTIIC30	UART3 transmission transfer end or buffer empty interrupt/CSI30 transfer end or buffer empty interrupt/IIC30 transfer end	Internal	00040H	(A)	√	√	√	√

Note 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 45 indicates the lowest priority.

Note 2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 21 - 1.

Table 21 - 3 Interrupt Source List (3/4)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type Note 2	Products with USB		Products without USB	
		Name	Trigger				100-pin	80/85-pin	100-pin	80/85-pin
Maskable	28	INTSR3	UART3 reception transfer end	Internal	00042H	(A)	√	√	√	√
	29	INTTM04	End of timer channel 04 count or capture		00046H		√	√	√	√
	30	INTTM05	End of timer channel 05 count or capture		00048H		√	√	√	√
	31	INTP6	Pin input edge detection	External	0004AH	(B)	√	√	√	√
	32	INTP7			0004CH		√	√	√	√
	33	INTCMP0	Comparator detection 0	Internal	00050H	(A)	√	√	√	√
	34	INTCMP1	Comparator detection 1		00052H		√	—	√	—
	35	INTTM06	End of timer channel 06 count or capture		00054H		√	√	√	√
	36	INTTM07	End of timer channel 07 count or capture		00056H		√	√	√	√
	37	INTUSB	USB INT interrupt		00058H		√	√	—	—
	38	INTRSUM	USB RESUME interrupt		0005AH		√	√	—	—
	39	INTSRE3	UART3 reception communication error occurrence		0005CH		√	√	√	√
	40	INTTKB2_0	End of timer KB2_0 count		0005EH		√	√	√	√
	41	INTTKB2_1	End of timer KB2_1 count		00060H		√	√	√	√
	42	INTFL	Reserved Note 3		00062H		√	√	√	√
	43	INTTKB2_2	End of timer KB2_2 count		00064H		√	√	√	√
	44	DTC0FIFO	End of DFIFO0 interrupt transfer		00066H		√	√	—	—
	45	DTC1FIFO	End of DFIFO1 interrupt transfer		00068H		√	√	—	—

Note 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 45 indicates the lowest priority.

Note 2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 21 - 1.

Note 3. Be used at the flash self programming library or the data flash library.

Table 21 - 4 Interrupt Source List (4/4)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type Note 2	Products with USB		Products without USB	
		Name	Trigger				100-pin	80/85-pin	100-pin	80/85-pin
Software	—	BRK	Execution of BRK instruction	—	0007EH	(D)	√	√	√	√
Reset	—	RESET	$\overline{\text{RESET}}$ pin input	—	00000H	—	√	√	√	√
		POR	Power-on-reset				√	√	√	√
		LVD	Voltage detection Note 3				√	√	√	√
		WDT	Overflow of watchdog timer				√	√	√	√
		TRAP	Execution of illegal instruction Note 4				√	√	√	√
		IAW	Illegal-memory access				√	√	√	√
		RPE	RAM parity error				√	√	√	√

Note 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 45 indicates the lowest priority.

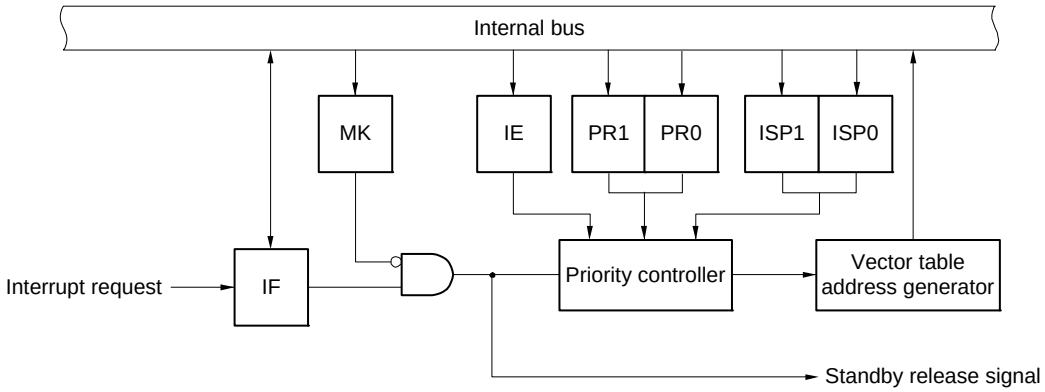
Note 2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 21 - 1.

Note 3. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is set to 1.

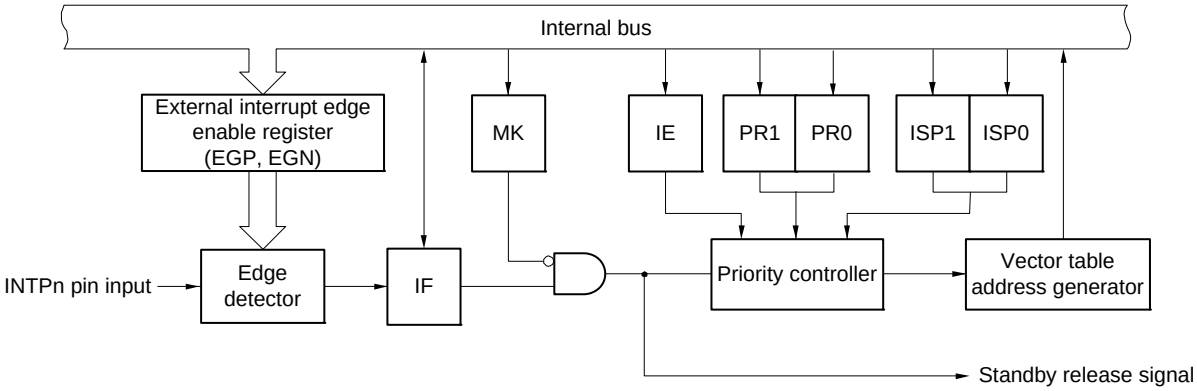
Note 4. When the instruction code in FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Figure 21 - 1 Basic Configuration of Interrupt Function

(A) Internal maskable interrupt



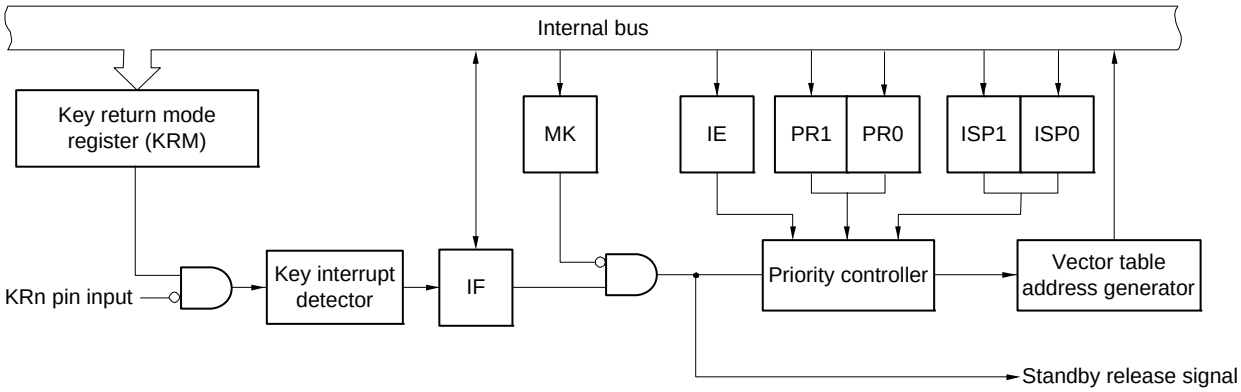
(B) External maskable interrupt (INTPn)



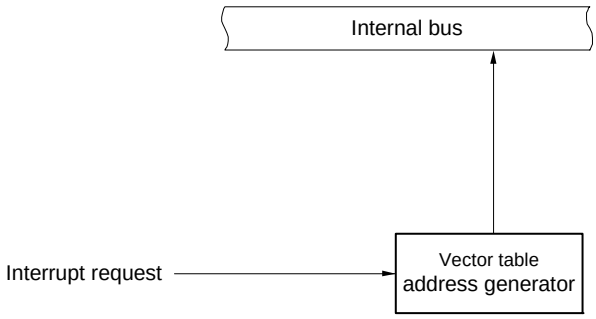
IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

Remark n = 0 to 7

(C) External maskable interrupt (INTKR)



(D) Software interrupt



IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

Remark n = 0 to 7

21.3 Registers Controlling Interrupt Functions

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L)
- Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L)
- Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L)
- External interrupt rising edge enable registers (EGP0)
- External interrupt falling edge enable registers (EGN0)
- Program status word (PSW)

Tables 21 - 5 to 21 - 8 show a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 21 - 5 Flags Corresponding to Interrupt Request Sources (1/4)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		Products with USB		Products without USB	
		Register		Register		Register	100-pin	80/85-pin	100-pin	80/85-pin
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L	√	√	√	√
INTLVI	LVIIF		LVIMK		LVIPR0, LVIPR1		√	√	√	√
INTP0	PIF0		PMK0		PPR00, PPR10		√	√	√	√
INTP1	PIF1		PMK1		PPR01, PPR11		√	√	√	√
INTP2	PIF2		PMK2		PPR02, PPR12		√	√	√	√
INTP3	PIF3		PMK3		PPR03, PPR13		√	√	√	√
INTP4	PIF4		PMK4		PPR04, PPR14		√	√	√	√
INTP5	PIF5		PMK5		PPR05, PPR15		√	√	√	√

Table 21 - 6 Flags Corresponding to Interrupt Request Sources (2/4)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		Products with USB		Products without USB	
		Register		Register		Register	100-pin	80/85-pin	100-pin	80/85-pin
INTST2 Note 1	STIF2 Note 1	IF0H	STMK2 Note 1	MK0H	STPR02, STPR12 Note 1	PR00H, PR10H	√	√	√	√
INTCSI20 Note 1	CSIIF20 Note 1		CSIMK20 Note 1		CSIPR020, CSIPR120 Note 1		√	√	√	√
INTIIC20 Note 1	IICIF20 Note 1		IICMK20 Note 1		IICPR020, IICPR120 Note 1		√	√	√	√
INTSR2	SRIF2		SRMK2		SRPR02, SRPR12		√	√	√	√
INTSRE2	SREIF2		SREMK2		SREPR02, SREPR12		√	√	√	√
INTST0 Note 2	STIF0 Note 2		STMK0 Note 2		STPR00, STPR10 Note 2		√	√	√	√
INTCSI00 Note 2	CSIIF00 Note 2		CSIMK00 Note 2		CSIPR000, CSIPR100 Note 2		√	√	√	√
INTIIC00 Note 2	IICIF00 Note 2		IICMK00 Note 2		IICPR000, IICPR100 Note 2		√	√	√	√
INTSR0	SRIF0		SRMK0		SRPR00, SRPR10		√	√	√	√
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100		√	√	√	√

Note 1. If one of the interrupt sources INTST2, INTCSI20, and INTIIC20 is generated, bit 0 of the IF0H register is set to 1. Bit 0 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.

Note 2. If one of the interrupt sources INTST0, INTCSI00, and INTIIC00 is generated, bit 5 of the IF0H register is set to 1. Bit 5 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.

Table 21 - 7 Flags Corresponding to Interrupt Request Sources (3/4)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		Products with USB		Products without USB	
		Register		Register		Register	100-pin	80/85-pin	100-pin	80/85-pin
INTSRE0 Note 1	SREIF0 Note 1	IF1L	SREMK0 Note 1	MK1L	SREPR00, SREPR10 Note 1	PR01L, PR11L	√	√	√	√
INTTM01H Note 1	TMIF01H Note 1		TMMK01H Note 1		TMPR001H, TMPR101H Note 1		√	√	√	√
INTCSI10 Note 2	CSIIF10 Note 2		CSIMK10 Note 2		CSIPR010, CSIPR110 Note 2		√	√	√	√
INTIIC10 Note 2	IICIF10 Note 2		IICMK10 Note 2		IICPR010, IICPR110 Note 2		√	√	√	√
INTST1 Note 2	STIF1 Note 2		STMK10 Note 2		STPR01, STPR11 Note 2		√	√	√	√
INTSR1	SRIF1		SRMK1		SRPR01, SRPR11		√	√	√	√
INTSRE1 Note 3	SREIF1 Note 3		SREMK1 Note 3		SREPR01, SREPR11 Note 3		√	√	√	√
INTTM03H Note 3	TMIF03H Note 3		TMMK03H Note 3		TMPR003H, TMPR103H Note 3		√	√	√	√
INTIICA0	IICAIF0		IICAMK0		IICAPR00, IICAPR10		√	√	√	√
INTRTIT	RTITIF		RTITMK		RTITPR0, RTITPR1		√	√	√	√
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101		√	√	√	√
INTTM02	TMIF02	IF1H	TMMK02	MK1H	TMPR002, TMPR102	PR01H, PR11H	√	√	√	√
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103		√	√	√	√
INTAD	ADIF		ADMK		ADPR0, ADPR1		√	√	√	√
INTRTC	RTCIF		RTCMK		RTCPR0, RTCPR1		√	√	√	√
INTIT	TMKAIF		TMKAMK		TMKAPR0, TMKAPR1		√	√	√	√
INTKR	KRIF		KRMK		KRPR0, KRPR1		√	√	√	√
INTST3 Note 4	STIF3 Note 4		STMK3 Note 4		STPR03, STPR13 Note 4		√	√	√	√
INTCSI30 Note 4	CSIIF30 Note 4		CSIMK30 Note 4		CSIPR030, CSIPR130 Note 4		√	√	√	√
INTIIC30 Note 4	IICIF30 Note 4		IICMK30 Note 4		IICPR030, IICPR130 Note 4		√	√	√	√
INTSR3	SRIF3		SRMK3		SRPR03, SRPR13		√	√	√	√

Note 1. 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 0 of the IF1L register is set to 1. Bit 0 of the MK1L, PR01L, and PR11L registers supports these two interrupt sources.

Note 2. If one of the interrupt sources INTST1, INTCSI10, and INTIIC10 is generated, bit 1 of the IF1L register is set to 1. Bit 1 of the MK1L, PR01L, and PR11L registers supports these three interrupt sources.

Note 3. Do not use a UART1 reception error interrupt and an interrupt of channel 3 of TAU0 (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. 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 3 of the IF1L register is set to 1. Bit 3 of the MK1L, PR01L, and PR11L registers supports these two interrupt sources.

Note 4. If one of the interrupt sources INTST3, INTCSI30, and INTIIC30 is generated, bit 6 of the IF1H register is set to 1. Bit 6 of the MK1H, PR01H, and PR11H registers supports these three interrupt sources.

Table 21 - 8 Flags Corresponding to Interrupt Request Sources (4/4)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		Products with USB		Products without USB	
		Register		Register		Register	100-pin	80/85-pin	100-pin	80/85-pin
INTTM04	TMIF04	IF2L	TMMK04	MK2L	TMPR004, TMPR104	PR02L, PR12L	√	√	√	√
INTTM05	TMIF05		TMMK05		TMPR005, TMPR105		√	√	√	√
INTP6	PIF6		PMK6		PPR06, PPR16		√	√	√	√
INTP7	PIF7		PMK7		PPR07, PPR17		√	√	√	√
INTCMP0	CMPIF0		CMPMK0		CMPPR00, CMPPR10		√	√	√	√
INTCMP1	CMPIF1		CMPMK1		CMPPR01, CMPPR11		√	—	√	—
INTTM06	TMIF06	IF2H	TMMK06	MK2H	TMPR006, TMPR106	PR02H, PR12H	√	√	√	√
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107		√	√	√	√
INTUSB	USBIF		USBMK		USBPR0, USBPR1		√	√	—	—
INTRSUM	RSUIF		RSUMK		RSUPR0, RSUPR1		√	√	—	—
INTSRE3	SREIF3		SREMK3		SREPR03, SREPR13		√	√	√	√
INTTKB20	TKB2IF0		TKB2MK0		TKB2PR00, TKB2PR10		√	√	√	√
INTTKB21	TKB2IF1		TKB2MK1		TKB2PR01, TKB2PR11		√	√	√	√
INTFL	FLIF		FLMK		FLPR0, FLPR1		√	√	√	√
INTTKB22	TKB2IF2	IF3L	TKB2MK2	MK3L	TKB2PR02, TKB2PR12	PR03L, PR13L	√	√	√	√
INTD0FIFO	FIFOIF0		FIFOMK0		FIFOPR00, FIFOPR10		√	√	—	—
INTD1FIFO	FIFOIF1		FIFOMK1		FIFOPR01, FIFOPR11		√	√	—	—

21.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L)

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, IF2H, and IF3L 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 21 - 2 Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L) (1/2)

Address: FFFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIF	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	3	<2>	<1>	<0>
IF0H	SRIF0	TMIF00	STIF0 CSIIF00 IICIF00	0	0	SREIF2	SRIF2	STIF2 CSIIF20 IICIF20

Address: FFFE2H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	TMIF01	0	RTITIF	IICAIF0	SREIF1 TMIF03H	SRIF1	CSIIF10 IICIF10 STIF1	SREIF0 TMMK01H

Address: FFFE3H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1H	SRIF3	CSIIF30 IICIF30 STIF3	KRIF	TMKAIF	RTCIF	ADIF	TMIF03	TMIF02

Address: FFFD0H After reset: 00H R/W

Symbol	<7>	<6>	5	<4>	<3>	<2>	<1>	0
IF2L	CMPIF1	CMPIF0	0	PIF7	PIF6	TMIF05	TMIF04	0

Figure 21 - 3 Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L) (2/2)

Address: FFFD1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2H	FLIF	TKB2IF1	TKB2IF0	SREIF3	RSUIF	USBIF	TMIF07	TMIF06

Address: FFFD2H After reset: 00H R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
IF3L	0	0	0	0	0	FIFOIF1	FIFOIF0	TKB2IF2

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Caution 1. The available registers and bits differ depending on the product. For details about the registers and bits available for each product, see Tables 21 - 5 to 21 - 8. Be sure to set bits that are not available to the initial value.

Caution 2. When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as "IF0L.0 = 0;" or "_asm ("clr1 IF0L, 0");" because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1).

If a program is described in C language using an 8-bit memory manipulation instruction such as "IF0L &= 0xfe;" and compiled, it becomes the assembler of three instructions.

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

In this case, even if the request flag of the another bit of the same interrupt request flag register (IF0L) is set to 1 at the timing between "mov a, IF0L" and "mov IF0L, a", the flag is cleared to 0 at "mov IF0L, a". Therefore, care must be exercised when using an 8-bit memory manipulation instruction in C language.

21.3.2 Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt.

The MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, and MK3L 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 21 - 4 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L) (1/2)

Address: FFFE4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK	WDTIMK

Address: FFFE5H After reset: FFH R/W

Symbol	<7>	6	<5>	4	3	<2>	<1>	<0>
MK0H	SRMK0	TMMK00	STMK0 CSIMK00 IICMK00	1	1	SREMK2	SRMK2	STMK2 CSIMK20 IICMK20

Address: FFFE6H After reset: FFH R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	TMMK01	1	RTITMK	IICAMK0	SREMK1 TMMK03H	SRMK1	CSIMK10 IICMK10 STMK1	SREMK0 TMMK01H

Address: FFFE7H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1H	SRMK3	CSIMK30 IICMK30 STMK3	KRMK	TMKAMK	RTCMK	ADMK	TMMK03	TMMK02

Address: FFFD4H After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	<3>	<2>	<1>	0
MK2L	CMPMK1	CMPMK0	1	PMK7	PMK6	TMMK05	TMMK04	1

Figure 21 - 5 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L) (2/2)

Address: FFFD5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2H	FLMK	TKB2MK1	TKB2MK0	SREMK3	RSUMK	USBMK	TMMK07	TMMK06

Address: FFFD6H After reset: FFH R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
MK3L	1	1	1	1	1	FIFOMK1	FIFOMK0	TKB2MK2

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 Tables 21 - 5 to 21 - 8. Be sure to set bits that are not available to the initial value.

21.3.3 Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L)

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, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, and PR13L 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 21 - 6 Format of Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L) (1/3)

Address: FFFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	PPR05	PPR04	PPR03	PPR02	PPR01	PPR00	LVIPR0	WDTIPR0

Address: FFFECH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	PPR15	PPR14	PPR13	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1

Address: FFFE9H After reset:FFH R/W

Symbol	<7>	<6>	<5>	4	3	<2>	<1>	<0>
PR00H	SRPR00	TMPR000	STPR00 CSIPR000 IICPR000	1	1	SREPR02	SRPR02	STPR02 CSIPR020 IICPR020

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	4	3	<2>	<1>	<0>
PR10H	SRPR10	TMPR100	STPR10 CSIPR100 IICPR100	1	1	SREPR12	SRPR12	STPR12 CSIPR120 IICPR120

Address: FFFEAH After reset: FFH R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PR01L	TMPR001	1	RTITPR0	IICAPR00	SREPR01 TMPR003H	SRPR01	CSIPR010 IICPR010 STPR01	SREPR00 TMPR001H

Figure 21 - 7 Format of Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L) (2/3)

Address: FFFEEH After reset: FFH R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	TMPR101	1	RTITPR1	IICAPR10	SREPR11 TMPR103H	SRPR11	CSIPR110 IICPR110 STPR11	SREPR10 TMPR101H

Address: FFFEBH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01H	SRPR03	CSIPR030 IICPR030 STPR03	KRPR0	TMKAPR0	RTCPR0	ADPR0	TMPR003	TMPR002

Address: FFFEFH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11H	SRPR13	CSIPR130 IICPR130 STPR13	KRPR1	TMKAPR1	RTCPR1	ADPR1	TMPR103	TMPR102

Address: FFFD8H After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	<3>	<2>	<1>	0
PR02L	CMPPR001	CMPPR00	1	PPR07	PPR06	TMPR005	TMPR004	1

Address: FFFDCH After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	<3>	<2>	<1>	0
PR12L	CMPPR11	CMPPR10	1	PPR17	PPR16	TMPR105	TMPR104	1

Address: FFFD9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02H	FLPR0	TKB2PR01	TKB2PR00	SREPR03	RSUPR0	USBPR0	TMPR007	TMPR006

Address: FFFDDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12H	FLPR1	TKB2PR11	TKB2PR10	SREPR13	RSUPR1	USBPR1	TMPR107	TMPR106

Figure 21 - 8 Format of Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L) (3/3)

Address: FFFDAH After reset: FFH R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
PR03L	1	1	1	1	1	FIFOPR01	FIFOPR00	TKB2PR02

Address: FFFDEH After reset: FFH R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
PR13L	1	1	1	1	1	FIFOPR11	FIFOPR10	TKB2PR12

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 Tables 21 - 5 to 21 - 8. Be sure to set bits that are not available to the initial value.

21.3.4 External interrupt rising edge enable registers (EGP0), external interrupt falling edge enable registers (EGN0)

These registers specify the valid edge for INTP0 to INTP7.

The EGP0 and EGN0 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 21 - 9 Format of External Interrupt Rising Edge Enable Registers (EGP0) and External Interrupt Falling Edge Enable Registers (EGN0)

Address: FFF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 7)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 21 - 9 shows the Ports Corresponding to EGPn and EGNn Bits.

Table 21 - 9 Ports Corresponding to EGPn and EGNn Bits

Detection Enable Bit		Interrupt Request Signal
EGP0	EGN0	INTP0
EGP1	EGN1	INTP1
EGP2	EGN2	INTP2
EGP3	EGN3	INTP3
EGP4	EGN4	INTP4
EGP5	EGN5	INTP5
EGP6	EGN6	INTP6
EGP7	EGN7	INTP7

Caution When the input port pins used for the external interrupt functions are switched to the output mode, the INTPn interrupt might be generated upon detection of a valid edge.
When switching the input port pins to the output mode, set the port mode register (PMxx) to 0 after disabling the edge detection (by setting EGPn and EGNn to 0).

Remark 1. For edge detection port, see **2.1 Port Function**.
Remark 2. n = 0 to 7

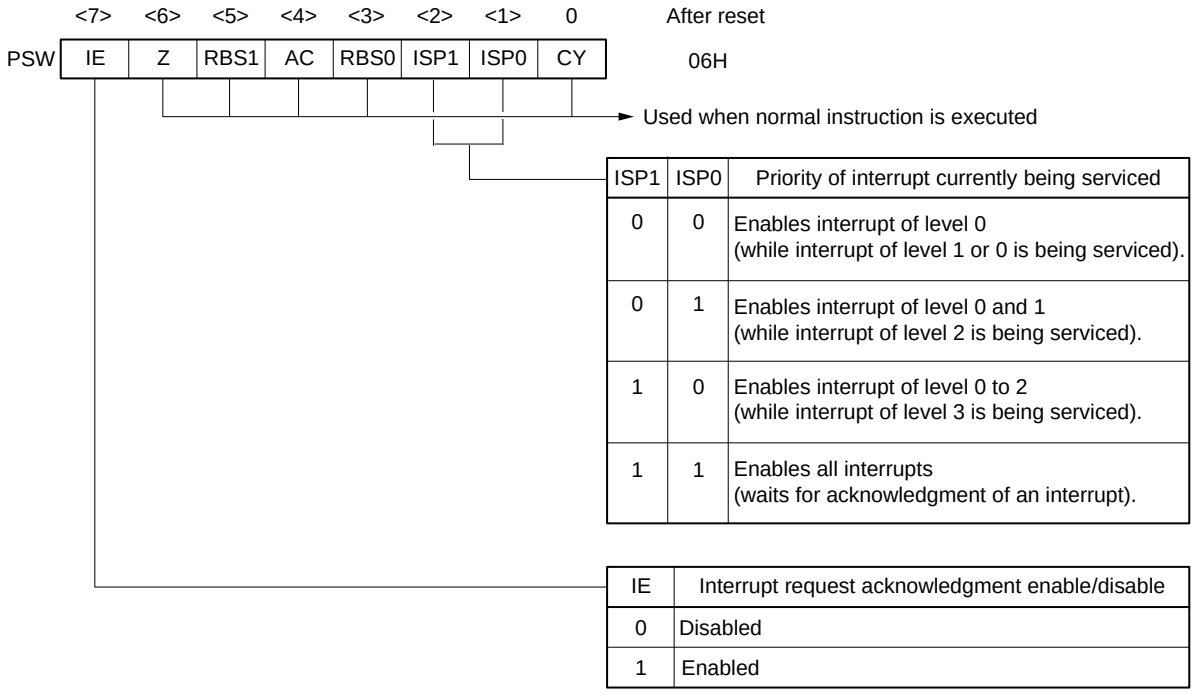
21.3.5 Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for an interrupt request. The IE flag that sets maskable interrupt enable/disable and the ISP0 and ISP1 flags that controls multiple interrupt servicing are mapped to the PSW.

Besides 8-bit read/write, this register can carry out operations using bit manipulation instructions and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged, if the BRK instruction is executed, the contents of the PSW are automatically saved into a stack and the IE flag is reset to 0. Upon acknowledgment of a maskable interrupt request, if the value of the priority specification flag register of the acknowledged interrupt is not 00, its value minus 1 is transferred to the ISP0 and ISP1 flags. The PSW contents are also saved into the stack with the PUSH PSW instruction. They are restored from the stack with the RETI, RETB, and POP PSW instructions.

Reset signal generation sets PSW to 06H.

Figure 21 - 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

21.4 Interrupt Servicing Operations

21.4.1 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 21 - 10 below.

For the interrupt request acknowledgment timing, see **Figures 21 - 12** and **21 - 13**.

Table 21 - 10 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.

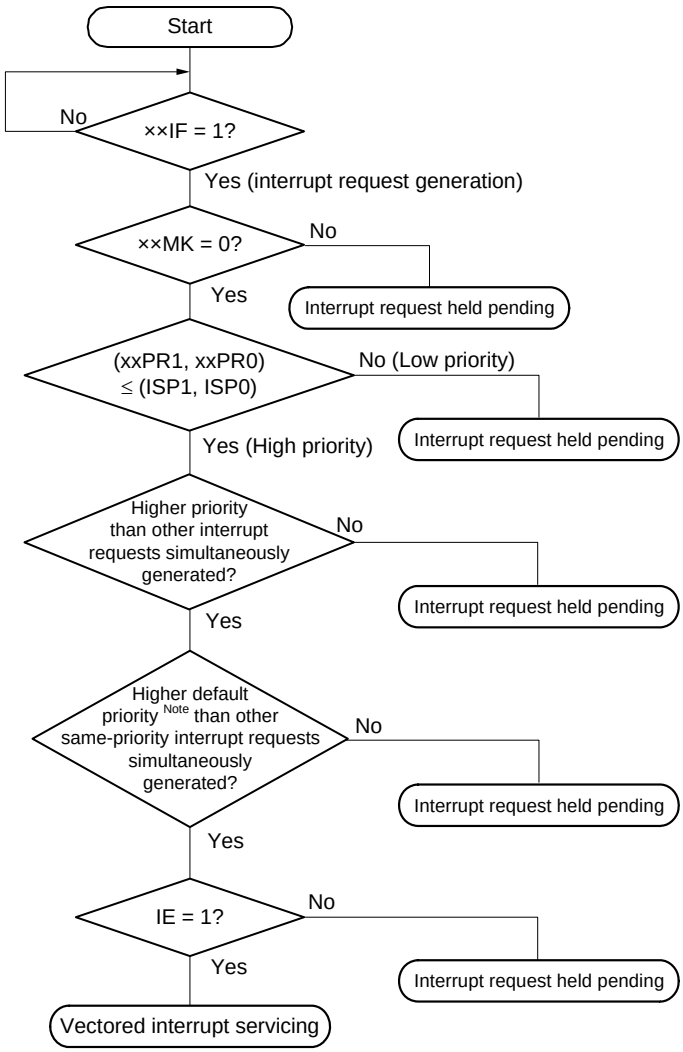
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 21 - 11 shows the Interrupt Request Acknowledgment Processing Algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset (0), and the contents of the priority specification flag corresponding to the acknowledged interrupt are transferred to the ISP1 and ISP0 flags. The vector table data determined for each interrupt request is the loaded into the PC and branched.

Restoring from an interrupt is possible by using the RETI instruction.

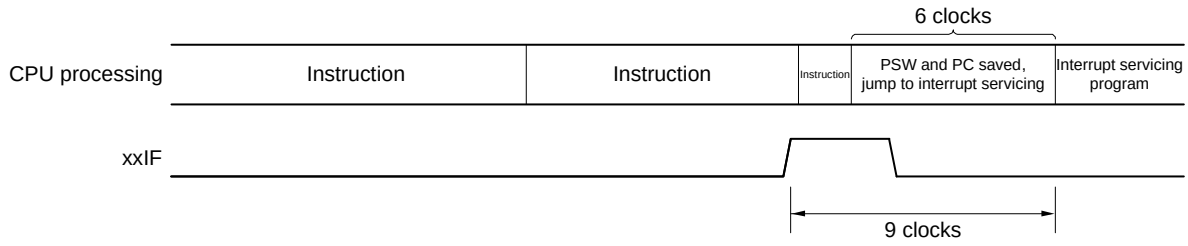
Figure 21 - 11 Interrupt Request Acknowledgment Processing Algorithm



xxIF: Interrupt request flag
xxMK: Interrupt mask flag
xxPR0: Priority specification flag 0
xxPR1: Priority specification flag 1
IE: Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)
ISP0, ISP1: Flag that indicates the priority level of the interrupt currently being serviced (see **Figure 21 - 10**)

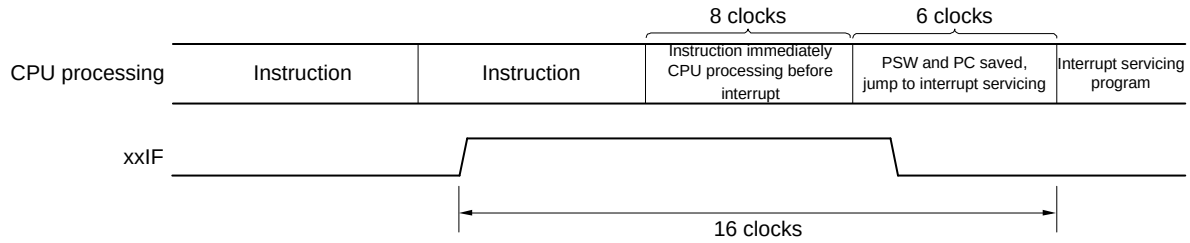
Note For the default priority, refer to **Tables 21 - 1** to **21 - 4 Interrupt Source List**.

Figure 21 - 12 Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/fCLK (fCLK: CPU clock)

Figure 21 - 13 Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/fCLK (fCLK: CPU clock)

21.4.2 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 The RETI instruction cannot be used for restoring from the software interrupt.

21.4.3 Multiple interrupt servicing

Multiple interrupt servicing occurs when another interrupt request is acknowledged during execution of an interrupt.

Multiple interrupt servicing does not occur unless the interrupt request acknowledgment enabled state is selected (IE = 1). When an interrupt request is acknowledged, interrupt request acknowledgment becomes disabled (IE = 0). Therefore, to enable multiple interrupt servicing, it is necessary to set (1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledgment.

Moreover, even if interrupts are enabled, multiple interrupt servicing may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for multiple interrupt servicing.

In the interrupt enabled state, if an interrupt request with a priority equal to or higher than that of the interrupt currently being serviced is generated, it is acknowledged for multiple interrupt servicing. If an interrupt with a priority equal to or lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for multiple interrupt servicing. However, when setting the IE flag to 1 during the interruption at level 0, other level 0 interruptions can be allowed.

Interrupt requests that are not enabled because interrupts are in the interrupt disabled state or because they have a lower priority are held pending. When servicing of the current interrupt ends, the pending interrupt request is acknowledged following execution of at least one main processing instruction execution.

Table 21 - 11 shows Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing and Figures 21 - 14 and 21 - 15 show multiple interrupt servicing examples.

Table 21 - 11 Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing

Multiple Interrupt Request / Interrupt Being Serviced		Maskable Interrupt Request								Software Interrupt Request
		Priority Level 0 (PR = 00)		Priority Level 1 (PR = 01)		Priority Level 2 (PR = 10)		Priority Level 3 (PR = 11)		
		IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	
Maskable interrupt	ISP1 = 0 ISP0 = 0	√	×	×	×	×	×	×	×	√
	ISP1 = 0 ISP0 = 1	√	×	√	×	×	×	×	×	√
	ISP1 = 1 ISP0 = 0	√	×	√	×	√	×	×	×	√
	ISP1 = 1 ISP0 = 1	√	×	√	×	√	×	√	×	√
Software interrupt		√	×	√	×	√	×	√	×	√

Remark 1. √: Multiple interrupt servicing enabled

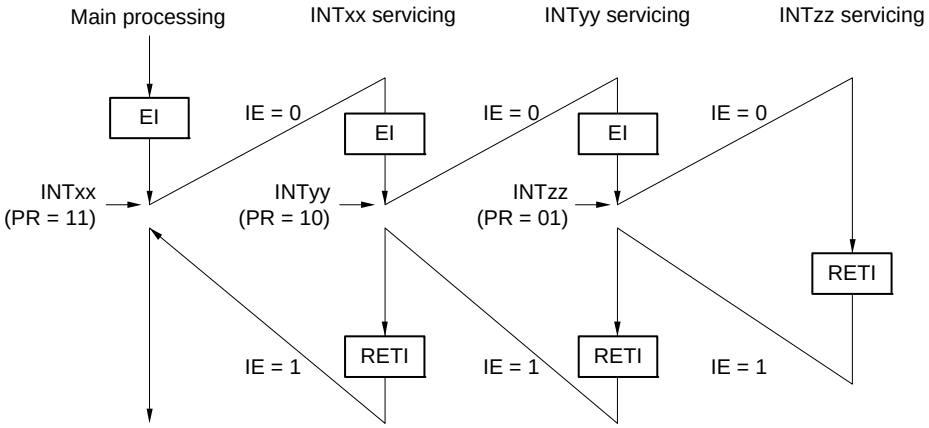
Remark 2. ×: Multiple interrupt servicing disabled

Remark 3. ISP0, ISP1, and IE are flags contained in the PSW.
ISP1 = 0, ISP0 = 0: An interrupt of level 1 or level 0 is being serviced.
ISP1 = 0, ISP0 = 1: An interrupt of level 2 is being serviced.
ISP1 = 1, ISP0 = 0: An interrupt of level 3 is being serviced.
ISP1 = 1, ISP0 = 1: Wait for An interrupt acknowledgment (all interrupts are enabled).
IE = 0: Interrupt request acknowledgment is disabled.
IE = 1: Interrupt request acknowledgment is enabled.

Remark 4. PR is a flag contained in the PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, and PR13L 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)

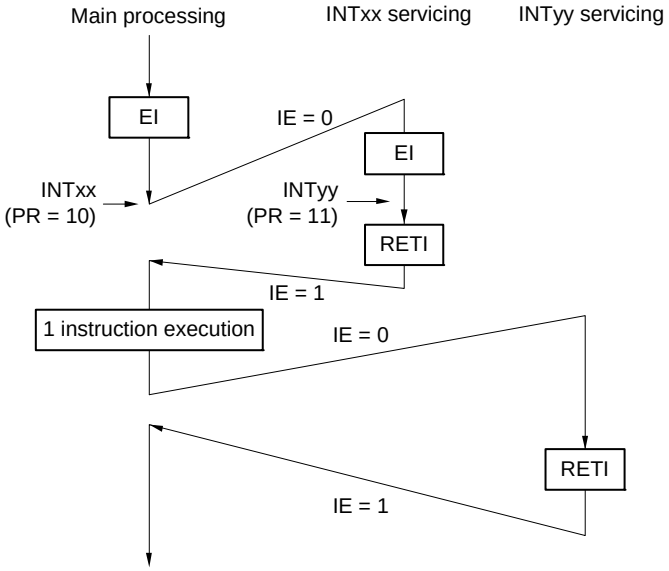
Figure 21 - 14 Examples of Multiple Interrupt Servicing (1/2)

Example 1. Multiple interrupt servicing occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgment.

Example 2. Multiple interrupt servicing does not occur due to priority control

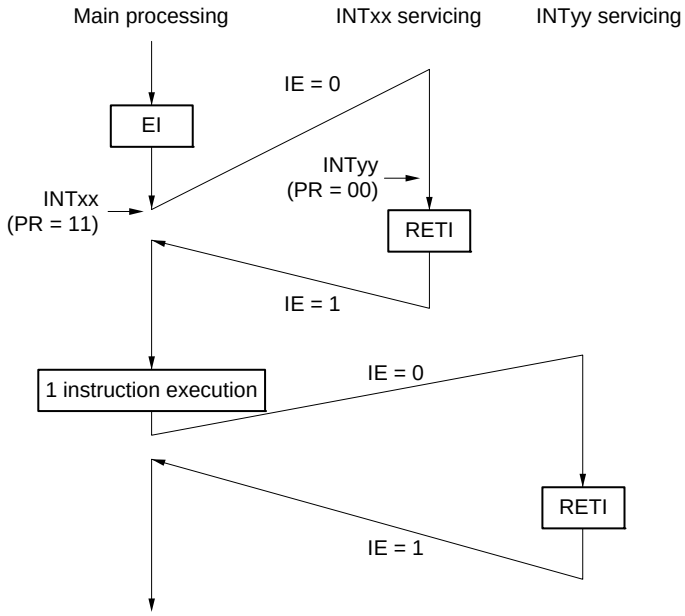


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with xxPR1x = 0, xxPR0x = 0 (higher priority level)
PR = 01: Specify level 1 with xxPR1x = 0, xxPR0x = 1
PR = 10: Specify level 2 with xxPR1x = 1, xxPR0x = 0
PR = 11: Specify level 3 with xxPR1x = 1, xxPR0x = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

Figure 21 - 15 Examples of Multiple Interrupt Servicing (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled



Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with xxPR1x = 0, xxPR0x = 0 (higher priority level)
PR = 01: Specify level 1 with xxPR1x = 0, xxPR0x = 1
PR = 10: Specify level 2 with xxPR1x = 1, xxPR0x = 0
PR = 11: Specify level 3 with xxPR1x = 1, xxPR0x = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

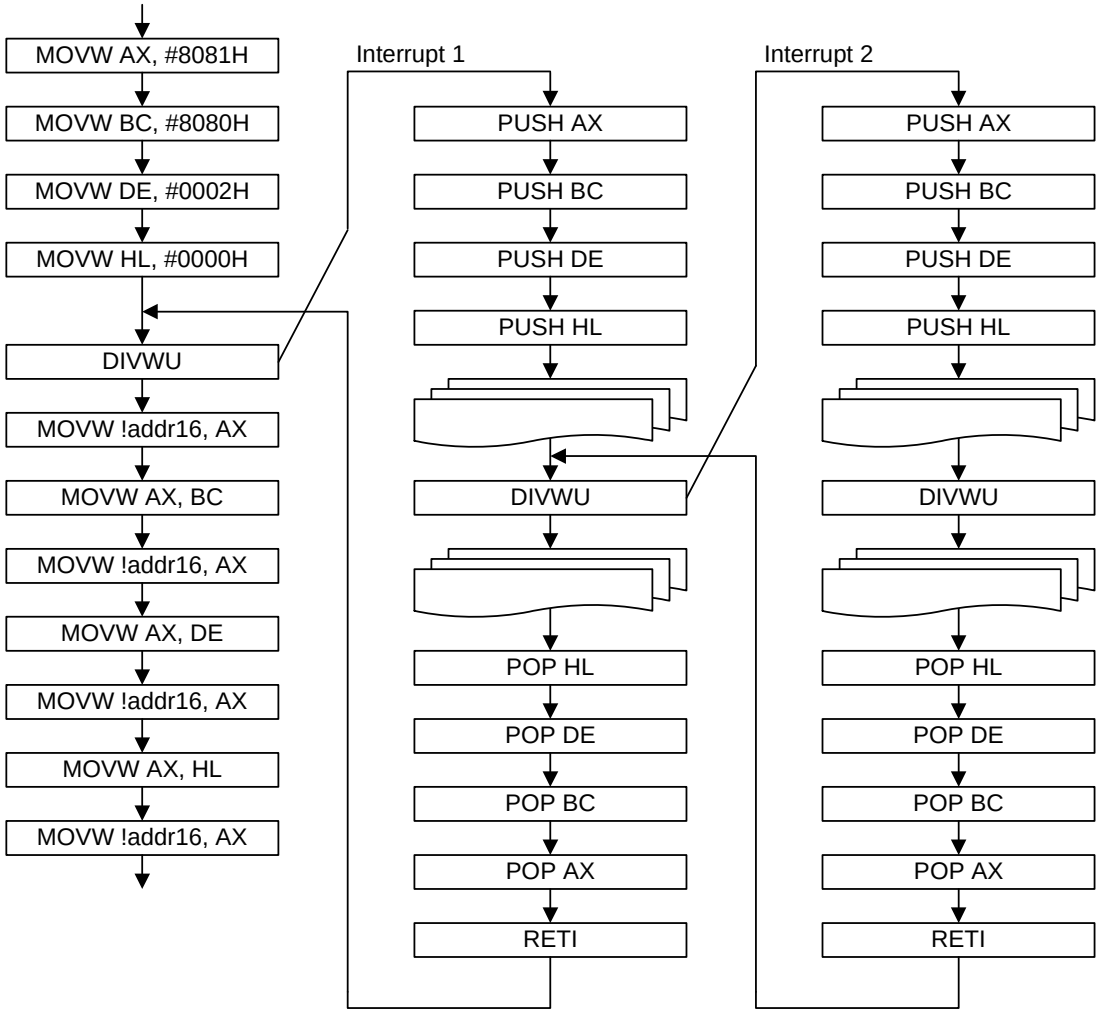
21.4.4 Interrupt servicing during division instruction

The RL78/L1C handles interrupts during the DIVHU/DIVWU instruction in order to enhance the interrupt response when a division instruction is executed.

- When an interrupt is generated while the DIVHU/DIVWU instruction is executed, the instruction is suspended
- After the instruction is suspended, the PC indicates the next instruction after DIVHU/DIVWU
- An interrupt is generated by the next instruction
- PC-3 is stacked to execute the DIVHU/DIVWU instruction again

Normal interrupt	Interrupts while Executing DIVHU/DIVWU Instruction
(SP-1) ← PSW	(SP-1) ← PSW
(SP-2) ← $(PC)_S$	(SP-2) ← $(PC-3)_S$
(SP-3) ← $(PC)_H$	(SP-3) ← $(PC-3)_H$
(SP-4) ← $(PC)_L$	(SP-4) ← $(PC-3)_L$
PC_S ← 0000	PC_S ← 0000
PC_H ← (Vector)	PC_H ← (Vector)
PC_L ← (Vector)	PC_L ← (Vector)
SP ← SP-4	SP ← SP-4
IE ← 0	IE ← 0

The AX, BC, DE, and HL registers are used for DIVHU/DIVWU. Use these registers by stacking them for interrupt servicing.



Caution Disable interrupts when executing the DIVHU or DIVWU instruction in an interrupt servicing routine.
Alternatively, unless they are executed in the RAM area, note that execution of a DIVHU or DIVWU instruction is possible even with interrupts enabled as long as a NOP instruction is added immediately after the DIVHU or DIVWU instruction in the assembly language source code. The following compilers automatically add a NOP instruction immediately after any DIVHU or DIVWU instruction output during the build process.

- **V. 1.71 and later versions of the CA78K0R (Renesas Electronics compiler), for both C and assembly language source code**
- **Service pack 1.40.6 and later versions of the EWRL78 (IAR compiler), for C language source code**
- **GNURL78 (KPIT compiler), for C language source code**

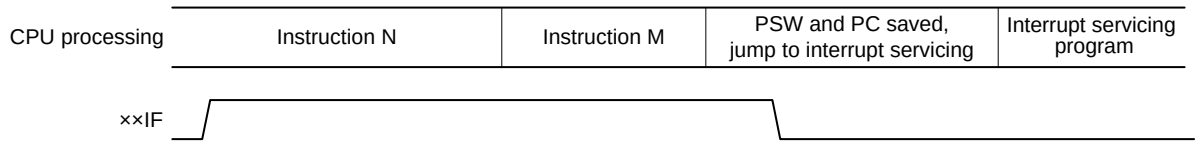
21.4.5 Interrupt request hold

There are instructions where, even if an interrupt request is issued while the instructions are being executed, interrupt request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, $addr20
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- SKH
- SKNH
- MULHU
- MULH
- MACHU
- MACH
- Write instructions for the IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, and PR13L registers

Figure 21 - 16 shows the timing at which interrupt requests are held pending.

Figure 21 - 16 Interrupt Request Hold



Remark 1. Instruction N: Interrupt request hold instruction
Remark 2. Instruction M: Instruction other than interrupt request hold instruction

CHAPTER 22 KEY INTERRUPT FUNCTION

22.1 Functions of Key Interrupt

A key interrupt (INTKR) can be generated by inputting a rising edge/falling edge to the key interrupt input pins (KR0 to KR7).

Table 22 - 1 Assignment of Key Interrupt Detection Pins

Key interrupt pins	Key return mode register (KRM0)
KR0	KRM00
KR1	KRM01
KR2	KRM02
KR3	KRM03
KR4	KRM04
KR5	KRM05
KR6	KRM06
KR7	KRM07

Remark KR1 to KR3: 80/85-pin

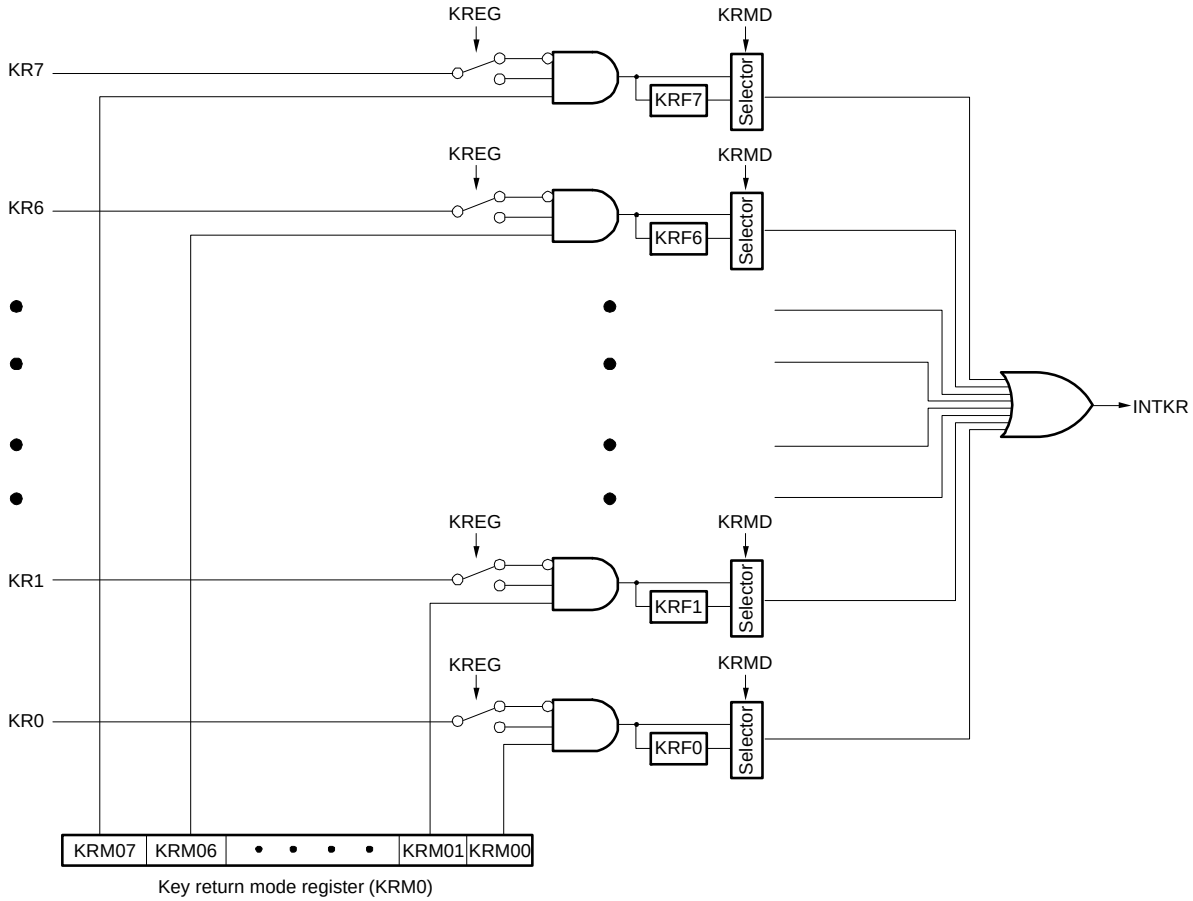
22.2 Configuration of Key Interrupt

The key interrupt includes the following hardware.

Table 22 - 2 Configuration of Key Interrupt

Item	Configuration
Input	KR0 to KR7
Control register	Key return control register (KRCTL) Key return mode register (KRM0) Key return flag register (KRF) Port mode register 7 (PM7)

Figure 22 - 1 Block Diagram of Key Interrupt



22.3 Registers Controlling Key Interrupt

The key interrupt function is controlled by the following five registers:

- Key return control register (KRCTL)
- Key return mode register (KRM0)
- Key return flag register (KRF)
- Port mode register 7 (PM7)

22.3.1 Key return control register (KRCTL)

This register controls the usage of the key return flags (KRF0 to KRF7) and sets the detection edge.

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

Reset signal generation clears this register to 00H.

Figure 22 - 2 Format of Key return control register (KRCTL)

Address: FFF34H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRCTL	KRMD	0	0	0	0	0	0	KREG

KRMD	Usage of key return flags (KRF0 to KRF7)
0	Does not use key return flags
1	Uses key return flags

KREG	Selection of detection edge (KR0 to KR7)
0	Falling edge
1	Rising edge

22.3.2 Key return mode register (KRM0)

This register sets the key interrupt mode.

The KRM0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 22 - 3 Format of Key return mode register (KRM0)

Address: FFF37H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRM0	KRM07	KRM06	KRM05	KRM04	KRM03	KRM02	KRM01	KRM00

KRM0n	Key interrupt mode control (n = 0 to 7)
0	Does not detect key interrupt signal
1	Detects key interrupt signal

Caution 1. When the bits to be used among the KRM00 to KRM07 bits are set to 1, pull up the relevant input pins to V_{DD} by an external resistor.

Caution 2. An interrupt will be generated if the target bit of the KRM0 register is changed. To ignore this interrupt, set the KRM0 register after disabling interrupt servicing by using the interrupt mask flag. Afterward, clear the interrupt request flag and enable interrupt servicing after waiting for the key interrupt input low-level width (see 34.4 or 35.4 AC Characteristics).

Caution 3. The bits not used in the key interrupt mode can be used as normal ports.

22.3.3 Key return flag register (KRF)

This register controls the key return flags (KRF0 to KRF7).

The KRF register can be set by a 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 22 - 4 Format of Key return flag register (KRF)

Address: FFF35H After reset: 00H R/W Note

Symbol	7	6	5	4	3	2	1	0
KRF	KRF7	KRF6	KRF5	KRF4	KRF3	KRF2	KRF1	KRF0

KRFn	Key interrupt flag (n = 0 to 7)
0	No key interrupt signal has been detected.
1	A key interrupt signal has been detected.

Note Writing to 1 is invalid. To clear the KRFn bit, write 0 to the corresponding bit and 1 to the other bits using an 8-bit memory manipulation instruction.

Caution When KRMD = 0, set to KRFn = 1 is prohibited.

22.3.4 Port mode register 7 (PM7)

When port 7 is used as the key interrupt input pins (KR0 to KR7), set the PM7n bit to 1. The output latches of P7n at this time may be 0 or 1.

The PM7 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to FFH.

Use of an on-chip pull-up resistor can be specified in 1-bit units by the pull-up resistor option register 7 (PU7).

Figure 22 - 5 Format of Port mode register 7 (PM7)

Address: FFF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	I/O mode selection for P7n/KRn pin (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

CHAPTER 23 STANDBY FUNCTION

23.1 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 CSIp or UARTq data reception, an A/D conversion request by the timer trigger signal (the interrupt request signal (INTRTC/INTIT) or ELC event input), and DTC start source, the STOP mode is exited, the CSIp or UARTq data is received without operating the CPU, A/D conversion is performed, and DTC start source. This can only be specified when the high-speed on-chip oscillator is selected for the CPU/peripheral hardware clock (f_{CLK}).

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

Caution 1. The STOP mode can be used only when the CPU is operating on the main system clock. The STOP mode cannot be set 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.

Caution 2. When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction (except SNOOZE mode setting unit).

Caution 3. When using CSIp, UARTq, or the A/D converter in the SNOOZE mode, set up serial standby control register m (SSCm) and A/D converter mode register 2 (ADM2) before switching to the STOP mode. For details, see 15.3 Registers Controlling Serial Array Unit and 12.3 Registers Controlling A/D Converter.

Caution 4. The following sequence is recommended for operating current reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS) and bit 0 (ADCE) of A/D converter mode register 0 (ADM0) to 0 to stop the A/D conversion operation, and then execute the STOP instruction.

Caution 5. It can be selected by the option byte whether the low-speed on-chip oscillator continues oscillating or stops in the HALT or STOP mode. For details, see CHAPTER 29 OPTION BYTE.

Remark p = 00, 20; q = 0, 2; m = 0, 1

23.2 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** and **CHAPTER 15 SERIAL ARRAY UNIT**.

23.3 Standby Function Operation

23.3.1 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 23 - 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_{HOCO})	When CPU is Operating on X1 Clock (fx)	When CPU is Operating on External Main System Clock (f_{EX})	When CPU is Operating on PLL clock (f_{PLL})
System clock			Clock supply to the CPU is stopped			
	Main system clock	f_{HOCO}	Operation continues (cannot be stopped)	Operation disabled		
		fx	Operation disabled	Operation continues (cannot be stopped)	Cannot operate	Cannot be stopped while the clock is supplied to the PLL
		f_{EX}		Cannot operate	Operation continues (cannot be stopped)	Cannot be stopped while the clock is supplied to the PLL
		f_{PLL}	Operation disabled	Operation disabled	Operation disabled	Operation continues (cannot be stopped)
	Subsystem clock	f_{XT}	Status before HALT mode was set is retained			
		f_{EXT}				
	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			Operation stopped (operable when DMA is executed)			
Port (latch)			Status before HALT mode was set is retained			
Timer array unit			Operable			
Timer KB2						
Real-time clock 2						
12-bit Interval timer						
Watchdog timer			See **CHAPTER 11 WATCHDOG TIMER**.			
Clock output/buzzer output			Operable			
A/D converter						
D/A converter						
Comparator						
Serial array unit (SAU)						
Serial interface (IICA)						
USB			Operable	Operation disabled	Operation disabled	Operable
LCD driver/controller			Operable (However, this depends on the status of the clock selected as the LCD source clock: operation is possible if the selected clock is operating, but operation will stop if the selected clock is stopped.)			
DTC			Operable			
ELC			Operable function blocks can be linked			
Power-on-reset function			Operable			
Voltage detection function						
External interrupt						
Key interrupt function						
CRC operation function		High-speed CRC				
		General-purpose CRC	Operation stopped (Operable while in the DTC is executed)			
RAM parity error detection function						
RAM guard function						
SFR guard function						
Illegal-memory access detection function						

Remark Operation stopped: Operation is automatically stopped before switching to the HALT mode.
Operation disabled: Operation is stopped before switching to the HALT mode.
f_{HOCO}: High-speed on-chip oscillator clock f_{EX}: External main system clock
f_{IL}: Low-speed on-chip oscillator clock f_{XT}: XT1 clock
fx: X1 clock f_{EXT}: External subsystem clock

Table 23 - 1 Operating Statuses in HALT Mode (2/2)

Item \ HALT Mode Setting			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_{EXT})
System clock			Clock supply to the CPU is stopped	
	Main system clock	f_{HOCO}	Operation disabled	
		f_X		
		f_{EX}		
		f_{PLL}		
	Subsystem clock	f_{XT}	Operation continues (cannot be stopped)	Cannot operate
		f_{EXT}	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: Setting prohibited • 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			Operation stopped (operable when DMA is executed)	
Port (latch)			Status before HALT mode was set is retained	
Timer array unit			Operable (Operation is disabled while in the low consumption RTC mode (when the RTCLPC bit of the OSMC register is 1))	
Timer KB2				
Real-time clock 2			Operable	
12-bit Interval timer				
Watchdog timer			See **CHAPTER 11 WATCHDOG TIMER**.	
Clock output/buzzer output			Operable (Operation is disabled while in the low consumption RTC mode (when the RTCLPC bit of the OSMC register is 1))	
A/D converter			Operation disabled	
D/A converter			Operable (Operation is disabled while in the low consumption RTC mode (when the RTCLPC bit of the OSMC register is 1))	
Comparator				
Serial array unit (SAU)			Operable (Operation is disabled while in the low consumption RTC mode (when the RTCLPC bit of the OSMC register is 1))	
Serial interface (IICA)			Operation disabled	
USB				
LCD driver/controller			Operable (However, this depends on the status of the clock selected as the LCD source clock: operation is possible if the selected clock is operating, but operation will stop if the selected clock is stopped.)	
DTC			Operable (Operation is disabled while in the low consumption RTC mode (when the RTCLPC bit of the OSMC register is 1))	
ELC			Operable function blocks can be linked	
Power-on-reset function			Operable	
Voltage detection function				
External interrupt				
Key interrupt function				
CRC operation function		High-speed CRC	Operation disabled	
		General-purpose CRC	Operation stopped (Operable while in the DTC is executed)	
RAM parity error detection function				
RAM guard function				
SFR guard function				
Illegal-memory access detection function				

Remark Operation stopped:Operation is automatically stopped before switching to the HALT mode.
Operation disabled: Operation is stopped before switching to the HALT mode.
f_{HOCO}: High-speed on-chip oscillator clock f_{EX}: External main system clock
f_{IL}: Low-speed on-chip oscillator clock f_{XT}: XT1 clock
f_X: X1 clock f_{EXT}: External subsystem clock

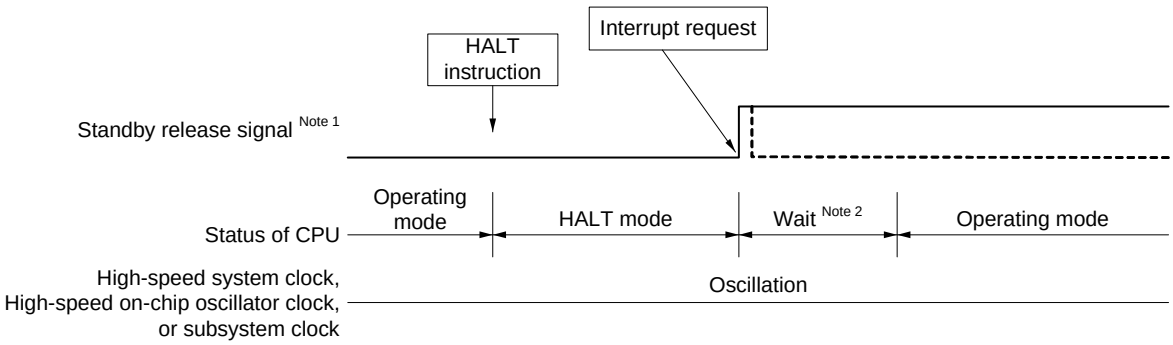
(2) HALT mode release

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 23 - 1 HALT Mode Release by Interrupt Request Generation



Note 1. For details of the standby release signal, see **Figure 21 - 1**.

Note 2. Wait time for HALT mode release

- When vectored interrupt servicing is carried out
  - Main system clock: 15 to 16 clocks
  - Subsystem clock (RTCLPC = 0): 10 to 11 clocks
  - Subsystem clock (RTCLPC = 1): 11 to 12 clocks
- When vectored interrupt servicing is not carried out
  - Main system clock: 9 to 10 clocks
  - Subsystem clock (RTCLPC = 0): 4 to 5 clocks
  - Subsystem clock (RTCLPC = 1): 5 to 6 clocks

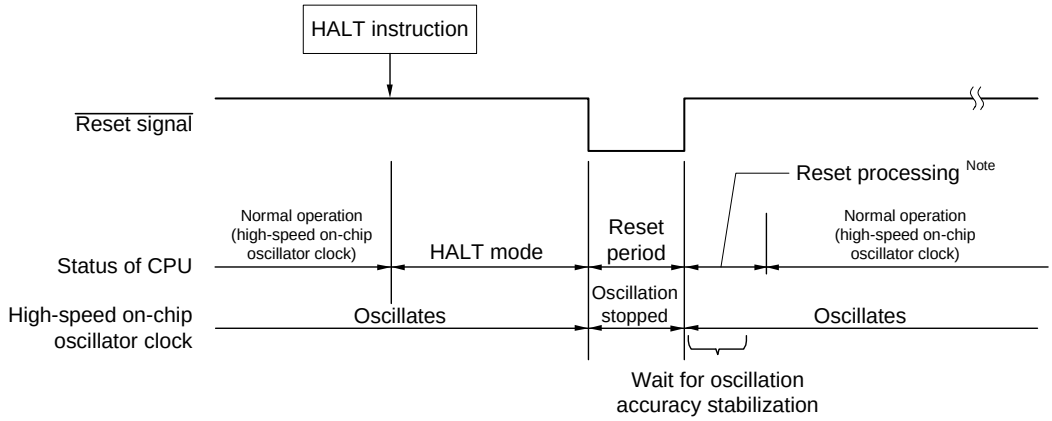
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

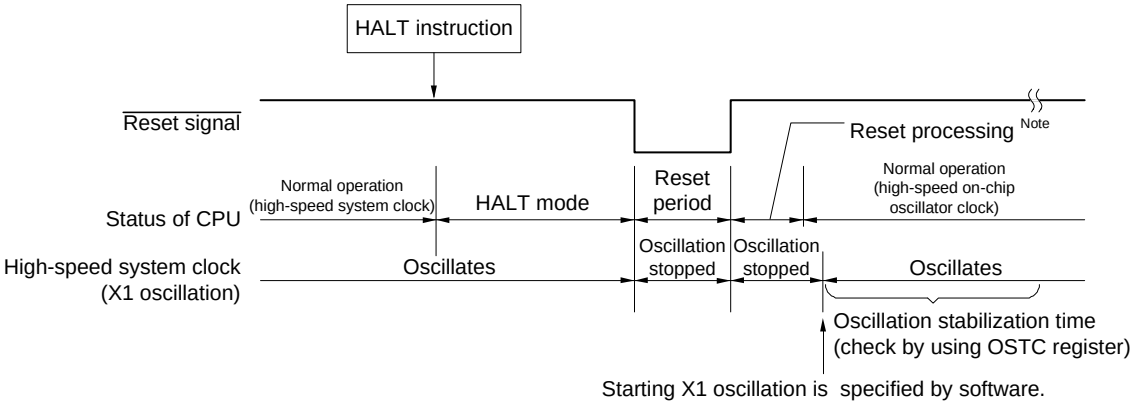
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 23 - 2 HALT Mode Release by Reset (1/2)

(1) When high-speed on-chip oscillator clock is used as CPU clock



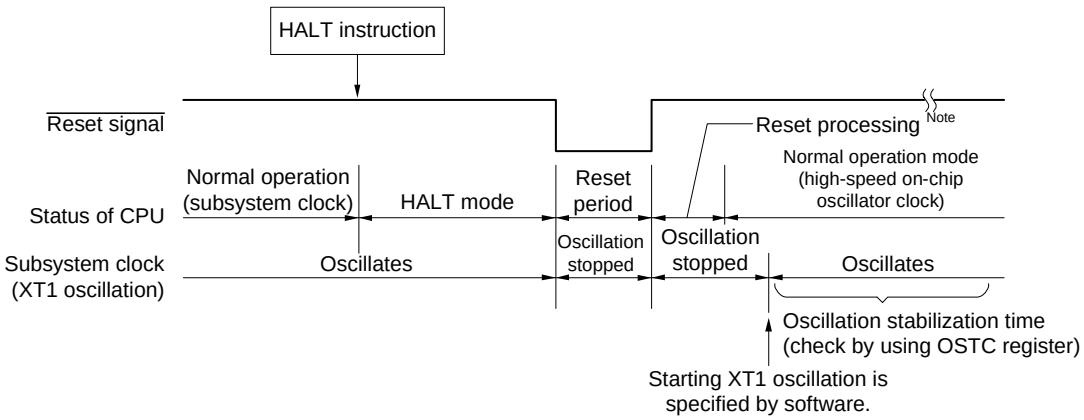
(2) When high-speed system clock is used as CPU clock



Note For the reset processing time, see **CHAPTER 24 RESET FUNCTION**. For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 25 POWER-ON-RESET CIRCUIT**.

Figure 23 - 3 HALT Mode Release by Reset (2/2)

(3) When subsystem clock is used as CPU clock



Note For the reset processing time, see **CHAPTER 24 RESET FUNCTION**. For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 25 POWER-ON-RESET CIRCUIT**.

23.3.2 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 23 - 2 Operating Statuses in STOP Mode

Item \ STOP Mode Setting			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_{HOCO})	When CPU is Operating on X1 Clock (fx)	When CPU is Operating on External Main System Clock (f_{EX})	When CPU is Operating on PLL clock (f_{PLL})
System clock			Clock supply to the CPU is stopped			
	Main system clock	f_{HOCO}	Stopped			
		f_X				
		f_{EX}				
		f_{PLL}	Operation disabled			
	Subsystem clock	f_{XT}	Status before STOP mode was set is retained			
		f_{EXT}				
	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			Operation stopped			
RAM			Operation stopped			
Port (latch)			Status before STOP mode was set is retained			
Timer array unit			Operation disabled			
Timer KB2						
Real-time clock 2			Operable			
12-bit Interval timer						
Watchdog timer			See **CHAPTER 11 WATCHDOG TIMER**.			
Clock output/buzzer output			Operable only when subsystem clock is selected as the count clock (when low-consumption RTC mode (set RTCLPC bit of OSMC register to 1), operation disabled)			
A/D converter			Wakeup operation is enabled (switching to the SNOOZE mode)			
D/A converter			Operable (status before STOP mode was set is retained)			
Comparator			Operable (when digital filter is not used)			
Serial array unit (SAU)			Wakeup operation is enabled only for CSIp and UARTq (switching to the SNOOZE mode) Operation is disabled for anything other than CSIp and UARTq			
Serial interface (IICA)			Wakeup by address match operable			
USB			Operation disabled			
LCD driver/controller			Operable (However, this depends on the status of the clock selected as the LCD source clock: operation is possible if the selected clock is operating, but operation will stop if the selected clock is stopped.)			
DTC			Operable (Status before transitioning to SNOOZE is retained)			
ELC			Operable function blocks can be linked			
Power-on-reset function			Operable			
Voltage detection function						
External interrupt						
Key interrupt function						
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						

(**Cautions** and **Remarks** are listed on the next page.)

Caution To use the peripheral hardware that stops operation in the STOP mode, and the peripheral hardware for which the clock that stops oscillating in the STOP mode after the STOP mode is released, restart the peripheral hardware.

Remark 1. Operation stopped: Operation is automatically stopped before switching to the STOP mode.
Operation disabled: Operation is stopped before switching to the STOP mode.
f_{HOCO}: High-speed on-chip oscillator clock
f_{EX}: External main system clock
f_{IL}: Low-speed on-chip oscillator clock
f_{XT}: XT1 clock
f_X: X1 clock
f_{EXT}: External subsystem clock

Remark 2. p = 00, 20; q = 0, 2

(2) STOP mode release

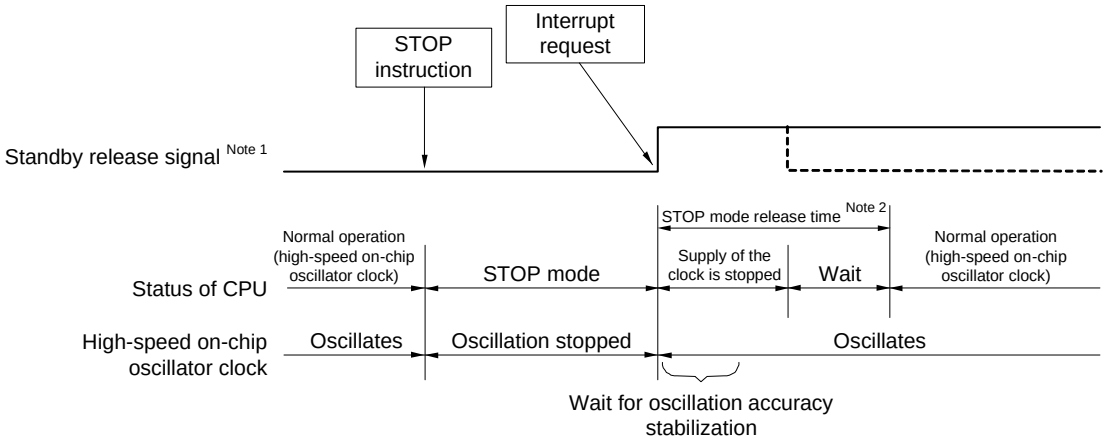
The STOP mode can be released by the following two sources.

(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 23 - 4 STOP Mode Release by Interrupt Request Generation (1/2)

(1) When high-speed on-chip oscillator clock is used as CPU clock



Note 1. For details of the standby release signal, see **Figure 21 - 1**.

Note 2. STOP mode release time

Supply of the clock is stopped:

- When FRQSEL4 = 0: 18 μs to 65 μs
- When FRQSEL4 = 1: 18 μs to 75 μs

Wait:

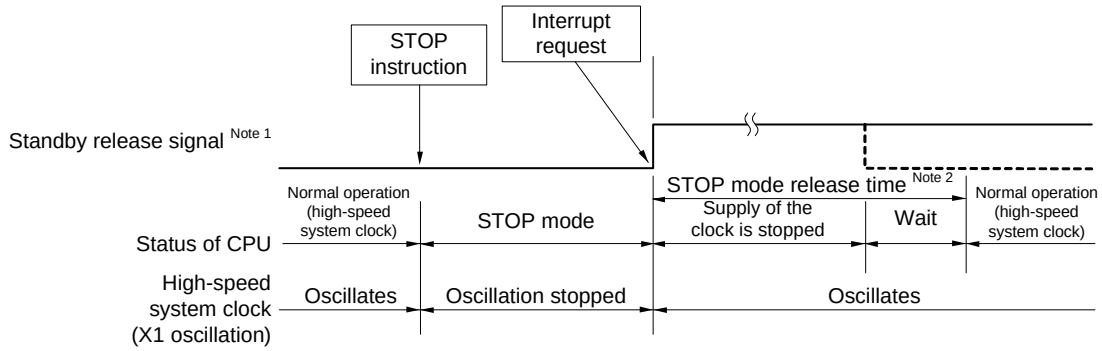
- When vectored interrupt servicing is carried out: 7 clocks
- When vectored interrupt servicing is not carried out: 1 clock

Remark 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

Remark 2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

Figure 23 - 5 STOP Mode Release by Interrupt Request Generation (2/2)

(2) When high-speed system clock (X1 oscillation) is used as CPU clock



Note 1. For details of the standby release signal, see **Figure 21 - 1 Basic Configuration of Interrupt Function**.

Note 2. STOP mode release time

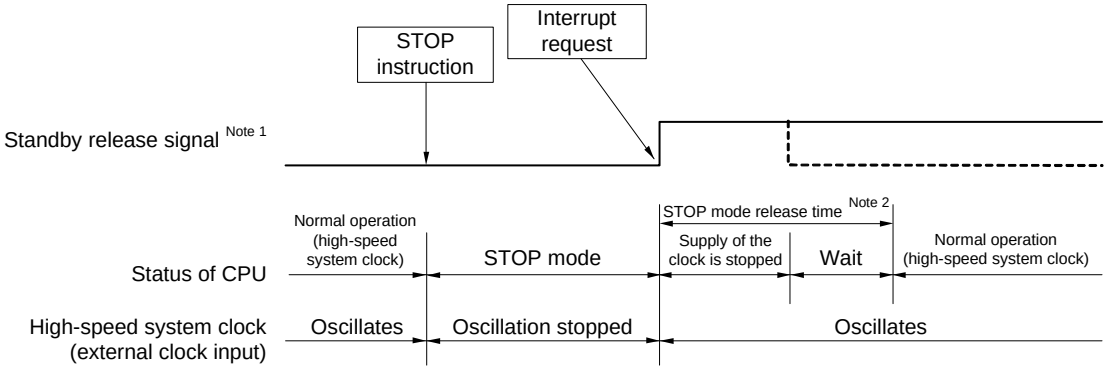
Supply of the clock is stopped:

- When FRQSEL4 = 0: 18 μs to “whichever is longer 65 μs or the oscillation stabilization time (set by OSTS)”
- When FRQSEL4 = 1: 18 μs to “whichever is longer 75 μs or the oscillation stabilization time (set by OSTS)”

Wait:

- When vectored interrupt servicing is carried out: 10 to 11 clocks
- When vectored interrupt servicing is not carried out: 4 to 5 clocks

(3) When high-speed system clock (external clock input) is used as CPU clock



Note 1. For details of the standby release signal, see **Figure 21 - 1**.

Note 2. STOP mode release time

Supply of the clock is stopped:

- When FRQSEL4 = 0: 18 μs to 65 μs
- When FRQSEL4 = 1: 18 μs to 75 μs

Wait:

- When vectored interrupt servicing is carried out: 7 clocks
- When vectored interrupt servicing is not carried out: 1 clock

Caution To reduce the oscillation stabilization time after release from the STOP mode while CPU operates based on the high-speed system clock (X1 oscillation), switch the clock to the high-speed on-chip oscillator clock temporarily before executing the STOP instruction.

Remark 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

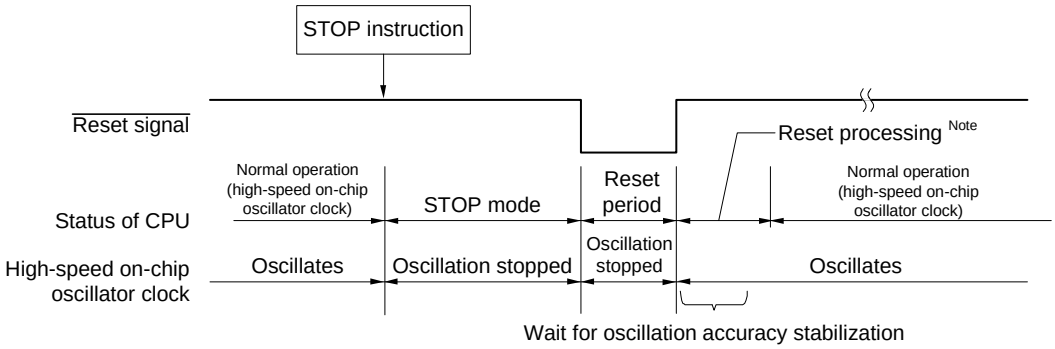
Remark 2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Release by reset signal generation

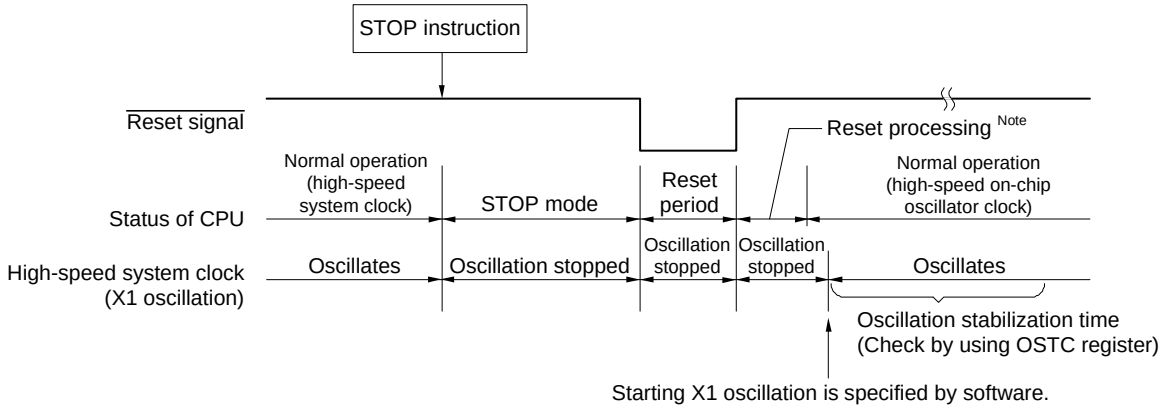
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 23 - 6 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 24 RESET FUNCTION**. For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 25 POWER-ON-RESET CIRCUIT**.

23.3.3 SNOOZE mode

(1) SNOOZE mode setting and operating statuses

The SNOOZE mode can only be specified for CSIp, UARTq, the A/D converter, and DTC. Note that this mode can only be specified if the CPU clock is the high-speed on-chip oscillator clock.

When using CSIp or UARTq in the SNOOZE mode, set up serial standby control register m (SSCm) before switching to the STOP mode. For details, see **15.3 Registers Controlling Serial Array Unit**.

When using the A/D converter in the SNOOZE mode, set up A/D converter mode register 2 (ADM2) before switching to the STOP mode. For details, see **12.3 Registers Controlling A/D Converter**.

When DTC transfer is used in SNOOZE mode, before switching to the STOP mode, allow DTC activation by interrupt to be used. During STOP mode, detecting DTC activation by interrupt enables DTC transit to SNOOZE mode, automatically. For details, see **19.3 Registers Controlling DTC**.

Remark p = 00, 20; q = 0, 2; m = 0, 1

In SNOOZE mode transition, wait status to be only following time.

Transition time from STOP mode to SNOOZE mode:

- When FRQSEL4 = 0: 18 μs to 65 μs
- When FRQSEL4 = 1: 18 μs to 75 μs

Remark Transition time from STOP mode to SNOOZE mode varies depending on the temperature conditions and the STOP mode period.

Transition time from SNOOZE mode to normal operation:

- When vectored interrupt servicing is carried out:
  HS (High-speed main) mode: “4.99 μs to 9.44 μs” + 7 clocks
  LS (Low-speed main) mode: “1.10 μs to 5.08 μs” + 7 clocks
  LV (Low-voltage main) mode: “16.58 μs to 25.40 μs” + 7 clocks

- When vectored interrupt servicing is not carried out:
  HS (High-speed main) mode: “4.99 μs to 9.44 μs” + 1 clock
  LS (Low-speed main) mode: “1.10 μs to 5.08 μs” + 1 clock
  LV (Low-voltage main) mode: “16.58 μs to 25.40 μs” + 1 clock

The operating statuses in the SNOOZE mode are shown next.

Table 23 - 3 Operating Statuses in SNOOZE Mode

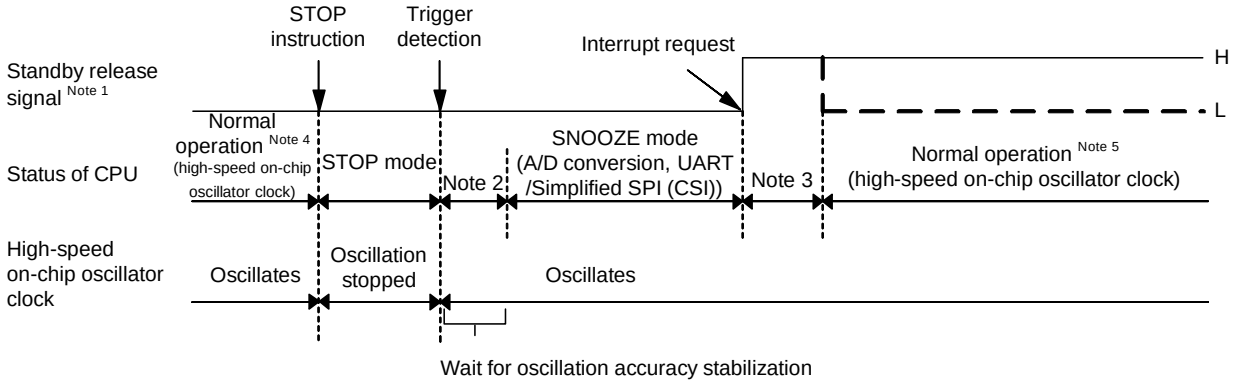
Item \ STOP Mode Setting		When Inputting CSIp/UARTq Data Reception Signal, A/D Converter Timer Trigger Signal, and Generating DTC Activation by Interrupt While in STOP Mode
		When CPU is Operating on High-speed On-chip Oscillator Clock (f_{HOCO})
System clock		Clock supply to the CPU is stopped
Main system clock	f_{HOCO}	Operation started
	f_X	Stopped
	f_{EX}	
	f_{PLL}	
Subsystem clock	f_{XT}	Use of the status while in the STOP mode continues
	f_{EXT}	
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 KB2		
Real-time clock 2		Operable
12-bit interval timer		
Watchdog timer		See **CHAPTER 11 WATCHDOG TIMER**.
Clock output/buzzer output		Operable only when subsystem clock is selected as the count clock (when low-consumption RTC mode (set RTCLPC bit of OSMC register to 1), operation disabled)
A/D converter		Operable
D/A converter		Operable (Status before SNOOZE mode was set is retained)
Comparator		Operable (when digital filter is not used)
Serial array unit (SAU)		Operable only CSIp and UARTq only. Operation disabled other than CSIp and UARTq.
Serial interface (IICA)		Operation disabled
USB		
LCD driver/controller		Operable (However, this depends on the status of the clock selected as the LCD source clock: operation is possible if the selected clock is operating, but operation will stop if the selected clock is stopped.)
DTC		Operable
ELC		Operable function blocks can be linked
Power-on-reset function		Operable
Voltage detection function		
External interrupt		
Key interrupt function		
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		

Remark 1. Operation stopped: Operation is automatically stopped before switching to the STOP mode.
Operation disabled: Operation is stopped before switching to the STOP mode.
f_{HOCO}: 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_{EXT}: External subsystem clock

Remark 2. p = 00, 20; q = 0, 2

(2) Timing diagram when the interrupt request signal is generated in the SNOOZE mode

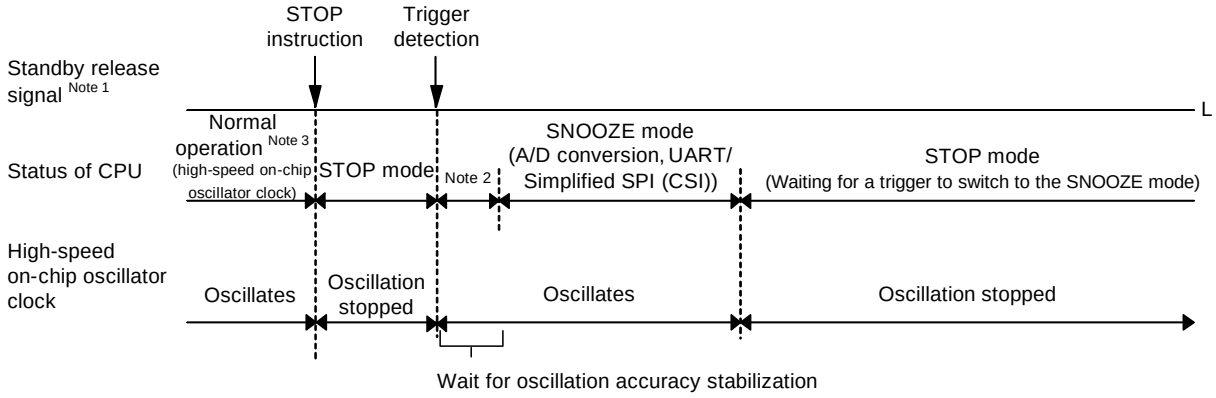
Figure 23 - 7 When the Interrupt Request Signal is Generated in the SNOOZE Mode



Note 1. For details of the standby release signal, see **Figure 21 - 1**.
Note 2. Transition time from STOP mode to SNOOZE mode
Note 3. Transition time from SNOOZE mode to normal operation
Note 4. Enable the SNOOZE mode (AWC = 1 or SWCm = 1) immediately before switching to the STOP mode.
Note 5. Be sure to release the SNOOZE mode (AWC = 0 or SWCm = 0) immediately after return to the normal operation.

(3) Timing diagram when the interrupt request signal is not generated in the SNOOZE mode

Figure 23 - 8 When the Interrupt Request Signal is not Generated in the SNOOZE Mode



Note 1. For details of the standby release signal, see **Figure 21 - 1**.
Note 2. Transition time from STOP mode to SNOOZE mode
Note 3. Enable the SNOOZE mode (AWC = 1 or SWCm = 1) immediately before switching to the STOP mode.

Remark For details of the SNOOZE mode function, see **CHAPTER 12 A/D CONVERTER** and **CHAPTER 15 SERIAL ARRAY UNIT**.

CHAPTER 24 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 00000H and 00001H 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 24 - 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.

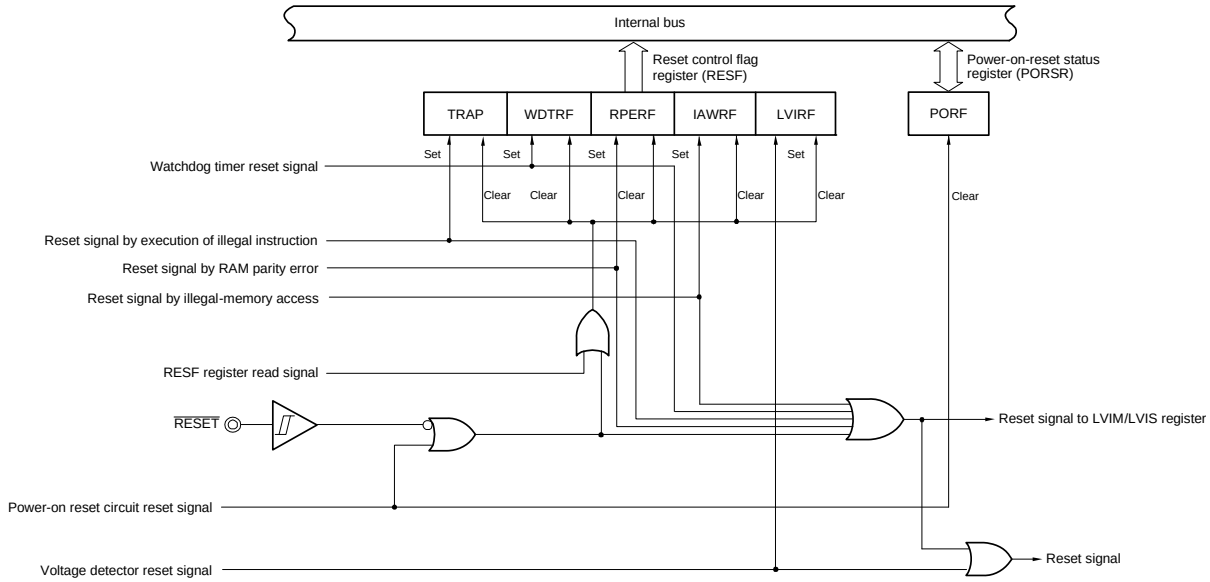
Caution 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
To perform an external reset upon power application, input a low level to the $\overline{\text{RESET}}$ pin, turn power on, continue to input a low level to the pin for 10 μs or more within the operating voltage range shown in 34.4 or 35.4 AC Characteristics, and then input a high level to the pin.

Caution 2. During reset input, the X1 clock, XT1 clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock oscillating. External main system clock input and external subsystem clock input become invalid.

Caution 3. The port pins become the following state because each SFR and 2nd SFR are initialized after reset.
- **P40: High-impedance during the external reset period or reset period by the POR. High level during other types of reset or after receiving a reset signal (connected to the internal pull-up resistance).**
- **P130: Low level during the reset period or after receiving a reset signal.**
- **Ports other than P40 and P130: 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 24 - 1 Block Diagram of Reset Function



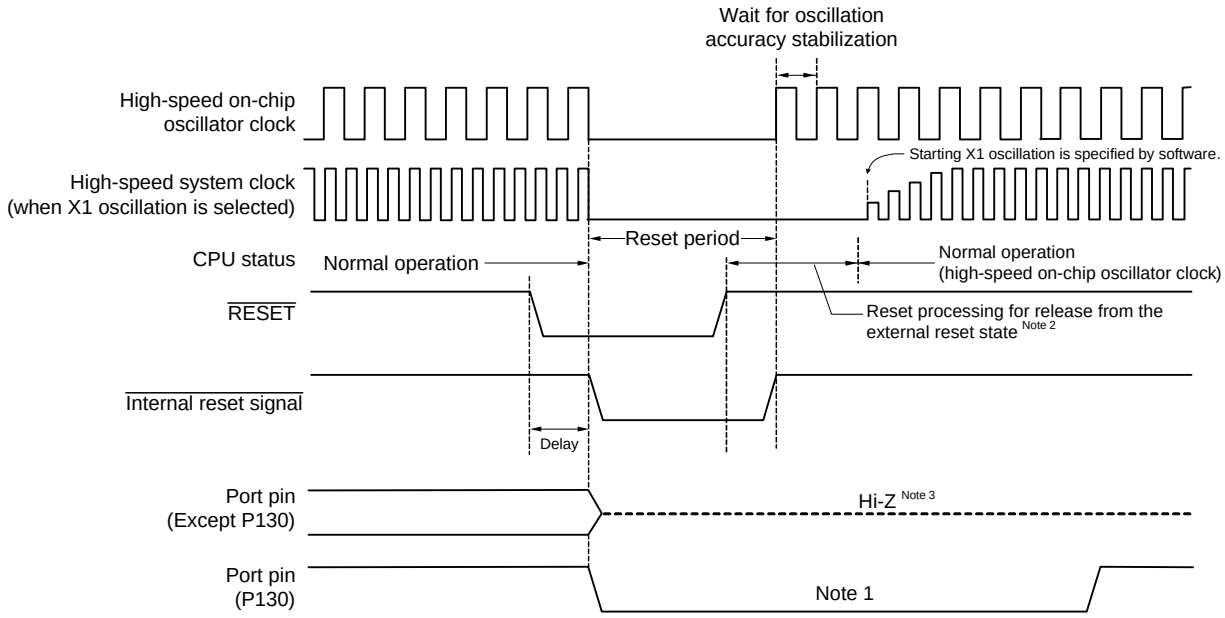
Caution An LVD circuit internal reset does not reset the LVD circuit.

Remark 1. LVIM: Voltage detection register
Remark 2. LVIS: Voltage detection level register

24.1 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.

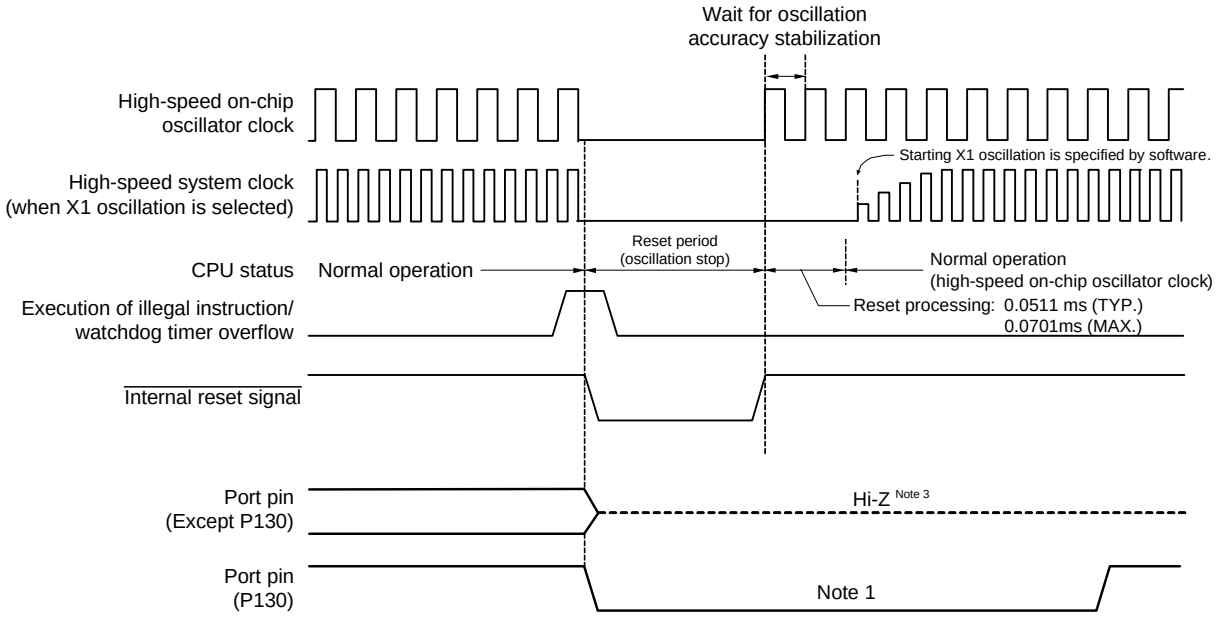
Figure 24 - 2 Timing of Reset by $\overline{\text{RESET}}$ Input



(**Notes** are listed on the next page.)

Release from the reset state is automatic in the case of a reset due to a watchdog timer overflow, execution of an illegal instruction, detection of a RAM parity error, or detection of illegal memory access. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

Figure 24 - 3 Timing of Reset Due to Watchdog Timer Overflow, Execution of Illegal Instruction, Detection of RAM Parity Error, or Detection of Illegal Memory Access



(**Notes** are listed on the next page.)

Note 1. When P130 is set to high-level output before reset is effected, the output signal of P130 can be dummy-output as a reset signal to an external device, because P130 outputs a low level when reset is effected. To release a reset signal to an external device, set P130 to high-level output by software.

Note 2. 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.

Note 3. 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 resistance).

24.2 States of Operation During Reset Periods

Table 24 - 1 shows the states of operation during reset periods. Table 24 - 2 shows the states of the hardware after receiving a reset signal.

Table 24 - 1 Operation Statuses During Reset Period

Item			During Reset Period
System clock			Clock supply to the CPU is stopped.
	Main system clock	f_{HOCO}	Operation stopped
		f_X	Operation stopped (the X1 and X2 pins are input port mode)
		f_{EX}	Clock input invalid (the pin is input port mode)
		f_{PLL}	Operation stopped
	Subsystem clock	f_{XT}	Operation possible
		f_{EXT}	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 (operable when DMA is executed)
Port (latch)			High impedance Note
Timer array unit			Operation stopped
16-bit timer KB2			
Real-time clock 2			During a reset other than the POR reset: Operation possible During a POR reset: Calendar operation possible; operation of the RTCC0, RTCC1, and SUBCUD registers stops.
12-bit Interval timer			Operation stopped
Watchdog timer			
Clock output/buzzer output			
A/D converter			
D/A converter			
Comparator			
Serial array unit (SAU)			
Serial interface (IICA)			
USB			
LCD controller/driver			Operation stopped (COM only pin, COM/SEG alternate pin: GND output, SEG/general-purpose port alternate pin: high-impedance output, V_{L1} to V_{L4} pins: high-impedance output, CAPH/P127 pin, CAPL/P126 pin: high-impedance output)
DTC			Operation stopped
ELC			
Power-on-reset function			Detection operation possible
Voltage detection function			Operation is possible in the case of an LVD reset and stopped in the case of other types of reset.
External interrupt			Operation stopped
Key interrupt function			
CRC operation function		High-speed CRC	
		General-purpose CRC	
RAM parity error detection function			
RAM guard function			
SFR guard function			
Illegal-memory access detection function			

Note P40 and P130 become 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 on-chip pull-up resistance).
- P130: Low level during the reset period

Remark f_{HOCO}: High-speed on-chip oscillator clock
f_X: X1 oscillation clock
f_{EX}: External main system clock
f_{XT}: XT1 oscillation clock
f_{EXT}: External subsystem clock
f_{IL}: Low-speed on-chip oscillator clock

Table 24 - 2 State of Hardware After Receiving a Reset Signal

Hardware		After Reset Acknowledgment Note
Program counter (PC)		The contents of the reset vector table (00000H, 00001H) 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 For the state of the special function register (SFR) after receiving a reset signal, see **3.1.4 Special function register (SFR) area** and **3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area**.

24.3 Register for Confirming Reset Source

24.3.1 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 24 - 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.

Note 1. The value after reset varies depending on the reset source. See **Table 24 - 3**.

Note 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.

Caution 1. Do not read data by a 1-bit memory manipulation instruction.

Caution 2. An instruction code fetched from RAM is not subject to parity error detection while it is being executed. However, the data read by the instruction is subject to parity error detection.

Caution 3. Because 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, when enabling RAM parity error resets (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

The status of the RESF register when a reset request is generated is shown in Table 24 - 3.

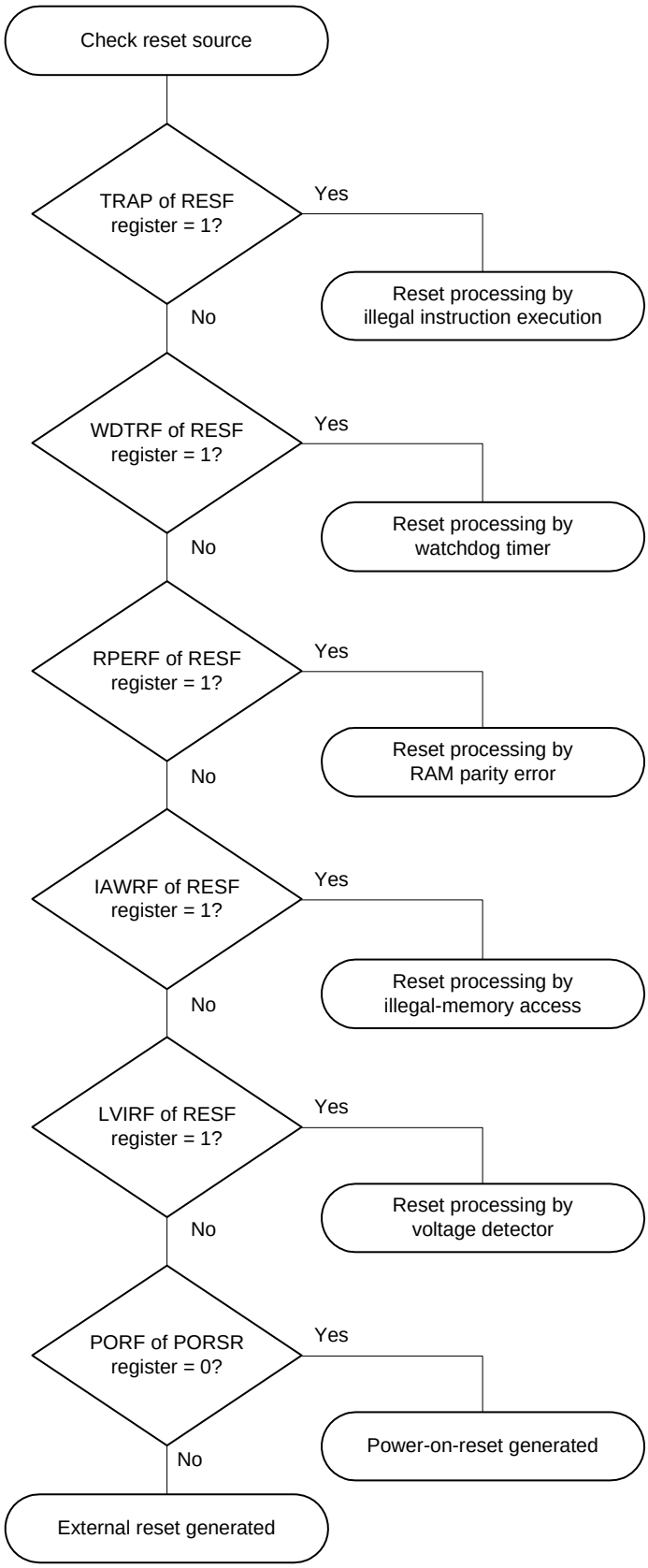
Table 24 - 3 RESF Register Status When Reset Request Is Generated

Reset Source / Flag	$\overline{\text{RESET}}$ Input	Reset by POR	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by RAM parity error	Reset by illegal-memory access	Reset by LVD
TRAP bit	Cleared (0)	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 24 - 5 shows the procedure for checking a reset source.

Figure 24 - 5 Example of Procedure for Checking Reset Source



Caution The flow described above is an example of the procedure for checking.

24.3.2 Power-on-reset status register (PORSR)

The PORSR register is used to check the occurrence of a power-on reset.

Writing 1 to bit 0 (PORF) of the PORSR register enables this function. Writing 0 disables this function.

Write 1 to the PORF bit in advance to enable checking of the occurrence of a power-on reset.

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

Power-on reset signal generation clears this register to 00H.

Caution 1. The PORSR register is reset only by a power-on reset; it retains the value when a reset caused by another factor occurs.

Caution 2. If the PORF bit is set to 1, it guarantees that no power-on reset has occurred, but it does not guarantee that the RAM value is retained.

Figure 24 - 6 Format of Power-on-reset status register (PORSR)

Address: F00F9H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PORSR	0	0	0	0	0	0	0	PORF

PORF	Checking occurrence of power-on reset
0	A value 1 has not been written, or a power-on reset has occurred.
1	No power-on reset has occurred.

CHAPTER 25 POWER-ON-RESET CIRCUIT

25.1 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 **34.4** or **35.4 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 **34.4** or **35.4 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) and the power-on-reset status register (PORSR) are cleared (00H).**

Remark 1. The RL78 microcontroller incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset source is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT), voltage-detector (LVD), illegal instruction execution, RAM parity error, or illegal-memory access. The RESF register is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by the watchdog timer (WDT), voltage-detector (LVD), illegal instruction execution, RAM parity error, or illegal-memory access.
For details of the RESF register, see **CHAPTER 24 RESET FUNCTION**.

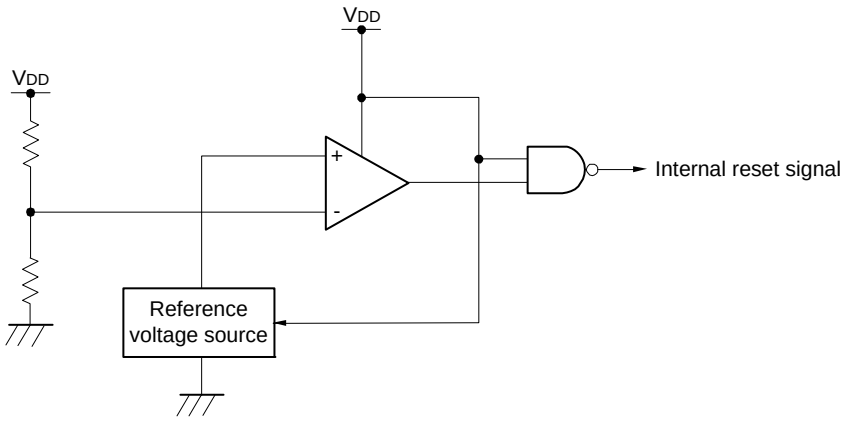
Remark 2. The occurrence of an internal reset in the power-on-reset circuit can be checked by the power-on reset status register (PORSR). For details on the PORSR register, refer to **CHAPTER 24 RESET FUNCTION**.

Remark 3. V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage
For details, see **34.6.5** or **35.6.5 POR circuit characteristics**.

25.2 Configuration of Power-on-reset Circuit

The block diagram of the power-on-reset circuit is shown in Figure 25 - 1.

Figure 25 - 1 Block Diagram of Power-on-reset Circuit

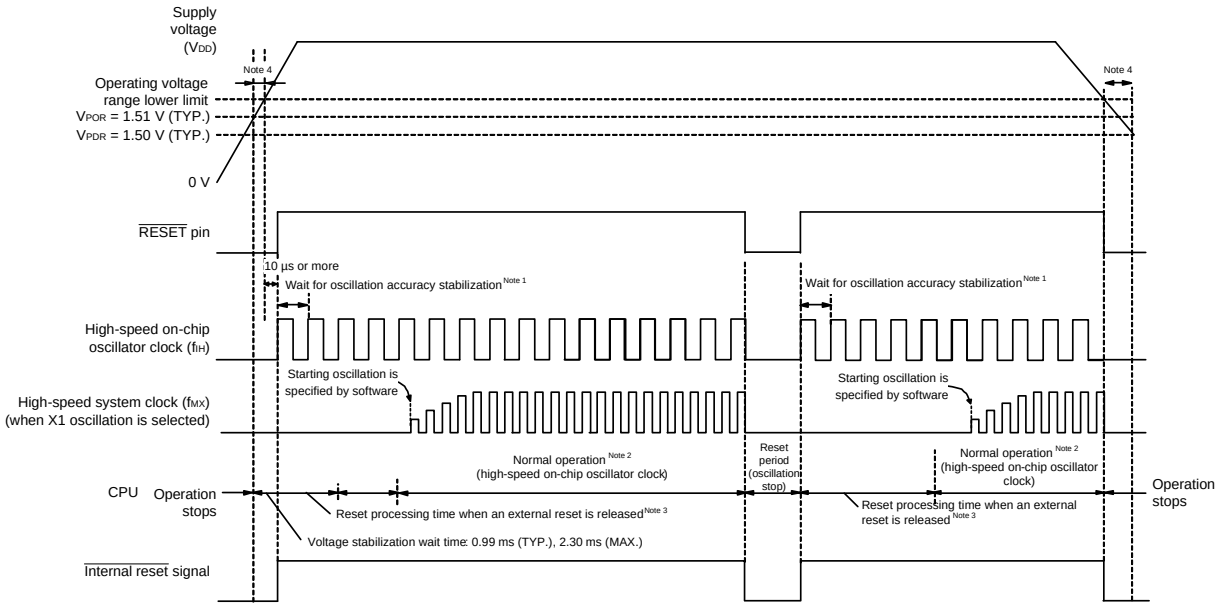


25.3 Operation of Power-on-reset Circuit

The timing of generation of the internal reset signal by the power-on-reset circuit and voltage detector is shown next.

Figure 25 - 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



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

Note 2. The high-speed on-chip oscillator clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

Note 3. The time until normal operation starts includes the following reset processing time when the external reset is released (in the first reset processing following release from the POR state) after the $\overline{\text{RESET}}$ signal is driven high (1) as well as the voltage stabilization wait time after VPOR (1.51 V, typ.) is reached.

The reset processing time when the external reset is released is shown below.

In the first reset processing following release from the POR state:

0.672 ms (TYP.), 0.832 ms (MAX.) (when the LVD is in use)

0.399 ms (TYP.), 0.519 ms (MAX.) (when the LVD is off)

The reset processing time when the external reset is released in the second reset processing following release from the POR state is shown below.

In the second reset processing following release from the POR state:

0.531 ms (TYP.), 0.675 ms (MAX.) (when the LVD is in use)

0.259 ms (TYP.), 0.362 ms (MAX.) (when the LVD is off)

Note 4. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **34.4** or **35.4 AC Characteristics**. This is done by controlling the externally input reset signal.

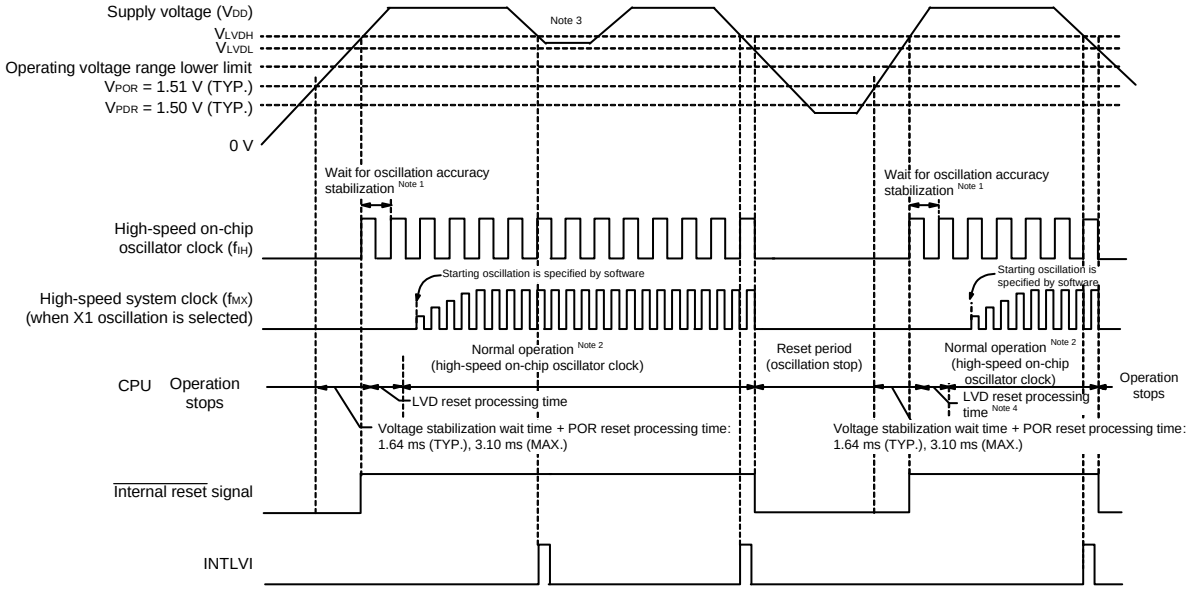
After power supply is turned off, this LSI should be placed in the STOP mode, or in the reset state by utilizing the voltage detection circuit or externally input reset signal, before the voltage falls below the operating range. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

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 26 VOLTAGE DETECTOR.**

Remark VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

Figure 25 - 3 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (2/3)

(2) When LVD is interrupt & reset mode (option byte 000C1: LVIMDS1, LVIMDS0 = 1, 0)



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

Note 2. The high-speed on-chip oscillator clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

Note 3. After the interrupt request signal (INTLVI) is generated, the LVILV and LVIMD bits of the voltage detection level register (LVIS) are automatically set to 1. After INTLVI is generated, appropriate settings should be made according to **Figure 26 - 10 Processing Procedure After an Interrupt Is Generated** and **Figure 26 - 11 Initial Setting of Interrupt and Reset Mode**, taking into consideration that the supply voltage might return to the high voltage detection level (V_{LVDH}) or higher without falling below the low voltage detection level (V_{LVDL}).

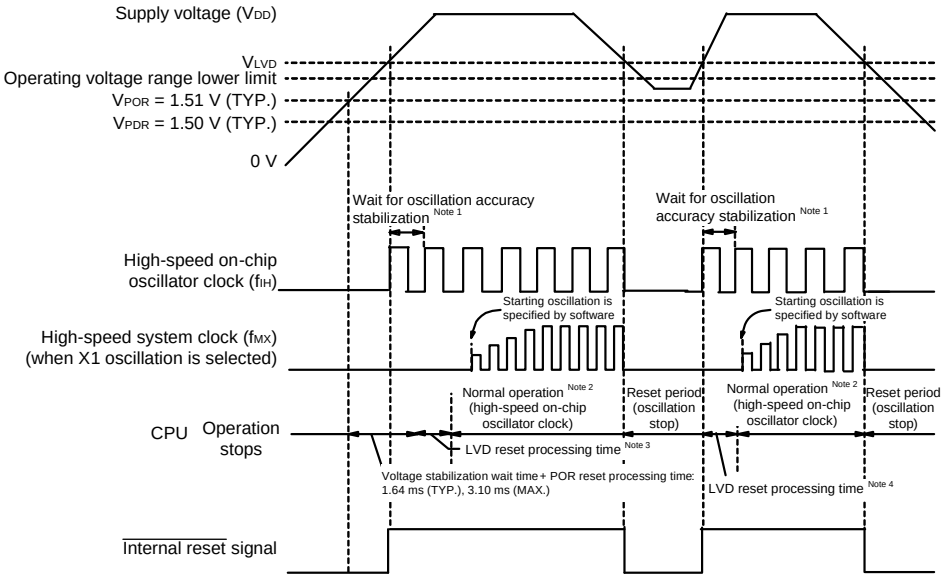
Note 4. The time until normal operation starts includes the following LVD reset processing time after the LVD detection level (V_{LVDH}) is reached as well as the voltage stabilization wait + POR reset processing time after the V_{POR} (1.51 V, TYP.) is reached.

LVD reset processing time: 0 ms to 0.0701 ms (MAX.)

Remark V_{LVDH}, V_{LVDL}: LVD detection voltage
V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Figure 25 - 4 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (3/3)

(3) LVD reset mode (option byte 000C1H: LVIMDS1, LVIMDS0 = 1, 1)



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

Note 2. The high-speed on-chip oscillator clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

Note 3. The time until normal operation starts includes the following LVD reset processing time after the LVD detection level (V_{LVD}) is reached as well as the voltage stabilization wait + POR reset processing time after the V_{POR} (1.51 V, TYP.) is reached.

LVD reset processing time: 0 ms to 0.0701 ms (MAX.)

Note 4. When the power supply voltage is below the lower limit for operation and the power supply voltage is then restored after an internal reset is generated only by the voltage detector (LVD), the following LVD reset processing time is required after the LVD detection level (V_{LVD}) is reached.

LVD reset processing time: 0.0511 ms (TYP.), 0.0701 ms (MAX.)

Remark 1. V_{LVDH}, V_{LVDL}: LVD detection voltage
V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Remark 2. When the LVD interrupt mode is selected (option byte 000C1H: LVIMD1 = 0, LVIMD0 = 1), the time until normal operation starts after power is turned on is the same as the time specified in Note 3 of Figure 25 - 4 (3).

CHAPTER 26 VOLTAGE DETECTOR

26.1 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 12 levels (For details, see **CHAPTER 29 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 **34.4** or **35.4 AC Characteristics**. This is done by utilizing the voltage detector or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage 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.

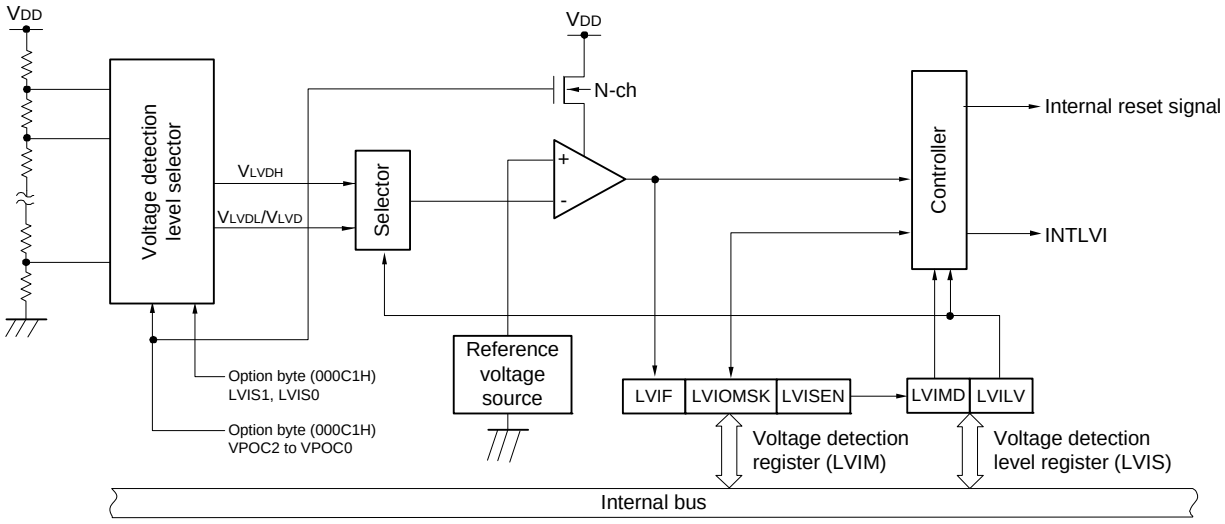
Interrupt & reset mode (LVIMDS1, LVIMDS0 = 1, 0)	Reset mode (LVIMDS1, LVIMDS0 = 1, 1)	Interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)
Generates an interrupt request signal by detecting $V_{DD} < V_{LVDH}$ when the operating voltage falls, and releases an internal reset by detecting $V_{DD} < V_{LVDL}$. Releases an internal reset by detecting $V_{DD} \geq V_{LVDH}$.	Releases an internal reset by detecting $V_{DD} \geq V_{LVD}$. Generates an internal reset by detecting $V_{DD} < V_{LVD}$.	Retains the state of an internal reset by the LVD immediately after a reset until $V_{DD} \geq V_{LVD}$. Releases the LVD internal reset by detecting $V_{DD} \geq V_{LVD}$. Generates an interrupt request signal (INTLVI) by detecting $V_{DD} < V_{LVD}$ or $V_{DD} \geq V_{LVD}$ after the LVD internal reset is released.

While the voltage detector is operating, whether the supply voltage is more than or less than the detection level can be checked by reading the voltage detection flag (LVIF: bit 0 of the voltage detection register (LVIM)).
Bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of the RESF register, see **CHAPTER 24 RESET FUNCTION**.

26.2 Configuration of Voltage Detector

The block diagram of the voltage detector is shown in Figure 26 - 1.

Figure 26 - 1 Block Diagram of Voltage Detector



26.3 Registers Controlling Voltage Detector

The voltage detector is controlled by the following registers.

- Voltage detection register (LVIM)
- Voltage detection level register (LVIS)

26.3.1 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 26 - 2 Format of Voltage detection register (LVIM)

Address: FFFA9H After reset: 00H Note 1 R/W Note 2

Symbol	<7>	6	5	4	3	2	<1>	<0>
LVIM	LVISEN Note 3	0	0	0	0	0	LVIOMSK	LVIF

LVISEN Note 3	Specification of whether to enable or disable rewriting the voltage detection level register (LVIS)
0	Disabling of rewriting the LVIS register (LVIOMSK = 0 (Mask of LVD output is invalid)
1	Enabling of rewriting the LVIS register Note 3 (LVIOMSK = 1 (Mask of LVD output is valid)

LVIOMSK	Mask status flag of LVD output
0	Mask of LVD output is invalid
1	Mask of LVD output is valid Note 4

LVIF	Voltage detection flag
0	Supply voltage (V_{DD}) ≥ detection voltage (V_{LVD}), or when LVD is off
1	Supply voltage (V_{DD}) < detection voltage (V_{LVD})

Note 1. The reset value changes depending on the reset source.
If the LVIS register is reset by LVD, it is not reset but holds the current value. The value of this register is reset to “00H” if a reset other than by LVD is effected.

Note 2. Bits 0 and 1 are read-only.

Note 3. Can only be set in the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0). Do not change the initial value in other modes.

Note 4. LVIOMSK bit is only automatically set to “1” when the interrupt & reset mode is selected (option byte LVIMDS1, LVIMDS0 = 1, 0) and reset or interrupt by LVD is masked.
- Period during LVISEN = 1
- Waiting period from the time when LVD interrupt is generated until LVD detection voltage becomes stable
- Waiting period from the time when the value of LVILV bit changes until LVD detection voltage becomes stable

26.3.2 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 26 - 3 Format of Voltage detection level register (LVIS)

Address: FFFAAH After reset:00H/01H/81H Note 1 R/W

Symbol	<7>	6	5	4	3	2	1	<0>
LVIS	LVIMD Note 2	0	0	0	0	0	0	LVILV Note 2

LVIMD Note 2	Operation mode of voltage detection
0	Interrupt mode
1	Reset mode

LVILV Note 2	LVD detection level
0	High-voltage detection level (V_{LVDH})
1	Low-voltage detection level (V_{LVDL} or V_{LVD})

Note 1. The reset value changes depending on the reset source and the setting of the option byte.
This register is not cleared (00H) by LVD reset.
The generation of reset signal other than an LVD reset sets as follows.
- When option byte LVIMDS1, LVIMDS0 = 1, 0: 00H
- When option byte LVIMDS1, LVIMDS0 = 1, 1: 81H
- When option byte LVIMDS1, LVIMDS0 = 0, 1: 01H

Note 2. Writing “0” can only be allowed in the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0). Do not set LVIMD and LVILV in other cases. The value is switched automatically when reset or interrupt is generated in the interrupt & reset mode.

Caution 1. Rewrite the value of the LVIS register according to Figures 26 - 9 and 26 - 10.

Caution 2. Specify the LVD operation mode and detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}) of each mode by using the option byte 000C1H. Figure 26 - 4 shows the format of the user option byte (000C1H/010C1H). For details about the option byte, see CHAPTER 29 OPTION BYTE.

Figure 26 - 4 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
1.77 V	1.73 V	1.63 V	0	0	0	1	0	1	0
1.88 V	1.84 V					0	1		
2.92 V	2.86 V					0	0		
1.98 V	1.94 V	1.84 V		0	1	1	0		
2.09 V	2.04 V					0	1		
3.13 V	3.06 V					0	0		
2.61 V	2.55 V	2.45 V		1	0	1	0		
2.71 V	2.65 V					0	1		
2.92 V	2.86 V	2.75 V		1	1	1	0		
3.02 V	2.96 V					0	1		
—			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
1.67 V	1.63 V	0	0	0	1	1	1	1
1.77 V	1.73 V		0	0	1	0		
1.88 V	1.84 V		0	1	1	1		
1.98 V	1.94 V		0	1	1	0		
2.09 V	2.04 V		0	1	0	1		
2.50 V	2.45 V		1	0	1	1		
2.61 V	2.55 V		1	0	1	0		
2.71 V	2.65 V		1	0	0	1		
2.81 V	2.75 V		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		
—		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.

Remark 1. For details on the LVD circuit, see **CHAPTER 26 VOLTAGE DETECTOR**.

Remark 2. The detection voltage is a TYP. value. For details, see **34.6.6** or **35.6.6 LVD circuit characteristics**.
(**Cautions** are listed on the next page.)

Figure 26 - 5 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 setting (interrupt mode)

Detection voltage		Option byte Setting Value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
1.67 V	1.63 V	0	0	0	1	1	0	1
1.77 V	1.73 V		0	0	1	0		
1.88 V	1.84 V		0	1	1	1		
1.98 V	1.94 V		0	1	1	0		
2.09 V	2.04 V		0	1	0	1		
2.50 V	2.45 V		1	0	1	1		
2.61 V	2.55 V		1	0	1	0		
2.71 V	2.65 V		1	0	0	1		
2.81 V	2.75 V		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		
—			Setting of values other than above is prohibited.					

• LVD off (use of external reset input via $\overline{\text{RESET}}$ pin)

Detection voltage		Option byte Setting Value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
—	—	1	×	×	×	×	×	1
—		Setting of values other than above is prohibited.						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Caution 1. Set bit 4 to 1.

Caution 2. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 34.4 or 35.4 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).

Remark 1. ×: don't care

Remark 2. For details on the LVD circuit, see **CHAPTER 26 VOLTAGE DETECTOR**.

Remark 3. The detection voltage is a TYP. value. For details, see **34.6.6** or **35.6.6 LVD circuit characteristics**.

26.4 Operation of Voltage Detector

26.4.1 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}).

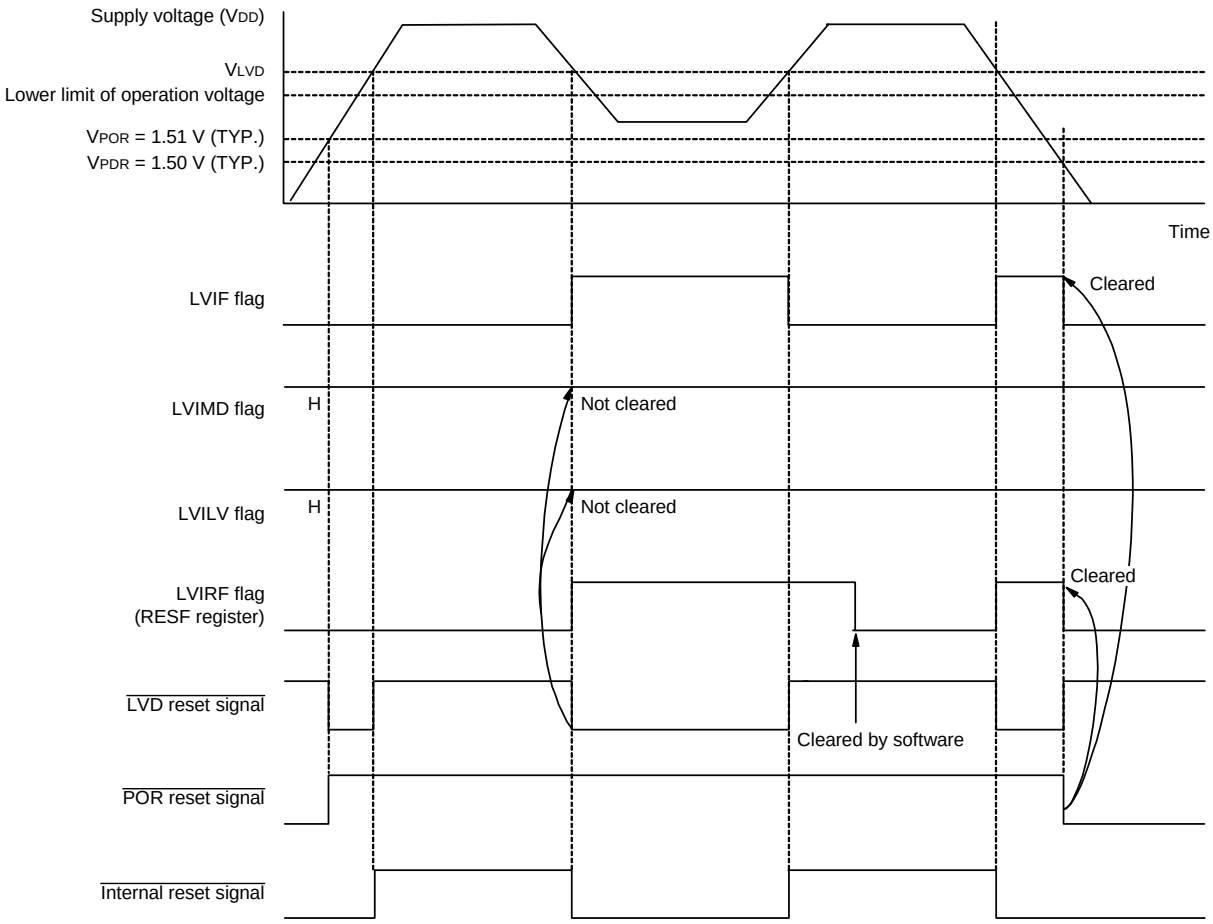
• Operation in LVD reset mode

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 26 - 6 shows the timing of the internal reset signal generated in the LVD reset mode.

Figure 26 - 6 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

26.4.2 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.

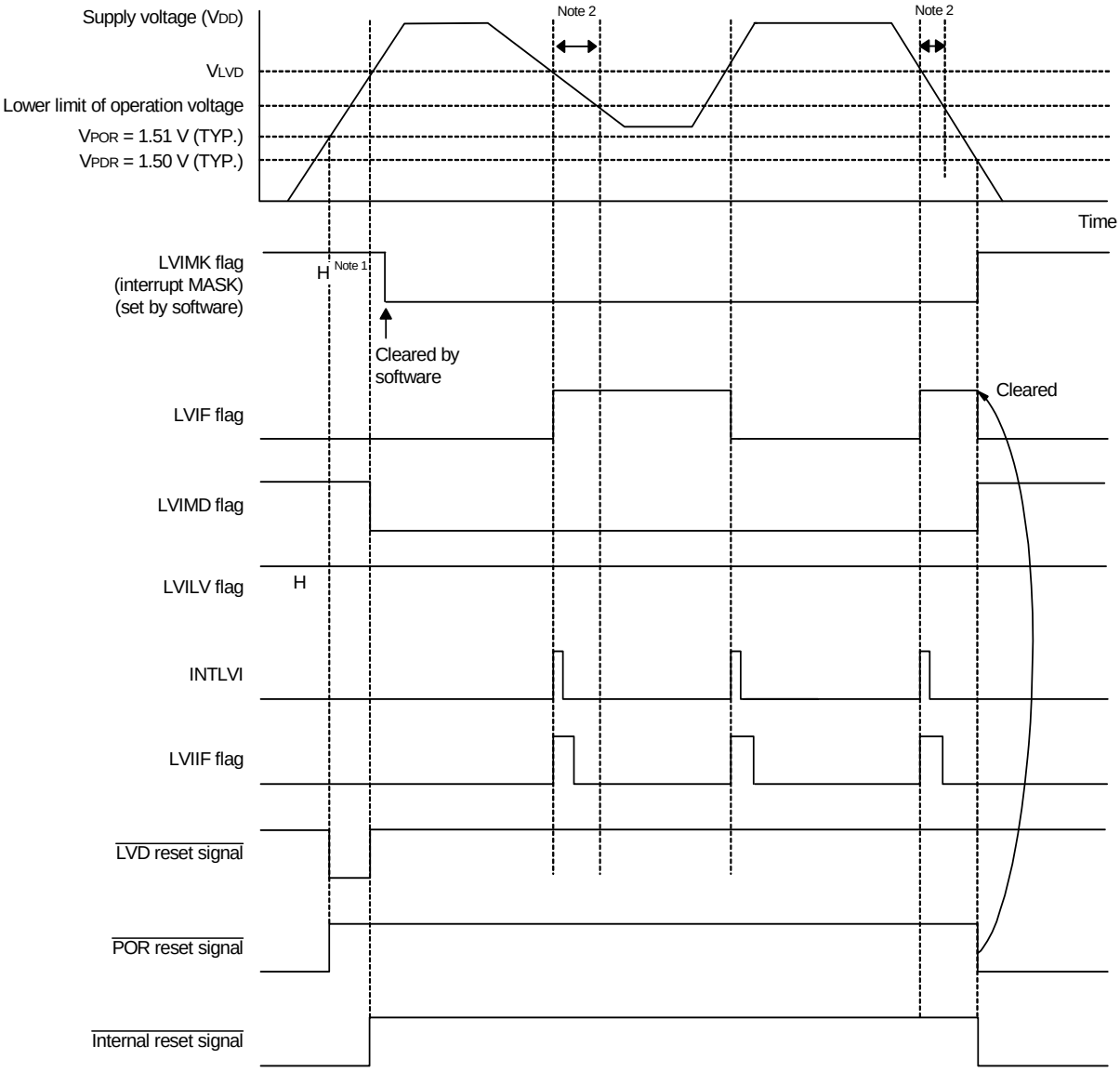
- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- The initial value of the voltage detection level select register (LVIS) is set to 01H.
  Bit 7 (LVIMD) is 0 (interrupt mode).
  Bit 0 (LVILV) is 1 (low-voltage detection level: V_{LVD}).

• Operation in LVD interrupt mode

In the interrupt mode (option byte LVIMDS1, LVIMDS0 = 0, 1), the state of an internal reset by LVD is retained immediately after a reset until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}). The internal reset is released when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}). After the LVD internal reset is released, an interrupt request signal (INTLVI) by the LVD is generated when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}). When the voltage falls, this LSI should be placed in the STOP mode, or placed in the reset state by controlling the externally input reset signal, before the voltage falls below the operating voltage range defined in **34.4** or **35.4 AC characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Figure 26 - 7 shows the timing of the interrupt request signal generated in the LVD interrupt mode.

Figure 26 - 7 Timing of Voltage Detector Internal Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 0, 1)



Note 1. The LVIMK flag is set to “1” by reset signal generation.

Note 2. When the voltage falls, this LSI should be placed in the STOP mode, or placed in the reset state by controlling the externally input reset signal, before the voltage falls below the operating voltage range defined in **34.4** or **35.4 AC characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Remark VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

26.4.3 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.

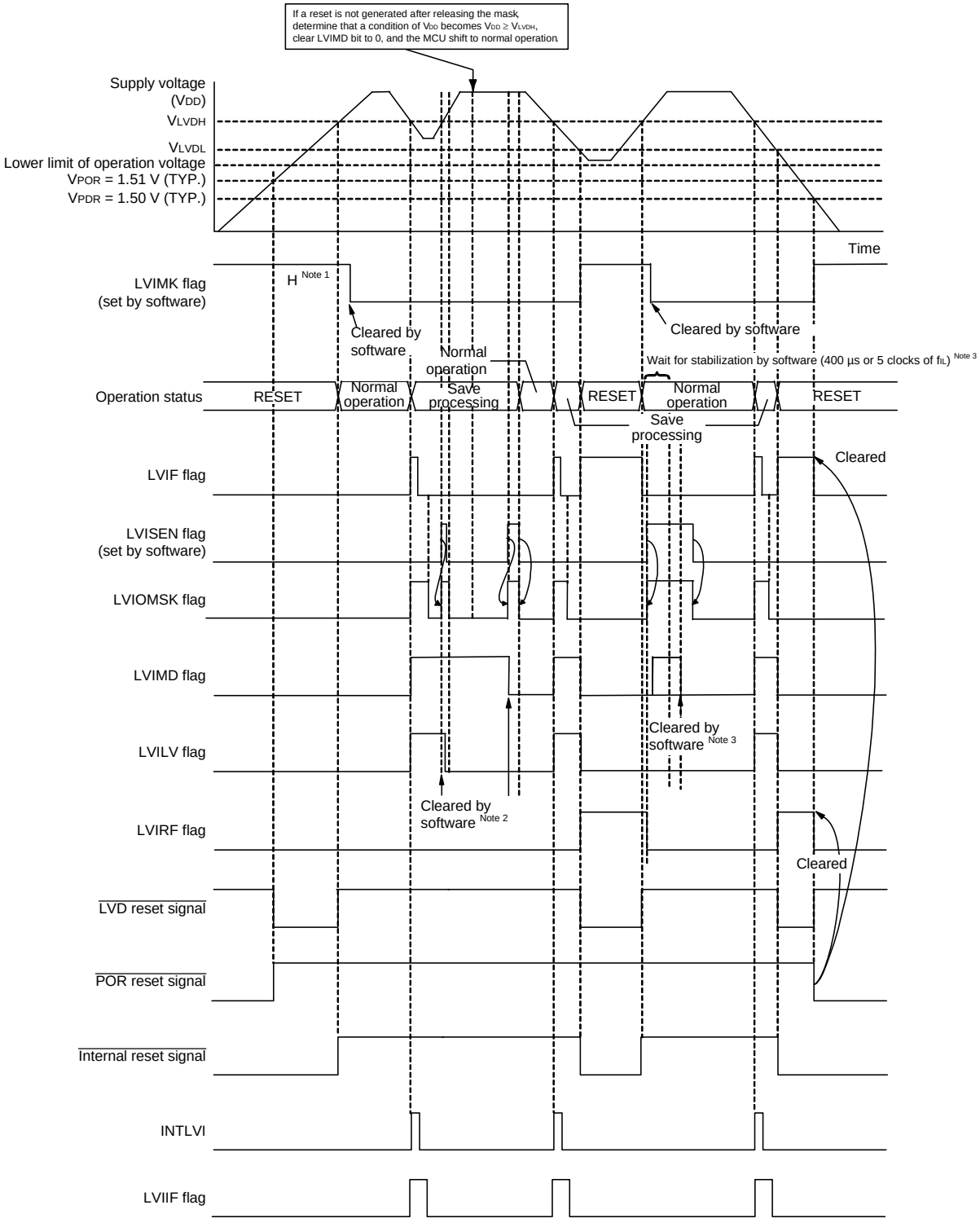
- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- The initial value of the voltage detection level select register (LVIS) is set to 00H.
  Bit 7 (LVIMD) is 0 (interrupt mode).
  Bit 0 (LVILV) is 0 (high-voltage detection 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 (INTLVI) is generated and arbitrary save processing is performed when the supply voltage (V_{DD}) falls below the high-voltage detection level (V_{LVDH}). After that, an internal reset by LVD is generated when the supply voltage (V_{DD}) falls below the low-voltage detection level (V_{LVDL}). After INTLVI is generated, an interrupt request signal is not generated even if the supply voltage becomes equal to or higher than the high-voltage detection voltage (V_{LVDH}) without falling below the low-voltage detection voltage (V_{LVDL}). To use the LVD reset & interrupt mode, perform the processing according to **Figure 26 - 10 Processing Procedure After an Interrupt Is Generated** and **Figure 26 - 11 Initial Setting of Interrupt and Reset Mode**.

Figure 26 - 8 to Figure 26 - 9 show the timing of the internal reset signal and interrupt signal generated in the LVD interrupt & reset mode.

Figure 26 - 8 Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (1/2)



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

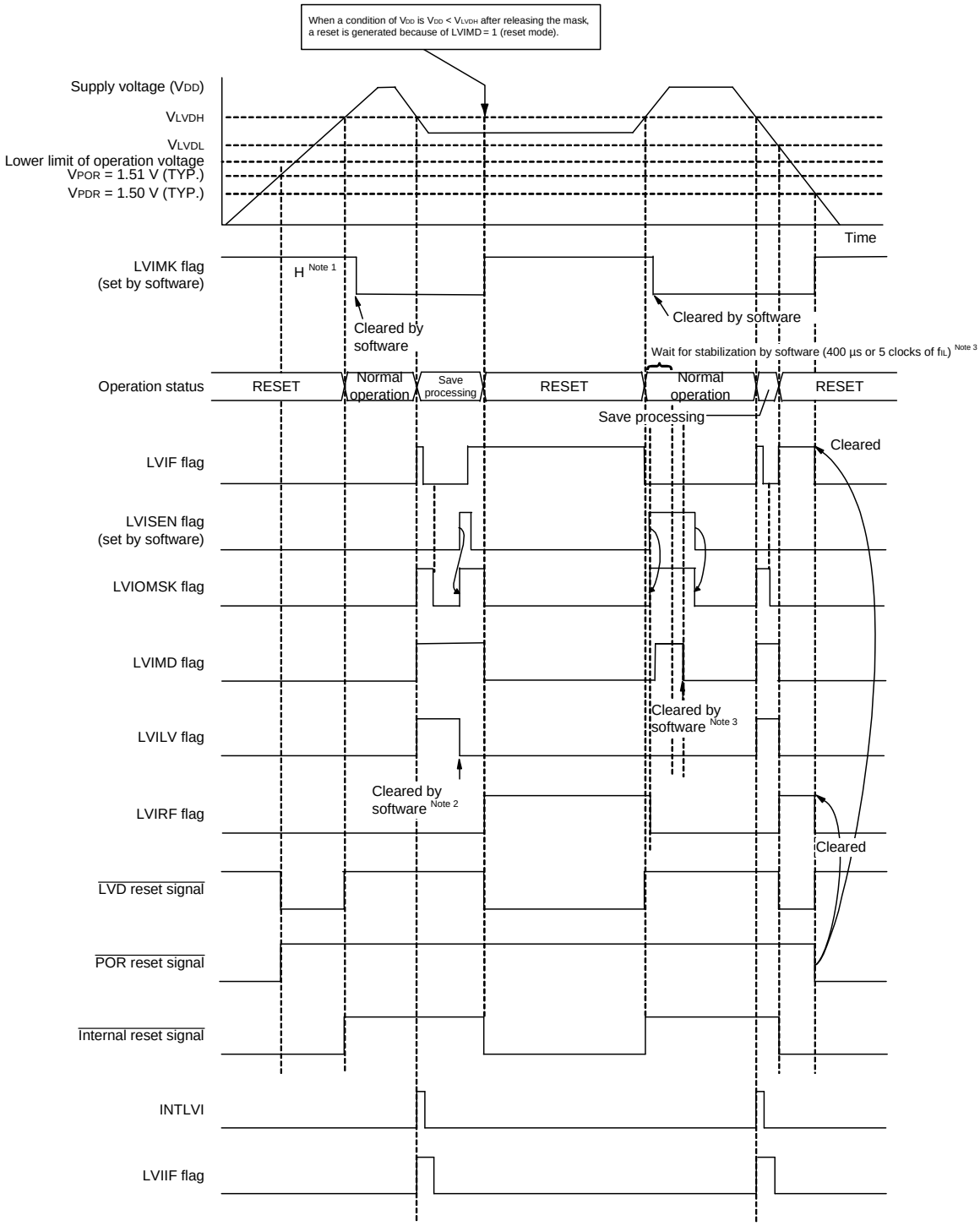
Note 1. The LVIMK flag is set to “1” by reset signal generation.

Note 2. After an interrupt is generated, perform the processing according to Figure 26 - 10 Processing Procedure After an Interrupt Is Generated in interrupt and reset mode.

Note 3. After a reset is released, perform the processing according to Figure 26 - 11 Initial Setting of Interrupt and Reset Mode in interrupt and reset mode.

Remark V_{POR}: POR power supply rise detection voltage
V_{POR}: POR power supply fall detection voltage

Figure 26 - 9 Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (2/2)



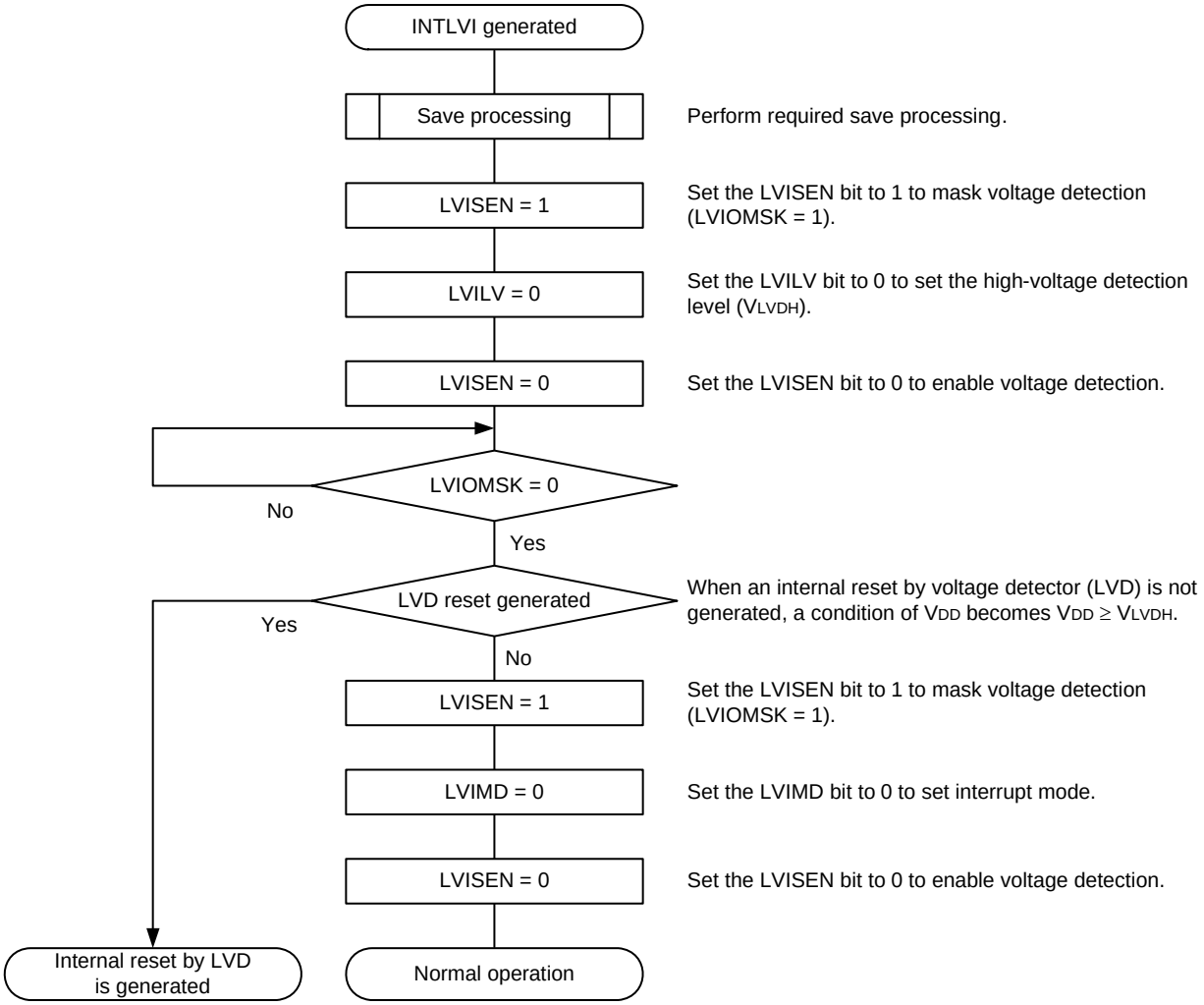
Note 1. The LVIMK flag is set to “1” by reset signal generation.

Note 2. After an interrupt is generated, perform the processing according to Figure 26 - 10 Processing Procedure After an Interrupt Is Generated in interrupt and reset mode.

Note 3. After a reset is released, perform the processing according to Figure 26 - 11 Initial Setting of Interrupt and Reset Mode in interrupt and reset mode.

Remark VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

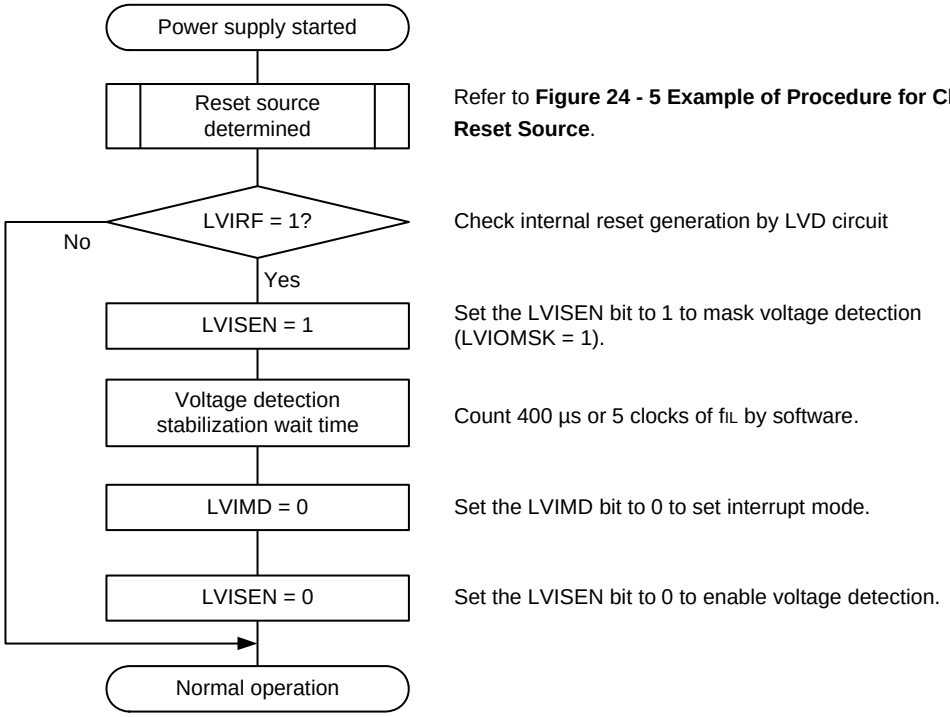
Figure 26 - 10 Processing Procedure After an Interrupt Is Generated



When setting an interrupt and reset mode (LVIMDS1, LVIMDS0 = 1, 0), voltage detection stabilization wait time for 400 µs or 5 clocks of f_{IL} is necessary after LVD reset is released (LVIRF = 1). After waiting until voltage detection stabilizes, (0) clear the LVIMD bit for initialization. While voltage detection stabilization wait time is being counted and when the LVIMD bit is rewritten, set LVISEN to 1 to mask a reset or interrupt generation by LVD.

Figure 26 - 11 shows the procedure for Initial Setting of Interrupt and Reset Mode.

Figure 26 - 11 Initial Setting of Interrupt and Reset Mode



Remark f_{IL}: Low-speed on-chip oscillator clock frequency

26.5 Cautions for Voltage Detector

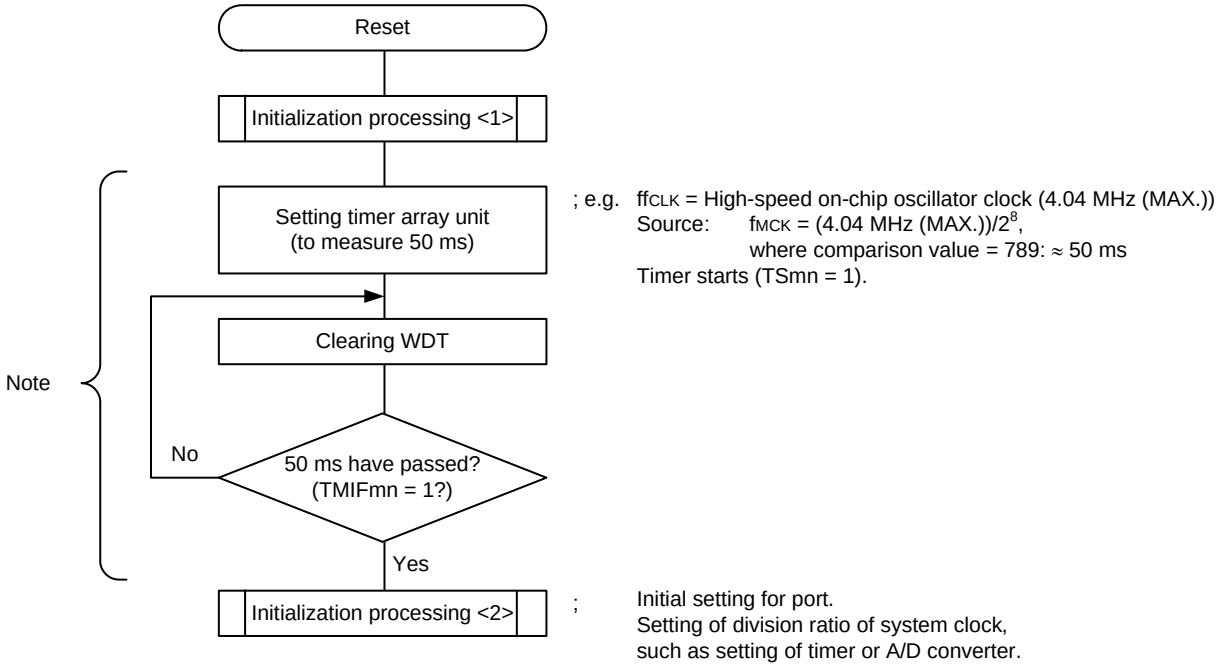
(1) Voltage fluctuation when power is supplied

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 26 - 12 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.

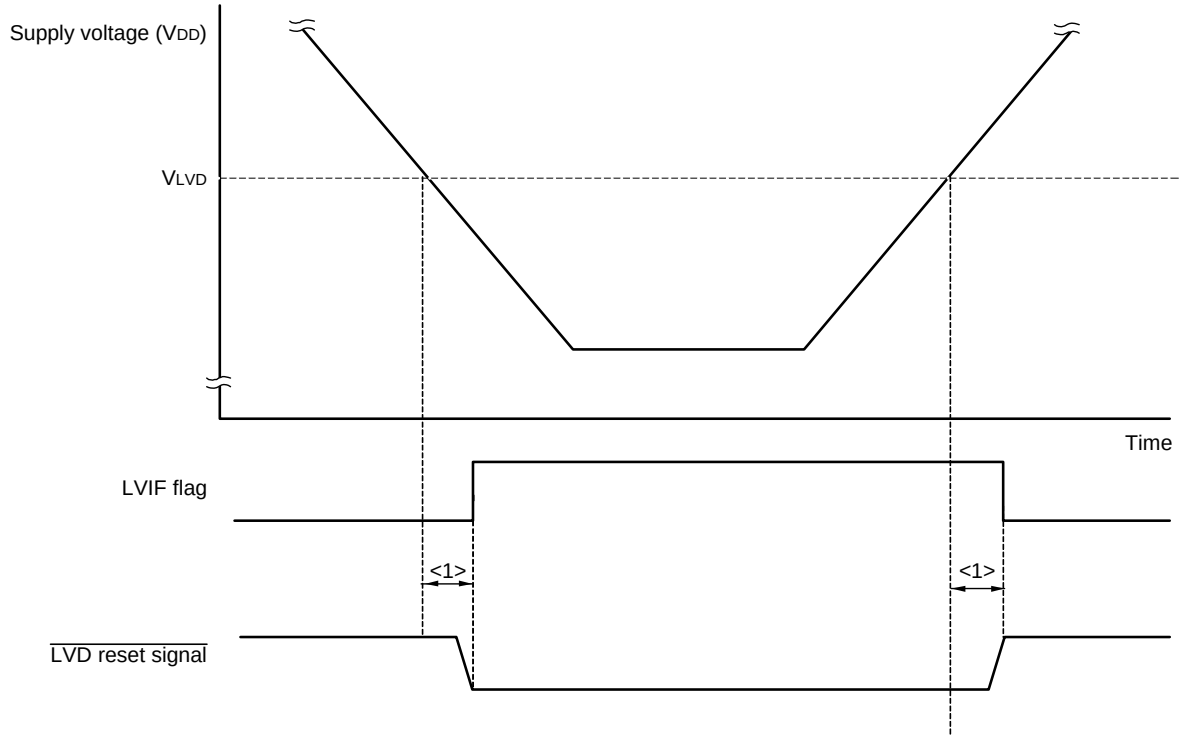
Remark m = 0, n = 0 to 7

(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 26 - 13**).

Figure 26 - 13 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 **34.4** or **35.4 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 **34.4** or **35.4 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

CHAPTER 27 SAFETY FUNCTIONS

27.1 Overview of Safety Functions

<R> The following safety functions are provided in the RL78/L1C to comply with the IEC60730 safety standard. 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/L1C that can be used according to the application or purpose of use.
- High-speed CRC: The CPU can be stopped and a high-speed check executed on its entire code flash memory area during the initialization routine.
- General CRC: This can be used for checking various data in addition to the code flash memory area while the CPU is running.

(2) RAM parity error detection function
This detects parity errors when the RAM is read as data.

(3) RAM guard function
This prevents RAM data from being rewritten when the CPU freezes.

(4) SFR guard function
This prevents SFRs from being rewritten when the CPU freezes.

(5) Invalid memory access detection function
This detects illegal accesses to invalid memory areas (such as areas where no memory is allocated and areas to which access is restricted).

(6) Frequency detection function
This 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.

(8) Digital output signal level detection function for I/O ports
When the I/O ports are output mode in which PMm bit of the port mode register (PMm) is 0, the output level of the pin can be read.

Remark 1. m = 0 to 8, 12, 14, 15; n = 0 to 7

Remark 2. For usage examples of the safety functions complying with the IEC60730 safety standards, refer to **RL78 MCU series IEC60730/60335 application notes (R01AN1062, R01AN1296)**.

27.2 Registers Used by Safety Functions

The safety functions use the following registers:

Register	Each Function of Safety Function
• Flash memory CRC control register (CRC0CTL) • Flash memory CRC operation result register (PGCRCL)	Flash memory CRC operation function (high-speed CRC)
• CRC input register (CRCIN) • CRC data register (CRCD)	CRC operation function (general-purpose CRC)
• RAM parity error control register (RPECTL)	RAM parity error detection function
• Invalid memory access detection control register (IAWCTL)	RAM guard function
	SFR guard function
	Invalid memory access detection function
• Timer input select register 0 (TIS0)	Frequency detection function
• A/D test register (ADTES)	A/D test function
• Port mode select register (PMS)	Digital output signal level detection function for I/O ports

The content of each register is described in 27.3 Operation of Safety Functions.

27.3 Operation of Safety Functions

27.3.1 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/L1C 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, 341 μs@24 MHz with 32 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.

27.3.1.1 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 27 - 1 Format of Flash memory CRC control register (CRC0CTL)

Address: F02F0H After reset:00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CRC0CTL	CRC0EN	0	FEA5	FEA4	FEA3	FEA2	FEA1	FEA0

CRC0EN	Control of high-speed CRC ALU operation
0	Stop the operation.
1	Start the operation according to HALT instruction execution.

FEA5	FEA4	FEA3	FEA2	FEA1	FEA0	High-speed CRC operation range
0	0	0	0	0	0	00000H to 03FFBH (16 K - 4 bytes)
0	0	0	0	0	1	00000H to 07FFBH (32 K - 4 bytes)
0	0	0	0	1	0	00000H to 0BFFBH (48 K - 4 bytes)
0	0	0	0	1	1	00000H to 0FFFBH (64 K - 4 bytes)
0	0	0	1	0	0	00000H to 13FFBH (80 K - 4 bytes)
0	0	0	1	0	1	00000H to 17FFBH (96 K - 4 bytes)
0	0	0	1	1	0	00000H to 1BFFBH (112 K - 4 bytes)
0	0	0	1	1	1	00000H to 1FFFBH (128 K - 4 bytes)
0	0	1	0	0	0	00000H to 23FFBH (144 K - 4 bytes)
0	0	1	0	0	1	00000H to 27FFBH (160 K - 4 bytes)
0	0	1	0	1	0	00000H to 2BFFBH (176 K - 4 bytes)
0	0	1	0	1	1	00000H to 2FFFBH (192 K - 4 bytes)
0	0	1	1	0	0	00000H to 33FFBH (208 K - 4 bytes)
0	0	1	1	0	1	00000H to 37FFBH (224 K - 4 bytes)
0	0	1	1	1	0	00000H to 3BFFBH (240 K - 4 bytes)
0	0	1	1	1	1	00000H to 3FFFBH (256 K - 4 bytes)
Other than the above						Setting prohibited

Remark Input the expected CRC operation result value to be used for comparison in the lowest 4 bytes of the flash memory. Note that the operation range will thereby be reduced by 4 bytes.

27.3.1.2 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 27 - 2 Format of Flash memory CRC operation result register (PGCRCL)

Address: F02F2H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
PGCRCL	PGCRC15	PGCRC14	PGCRC13	PGCRC12	PGCRC11	PGCRC10	PGCRC9	PGCRC8

7	6	5	4	3	2	1	0
PGCRC7	PGCRC6	PGCRC5	PGCRC4	PGCRC3	PGCRC2	PGCRC1	PGCRC0

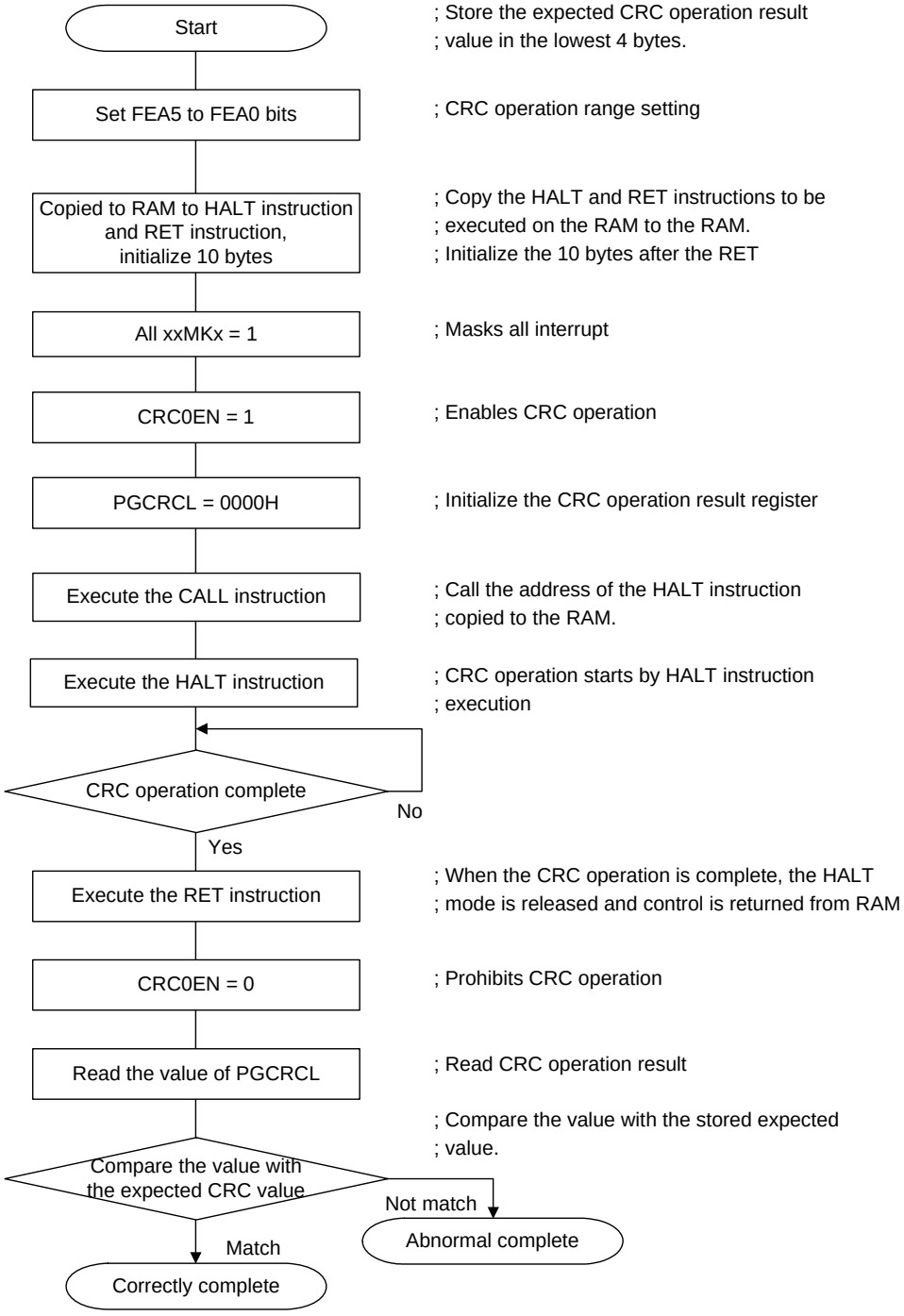
PGCRC15 to 0	High-speed CRC operation results
0000H to FFFFH	Store the high-speed CRC operation results.

Caution The PGCRCL register can only be written if CRC0EN (bit 7 of the CRC0CTL register) = 1.

Figure 27 - 3 shows the Flowchart of Flash Memory CRC Operation Function (High-speed CRC).

<Operation flow>

Figure 27 - 3 Flowchart of Flash Memory CRC Operation Function (High-speed CRC)



Caution 1. The CRC operation is executed only on the code flash.
Caution 2. Store the expected CRC operation value in the area below the operation range in the code flash.
Caution 3. The CRC operation is enabled by executing the HALT instruction in the RAM area.
Be sure to execute the HALT instruction in RAM area.

The expected CRC value can be calculated by using the Integrated Development Environment CubeSuite+. See the Integrated Development Environment CubeSuite+ user's manual for details.

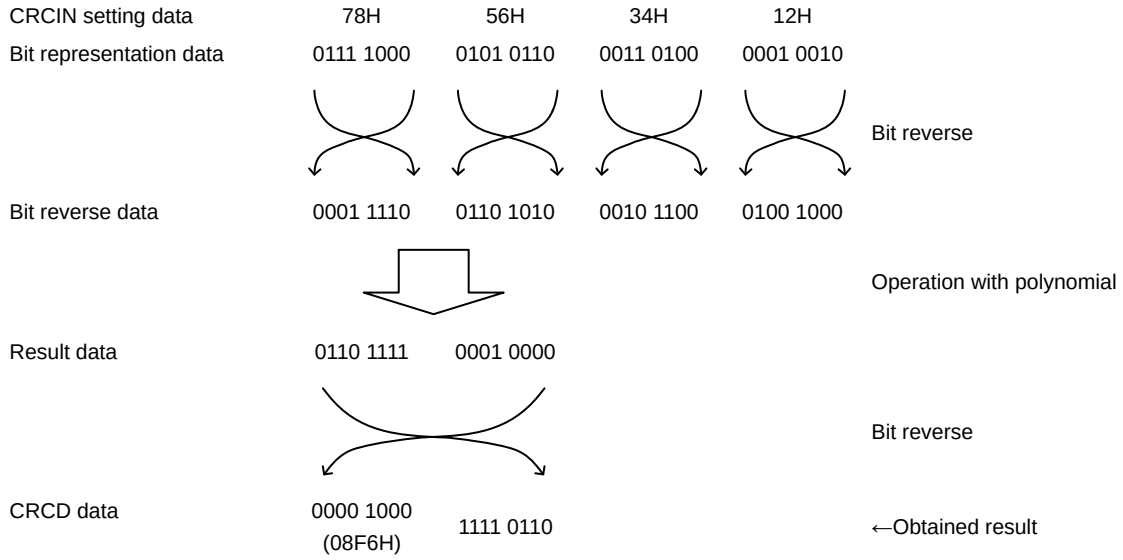
27.3.2 CRC operation function (general-purpose CRC)

<R>

In the RL78/L1C, 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.

27.3.2.1 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 27 - 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.

27.3.2.2 CRC data register (CRCD)

This register is used to store the CRC operation result of the general-purpose CRC.

The possible setting range is 0000H to FFFFH.

After 1 clock of CPU/peripheral hardware clock (f_{CLK}) has elapsed from the time CRCIN register is written, the CRC operation result is stored to the CRCD register.

The CRCD register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 27 - 5 Format of CRC data register (CRCD)

Address: F02FAH After reset: 0000H R/W

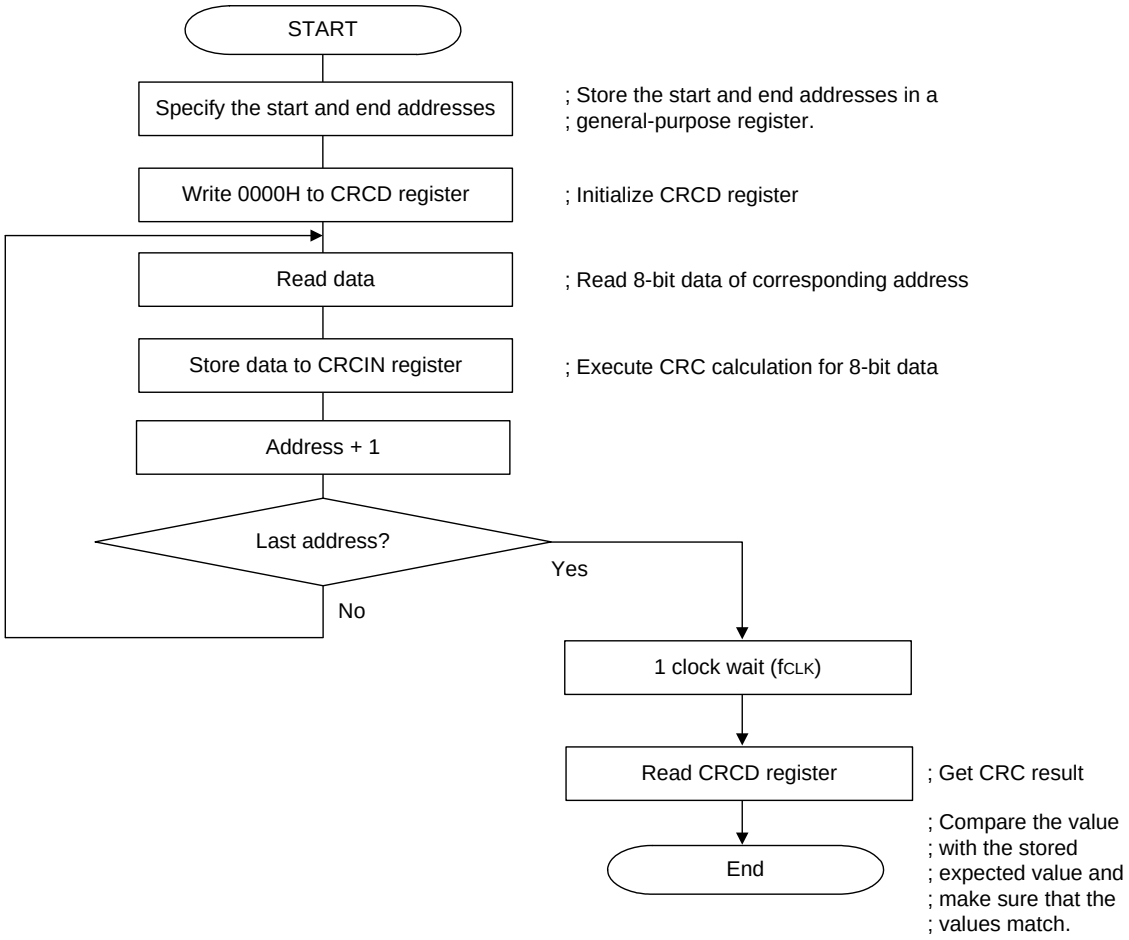
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCD																

Caution 1. Read the value written to CRCD register before writing to CRCIN register.

Caution 2. If writing and storing operation result to CRCD register conflict, the writing is ignored.

<Operation flow>

Figure 27 - 6 CRC Operation Function (General-Purpose CRC)



27.3.3 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/L1C'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.

27.3.3.1 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 27 - 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

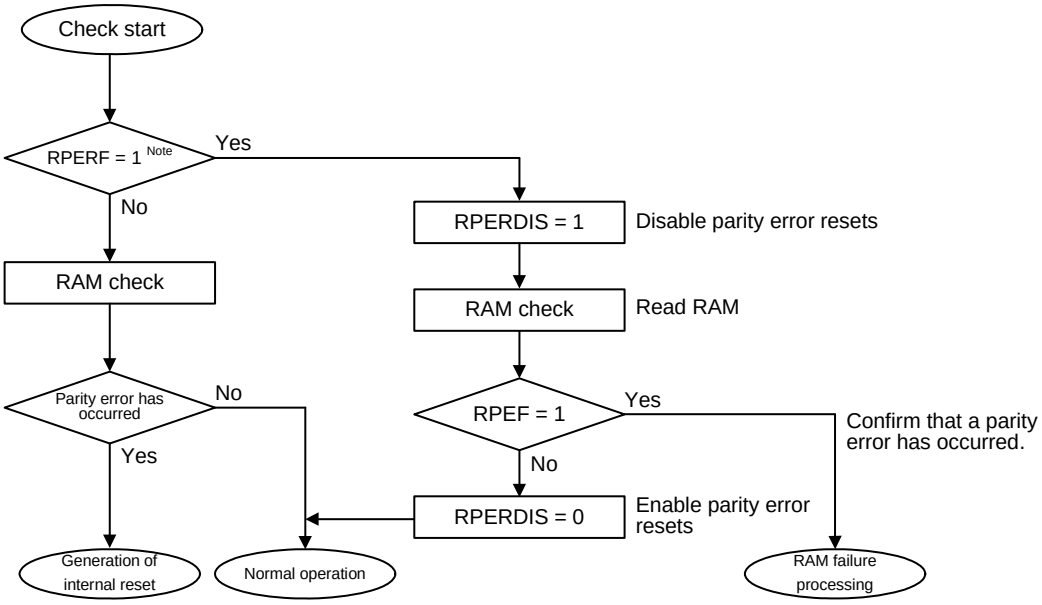
RPERDIS	Parity error reset mask flag
0	Enable parity error resets.
1	Disable parity error resets.

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. When using the self-programming function while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the RAM area to overwrite + 10 bytes before overwriting.

Remark 1. The parity error reset is enabled by default (RPERDIS = 0).
Remark 2. Even if the parity error reset is disabled (RPERDIS = 1), the RPEF flag will be set (1) if a parity error occurs. If parity error resets are enabled (RPERDIS = 0) with RPEF set to 1, a parity error reset is generated when the RPERDIS bit is cleared to 0.
Remark 3. The RPEF flag in the RPECTL register is set (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.
Remark 4. The general registers are not included for RAM parity error detection.

Figure 27 - 8 RAM Parity Error Check Flow



Note See **CHAPTER 24 RESET FUNCTION** for details on how to confirm internal resets due to RAM parity errors.

27.3.4 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.

27.3.4.1 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 27 - 9 Format of Invalid memory access detection control register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GRAM1	GRAM0	RAM guard space Note
0	0	Disabled. RAM can be written to.
0	1	The 128 bytes of space starting at the start address in the RAM
1	0	The 256 bytes of space starting at the start address in the RAM
1	1	The 512 bytes of space starting at the start address in the RAM

Note The RAM start address differs depending on the size of the RAM provided with the product.

27.3.5 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.

27.3.5.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.
GPORT, GINT and GCSC bits are used in SFR guard function.
The IAWCTL register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 27 - 10 Format of Invalid memory access detection control register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GPORT	Control registers of port function guard
0	Disabled. Control registers of port function can be read or written to.
1	Enabled. Writing to control registers of port function is disabled. Reading is enabled. [Guarded SFR] PMxx, PUxx, PIMxx, POMxx, PMCxx, ADPC, PIOR, PFSEGxx, ISCLCD Note

GINT	Registers of interrupt function guard
0	Disabled. Registers of interrupt function can be read or written to.
1	Enabled. Writing to registers of interrupt function is disabled. Reading is enabled. [Guarded SFR] IFxx, MKxx, PRxx, EGPx, EGNx

GCSC	Control registers of clock control function, voltage detector and RAM parity error detection function guard
0	Disabled. Control registers of clock control function, voltage detector and RAM parity error detection function can be read or written to.
1	Enabled. Writing to control registers of clock control function, voltage detector and RAM parity error detection function is disabled. Reading is enabled. [Guarded SFR] CMC, CSC, OSTS, CKC, PERx, OSMC, LVIM, LVIS, RPECTL, DSCCTL

Note Pxx (Port register) and TOS (Timer output select register) are not guarded.

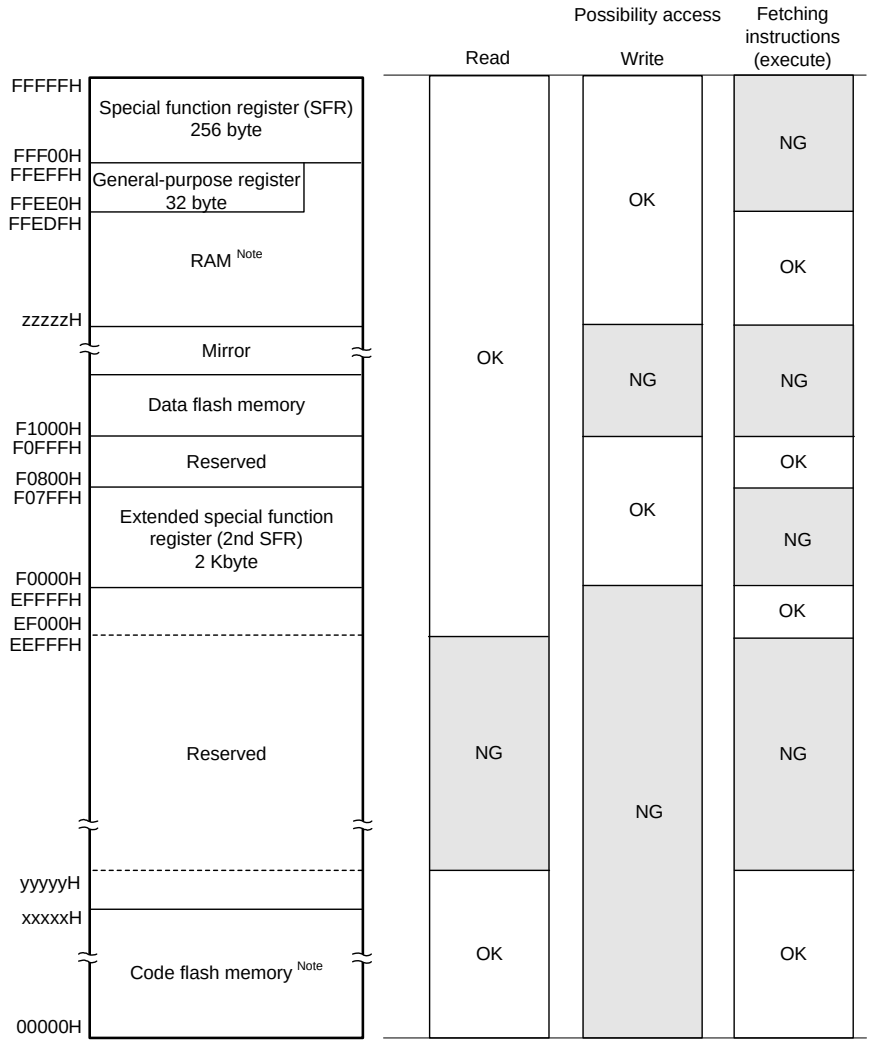
27.3.6 Invalid memory access detection function

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 27 - 11.

Figure 27 - 11 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)
R5F110xE, R5F111xE (x = M, N, P)	65536 × 8 bits (00000H to 0FFFFH)	8192 × 8 bits (FDF00H to FFEFFH)	10000H
R5F110xF, R5F111xF (x = M, N, P)	98304 × 8 bits (00000H to 1FFFFH)	10240 × 8 bits (FD700H to FFEFFH)	20000H
R5F110xG, R5F111xG (x = M, N, P)	131072 × 8 bits (00000H to 1FFFFH)	12288 × 8 bits (FCF00H to FFEFFH)	20000H
R5F110xH, R5F111xH (x = M, N, P)	196608 × 8 bits (00000H to 2FFFFH)	16384 × 8 bits (FBF00H to FFEFFH)	30000H
R5F110xJ, R5F111xJ (x = M, N, P)	262144 × 8 bits (00000H to 3FFFFH)	16384 × 8 bits (FBF00H to FFEFFH)	40000H

27.3.6.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

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 27 - 12 Format of Invalid memory access detection control register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN Note	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

IAWEN Note	Control of invalid memory access detection
0	Disable the detection of invalid memory access.
1	Enable the detection of invalid memory access.

Note Only writing 1 to the IAWEN bit is enabled, not writing 0 to it after setting it to 1.

Remark By specifying WDTON = 1 (watchdog timer operation enabled) in the option byte (000C0H), the invalid memory access detection function is enabled even IAWEN = 0.

27.3.7 Frequency detection function

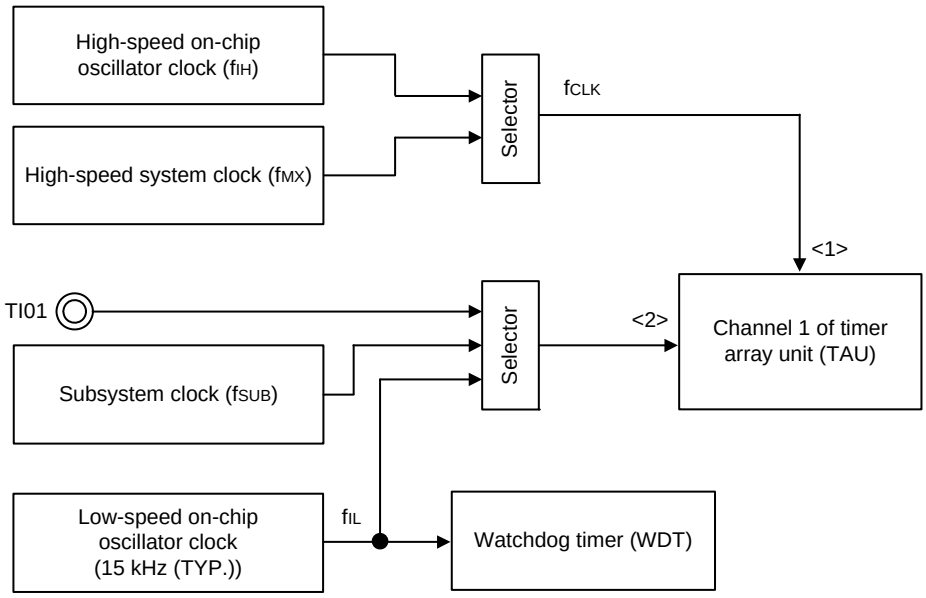
The IEC60730 standard mandates checking that the oscillation frequency is correct.
By using the CPU/peripheral hardware clock frequency (f_{CLK}) and measuring the pulse width of the input signal to channel 1 of the timer array unit (TAU), whether the proportional relationship between the two clock frequencies is correct can be determined. Note that, however, if one or both clock operations are completely stopped, the proportional relationship between the clocks cannot be determined.

<Clocks to be compared>

<1> CPU/peripheral hardware clock frequency (f_{CLK}):
- High-speed on-chip oscillator clock (f_{IH})
- High-speed system clock (f_{MX})

<2> Input to channel 1 of the timer array unit
- Timer input to channel 1 (TI01)
- Low-speed on-chip oscillator clock (f_{IL}: 15 kHz (typ.))
- Subsystem clock (f_{SUB})

Figure 27 - 13 Configuration of Frequency Detection Function



If pulse interval measurement results in an abnormal value, it can be concluded that the clock frequency is abnormal.
For how to execute pulse interval measurement, see **6.8.3 Operation as input pulse interval measurement**.

27.3.7.1 Timer input select register 0 (TIS0)

This register is used to select the timer input of channels 0, 1.

By selecting the low-speed on-chip oscillator clock for the timer input, its pulse width can be measured to determine whether the proportional relationship between the low-speed on-chip oscillator clock and the timer operation clock is correct.

The TIS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 27 - 14 Format of Timer input select register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	TIS04	0	TIS02	TIS01	TIS00

TIS04	Selection of timer input used with channel 0
0	Input signal of timer input pin (TI00)
1	Event input signal from ELC

TIS02	TIS01	TIS00	Selection of timer input used with channel 1
0	0	0	Input signal of timer input pin (TI01)
0	1	0	
0	1	1	
0	0	1	Event input signal from ELC
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
1	0	1	Subsystem clock (f_{SUB})
Other than above			Setting prohibited

27.3.8 A/D test function

The IEC60730 standard mandates testing the A/D converter. The A/D test function is used to check whether the A/D converter is operating normally by executing A/D conversions of the positive reference voltage and negative reference voltage of the A/D converter, analog input channel (ANI), temperature sensor output voltage, and internal reference voltage. For details on the checking method, refer to the safety function (A/D test) application note (R01AN0955).

The analog multiplexer can be checked using the following procedure.

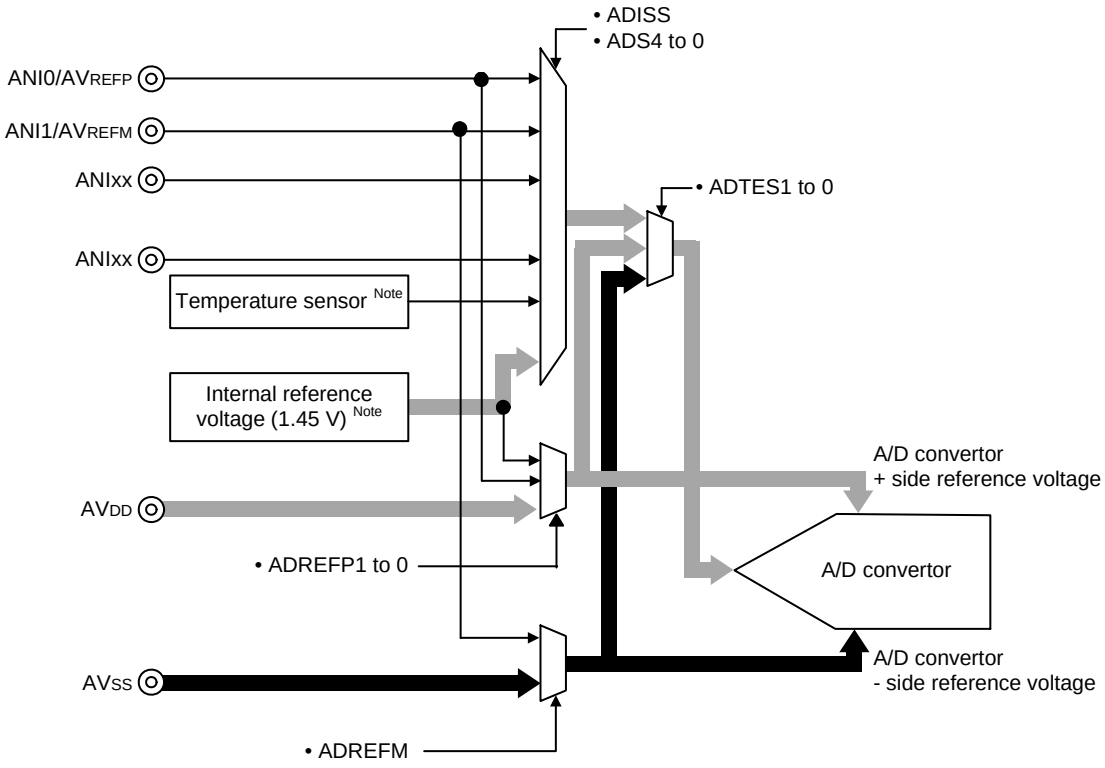
(1) Select the ANIx pin as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 0, 0).
(2) Perform A/D conversion for the ANIx pin (conversion result 1-1).
(3) Select the negative reference voltage of the A/D converter as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 1, 0).
(4) Perform A/D conversion of the negative reference voltage of the A/D converter (conversion result 2-1).
(5) Select the ANIx pin as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 0, 0).
(6) Perform A/D conversion for the ANIx pin (conversion result 1-2).
(7) Select the positive reference voltage of the A/D converter as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 1, 1).
(8) Perform A/D conversion of the positive reference voltage of the A/D converter (conversion result 2-2).
(9) Select the ANIx pin as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 0, 0).
(10) Perform A/D conversion for the ANIx pin (conversion result 1-3).
(11) Make sure that “conversion result 1-1” = “conversion result 1-2” = “conversion result 1-3”.
(12) Make sure that the A/D conversion results of “conversion result 2-1” are all 0 and those of “conversion result 2-2” are all 1.

Using the procedure above can confirm that the analog multiplexer is selected and all wiring is connected.

Remark 1. If the analog input voltage is variable during conversion in steps (1) to (10) above, use another method to check the analog multiplexer.

Remark 2. The conversion results might contain an error. Consider an appropriate level of error when comparing the conversion results.

Figure 27 - 15 Configuration of A/D Test Function



Note This setting can be used only in HS (high-speed main) mode.

27.3.8.1 A/D test register (ADTES)

This register is used to select the A/D converter positive reference voltage, 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 the negative reference voltage as the target of A/D conversion when measuring the zero-scale.
- Select the positive reference voltage as the target of A/D conversion when measuring the full-scale.

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

Reset signal generation clears this register to 00H.

Figure 27 - 16 Format of A/D test register (ADTES)

Address: F0013H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	0	ADTES1	ADTES0

ADTES1	ADTES0	A/D conversion target
0	0	ANIxx/temperature sensor output voltage Note/internal reference voltage (1.45 V) Note (This is specified using the analog input channel specification register (ADS).)
1	0	Negative reference voltage (selected by the ADREFM bit in the ADM2 register)
1	1	Positive reference voltage (selected by the ADREFP1 and ADREFP0 bits in the ADM2 register)
Other than the 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.

27.3.8.2 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 27 - 17 Format of Analog input channel specification register (ADS)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P150/ANI0/AV_{REFP} pin
0	0	0	0	0	1	ANI1	P151/ANI1/AV_{REFM} pin
0	0	0	0	1	0	ANI2	P152/ANI2 pin
0	0	0	0	1	1	ANI3	P153/ANI3 pin
0	0	0	1	0	0	ANI4	P154/ANI4 pin
0	0	0	1	0	1	ANI5	P155/ANI5 pin
0	0	0	1	1	0	ANI6	P156/ANI6 pin
0	1	0	0	0	0	ANI16	P140/ANI16 pin
0	1	0	0	0	1	ANI17	P141/ANI17 pin
0	1	0	0	1	0	ANI18	P142/ANI18 pin
0	1	0	0	1	1	ANI19	P143/ANI19 pin
0	1	0	1	0	0	ANI20	P20/ANI20 pin
0	1	0	1	0	1	ANI21	P21/ANI21 pin
1	0	0	0	0	0	—	Temperature sensor output voltage Note
1	0	0	0	0	1	—	Internal reference voltage output (1.45 V) Note
Other than the above						Setting prohibited	

Note This setting can be used only in HS (high-speed main) mode.

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

Caution 2. For ports that set to analog input using the ADPC and PMC registers, select input mode using port mode register 2, 14, or 15 (PM2, PM14, PM15).

Caution 3. Do not use the ADS register to set ports that to be set as digital I/O using the A/D port configuration register (ADPC).

Caution 4. Do not use the ADS register to set ports that to be set as digital I/O using port mode control register 2 or 14 (PMC2, PMC14).

Caution 5. Only rewrite the value of the ADISS bit while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 6. When using AV_{REFP} as the positive reference voltage of the A/D converter, do not select ANI0 as an A/D conversion channel.

Caution 7. When using AV_{REFM} as the negative reference voltage of the A/D converter, do not select ANI1 as an A/D conversion channel.

Caution 8. If ADISS is set to 1, the internal reference voltage output (1.45 V) cannot be used for the positive reference voltage. Also, the first conversion result cannot be used after ADISS is set to 1. For details on the setup flow, see 12.7.4 Setup when temperature sensor output/internal reference voltage output is selected.

Caution 9. Do not set ADISS to 1 when entering HALT mode while in STOP mode or while the CPU operates on the subsystem clock. With ADISS = 1, the current value of the A/D converter reference voltage current (I_{ADREF}) listed in 34.3.2 or 35.3.2 Supply current characteristics is added.

27.3.9 Digital output signal level detection function for I/O ports

In the IEC60730, it is required to check that the I/O function correctly operates.

By using the digital output signal level detection function for I/O ports, the digital output level of the pin can be read when the port is set to output mode (the PMmn bit in the port mode register (PMm) is 0).

27.3.9.1 Port mode select register (PMS)

This register is used to select the output level from output latch level or pin output level when the port is output mode in which PMm bit of port mode register (PMm) is 0.

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

Reset signal generation clears these registers to 00H.

Figure 27 - 18 Format of Port mode select register (PMS)

Address: F007BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PMS	0	0	0	0	0	0	0	PMS0

PMS0	Method for selecting output level to be read when port is output mode (PMmn = 0)
0	Pmn register value is read.
1	Digital output level of the pin is read.

Caution 1. When setting the PMS0 bit to 1 and rewriting the port register (Pm register), use an 8-bit memory manipulation instruction only.

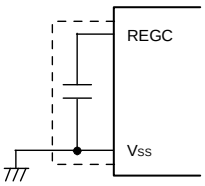
Caution 2. When using P60 and P61 as general-purpose ports, the output level of these pins cannot be read by setting PMS0 (However, only when the IICA0EN bit in the PER0 register is set to 1, the output level of P60 and P61 can be read by setting the PMS0 bit).

Remark m = 0 to 8, 12, 14, 15
n = 0 to 7

CHAPTER 28 REGULATOR

28.1 Regulator Overview

The RL78/L1C contains a circuit for operating the device with a constant voltage. At this time, in order to stabilize the regulator output voltage, connect the REGC pin to Vss via a capacitor (0.47 to 1 μF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.



Caution Keep the wiring length as short as possible for the broken-line part in the above figure.

Table 28 - 1 Regulator Output Voltage Conditions

Mode	Output Voltage	Condition
LV (low voltage main) mode	1.8 V	—
LS (low-speed main) mode		
HS (high-speed main) mode	1.8 V	In STOP mode
		When both the high-speed system clock (fMX) and the high-speed on-chip oscillator clock (fHOCO) are stopped during CPU operation with the subsystem clock (fSUB)
		When both the high-speed system clock (fMX) and the high-speed on-chip oscillator clock (fHOCO) are stopped during the HALT mode when the CPU operation with the subsystem clock (fSUB) 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 29 OPTION BYTE

29.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the RL78/L1C 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.

Remark The option bytes should always be set regardless of whether each function is used.

29.1.1 User option byte (000C0H to 000C2H/010C0H to 010C2H)

(1) 000C0H/010C0H

- ○ Operation of watchdog timer
  - • Enabling or disabling of counter operation
  - • Enabling or disabling of counter operation in the HALT or STOP mode
- ○ Setting of interval time of watchdog timer
- ○ Setting of window open period of watchdog timer
- ○ 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

○ 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)

○ Setting of LVD detection level (V_{LVDH}, V_{LVDL}, V_{LVD})

Caution 1. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 34.4 or 35.4 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).

Caution 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

○ Setting of flash operation mode

Make the setting depending on the main system clock frequency (f_{MAIN}) and power supply voltage (V_{DD}) to be used.
- LV (low voltage main) mode
- LS (low speed main) mode
- HS (high speed main) mode

○ Setting of the frequency of the high-speed on-chip oscillator
- Select from 48 MHz, 24 MHz, 16 MHz, 12 MHz, 8 MHz, 6 MHz, 4 MHz, 3 MHz, 2 MHz, and 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.

29.1.2 On-chip debug option byte (000C3H/010C3H)

○ Control of on-chip debug operation
- On-chip debug operation is disabled or enabled.

○ Handling of data of flash memory in case of failure in on-chip debug security ID authentication
- Data of flash memory is erased or not erased in case of failure in on-chip debug security ID authentication.

Caution Set the same value as 000C3H to 010C3H when the boot swap operation is used because 000C3H is replaced by 010C3H.

29.2 Format of User Option Byte

The format of user option byte is shown below.

Figure 29 - 1 Format of User Option Byte (000C0H/010C0H)

Address: 000C0H/010C0H Note 1

7	6	5	4	3	2	1	0
WDTINT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINT	Use of interval interrupt of watchdog timer
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% + 1/2fIL 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 (fIL = 17.25 kHz (MAX.))
0	0	0	2^6/fIL (3.71 ms)
0	0	1	2^7/fIL (7.42 ms)
0	1	0	2^8/fIL (14.84 ms)
0	1	1	2^9/fIL (29.68 ms)
1	0	0	2^{11}/fIL (118.72 ms)
1	0	1	2^{13}/fIL (474.89 ms)
1	1	0	2^{14}/fIL (949.79 ms)
1	1	1	2^{16}/fIL (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

Note 1. Set the same value as 000C0H to 010C0H when the boot swap operation is used because 000C0H is replaced by 010C0H.

Note 2. The window open period is 100% when WDSTBYON = 0, regardless the value of the WINDOW1 and WINDOW0 bits.

Note 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	$2^{13}/f_{IL}$ (474.90 ms)	237.44 ms to 321.26 ms
1	1	0	$2^{14}/f_{IL}$ (949.80 ms)	474.89 ms to 642.51 ms
1	1	1	$2^{16}/f_{IL}$ (3799.19 ms)	1899.59 ms to 2570.04 ms

Remark f_{IL}: Low-speed on-chip oscillator clock frequency

Figure 29 - 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
1.77 V	1.73 V	1.63 V	0	0	0	1	0	1	0
1.88 V	1.84 V					0	1		
2.92 V	2.86 V					0	0		
1.98 V	1.94 V	1.84 V		0	1	1	0		
2.09 V	2.04 V					0	1		
3.13 V	3.06 V					0	0		
2.61 V	2.55 V	2.45 V		1	0	1	0		
2.71 V	2.65 V					0	1		
2.92 V	2.86 V	2.75 V		1	1	1	0		
3.02 V	2.96 V					0	1		
—			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
1.67 V	1.63 V	0	0	0	1	1	1	1
1.77 V	1.73 V				1	0		
1.88 V	1.84 V		0	1	1	1		
1.98 V	1.94 V				1	0		
2.09 V	2.04 V				0	1		
2.50 V	2.45 V		1	0	1	1		
2.61 V	2.55 V				1	0		
2.71 V	2.65 V				0	1		
2.81 V	2.75 V		1	1	1	1		
2.92 V	2.86 V				1	0		
3.02 V	2.96 V				0	1		
3.13 V	3.06 V		0	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.

Remark 1. For details on the LVD circuit, see **CHAPTER 26 VOLTAGE DETECTOR**.

Remark 2. The detection voltage is a typical value. For details, see **34.6.6** or **35.6.6 LVD circuit characteristics**. (**Cautions** are listed on the next page.)

Figure 29 - 3 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 setting (interrupt mode)

Detection voltage		Option byte Setting Value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
1.67 V	1.63 V	0	0	0	1	1	0	1
1.77 V	1.73 V				1	0		
1.88 V	1.84 V		0	1	1	1		
1.98 V	1.94 V				1	0		
2.09 V	2.04 V				0	1		
2.50 V	2.45 V		1	0	1	1		
2.61 V	2.55 V				1	0		
2.71 V	2.65 V				0	1		
2.81 V	2.75 V		1	1	1	1		
2.92 V	2.86 V				1	0		
3.02 V	2.96 V				0	1		
3.13 V	3.06 V		0	1	0	0		
—		Setting of values other than above is prohibited						

• LVD off (by controlling the externally input reset signal on the $\overline{\text{RESET}}$ pin)

Detection voltage		Option byte Setting Value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
—	—	1	×	×	×	×	×	1
—		Setting of values other than above is prohibited						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Caution 1. Be sure to set bit 4 to "1".

Caution 2. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 34.4 or 35.4 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).

Remark 1. ×: don't care

Remark 2. For details on the LVD circuit, see **CHAPTER 26 VOLTAGE DETECTOR**.

Remark 3. The detection voltage is a typical value. For details, see **34.6.6** or **35.6.6 LVD circuit** characteristics.

Figure 29 - 4 Format of User Option Byte (000C2H/010C2H)

Address: 000C2H/010C2H Note 1

7	6	5	4	3	2	1	0
CMODE1	CMODE0	1	FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

CMODE1	CMODE0	Setting of flash operation mode		
			Operating Frequency Range (f_{MAIN})	Operating Voltage Range (V_{DD})
0	0	LV (low voltage main) mode	1 to 4 MHz	1.6 to 3.6 V
1	0	LS (low speed main) mode	1 to 8 MHz	1.8 to 3.6 V
1	1	HS (high speed main) mode	1 to 16 MHz	2.4 to 3.6 V
			1 to 24 MHz	2.7 to 3.6 V
Other than above		Setting prohibited		

FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator clock	
					f_{HOCO}	f_{IH}
1	0	0	0	0	48 MHz Note 3	24 MHz, 12 MHz, 6 MHz Note 2
0	0	0	0	0	24 MHz	24 MHz
0	1	0	0	1	16 MHz	16 MHz
0	0	0	0	1	12 MHz	12 MHz
0	1	0	1	0	8 MHz	8 MHz
0	0	0	1	0	6 MHz	6 MHz
0	1	0	1	1	4 MHz	4 MHz
0	0	0	1	1	3 MHz	3 MHz
0	1	1	0	0	2 MHz	2 MHz
0	1	1	0	1	1 MHz	1 MHz
Other than above					Setting prohibited	

Note 1. Set the same value as 000C2H to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

Note 2. See the MCKC register for f_{IH} division ratio settings.

Note 3. When using the high-speed on-chip oscillator clock (f_{HOCO}) to operate the USB/function controller, be sure to set f_{HOCO} = 48 MHz.

29.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 29 - 5 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 write 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.

29.4 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      1AH     ; Select 1.63 V for VLVDL
                        ; Select rising edge 1.77 V, falling edge 1.73 V for VLVDH
                        ; Select the interrupt & reset mode as the LVD operation mode
        DB      2DH     ; Select the LV (low voltage main) mode as the flash operation mode
                          and 1 MHz as the frequency of the high-speed on-chip oscillator clock
        DB      85H     ; Enables on-chip debug operation, does not erase flash memory
                          data when security ID authorization fails
```

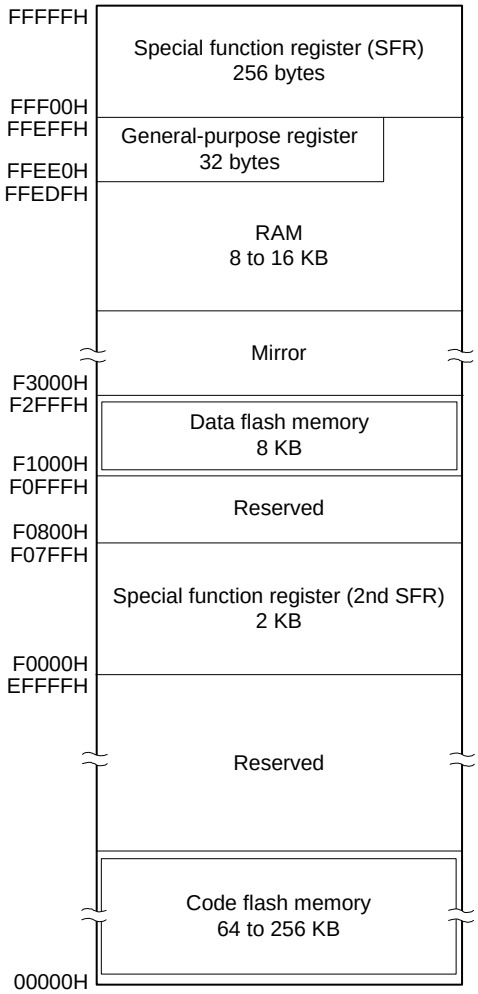
When the boot swap function is used during self programming, 000C0H to 000C3H is switched to 010C0H to 010C3H. Describe to 010C0H to 010C3H, therefore, the same values as 000C0H to 000C3H as follows.

```
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      1AH     ; Select 1.63 V for VLVDL
                        ; Select rising edge 1.77 V, falling edge 1.73 V for VLVDH
                        ; Select the interrupt & reset mode as the LVD operation mode
        DB      2DH     ; Select the LV (low main voltage) mode as the flash operation mode
                          and 1 MHz as the frequency of the high-speed on-chip oscillator clock
        DB      85H     ; Enables on-chip debug operation, does not erase flash memory
                          data when security ID authorization fails
```

Caution To specify the option byte by using assembly language, use OPT_BYTE as the relocation attribute name of the CSEG pseudo instruction. 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 30 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 an external device (UART communication), or through self-programming.

- Serial programming using flash memory programmer (see **30.4**)
  Data can be written to the flash memory on-board or off-board by using a dedicated flash memory programmer.

- Serial programming using external device (UART communication) (see **30.2**)
  Data can be written to the flash memory on-board through UART communication with an external device (microcontroller or ASIC).

- Self-programming (see **30.6**)
  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 **30.8 Data Flash**.

30.1 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-FP6, FL-PR6
- E1, E2, E2 Lite, E20 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-PR6 and FA series are products of Naito Densei Machida Mfg. Co., Ltd.

Table 30 - 1 Wiring Between RL78/L1C and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin No.		
					80-pin	85-pin	100-pin
Signal Name		I/O	Pin Function		LQFP (12 × 12)	LGA (7 × 7)	LQFP (14 × 14)
PG-FP6, FL-PR6	E1, E2, E2 Lite, E20 on-chip debugging emulator						
—	TOOL0	I/O	Transmit/receive signal	TOOL0/ P40	9	E8	12
SI/RxD	—	I/O	Transmit/receive signal				
—	$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	10	F9	13
/RESET	—	Output					
V_{DD}		I/O	V_{DD} voltage generation/power monitoring	V_{DD}	18	C10	21, 90
GND		—	Ground	V_{SS}	17	F10	20, 91
				REGC Note	16	D9	19
FLMD1	EMV_{DD}	—	Driving power for TOOL pin	V_{DD}	18	C10	21, 90

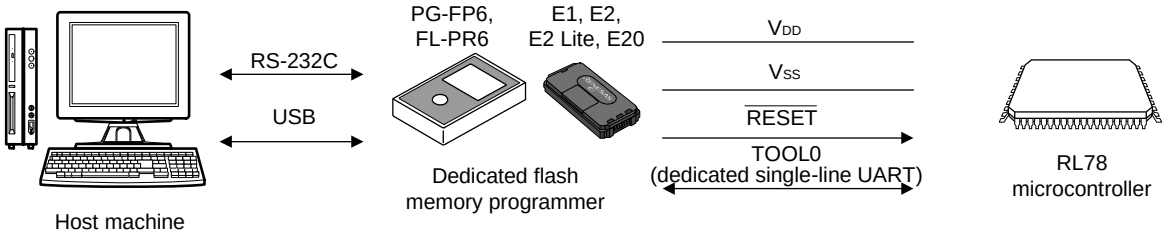
Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

Remark Pins that are not indicated in the above table can be left open when using the flash memory programmer for flash programming.

30.1.1 Programming Environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 30 - 1 Environment for Writing Program to Flash Memory



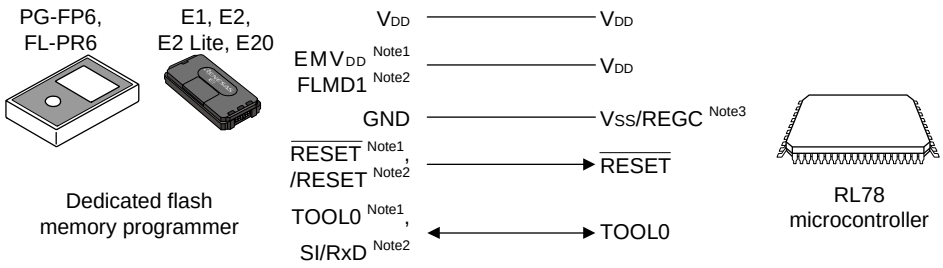
A host machine that controls the dedicated flash memory programmer is necessary.
To interface between the dedicated flash memory programmer and the RL78 microcontroller, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART.

30.1.2 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 30 - 2 Communication with Dedicated Flash Memory Programmer



Note 1. When using E1, E2, E2 Lite, E20 on-chip debugging emulator.
Note 2. When using PG-FP6 or FL-PR6.
Note 3. Connect REGC pin to ground via a capacitor (0.47 to 1 µF).

The dedicated flash memory programmer generates the following signals for the RL78 microcontroller. See each manual of PG-FP6, FL-PR6, or E1, E2, E2 Lite, E20 on-chip debugging emulator for details.

Table 30 - 2 Pin Connection

Dedicated Flash Memory Programmer				RL78 microcontroller
Signal Name		I/O	Pin Function	Pin Name Note 2
PG-FP6, FL-PR6	E1, E2, E2 Lite, E20 on-chip debugging emulator			
V_{DD}		I/O	V_{DD} voltage generation/power monitoring	V_{DD}
GND		—	Ground	V_{SS}, REGC Note 1
FLMD1	EMVDD	—	Driving power for TOOL0 pin	V_{DD}
/RESET	—	Output	Reset signal	$\overline{\text{RESET}}$
—	$\overline{\text{RESET}}$	Output		
—	TOOL0	I/O	Transmit/receive signal	TOOL0
SI/RxD	—	I/O	Transmit/receive signal	

Note 1. Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

Note 2. Pins to be connected differ with the product. For details, see **Table 30 - 1**.

30.2 Serial Programming Using External Device (that Incorporates UART)

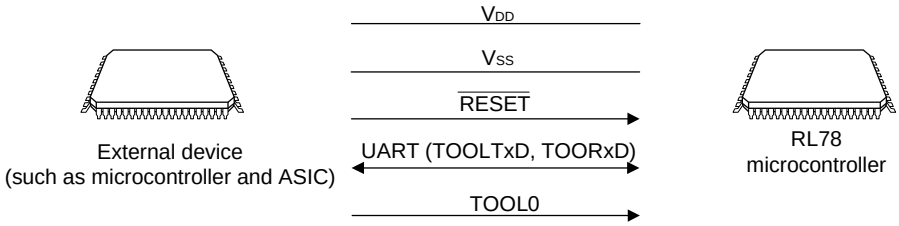
On-board data writing to the internal flash memory is possible by using the RL78 microcontroller and an external device (a microcontroller or ASIC) connected to a UART.

On the development of flash memory programmer by user, refer to **RL78 microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

30.2.1 Programming Environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 30 - 3 Environment for Writing Program to Flash Memory



Processing to write data to or delete data from the RL78 microcontroller by using an external device is performed on-board. Off-board writing is not possible.

30.2.2 Communication Mode

Communication between the external device and the RL78 microcontroller is established by serial communication using the TOOLTxD and TOOLRxD pins via the dedicated UART of the RL78 microcontroller.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 30 - 4 Communication with External Device

Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

The external device generates the following signals for the RL78 microcontroller.

Table 30 - 3 Pin Connection

External Device			RL78 microcontroller
Signal Name	I/O	Pin Function	Pin Name
VDD	I/O	VDD voltage generation/power monitoring	VDD
GND	—	Ground	VSS, REGC Note
RESETOUT	Output	Reset signal output	$\overline{\text{RESET}}$
RxD	Input	Receive signal	TOOLTxD
TxD	Output	Transmit signal	TOOLRxD
PORT	Output	Mode signal	TOOL0

Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

30.3 Connection of Pins on Board

To write the flash memory on-board by using the flash memory programmer, connectors that connect the dedicated flash memory programmer must be provided on the target system. First provide a function that selects the normal operation mode or flash memory programming mode on the board.

When the flash memory programming mode is set, all the pins not used for programming the flash memory are in the same status as immediately after reset. Therefore, if the external device does not recognize the state immediately after reset, the pins must be handled as described below.

Remark For the flash programming mode, see **30.4.2 Flash memory programming mode**.

30.3.1 P40/TOOL0 pin

In the flash memory programming mode, connect this pin to the dedicated flash memory programmer via an external 1 kΩ pull-up resistor.

When this pin is used as the port pin, use that by the following method.

When used as an input pin:	Input of low-level is prohibited for t_{HD} period after external pin reset release. However, when this pin is used via pull-down resistors, use the 500 kΩ or more resistors.
When used as an output pin:	When this pin is used via pull-down resistors, use the 500 kΩ or more resistors.

Remark 1. t_{HD}: How long to keep the TOOL0 pin at the low level from when the external and internal resets end for setting of the flash memory programming mode (see **34.12** or **35.12 Timing of Entry to Flash Memory Programming Modes**)

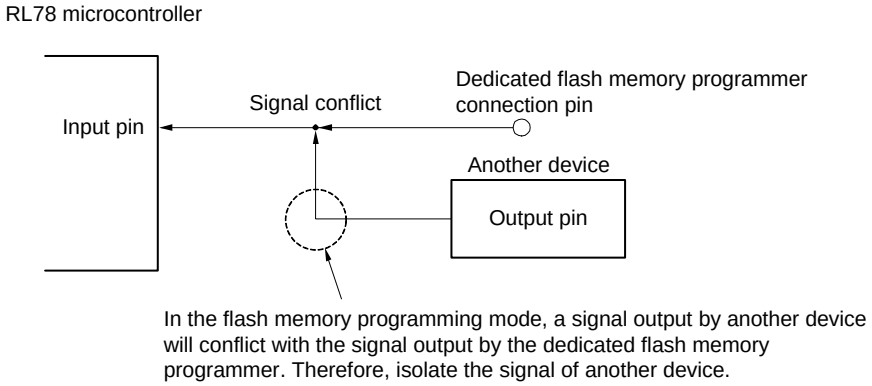
Remark 2. The SAU and IICA pins are not used for communication between the RL78 microcontroller and dedicated flash memory programmer, because single-line UART (TOOL0 pin) is used.

30.3.2 $\overline{\text{RESET}}$ pin

Signal conflict will occur if the reset signal of the dedicated flash memory programmer and external device are connected to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on the board. To prevent this conflict, isolate the connection with the reset signal generator.

The flash memory will not be correctly programmed if the reset signal is input from the user system while the flash memory programming mode is set. Do not input any signal other than the reset signal of the dedicated flash memory programmer and external device.

Figure 30 - 5 Signal Conflict ($\overline{\text{RESET}}$ Pin)



30.3.3 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.

30.3.4 REGC pin

Connect the REGC pin to GND via a capacitor having excellent characteristics (0.47 to 1 μF) in the same manner as during normal operation. Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.

30.3.5 X1 and X2 pins

Connect X1 and X2 in the same status as in the normal operation mode.

Remark In the flash memory programming mode, the high-speed on-chip oscillator clock (f_{HOCO}) is used.

30.3.6 Power supply

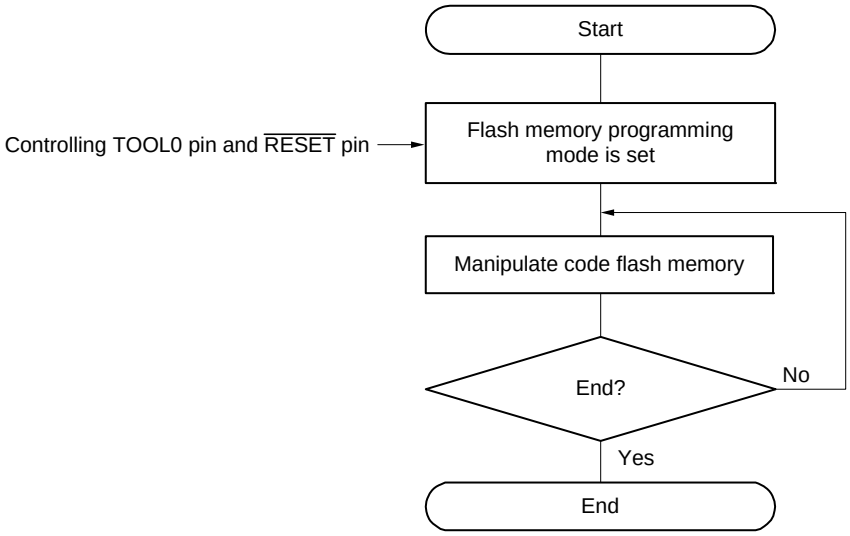
To use the supply voltage output of the flash memory programmer, connect the V_{DD} pin to V_{DD} of the flash memory programmer, and the V_{SS} pin to GND of the flash memory programmer.
To use the on-board supply voltage, connect in compliance with the normal operation mode.
However, when writing to the flash memory by using the flash memory programmer and using the on-board supply voltage, be sure to connect the V_{DD} and V_{SS} pins to V_{DD} and GND of the flash memory programmer to use the power monitor function with the flash memory programmer.

30.4 Serial Programming Method

30.4.1 Serial programming procedure

The following figure illustrates a flow for rewriting the code flash memory through serial programming.

Figure 30 - 6 Code Flash Memory Manipulation Procedure



30.4.2 Flash memory programming mode

To rewrite the contents of the code flash memory through serial programming, specify the flash memory programming mode. To enter the mode, set as follows.

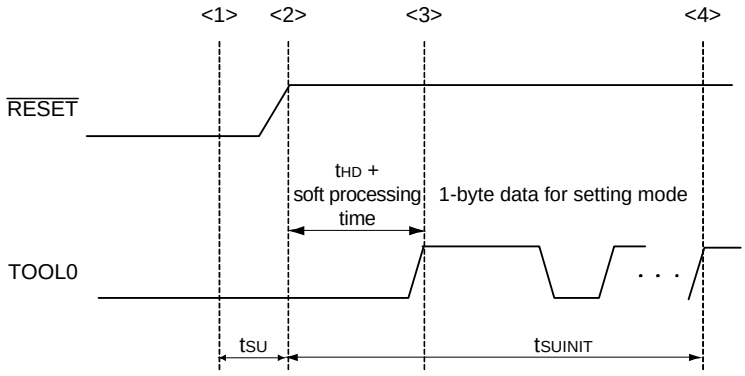
<Serial programming using the dedicated flash memory programmer>
Connect the RL78 microcontroller to a dedicated flash memory programmer. Communication from the dedicated flash memory programmer is performed to automatically switch to the flash memory programming mode.

<Serial programming using an external device (UART communication)>
Set the TOOL0 pin to the low level, and then cancel the reset (see **Table 30 - 4**). After that, enter flash memory programming mode according to the procedures <1> to <4> shown in **Figure 30 - 7**. For details, refer to **RL78 Microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

Table 30 - 4 Relationship Between TOOL0 Pin and Operation Mode After Reset Release

TOOL0	Operation Mode
V_{DD}	Normal operation mode
0 V	Flash memory programming mode

Figure 30 - 7 Setting of Flash Memory Programming Mode



<1> The low level is input to the TOOL0 pin.
<2> The pins 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 a pin reset ends
t_{HD}: How long to keep the TOOL0 pin at the low level from when the external and internal resets end (the flash firmware processing time is excluded)

For details, see **34.12** or **35.12 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 30 - 5 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} ≤ 3.6 V	Blank state		Full speed mode
	HS (high speed main) mode	1 MHz to 24 MHz	Full speed mode
	LS (low speed main) mode	1 MHz to 8 MHz	Wide voltage mode
	LV (low voltage main) mode	1 MHz to 4 MHz	Wide voltage mode
2.4 V ≤ V_{DD} < 2.7 V	Blank state		Full speed mode
	HS (high speed main) mode	1 MHz to 16 MHz	Full speed mode
	LS (low speed main) mode	1 MHz to 8 MHz	Wide voltage mode
	LV (low voltage main) mode	1 MHz to 4 MHz	Wide voltage mode
1.8 V ≤ V_{DD} < 2.4 V	Blank state		Wide voltage mode
	LS (low speed main) mode	1 MHz to 8 MHz	Wide voltage mode
	LV (low voltage main) mode	1 MHz to 4 MHz	Wide voltage mode

Remark 1. Using both the wide voltage mode and full speed mode imposes no restrictions on writing, deletion, or verification.

Remark 2. For details about communication commands, see **30.4.4 Communication commands**.

30.4.3 Selecting communication mode

Communication mode of the RL78 microcontroller as follows.

Table 30 - 6 Communication Modes

Communication Mode	Standard Setting Note 1				Pins Used
	Port	Speed Note 2	Frequency	Multiply Rate	
1-line mode (when flash memory programmer is used, or when external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	—	—	TOOL0
Dedicated UART (when external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	—	—	TOOLTxD, TOOLRxD

Note 1. Selection items for Standard settings on GUI of the flash memory programmer.

Note 2. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

30.4.4 Communication commands

The RL78 microcontroller executes serial programming through the commands listed in **Table 30 - 7**.

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 **RL78 Microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

Table 30 - 7 Flash Memory Control Commands

Classification	Command Name	Function
Verify	Verify	Compares the contents of a specified area of the flash memory with data transmitted from the programmer.
Erase	Block Erase	Erases a specified area in the flash memory.
Blank check	Block Blank Check	Checks if a specified block in the flash memory has been correctly erased
Write	Programming	Writes data to a specified area in the flash memory.Note
Getting information	Silicon Signature	Gets the RL78 microcontroller information (such as the part number, flash memory configuration, and programming firmware version).
	Checksum	Gets the checksum data for a specified area.
Security	Security Set	Sets security information.
	Security Get	Gets security information.
	Security Release	Release setting of prohibition of writing.
Others	Reset	Used to detect synchronization status of communication.
	Baud Rate Set	Sets baud rate when UART communication mode is selected.

Note Confirm that no data has been written to the write area. Because data cannot be erased after block erase is prohibited, do not write data if the data has not been erased.

Product information (such as product name and firmware version) can be obtained by executing the “Silicon Signature” command.

Tables 30 - 8 and 30 - 9 show signature data list and example of signature data list.

Table 30 - 8 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 F2FFFH (8 KB) → FFH, 2FH, 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 30 - 9 Signature Data List

Field name	Description	Number of transmit data	Data (hexadecimal)
Device code	RL78 protocol A	3 bytes	10 00 06
Device name	R5F110PE	10 bytes	52 = “R” 35 = “5” 46 = “F” 31 = “1” 31 = “1” 30 = “0” 50 = “P” 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 F2FFFH (8 KB)	3 bytes	FF 2F 0F
Firmware version	Ver.1.23	3 bytes	01 02 03

30.5 Processing Time for Each Command When PG-FP6 Is in Use (Reference Value)

The following shows the processing time for each command (reference value) when PG-FP6 is used as a dedicated flash memory programmer.

Table 30 - 10 Processing Time for Each Command When PG-FP6 Is in Use (Reference Value)

PG-FP6 Command	Code Flash				
	64 Kbytes	96 Kbytes	128 Kbytes	192 Kbytes	256 Kbytes
Erasing	1.5 s	1.5 s	2 s	2 s	2.5 s
Writing	2.3 s	2.7 s	3.2 s	4.6 s	5.5 s
Verification	2 s	3 s	3.5 s	4.5 s	5.5 s
Writing after erasing	3 s	4 s	4.5 s	6.5 s	8 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)

30.6 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.

Caution 1. The self-programming function cannot be used when the CPU operates with the subsystem clock.

Caution 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 self-programming library.

Caution 3. The high-speed on-chip oscillator should be kept operating during self-programming. If it is kept stopping, it should be operated (HIOSTOP = 0). The flash self-programming library should be executed after 30 μs have elapsed.

Remark 1. For details of the self-programming function, refer to **RL78 Microcontroller Self-Programming Library Type01 User's Manual (R01US0050)**.

Remark 2. For details of the time required to execute self-programming, see the notes on use that accompany the flash self-programming library tool.

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 or LV (low voltage main) mode is specified.

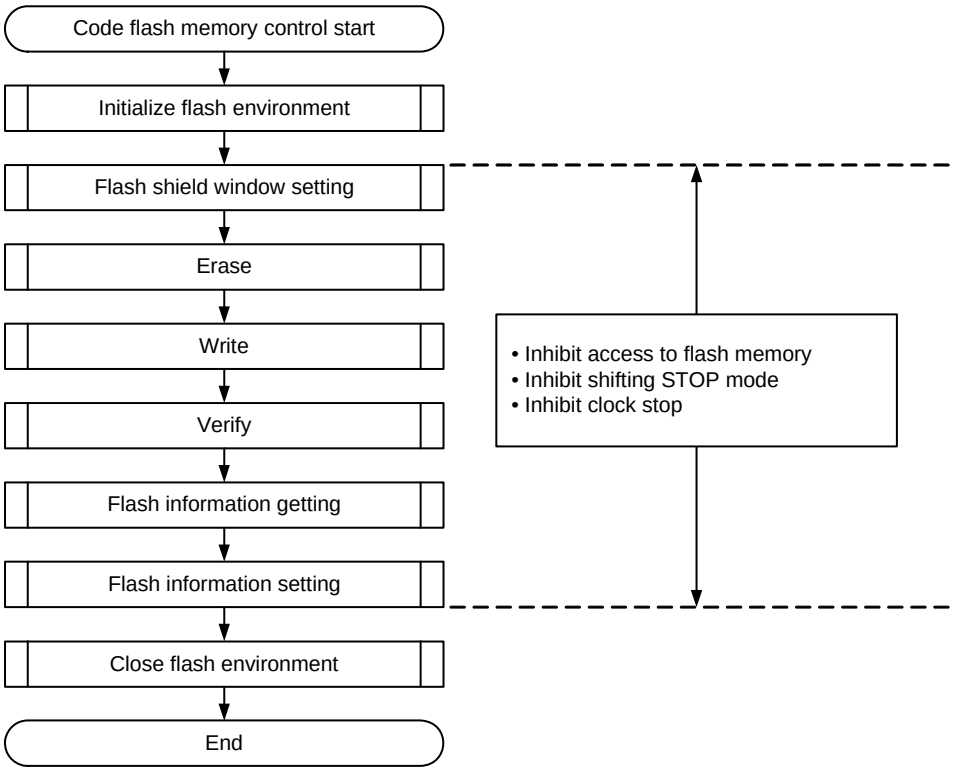
If the argument fsl_flash_voltage_u08 is 00H when the FSL_Init function of the flash self-programming library provided by Renesas Electronics is executed, full speed mode is specified. If the argument is other than 00H, the wide voltage mode is specified.

Remark Using both the wide voltage mode and full speed mode imposes no restrictions on writing, deletion, or verification.

30.6.1 Self-programming procedure

The following figure illustrates a flow for rewriting the code flash memory by using a flash self-programming library.

Figure 30 - 8 Flow of Self-Programming (Rewriting Flash Memory)



30.6.2 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.

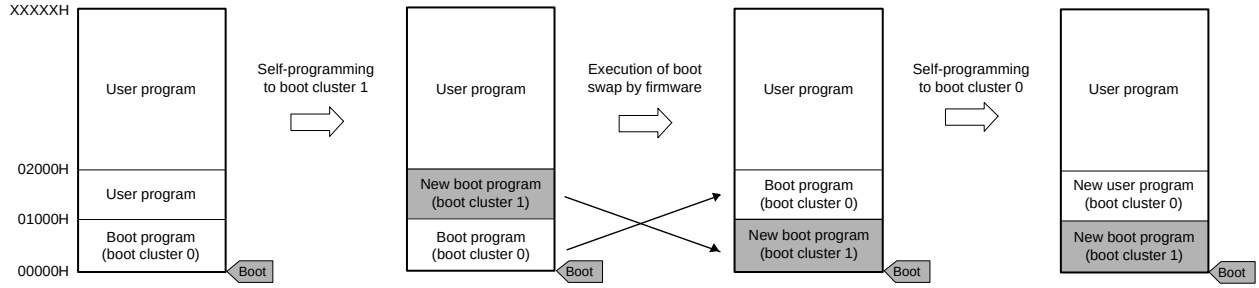
The boot swap function is used to avoid this problem.

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 boot program area, boot cluster 0.

As a result, even if a power failure occurs while the boot programming 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 30 - 9 Boot Swap Function

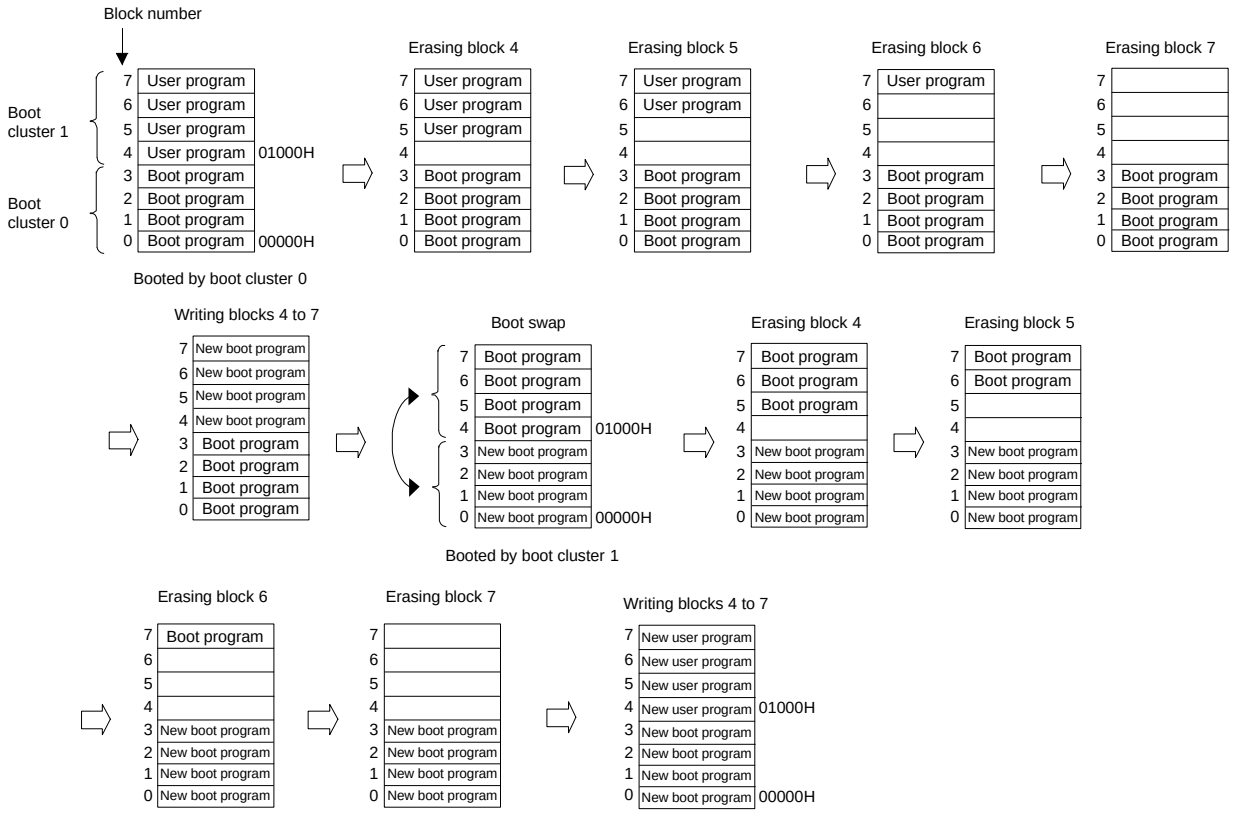


In an example of above figure, it is as follows.

Boot cluster 0: Boot area before boot swap

Boot cluster 1: Boot area after boot swap

Figure 30 - 10 Example of Executing Boot Swapping



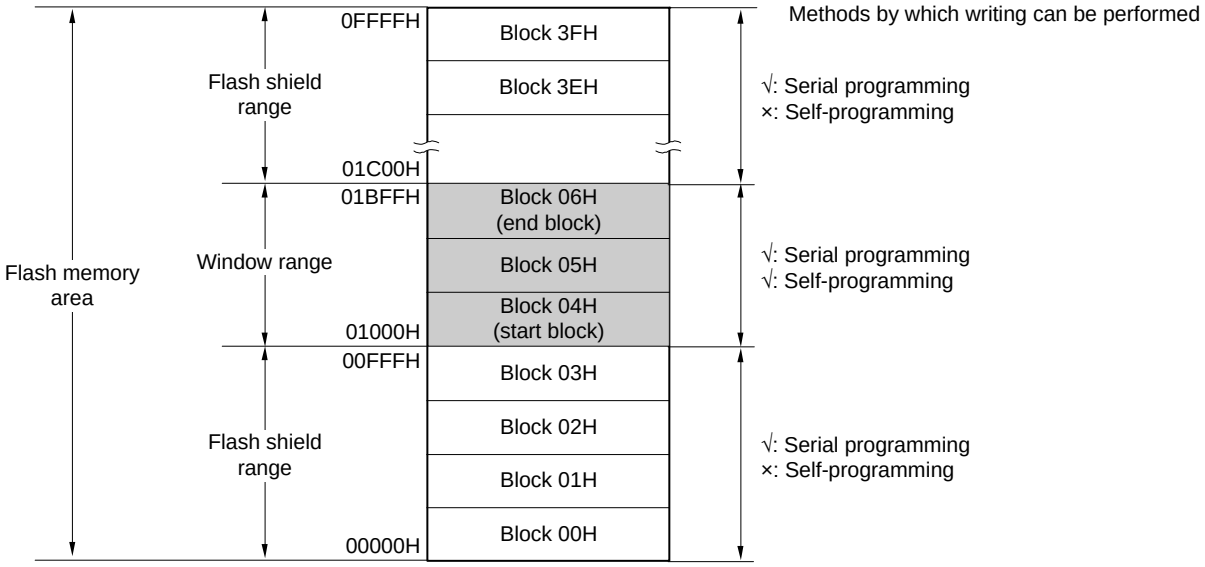
30.6.3 Flash shield window function

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 30 - 11 Flash Shield Window Setting Example
(Target Devices: R5F110PE, Start Block: 04H, End Block: 06H)



Caution 1. If the rewrite-prohibited area of the boot cluster 0 overlaps with the flash shield window range, prohibition to rewrite the boot cluster 0 takes priority.

Caution 2. The flash shield window can only be used for the code flash memory (and is not supported for the data flash memory).

Table 30 - 11 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 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 **30.7 Security Settings** to prohibit writing/erasing during serial programming.

30.7 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.

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 30 - 12 shows the relationship between the erase and write commands when the RL78 microcontroller security function is enabled.

Caution The security function of the dedicated flash programmer does not support self-programming.

Remark To prohibit writing and erasing during self-programming, use the flash shield window function (see **30.6.3** for detail).

Table 30 - 12 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 **30.6.3** for detail).

Table 30 - 13 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) 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.

30.8 Data Flash

30.8.1 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 data flash library. For details, refer to **RL78 Family Data Flash Library User's Manual**.
- The data flash memory can also be rewritten to through serial programming using the dedicated flash memory programmer or an external device.
- The data flash can be erased in 1-block (1-Kbyte) 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, background 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.

Caution 1. The data flash memory is stopped after a reset is canceled. The data flash control register (DFLCTL) must be set up in order to use the data flash memory.

Caution 2. The high-speed on-chip oscillator should be kept operating during data flash rewrite. If it is kept stopping, the high-speed on-chip oscillator clock should be operated (HIOSTOP = 0). The data flash library should be executed after 30 μs have elapsed.

Remark For rewriting the code flash memory via a user program, see **30.6 Self-Programming**.

30.8.2 Register controlling data flash memory

30.8.2.1 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 30 - 12 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.

30.8.3 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.

After initial setting, the data flash can be read through CPU instructions and can be read or rewritten to by using the data flash library.

<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 flash operation mode for the main clock.

<Setup time for each flash operation mode>

- HS (High-speed main): 5 μs
- LS (Low-speed main): 720 ns
- LV (Low-voltage main): 10 μs

<3> After the wait, the data flash memory can be accessed.

Caution 1. Accessing the data flash memory is not possible during the setup time.

Caution 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.

Caution 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 data flash library should be executed after 30 μs have elapsed.

<R> **Caution 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
LV (low-voltage main) mode: 10 μs

After initialized, the data flash memory can be read by using a CPU instruction or can be read/written by using a data flash library.

If the DMA controller operates when the data flash memory is accessed, however, follow one of these procedures:

(A) Suspending/forcibly terminating DMA transfer
Before reading the data flash memory, suspend DMA transfer of all the channels used.
After setting the DWAITn bit to 1, however, wait at least for the duration of three clocks (f_{CLK}) before reading the data flash memory. After reading the data flash memory, lift the suspension of transfer by clearing the DWAITn bit to 0.
Or, forcibly terminate DMA transfer in accordance with the procedure in 15.5.5 Forced termination by software
before reading the data flash memory. Resume DMA transfer after the data flash memory has been read.

(B) Access the data flash memory
Access the data flash memory by using the newest data flash library.

(C) Insertion of NOP
Insert an NOP instruction immediately before the instruction that reads the data flash memory.

<Example>

```
MOVW    HL,!addr16      ; Reads RAM.
NOP                     ; Insert NOP instruction before reading data flash memory.
MOV     A,[DE]          ; Read data flash memory.
```

If a high-level language such as C is used, however, the compiler may generate two instructions for one code. In this case, the NOP instruction is not inserted immediately before the data flash memory read instruction. Therefore, read the data flash memory by (A) or (B) above.

Remarks f_{CLK}: CPU/peripheral hardware clock frequency

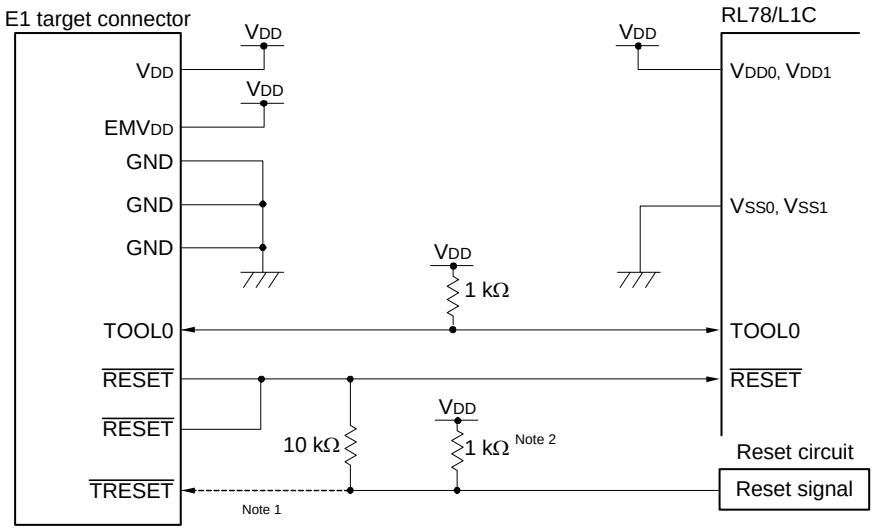
CHAPTER 31 ON-CHIP DEBUG FUNCTION

31.1 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 31 - 1 Connection Example of E1 On-chip Debugging Emulator



Note 1. Connecting the dotted line is not necessary during serial programming.

Note 2. If the reset circuit on the target system does not have a buffer and generates a reset signal only with resistors and capacitors, this pull-up resistor is not necessary.

Caution This circuit diagram is assumed that the reset signal outputs from an N-ch O.D. buffer (output resistor: 100 Ω or less)

31.2 On-Chip Debug Security ID

The RL78 microcontroller has an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 29 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 31 - 1 On-Chip Debug Security ID

Address	On-Chip Debug Security ID
000C4H to 000CDH	Any ID code Note of 10 bytes (excluding all FFH)
010C4H to 010CDH	

Note The setting FFFFFFFFFFFFFFFFFFFFH for the ID code is not possible.

31.3 Securing of User Resources

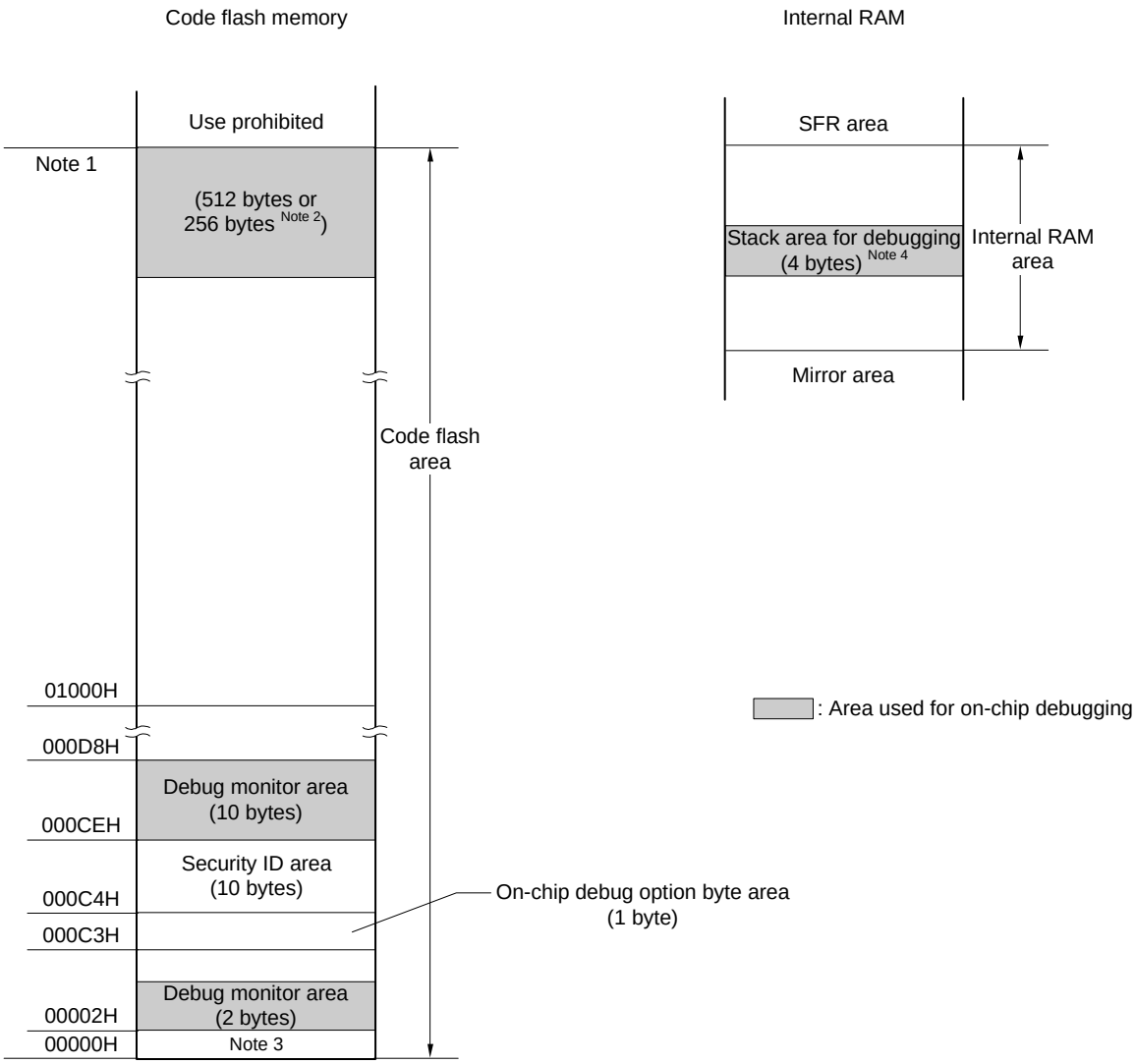
To perform communication between the RL78 microcontroller and E1 on-chip debugging emulator, as well as each debug function, the securing of memory space must be done beforehand.

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 31 - 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 31 - 2 Memory Spaces Where Debug Monitor Programs Are Allocated



Note 1. Address differs depending on products as follows.

Products (code flash memory capacity)	Address of **Note 1.**
R5F110xE, R5F111xE (x = M, N, P)	0FFFFH
R5F110xF, R5F111xF (x = M, N, P)	17FFFH
R5F110xG, R5F111xG (x = M, N, P)	1FFFFH
R5F110xH, R5F111xH (x = M, N, P)	2FFFFH
R5F110xJ, R5F111xJ (x = M, N, P)	3FFFFH

Note 2. When real-time RAM monitor (RRM) function and dynamic memory modification (DMM) function are not used, it is 256 bytes.

Note 3. In debugging, reset vector is rewritten to address allocated to a monitor program.

Note 4. Since this area is allocated immediately before the stack area, the address of this area varies depending on the stack increase and decrease. That is, 4 extra bytes are consumed for the stack area used.
When using self-programming, 12 extra bytes are consumed for the stack area used.

CHAPTER 32 BCD CORRECTION CIRCUIT

32.1 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).

32.2 Registers Used by BCD Correction Circuit

The BCD correction circuit uses the following registers.

- BCD correction result register (BCDADJ)

32.2.1 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 32 - 1 Format of BCD correction result register (BCDADJ)

Address: F00FEH After reset: Undefined R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

32.3 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 33 INSTRUCTION SET

This chapter lists the instructions in the RL78 microcontroller instruction set. For details of each operation and operation code, refer to the separate document **RL78 Family User's Manual Software (R01US0015)**.

33.1 Conventions Used in Operation List

33.1.1 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 33 - 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 FFF1FH Immediate data or labels (even addresses only Note)
addr20 addr16 addr5	00000H to FFFFFH Immediate data or labels 0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions Note) 0080H to 00BFH Immediate data or labels (even addresses only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label
RBn	RB0 to RB3

Note Bit 0 = 0 when an odd address is specified.

Remark The special function registers can be described to operand sfr as symbols. See **Tables 3 - 7** to **3 - 10 SFR List** for the symbols of the special function registers. The extended special function registers can be described to operand !addr16 as symbols. See **Tables 3 - 11** to **3 - 25 Extended SFR (2nd SFR) List** for the symbols of the extended special function registers.

33.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 33 - 2 Symbols in “Operation” Column

Symbol	Function
A	A register; 8-bit accumulator
X	X register
B	B register
C	C register
D	D register
E	E register
H	H register
L	L register
ES	ES register
CS	CS register
AX	AX register pair; 16-bit accumulator
BC	BC register pair
DE	DE register pair
HL	HL register pair
PC	Program counter
SP	Stack pointer
PSW	Program status word
CY	Carry flag
AC	Auxiliary carry flag
Z	Zero flag
RBS	Register bank select flag
IE	Interrupt request enable flag
()	Memory contents indicated by address or register contents in parentheses
X_H, X_L X_S, X_H, X_L	16-bit registers: X_H = higher 8 bits, X_L = lower 8 bits 20-bit registers: X_S = (bits 19 to 16), X_H = (bits 15 to 8), X_L = (bits 7 to 0)
∧	Logical product (AND)
∨	Logical sum (OR)
∨̸	Exclusive logical sum (exclusive OR)
—	Inverted data
addr5	16-bit immediate data (even addresses only in 0080H to 00BFH)
addr16	16-bit immediate data
addr20	20-bit immediate data
jdisp8	Signed 8-bit data (displacement value)
jdisp16	Signed 16-bit data (displacement value)

33.1.3 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 33 - 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

33.1.4 PREFIX instruction

Instructions with “ES:” have a PREFIX operation code as a prefix to extend the accessible data area to the 1 MB space (00000H to FFFFFH), by adding the ES register value to the 64 KB space from F0000H to FFFFFH. When a PREFIX operation code is attached as a prefix to the target instruction, only one instruction immediately after the PREFIX operation code is executed as the addresses with the ES register value added.

A interrupt and DTC transfer are not acknowledged between a PREFIX instruction code and the instruction immediately after.

Table 33 - 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.

33.2 Operation List

Table 33 - 5 Operation List (1/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	r, #byte	2	1	—	r ← byte			
		PSW, #byte	3	3	—	PSW ← byte	×	×	×
		CS, #byte	3	1	—	CS ← byte			
		ES, #byte	2	1	—	ES ← byte			
		!addr16, #byte	4	1	—	(addr16) ← byte			
		ES:!addr16, #byte	5	2	—	(ES, addr16) ← byte			
		saddr, #byte	3	1	—	(saddr) ← byte			
		sfr, #byte	3	1	—	sfr ← byte			
		[DE+byte], #byte	3	1	—	(DE + byte) ← byte			
		ES:[DE+byte], #byte	4	2	—	((ES, DE) + byte) ← byte			
		[HL+byte], #byte	3	1	—	(HL + byte) ← byte			
		ES:[HL+byte], #byte	4	2	—	((ES, HL) + byte) ← byte			
		[SP+byte], #byte	3	1	—	(SP + byte) ← byte			
		word[B], #byte	4	1	—	(B + word) ← byte			
		ES:word[B], #byte	5	2	—	((ES, B) + word) ← byte			
		word[C], #byte	4	1	—	(C+word) ← byte			
		ES:word[C], #byte	5	2	—	((ES, C) + word) ← byte			
		word[BC], #byte	4	1	—	(BC+word) ← byte			
		ES:word[BC], #byte	5	2	—	((ES, BC) + word) ← byte			
		A, r Note 3	1	1	—	A ← r			
		r, A Note 3	1	1	—	r ← A			
		A, PSW	2	1	—	A ← PSW			
		PSW, A	2	3	—	PSW ← A	×	×	×
		A, CS	2	1	—	A ← CS			
		CS, A	2	1	—	CS ← A			
		A, ES	2	1	—	A ← ES			
		ES, A	2	1	—	ES ← A			
		A, !addr16	3	1	4	A ← (addr16)			
		A, ES:!addr16	4	2	5	A ← (ES, addr16)			
		!addr16, A	3	1	—	(addr16) ← A			
		ES:!addr16, A	4	2	—	(ES, addr16) ← A			
		A, saddr	2	1	—	A ← (saddr)			
		saddr, A	2	1	—	(saddr) ← A			

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 6 Operation List (2/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	A, sfr	2	1	—	A ← sfr			
		sfr, A	2	1	—	sfr ← A			
		A, [DE]	1	1	4	A ← (DE)			
		[DE], A	1	1	—	(DE) ← A			
		A, ES:[DE]	2	2	5	A ← (ES, DE)			
		ES:[DE], A	2	2	—	(ES, DE) ← A			
		A, [HL]	1	1	4	A ← (HL)			
		[HL], A	1	1	—	(HL) ← A			
		A, ES:[HL]	2	2	5	A ← (ES, HL)			
		ES:[HL], A	2	2	—	(ES, HL) ← A			
		A, [DE+byte]	2	1	4	A ← (DE + byte)			
		[DE+byte], A	2	1	—	(DE + byte) ← A			
		A, ES:[DE+byte]	3	2	5	A ← ((ES, DE) + byte)			
		ES:[DE+byte], A	3	2	—	((ES, DE) + byte ← A			
		A, [HL+byte]	2	1	4	A ← (HL + byte)			
		[HL+byte], A	2	1	—	(HL + byte) ← A			
		A, ES:[HL+byte]	3	2	5	A ← ((ES, HL) + byte)			
		ES:[HL+byte], A	3	2	—	((ES, HL) + byte) ← A			
		A, [SP+byte]	2	1	—	A ← (SP + byte)			
		[SP+byte], A	2	1	—	(SP + byte) ← A			
		A, word[B]	3	1	4	A ← (B + word)			
		word[B], A	3	1	—	(B + word) ← A			
		A, ES:word[B]	4	2	5	A ← ((ES, B) + word)			
		ES:word[B], A	4	2	—	((ES, B) + word) ← A			
		A, word[C]	3	1	4	A ← (C + word)			
		word[C], A	3	1	—	(C + word) ← A			
		A, ES:word[C]	4	2	5	A ← ((ES, C) + word)			
		ES:word[C], A	4	2	—	((ES, C) + word) ← A			
		A, word[BC]	3	1	4	A ← (BC + word)			
		word[BC], A	3	1	—	(BC + word) ← A			
		A, ES:word[BC]	4	2	5	A ← ((ES, BC) + word)			
		ES:word[BC], A	4	2	—	((ES, BC) + word) ← A			

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 7 Operation List (3/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	A, [HL+B]	2	1	4	A ← (HL + B)			
		[HL+B], A	2	1	—	(HL + B) ← A			
		A, ES:[HL+B]	3	2	5	A ← ((ES, HL) + B)			
		ES:[HL+B], A	3	2	—	((ES, HL) + B) ← A			
		A, [HL+C]	2	1	4	A ← (HL + C)			
		[HL+C], A	2	1	—	(HL + C) ← A			
		A, ES:[HL+C]	3	2	5	A ← ((ES, HL) + C)			
		ES:[HL+C], A	3	2	—	((ES, HL) + C) ← A			
		X, !addr16	3	1	4	X ← (addr16)			
		X, ES:!addr16	4	2	5	X ← (ES, addr16)			
		X, saddr	2	1	—	X ← (saddr)			
		B, !addr16	3	1	4	B ← (addr16)			
		B, ES:!addr16	4	2	5	B ← (ES, addr16)			
		B, saddr	2	1	—	B ← (saddr)			
		C, !addr16	3	1	4	C ← (addr16)			
		C, ES:!addr16	4	2	5	C ← (ES, addr16)			
		C, saddr	2	1	—	C ← (saddr)			
		ES, saddr	3	1	—	ES ← (saddr)			
	XCH	A, r Note 3	1 (r = X) 2 (other than r = X)	1	—	A ←→ r			
		A, !addr16	4	2	—	A ←→ (addr16)			
		A, ES:!addr16	5	3	—	A ←→ (ES, addr16)			
		A, saddr	3	2	—	A ←→ (saddr)			
		A, sfr	3	2	—	A ←→ sfr			
		A, [DE]	2	2	—	A ←→ (DE)			
		A, ES:[DE]	3	3	—	A ←→ (ES, DE)			
		A, [HL]	2	2	—	A ←→ (HL)			
		A, ES:[HL]	3	3	—	A ←→ (ES, HL)			
		A, [DE+byte]	3	2	—	A ←→ (DE + byte)			
		A, ES:[DE+byte]	4	3	—	A ←→ ((ES, DE) + byte)			
		A, [HL+byte]	3	2	—	A ←→ (HL + byte)			
		A, ES:[HL+byte]	4	3	—	A ←→ ((ES, HL) + byte)			

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 8 Operation List (4/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	XCH	A, [HL+B]	2	2	—	A ←→ (HL + B)			
		A, ES:[HL+B]	3	3	—	A ←→ ((ES, HL) + B)			
		A, [HL+C]	2	2	—	A ←→ (HL + C)			
		A, ES:[HL+C]	3	3	—	A ←→ ((ES, HL) + C)			
	ONEB	A	1	1	—	A ← 01H			
		X	1	1	—	X ← 01H			
		B	1	1	—	B ← 01H			
		C	1	1	—	C ← 01H			
		!addr16	3	1	—	(addr16) ← 01H			
		ES:!addr16	4	2	—	(ES, addr16) ← 01H			
		saddr	2	1	—	(saddr) ← 01H			
	CLRB	A	1	1	—	A ← 00H			
		X	1	1	—	X ← 00H			
		B	1	1	—	B ← 00H			
		C	1	1	—	C ← 00H			
		!addr16	3	1	—	(addr16) ← 00H			
		ES:!addr16	4	2	—	(ES,addr16) ← 00H			
		saddr	2	1	—	(saddr) ← 00H			
	MOVS	[HL+byte], X	3	1	—	(HL + byte) ← X	×		×
		ES:[HL+byte], X	4	2	—	(ES, HL + byte) ← X	×		×
16-bit data transfer	MOVW	rp, #word	3	1	—	rp ← word			
		saddrp, #word	4	1	—	(saddrp) ← word			
		sfrp, #word	4	1	—	sfrp ← word			
		AX, rp Note 3	1	1	—	AX ← rp			
		rp, AX Note 3	1	1	—	rp ← AX			
		AX, !addr16	3	1	4	AX ← (addr16)			
		!addr16, AX	3	1	—	(addr16) ← AX			
		AX, ES:!addr16	4	2	5	AX ← (ES, addr16)			
		ES:!addr16, AX	4	2	—	(ES, addr16) ← AX			
		AX, saddrp	2	1	—	AX ← (saddrp)			
		saddrp, AX	2	1	—	(saddrp) ← AX			
		AX, sfrp	2	1	—	AX ← sfrp			
		sfrp, AX	2	1	—	sfrp ← AX			

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except rp = AX

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 9 Operation List (5/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	MOVW	AX, [DE]	1	1	4	AX ← (DE)			
		[DE], AX	1	1	—	(DE) ← AX			
		AX, ES:[DE]	2	2	5	AX ← (ES, DE)			
		ES:[DE], AX	2	2	—	(ES, DE) ← AX			
		AX, [HL]	1	1	4	AX ← (HL)			
		[HL], AX	1	1	—	(HL) ← AX			
		AX, ES:[HL]	2	2	5	AX ← (ES, HL)			
		ES:[HL], AX	2	2	—	(ES, HL) ← AX			
		AX, [DE+byte]	2	1	4	AX ← (DE + byte)			
		[DE+byte], AX	2	1	—	(DE + byte) ← AX			
		AX, ES:[DE+byte]	3	2	5	AX ← ((ES, DE) + byte)			
		ES:[DE+byte], AX	3	2	—	((ES, DE) + byte) ← AX			
		AX, [HL+byte]	2	1	4	AX ← (HL + byte)			
		[HL+byte], AX	2	1	—	(HL + byte) ← AX			
		AX, ES:[HL+byte]	3	2	5	AX ← ((ES, HL) + byte)			
		ES:[HL+byte], AX	3	2	—	((ES, HL) + byte) ← AX			
		AX, [SP+byte]	2	1	—	AX ← (SP + byte)			
		[SP+byte], AX	2	1	—	(SP + byte) ← AX			
		AX, word[B]	3	1	4	AX ← (B + word)			
		word[B], AX	3	1	—	(B + word) ← AX			
		AX, ES:word[B]	4	2	5	AX ← ((ES, B) + word)			
		ES:word[B], AX	4	2	—	((ES, B) + word) ← AX			
		AX, word[C]	3	1	4	AX ← (C + word)			
		word[C], AX	3	1	—	(C + word) ← AX			
		AX, ES:word[C]	4	2	5	AX ← ((ES, C) + word)			
		ES:word[C], AX	4	2	—	((ES, C) + word) ← AX			
		AX, word[BC]	3	1	4	AX ← (BC + word)			
		word[BC], AX	3	1	—	(BC + word) ← AX			
		AX, ES:word[BC]	4	2	5	AX ← ((ES, BC) + word)			
		ES:word[BC], AX	4	2	—	((ES, BC) + word) ← AX			

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the 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 33 - 10 Operation List (6/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	MOVW	BC, !addr16	3	1	4	BC ← (addr16)			
		BC, ES:!addr16	4	2	5	BC ← (ES, addr16)			
		DE, !addr16	3	1	4	DE ← (addr16)			
		DE, ES:!addr16	4	2	5	DE ← (ES, addr16)			
		HL, !addr16	3	1	4	HL ← (addr16)			
		HL, ES:!addr16	4	2	5	HL ← (ES, addr16)			
		BC, saddrp	2	1	—	BC ← (saddrp)			
		DE, saddrp	2	1	—	DE ← (saddrp)			
		HL, saddrp	2	1	—	HL ← (saddrp)			
	XCHW	AX, rp Note 3	1	1	—	AX ←→ rp			
	ONEW	AX	1	1	—	AX ← 0001H			
		BC	1	1	—	BC ← 0001H			
	CLRW	AX	1	1	—	AX ← 0000H			
		BC	1	1	—	BC ← 0000H			
8-bit operation	ADD	A, #byte	2	1	—	A, CY ← A + byte	×	×	×
		saddr, #byte	3	2	—	(saddr), CY ← (saddr) + byte	×	×	×
		A, r Note 4	2	1	—	A, CY ← A + r	×	×	×
		r, A	2	1	—	r, CY ← r + A	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16)	×	×	×
		A, saddr	2	1	—	A, C ← A + (saddr)	×	×	×
		A, [HL]	1	1	4	A, CY ← A + (HL)	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A + (ES, HL)	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A + (HL + byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A + ((ES, HL) + byte)	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A + (HL + B)	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A + ((ES, HL) + B)	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A + (HL + C)	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A + ((ES, HL) + C)	×	×	×

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except rp = AX

Note 4. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 11 Operation List (7/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	ADDC	A, #byte	2	1	—	A, CY ← A + byte + CY	×	×	×
		saddr, #byte	3	2	—	(saddr), CY ← (saddr) + byte + CY	×	×	×
		A, r Note 3	2	1	—	A, CY ← A + r + CY	×	×	×
		r, A	2	1	—	r, CY ← r + A + CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16) + CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16) + CY	×	×	×
		A, saddr	2	1	—	A, CY ← A + (saddr) + CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A + (HL) + CY	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A + (ES, HL) + CY	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A + (HL + byte) + CY	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A + ((ES, HL) + byte) + CY	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A + (HL + B) + CY	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A + ((ES, HL) + B) + CY	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A + (HL + C) + CY	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A + ((ES, HL) + C) + CY	×	×	×
	SUB	A, #byte	2	1	—	A, CY ← A - byte	×	×	×
		saddr, #byte	3	2	—	(saddr), CY ← (saddr) - byte	×	×	×
		A, r Note 3	2	1	—	A, CY ← A - r	×	×	×
		r, A	2	1	—	r, CY ← r - A	×	×	×
		A, !addr16	3	1	4	A, CY ← A - (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A - (ES, addr16)	×	×	×
		A, saddr	2	1	—	A, CY ← A - (saddr)	×	×	×
		A, [HL]	1	1	4	A, CY ← A - (HL)	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A - (ES, HL)	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A - (HL + byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A - ((ES, HL) + byte)	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A - (HL + B)	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A - ((ES, HL) + B)	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A - (HL + C)	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A - ((ES, HL) + C)	×	×	×

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 12 Operation List (8/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	SUBC	A, #byte	2	1	—	A, CY ← A - byte - CY	×	×	×
		saddr, #byte	3	2	—	(saddr), CY ← (saddr) - byte - CY	×	×	×
		A, r Note 3	2	1	—	A, CY ← A - r - CY	×	×	×
		r, A	2	1	—	r, CY ← r - A - CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A - (addr16) - CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A - (ES, addr16) - CY	×	×	×
		A, saddr	2	1	—	A, CY ← A - (saddr) - CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A - (HL) - CY	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A - (ES, HL) - CY	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A - (HL + byte) - CY	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A - ((ES, HL) + byte) - CY	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A - (HL + B) - CY	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A - ((ES, HL) + B) - CY	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A - (HL + C) - CY	×	×	×
		A, ES:[HL+C]	3	2	5	A, CY ← A - ((ES:HL) + C) - CY	×	×	×
	AND	A, #byte	2	1	—	A ← A ∧ byte	×		
		saddr, #byte	3	2	—	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 3	2	1	—	A ← A ∧ r	×		
		r, A	2	1	—	r ← r ∧ A	×		
		A, !addr16	3	1	4	A ← A ∧ (addr16)	×		
		A, ES:!addr16	4	2	5	A ← A ∧ (ES:addr16)	×		
		A, saddr	2	1	—	A ← A ∧ (saddr)	×		
		A, [HL]	1	1	4	A ← A ∧ (HL)	×		
		A, ES:[HL]	2	2	5	A ← A ∧ (ES:HL)	×		
		A, [HL+byte]	2	1	4	A ← A ∧ (HL + byte)	×		
		A, ES:[HL+byte]	3	2	5	A ← A ∧ ((ES:HL) + byte)	×		
		A, [HL+B]	2	1	4	A ← A ∧ (HL + B)	×		
		A, ES:[HL+B]	3	2	5	A ← A ∧ ((ES:HL) + B)	×		
		A, [HL+C]	2	1	4	A ← A ∧ (HL + C)	×		
		A, ES:[HL+C]	3	2	5	A ← A ∧ ((ES:HL) + C)	×		

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 13 Operation List (9/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	OR	A, #byte	2	1	—	A ← A ∨ byte	×		
		saddr, #byte	3	2	—	(saddr) ← (saddr) ∨ byte	×		
		A, r Note 3	2	1	—	A ← A ∨ r	×		
		r, A	2	1	—	r ← r ∨ A	×		
		A, !addr16	3	1	4	A ← A ∨ (addr16)	×		
		A, ES:!addr16	4	2	5	A ← A ∨ (ES:addr16)	×		
		A, saddr	2	1	—	A ← A ∨ (saddr)	×		
		A, [HL]	1	1	4	A ← A ∨ (HL)	×		
		A, ES:[HL]	2	2	5	A ← A ∨ (ES:HL)	×		
		A, [HL+byte]	2	1	4	A ← A ∨ (HL + byte)	×		
		A, ES:[HL+byte]	3	2	5	A ← A ∨ ((ES:HL) + byte)	×		
		A, [HL+B]	2	1	4	A ← A ∨ (HL + B)	×		
		A, ES:[HL+B]	3	2	5	A ← A ∨ ((ES:HL) + B)	×		
		A, [HL+C]	2	1	4	A ← A ∨ (HL + C)	×		
		A, ES:[HL+C]	3	2	5	A ← A ∨ ((ES:HL) + C)	×		
	XOR	A, #byte	2	1	—	A ← A ⩛ byte	×		
		saddr, #byte	3	2	—	(saddr) ← (saddr) ⩛ byte	×		
		A, r Note 3	2	1	—	A ← A ⩛ r	×		
		r, A	2	1	—	r ← r ⩛ A	×		
		A, !addr16	3	1	4	A ← A ⩛ (addr16)	×		
		A, ES:!addr16	4	2	5	A ← A ⩛ (ES:addr16)	×		
		A, saddr	2	1	—	A ← A ⩛ (saddr)	×		
		A, [HL]	1	1	4	A ← A ⩛ (HL)	×		
		A, ES:[HL]	2	2	5	A ← A ⩛ (ES:HL)	×		
		A, [HL+byte]	2	1	4	A ← A ⩛ (HL + byte)	×		
		A, ES:[HL+byte]	3	2	5	A ← A ⩛ ((ES:HL) + byte)	×		
		A, [HL+B]	2	1	4	A ← A ⩛ (HL + B)	×		
		A, ES:[HL+B]	3	2	5	A ← A ⩛ ((ES:HL) + B)	×		
		A, [HL+C]	2	1	4	A ← A ⩛ (HL + C)	×		
		A, ES:[HL+C]	3	2	5	A ← A ⩛ ((ES:HL) + C)	×		

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 14 Operation List (10/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	CMP	A, #byte	2	1	—	A - byte	×	×	×
		!addr16, #byte	4	1	4	(addr16) - byte	×	×	×
		ES:!addr16, #byte	5	2	5	(ES:addr16) - byte	×	×	×
		saddr, #byte	3	1	—	(saddr) - byte	×	×	×
		A, r Note 3	2	1	—	A - r	×	×	×
		r, A	2	1	—	r - A	×	×	×
		A, !addr16	3	1	4	A - (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A - (ES:addr16)	×	×	×
		A, saddr	2	1	—	A - (saddr)	×	×	×
		A, [HL]	1	1	4	A - (HL)	×	×	×
		A, ES:[HL]	2	2	5	A - (ES:HL)	×	×	×
		A, [HL+byte]	2	1	4	A - (HL + byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A - ((ES:HL) + byte)	×	×	×
		A, [HL+B]	2	1	4	A - (HL + B)	×	×	×
		A, ES:[HL+B]	3	2	5	A - ((ES:HL) + B)	×	×	×
		A, [HL+C]	2	1	4	A - (HL + C)	×	×	×
		A, ES:[HL+C]	3	2	5	A - ((ES:HL) + C)	×	×	×
	CMP0	A	1	1	—	A - 00H	×	0	0
		X	1	1	—	X - 00H	×	0	0
		B	1	1	—	B - 00H	×	0	0
		C	1	1	—	C - 00H	×	0	0
		!addr16	3	1	4	(addr16) - 00H	×	0	0
		ES:!addr16	4	2	5	(ES:addr16) - 00H	×	0	0
		saddr	2	1	—	(saddr) - 00H	×	0	0
	CMPS	X, [HL+byte]	3	1	4	X - (HL + byte)	×	×	×
		X, ES:[HL+byte]	4	2	5	X - ((ES:HL) + byte)	×	×	×

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 15 Operation List (11/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	ADDW	AX, #word	3	1	—	AX, CY ← AX + word	×	×	×
		AX, AX	1	1	—	AX, CY ← AX + AX	×	×	×
		AX, BC	1	1	—	AX, CY ← AX + BC	×	×	×
		AX, DE	1	1	—	AX, CY ← AX + DE	×	×	×
		AX, HL	1	1	—	AX, CY ← AX + HL	×	×	×
		AX, !addr16	3	1	4	AX, CY × AX + (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX + (ES:addr16)	×	×	×
		AX, saddrp	2	1	—	AX, CY ← AX + (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX + (HL + byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX + ((ES:HL) + byte)	×	×	×
	SUBW	AX, #word	3	1	—	AX, CY ← AX - word	×	×	×
		AX, BC	1	1	—	AX, CY ← AX - BC	×	×	×
		AX, DE	1	1	—	AX, CY ← AX - DE	×	×	×
		AX, HL	1	1	—	AX, CY ← AX - HL	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX - (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX - (ES:addr16)	×	×	×
		AX, saddrp	2	1	—	AX, CY ← AX - (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX - (HL + byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX - ((ES:HL) + byte)	×	×	×
	CMPW	AX, #word	3	1	—	AX - word	×	×	×
		AX, BC	1	1	—	AX - BC	×	×	×
		AX, DE	1	1	—	AX - DE	×	×	×
		AX, HL	1	1	—	AX - HL	×	×	×
		AX, !addr16	3	1	4	AX - (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX - (ES:addr16)	×	×	×
		AX, saddrp	2	1	—	AX - (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX - (HL + byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX - ((ES:HL) + byte)	×	×	×

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 16 Operation List (12/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Multiply, Divide, Multiply & accumulate	MULU	X	1	1	—	AX ← A × X			
	MULHU		3	2	—	BCAX ← AX × BC (unsigned)			
	MULH		3	2	—	BCAX ← AX × BC (signed)			
	DIVHU		3	9	—	AX (quotient), DE (remainder) ← AX ÷ DE (unsigned)			
	DIVWU		3	17	—	BCAX (quotient), HLDE (remainder) ← BCAX ÷ HLDE (unsigned)			
	MACHU		3	3	—	MACR ← MACR + AX × BC (unsigned)		×	×
	MACH		3	3	—	MACR ← MACR + AX × BC(signed)		×	×

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Caution Disable interrupts when executing the DIVHU or DIVWU instruction in an interrupt servicing routine. Alternatively, unless they are executed in the RAM area, note that execution of a DIVHU or DIVWU instruction is possible even with interrupts enabled as long as a NOP instruction is added immediately after the DIVHU or DIVWU instruction in the assembly language source code. The following compilers automatically add a NOP instruction immediately after any DIVHU or DIVWU instruction output during the build process.

- **V. 1.71 and later versions of the CA78K0R (Renesas Electronics compiler), for both C and assembly language source code**
- **Service pack 1.40.6 and later versions of the EWRL78 (IAR compiler), for C language source code**
- **GNURL78 (KPIT compiler), for C language source code**

Remark 1. Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Remark 2. MACR indicates the multiplication and accumulation register (MACRH, MACRL).

Table 33 - 17 Operation List (13/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Increment/ decrement	INC	r	1	1	—	r ← r + 1	×	×	
		!addr16	3	2	—	(addr16) ← (addr16) + 1	×	×	
		ES:!addr16	4	3	—	(ES, addr16) ← (ES, addr16) + 1	×	×	
		saddr	2	2	—	(saddr) ← (saddr) + 1	×	×	
		[HL+byte]	3	2	—	(HL + byte) ← (HL + byte) + 1	×	×	
		ES: [HL+byte]	4	3	—	((ES:HL) + byte) ← ((ES:HL) + byte) + 1	×	×	
	DEC	r	1	1	—	r ← r - 1	×	×	
		!addr16	3	2	—	(addr16) ← (addr16) - 1	×	×	
		ES:!addr16	4	3	—	(ES, addr16) ← (ES, addr16) - 1	×	×	
		saddr	2	2	—	(saddr) ← (saddr) - 1	×	×	
		[HL+byte]	3	2	—	(HL + byte) ← (HL + byte) - 1	×	×	
		ES: [HL+byte]	4	3	—	((ES:HL) + byte) ← ((ES:HL) + byte) - 1	×	×	
	INCW	rp	1	1	—	rp ← rp + 1			
		!addr16	3	2	—	(addr16) ← (addr16) + 1			
		ES:!addr16	4	3	—	(ES, addr16) ← (ES, addr16) + 1			
		saddrp	2	2	—	(saddrp) ← (saddrp) + 1			
		[HL+byte]	3	2	—	(HL + byte) ← (HL + byte) + 1			
		ES: [HL+byte]	4	3	—	((ES:HL) + byte) ← ((ES:HL) + byte) + 1			
	DECW	rp	1	1	—	rp ← rp - 1			
		!addr16	3	2	—	(addr16) ← (addr16) - 1			
		ES:!addr16	4	3	—	(ES, addr16) ← (ES, addr16) - 1			
		saddrp	2	2	—	(saddrp) ← (saddrp) - 1			
		[HL+byte]	3	2	—	(HL + byte) ← (HL + byte) - 1			
		ES: [HL+byte]	4	3	—	((ES:HL) + byte) ← ((ES:HL) + byte) - 1			
Shift	SHR	A, cnt	2	1	—	(CY ← A_0, A_{m-1} ← A_m, A_7 ← 0) × cnt			×
	SHRW	AX, cnt	2	1	—	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← 0) × cnt			×
	SHL	A, cnt	2	1	—	(CY ← A_7, A_m ← A_{m-1}, A_0 ← 0) × cnt			×
		B, cnt	2	1	—	(CY ← B_7, B_m ← B_{m-1}, B_0 ← 0) × cnt			×
		C, cnt	2	1	—	(CY ← C_7, C_m ← C_{m-1}, C_0 ← 0) × cnt			×
	SHLW	AX, cnt	2	1	—	(CY ← AX_{15}, AX_m ← AX_{m-1}, AX_0 ← 0) × cnt			×
		BC, cnt	2	1	—	(CY ← BC_{15}, BC_m ← BC_{m-1}, BC_0 ← 0) × cnt			×
	SAR	A, cnt	2	1	—	(CY ← A_0, A_{m-1} ← A_m, A_7 ← A_7) × cnt			×
	SARW	AX, cnt	2	1	—	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← AX_{15}) × cnt			×

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark 1. Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Remark 2. cnt indicates the bit shift count.

Table 33 - 18 Operation List (14/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Rotate	ROR	A, 1	2	1	—	(CY, $A_7 \leftarrow A_0$, $A_{m-1} \leftarrow A_m$) × 1			×
	ROL	A, 1	2	1	—	(CY, $A_0 \leftarrow A_7$, $A_{m+1} \leftarrow A_m$) × 1			×
	RORC	A, 1	2	1	—	(CY $\leftarrow A_0$, $A_7 \leftarrow$ CY, $A_{m-1} \leftarrow A_m$) × 1			×
	ROLC	A, 1	2	1	—	(CY $\leftarrow A_7$, $A_0 \leftarrow$ CY, $A_{m+1} \leftarrow A_m$) × 1			×
	ROLWC	AX,1	2	1	—	(CY $\leftarrow AX_{15}$, $AX_0 \leftarrow$ CY, $AX_{m+1} \leftarrow AX_m$) × 1			×
		BC,1	2	1	—	(CY $\leftarrow BC_{15}$, $BC_0 \leftarrow$ CY, $BC_{m+1} \leftarrow BC_m$) × 1			×
Bit manipulate	MOV1	CY, A.bit	2	1	—	CY ← A.bit			×
		A.bit, CY	2	1	—	A.bit ← CY			
		CY, PSW.bit	3	1	—	CY ← PSW.bit			×
		PSW.bit, CY	3	4	—	PSW.bit ← CY	×	×	
		CY, saddr.bit	3	1	—	CY ← (saddr).bit			×
		saddr.bit, CY	3	2	—	(saddr).bit ← CY			
		CY, sfr.bit	3	1	—	CY ← sfr.bit			×
		sfr.bit, CY	3	2	—	sfr.bit ← CY			
		CY,[HL].bit	2	1	4	CY ← (HL).bit			×
		[HL].bit, CY	2	2	—	(HL).bit ← CY			
		CY, ES:[HL].bit	3	2	5	CY ← (ES, HL).bit			×
		ES:[HL].bit, CY	3	3	—	(ES, HL).bit ← CY			
	AND1	CY, A.bit	2	1	—	CY ← CY ∧ A.bit			×
		CY, PSW.bit	3	1	—	CY ← CY ∧ PSW.bit			×
		CY, saddr.bit	3	1	—	CY ← CY ∧ (saddr).bit			×
		CY, sfr.bit	3	1	—	CY ← CY ∧ sfr.bit			×
		CY,[HL].bit	2	1	4	CY ← CY ∧ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∧ (ES, HL).bit			×
	OR1	CY, A.bit	2	1	—	CY ← CY ∨ A.bit			×
		CY, PSW.bit	3	1	—	CY ← CY ∨ PSW.bit			×
		CY, saddr.bit	3	1	—	CY ← CY ∨ (saddr).bit			×
		CY, sfr.bit	3	1	—	CY ← CY ∨ sfr.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ∨ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∨ (ES, HL).bit			×

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 19 Operation List (15/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipulate	XOR1	CY, A.bit	2	1	—	CY ← CY ⨯ bit			×
		CY, PSW.bit	3	1	—	CY ← CY ⨯ PSW.bit			×
		CY, saddr.bit	3	1	—	CY ← CY ⨯ (saddr).bit			×
		CY, sfr.bit	3	1	—	CY ← CY ⨯ sfr.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ⨯ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ⨯ (ES, HL).bit			×
	SET1	A.bit	2	1	—	A.bit ← 1			
		PSW.bit	3	4	—	PSW.bit ← 1	×	×	×
		!addr16.bit	4	2	—	(addr16).bit ← 1			
		ES:!addr16.bit	5	3	—	(ES, addr16).bit ← 1			
		saddr.bit	3	2	—	(saddr).bit ← 1			
		sfr.bit	3	2	—	sfr.bit ← 1			
		[HL].bit	2	2	—	(HL).bit ← 1			
		ES:[HL].bit	3	3	—	(ES, HL).bit ← 1			
	CLR1	A.bit	2	1	—	A.bit ← 0			
		PSW.bit	3	4	—	PSW.bit ← 0	×	×	×
		!addr16.bit	4	2	—	(addr16).bit ← 0			
		ES:!addr16.bit	5	3	—	(ES, addr16).bit ← 0			
		saddr.bit	3	2	—	(saddr.bit) ← 0			
		sfr.bit	3	2	—	sfr.bit ← 0			
		[HL].bit	2	2	—	(HL).bit ← 0			
		ES:[HL].bit	3	3	—	(ES, HL).bit ← 0			
	SET1	CY	2	1	—	CY ← 1			1
	CLR1	CY	2	1	—	CY ← 0			0
	NOT1	CY	2	1	—	CY ← $\overline{CY}$			×

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 20 Operation List (16/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Call/return	CALL	rp	2	3	—	(SP - 2) ← (PC + 2)S, (SP - 3) ← (PC + 2)H, (SP - 4) ← (PC + 2)L, PC ← CS, rp, SP ← SP - 4			
		$!addr20	3	3	—	(SP - 2) ← (PC + 3)S, (SP - 3) ← (PC + 3)H, (SP - 4) ← (PC + 3)L, PC ← PC + 3 + jdisp16, SP ← SP - 4			
		!addr16	3	3	—	(SP - 2) ← (PC + 3)S, (SP - 3) ← (PC + 3)H, (SP - 4) ← (PC + 3)L, PC ← 0000, addr16, SP ← SP - 4			
		!!addr20	4	3	—	(SP - 2) ← (PC + 4)S, (SP - 3) ← (PC + 4)H, (SP - 4) ← (PC + 4)L, PC ← addr20, SP ← SP - 4			
	CALLT	[addr5]	2	5	—	(SP - 2) ← (PC + 2)S, (SP - 3) ← (PC + 2)H, (SP - 4) ← (PC + 2)L , 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

Note 1. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (fCLK) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 21 Operation List (17/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Stack manipulate	PUSH	PSW	2	1	—	(SP - 1) ← PSW, (SP - 2) ← 00H, SP ← SP - 2			
		rp	1	1	—	(SP - 1) ← rp_H, (SP - 2) ← rp_L, SP ← SP - 2			
	POP	PSW	2	3	—	PSW ← (SP + 1), SP ← SP + 2	R	R	R
		rp	1	1	—	rp_L ← (SP), rp_H ← (SP + 1), SP ← SP + 2			
	MOVW	SP, #word	4	1	—	SP ← word			
		SP, AX	2	1	—	SP ← AX			
		AX, SP	2	1	—	AX ← SP			
		HL, SP	3	1	—	HL ← SP			
		BC, SP	3	1	—	BC ← SP			
		DE, SP	3	1	—	DE ← SP			
	ADDW	SP, #byte	2	1	—	SP ← SP + byte			
	SUBW	SP, #byte	2	1	—	SP ← SP - byte			
Unconditional branch	BR	AX	2	3	—	PC ← CS, AX			
		$addr20	2	3	—	PC ← PC + 2 + jdisp8			
		$!addr20	3	3	—	PC ← PC + 3 + jdisp16			
		!addr16	3	3	—	PC ← 0000, addr16			
		!!addr20	4	3	—	PC ← addr20			
Conditional branch	BC	$addr20	2	2/4 Note 3	—	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	$addr20	2	2/4 Note 3	—	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	$addr20	2	2/4 Note 3	—	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	$addr20	2	2/4 Note 3	—	PC ← PC + 2 + jdisp8 if Z = 0			
	BH	$addr20	3	2/4 Note 3	—	PC ← PC + 3 + jdisp8 if (Z ∨ CY) = 0			
	BNH	$addr20	3	2/4 Note 3	—	PC ← PC + 3 + jdisp8 if (Z ∨ CY) = 1			
	BT	saddr.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 1			
		sfr.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
		A.bit, $addr20	3	3/5 Note 3	—	PC ← PC + 3 + jdisp8 if A.bit = 1			
		PSW.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if PSW.bit = 1			
		[HL].bit, $addr20	3	3/5 Note 3	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 1			
		ES:[HL].bit, $addr20	4	4/6 Note 3	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1			

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. This indicates the number of clocks “when condition is not met/when condition is met”.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 22 Operation List (18/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Conditional branch	BF	saddr.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 0			
		sfr.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
		A.bit, $addr20	3	3/5 Note 3	—	PC ← PC + 3 + jdisp8 if A.bit = 0			
		PSW.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
		[HL].bit, $addr20	3	3/5 Note 3	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
		ES:[HL].bit, $addr20	4	4/6 Note 3	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0			
	BTCLR	saddr.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit			
		sfr.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit			
		A.bit, $addr20	3	3/5 Note 3	—	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit			
		PSW.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	×	×	×
		[HL].bit, $addr20	3	3/5 Note 3	—	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit			
		ES:[HL].bit, $addr20	4	4/6 Note 3	—	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1 then reset (ES, HL).bit			
Conditional skip	SKC	—	2	1	—	Next instruction skip if CY = 1			
	SKNC	—	2	1	—	Next instruction skip if CY = 0			
	SKZ	—	2	1	—	Next instruction skip if Z = 1			
	SKNZ	—	2	1	—	Next instruction skip if Z = 0			
	SKH	—	2	1	—	Next instruction skip if (Z ∨ CY) = 0			
	SKNH	—	2	1	—	Next instruction skip if (Z ∨ CY) = 1			
CPU control	SEL Note 4	RBn	2	1	—	RBS[1:0] ← n			
	NOP	—	1	1	—	No Operation			
	EI	—	3	4	—	IE ← 1 (Enable Interrupt)			
	DI	—	3	4	—	IE ← 0 (Disable Interrupt)			
	HALT	—	2	3	—	Set HALT Mode			
	STOP	—	2	3	—	Set STOP Mode			

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. This indicates the number of clocks “when condition is not met/when condition is met”.

Note 4. n indicates the number of register banks (n = 0 to 3)

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

CHAPTER 34 ELECTRICAL SPECIFICATIONS (T_A = -40 to +85°C)

This chapter describes the electrical specifications for the products A: Consumer applications (T_A = -40 to +85°C) and G: Industrial applications (when used in the range of T_A = -40 to +85°C).

Caution 1. The RL78 microcontroller has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

Caution 2. The pins mounted depend on the product. Refer to 2.1 Port Function to 2.2.1 With functions for each product.

34.1 Absolute Maximum Ratings

Absolute Maximum Ratings (T_A = 25°C) (1/3)

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}		-0.5 to + 6.5	V
	UV_{BUS}		-0.5 to + 6.5	V
	AV_{DD}	$AV_{DD} \leq V_{DD}$	-0.5 to + 4.6	V
REGC pin input voltage	V_{IREGC}	REGC	-0.3 to + 2.8 and -0.3 to V_{DD} + 0.3 Note 1	V
U_{REGC} pin input voltage	V_{IUREGC}	U_{REGC}	-0.3 to UV_{BUS} + 0.3 Note 2	V
Input voltage	V_{I1}	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P70 to P77, P80 to P83, P125 to P127, P137, P140 to P143, EXCLK, EXCLKS, $\overline{\text{RESET}}$	-0.3 to V_{DD} + 0.3 Note 3	V
	V_{I2}	P60, P61 (N-ch open-drain)	-0.3 to + 6.5	V
	V_{I3}	UDP, UDM	-0.3 to + 6.5	V
	V_{I4}	P150 to P156	-0.3 to AV_{DD} + 0.3 Note 4	V
Output voltage	V_{O1}	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P60, P61, P70 to P77, P80 to P83, P125 to P127, P130, P140 to P143	-0.3 to V_{DD} + 0.3 Note 3	V
	V_{O2}	P150 to P156	-0.3 to AV_{DD} + 0.3 Note 3	V
	V_{O3}	UDP, UDM	-0.3 to + 3.8	V
Analog input voltage	V_{AI1}	ANI16 to ANI21	-0.3 to V_{DD} + 0.3 and $AV_{REF(+)}$ + 0.3 Notes 3, 5	V
	V_{AI2}	ANI0 to ANI6	-0.3 to AV_{DD} + 0.3 and $AV_{REF(+)}$ + 0.3 Notes 3, 5	V

Note 1. Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

Note 2. Connect the U_{REGC} pin to Vss via a capacitor (0.33 μF). This value regulates the absolute maximum rating of the U_{REGC} pin. Do not use this pin with voltage applied to it.

Note 3. Must be 6.5 V or lower.

Note 4. Must be 4.6 V or lower.

Note 5. Do not exceed $AV_{REF(+)}$ + 0.3 V in case of A/D conversion target pin.

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

Remark 2. $AV_{REF(+)}$: + side reference voltage of the A/D converter.

Remark 3. V_{SS}: Reference voltage

Absolute Maximum Ratings (T_A = 25°C) (2/3)

Parameter	Symbols	Conditions		Ratings	Unit
LCD voltage	V_{LI1}	V_{L1} input voltage Note 1		-0.3 to +2.8	V
	V_{LI2}	V_{L2} input voltage Note 1		-0.3 to +6.5	V
	V_{LI3}	V_{L3} input voltage Note 1		-0.3 to +6.5	V
	V_{LI4}	V_{L4} input voltage Note 1		-0.3 to +6.5	V
	V_{LI5}	CAPL, CAPH input voltage Note 1		-0.3 to +6.5	V
	V_{LO1}	V_{L1} output voltage		-0.3 to +2.8	V
	V_{LO2}	V_{L2} output voltage		-0.3 to +6.5	V
	V_{LO3}	V_{L3} output voltage		-0.3 to +6.5	V
	V_{LO4}	V_{L4} output voltage		-0.3 to +6.5	V
	V_{LO5}	CAPL, CAPH output voltage		-0.3 to +6.5	V
	V_{LO6}	COM0 to COM7 SEG0 to SEG55 output voltage	External resistance division method	-0.3 to V_{DD} + 0.3 Note 2	V
			Capacitor split method	-0.3 to V_{DD} + 0.3 Note 2	V
			Internal voltage boosting method	-0.3 to V_{LI4} + 0.3 Note 2	V

Note 1. This value only indicates the absolute maximum ratings when applying voltage to the V_{L1}, V_{L2}, V_{L3}, and V_{L4} pins; it does not mean that applying voltage to these pins is recommended. When using the internal voltage boosting method or capacitance split method, connect these pins to V_{SS} via a capacitor (0.47 ± 30%) and connect a capacitor (0.47 ± 30%) between the CAPL and CAPH pins.

Note 2. Must be 6.5 V or lower.

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.

Absolute Maximum Ratings (T_A = 25°C) (3/3)

Parameter	Symbols	Conditions		Ratings	Unit
Output current, high	I_{OH1}	Per pin	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P70 to P77, P80 to P83, P125 to P127, P130, P140 to P143	-40	mA
		Total of all pins -170 mA	P40 to P46	-70	mA
			P00 to P07, P10 to P17, P20 to P27, P30 to P37, P50 to P57, P70 to P77, P80 to P83, P125 to P127, P130, P140 to P143	-100	mA
	I_{OH2}	Per pin	P150 to P156	-0.1	mA
		Total of all pins		-0.7	mA
	I_{OH3}	Per pin	UDP, UDM	-3	mA
Output current, low	I_{OL1}	Per pin	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P60, P61, P70 to P77, P80 to P83, P125 to P127, P130, P140 to P143	40	mA
		Total of all pins 170 mA	P40 to P46	70	mA
			P00 to P07, P10 to P17, P20 to P27, P30 to P37, P50 to P57, P70 to P77, P80 to P83, P125 to P127, P130, P140 to P143	100	mA
	I_{OL2}	Per pin	P150 to P156	0.4	mA
		Total of all pins		2.8	mA
	I_{OL3}	Per pin	UDP, UDM	3	mA
Operating ambient temperature	T_A	In normal operation mode		-40 to +85	°C
		In flash memory programming mode			
Storage temperature	T_{stg}			-65 to +150	°C

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

34.2 Oscillator Characteristics

34.2.1 X1 and XT1 oscillator characteristics

(T_A = -40 to +85°C, 1.6 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Resonator	Conditions	MIN.	TYP.	MAX.	Unit
X1 clock oscillation frequency (f_X) Note	Ceramic resonator/crystal resonator	2.7 V ≤ V_{DD} ≤ 3.6 V	1.0		20.0	MHz
		2.4 V ≤ V_{DD} < 2.7 V	1.0		16.0	
		1.8 V ≤ V_{DD} < 2.4 V	1.0		8.0	
		1.6 V ≤ V_{DD} < 1.8 V	1.0		4.0	
XT1 clock oscillation frequency (f_{XT}) Note	Crystal resonator		32	32.768	35	kHz

Note Indicates only permissible oscillator frequency ranges. Refer to **AC Characteristics** for instruction execution time. Request evaluation by the manufacturer of the oscillator circuit mounted on a board to check the oscillator characteristics.

Caution **Since the CPU is started by the high-speed on-chip oscillator clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and the oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.**

Remark When using the X1 and XT1 oscillator, refer to **5.4 System Clock Oscillator**.

34.2.2 On-chip oscillator characteristics

(T_A = -40 to +85°C, 1.6 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Oscillators	Parameters	Conditions		MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency Notes 1, 2	f_{HOCO}			1		48	MHz
High-speed on-chip oscillator clock frequency accuracy		-20 to +85°C	1.8 V ≤ V_{DD} ≤ 3.6 V	-1.0		+1.0	%
			1.6 V ≤ V_{DD} ≤ 1.8 V	-5.0		+5.0	%
		-40 to -20°C	1.8 V ≤ V_{DD} < 3.6 V	-1.5		+1.5	%
			1.6 V ≤ V_{DD} ≤ 1.8 V	-5.5		+5.5	%
Low-speed on-chip oscillator clock frequency	f_{IL}				15		kHz
Low-speed on-chip oscillator clock frequency accuracy				-15		+15	%

Note 1. High-speed on-chip oscillator frequency is selected with bits 0 to 4 of the option byte (000C2H) and bits 0 to 2 of the HOCODIV register.

Note 2. This only indicates the oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

34.2.3 PLL oscillator characteristics

(T_A = -40 to +85°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Oscillators	Parameters	Conditions	MIN.	TYP.	MAX.	Unit
PLL input frequency Note	f_{PLLIN}	High-speed system clock	6.00		16.00	MHz
PLL output frequency Note	f_{PLL}			48.00		MHz

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

34.3 DC Characteristics

34.3.1 Pin characteristics

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high Note 1	I_{OH1}	Per pin for P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P70 to P77, P80 to P83, P125 to P127, P130, P140 to P143				-10.0 Note 2	mA
		Total of P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P70 to P77, P80 to P83, P125 to P127, P130, P140 to P143 (When duty ≤ 70% Note 3)	2.7 V ≤ V_{DD} ≤ 3.6 V			-15.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			-7.0	mA
			1.6 V ≤ V_{DD} < 1.8 V			-3.0	mA
	I_{OH2}	Per pin for P150 to P156	1.6 V ≤ V_{DD} ≤ 3.6 V			-0.1 Note 2	mA
		Total of all pins	1.6 V ≤ V_{DD} ≤ 3.6 V			-0.7	mA

Note 1. Value of current at which the device operation is guaranteed even if the current flows from the V_{DD} pin to an output pin.

Note 2. However, do not exceed the total current value.

Note 3. Specification under conditions where the duty factor ≤ 70%.
The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).

- Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OH} = -10.0 mA
Total output current of pins = (-10.0 × 0.7)/(80 × 0.01) ≈ -8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Caution P00 to P02, P10 to P12, P24 to P26, P33 to P35, and P42 to P44 do not output high level in N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low Note 1	I_{OL1}	Per pin for P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P70 to P77, P80 to P83, P125 to P127, P130, P140 to P143				20.0 Note 2	mA
		Per pin for P60 and P61				15.0 Note 2	mA
		Total of P40 to P46, P130 (When duty ≤ 70% Note 3)	2.7 V ≤ V_{DD} ≤ 3.6 V			15.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			9.0	mA
			1.6 V ≤ V_{DD} < 1.8 V			4.5	mA
		Total of P00 to P07, P10 to P17, P20 to P27, P30 to P37, P50 to P57, P60, P61, P70 to P77, P80 to P83, P125 to P127, P140 to P143 (When duty ≤ 70% Note 3)	2.7 V ≤ V_{DD} ≤ 3.6 V			35.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			20.0	mA
			1.6 V ≤ V_{DD} < 1.8 V			10.0	mA
		Total of all pins (When duty ≤ 70% Note 3)				50.0	mA
	I_{OL2}	Per pin for P150 to P156				0.4 Note 2	mA
		Total of all pins	1.6 V ≤ V_{DD} ≤ 3.6 V			2.8	mA

Note 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to the V_{SS} pin.

Note 2. However, do not exceed the total current value.

Note 3. Specification under conditions where the duty factor ≤ 70%.
The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression
(when changing the duty factor from 70% to n%).

- Total output current of pins = (I_{OL} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OL} = 10.0 mA
Total output current of pins = (10.0 × 0.7)/(80 × 0.01) ≈ 8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P70 to P77, P80 to P83, P125 to P127, P140 to P143	Normal input buffer	0.8 V_{DD}		V_{DD}	V
	V_{IH2}	P00, P01, P10, P11, P24, P25, P33, P34, P43, P44	TTL input buffer 3.3 V ≤ V_{DD} ≤ 3.6 V	2.0		V_{DD}	V
			TTL input buffer 1.6 V ≤ V_{DD} < 3.3 V	1.50		V_{DD}	V
	V_{IH3}	P150 to P156		0.7 AV_{DD}		AV_{DD}	V
	V_{IH4}	P60, P61		0.7 V_{DD}		6.0	V
	V_{IH5}	P121 to P124, P137, EXCLK, EXCLKS, $\overline{\text{RESET}}$		0.8 V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P70 to P77, P80 to P83, P125 to P127, P140 to P143	Normal input buffer	0		0.2 V_{DD}	V
	V_{IL2}	P00, P01, P10, P11, P24, P25, P33, P34, P43, P44	TTL input buffer 3.3 V ≤ V_{DD} ≤ 3.6 V	0		0.5	V
			TTL input buffer 1.6 V ≤ V_{DD} < 3.3 V	0		0.32	V
	V_{IL3}	P150 to P156		0		0.3 AV_{DD}	V
	V_{IL4}	P60, P61		0		0.3 V_{DD}	V
	V_{IL5}	P121 to P124, P137, EXCLK, EXCLKS, $\overline{\text{RESET}}$		0		0.2 V_{DD}	V

Caution The maximum value of V_{IH} of pins P00 to P02, P10 to P12, P24 to P26, P33 to P35, and P42 to P44 is V_{DD}, even in the N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH1}	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P70 to P77, P80 to P83, P125 to P127, P130, P140 to P143	2.7 V ≤ V_{DD} ≤ 3.6 V, I_{OH1} = -2.0 mA	V_{DD} - 0.6			V
			1.8 V ≤ V_{DD} ≤ 3.6 V, I_{OH1} = -1.5 mA	V_{DD} - 0.5			V
			1.6 V ≤ V_{DD} < 3.6 V, I_{OH1} = -1.0 mA	V_{DD} - 0.5			V
	V_{OH2}	P150 to P156	1.6 V ≤ V_{DD} ≤ 3.6 V, I_{OH2} = -100 µA	AV_{DD} - 0.5			V
Output voltage, low	V_{OL1}	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P70 to P77, P80 to P83, P125 to P127, P130, P140 to P143	2.7 V ≤ V_{DD} ≤ 3.6 V, I_{OL1} = 3.0 mA			0.6	V
			2.7 V ≤ V_{DD} ≤ 3.6 V, I_{OL1} = 1.5 mA			0.4	V
			1.8 V ≤ V_{DD} ≤ 3.6 V, I_{OL1} = 0.6 mA			0.4	V
			1.6 V ≤ V_{DD} < 1.8 V, I_{OL1} = 0.3 mA			0.4	V
	V_{OL2}	P150 to P156	1.6 V ≤ V_{DD} ≤ 3.6 V, I_{OL2} = 400 µA			0.4	V
	V_{OL3}	P60, P61	2.7 V ≤ V_{DD} ≤ 3.6 V, I_{OL3} = 3.0 mA			0.4	V
			1.8 V ≤ V_{DD} ≤ 3.6 V, I_{OL3} = 2.0 mA			0.4	V
			1.6 V ≤ V_{DD} ≤ 1.8 V, I_{OL3} = 1.0 mA			0.4	V

Caution P00 to P02, P10 to P12, P24 to P26, P33 to P35, and P42 to P44 do not output high level in N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P60, P61, P70 to P77, P80 to P83, P125 to P127, P137, P140 to P143, $\overline{RESET}$	V_I = V_{DD}				1	μA
	I_{LIH2}	P20, P21, P140 to P143	V_I = V_{DD}				1	μA
	I_{LIH3}	P121 to P124 (X1, X2, EXCLK, XT1, XT2, EXCLKS)	V_I = V_{DD}	In input port or external clock input			1	μA
				In resonator connection			10	μA
	I_{LIH4}	P150 to P156	V_I = AV_{DD}				1	μA
Input leakage current, low	I_{LIL1}	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P60, P61, P70 to P77, P80 to P83, P125 to P127, P137, P140 to P143, $\overline{RESET}$	V_I = V_{SS}				-1	μA
	I_{LIL2}	P20, P21, P140 to P143	V_I = V_{SS}				-1	μA
	I_{LIL3}	P121 to P124 (X1, X2, EXCLK, XT1, XT2, EXCLKS)	V_I = V_{SS}	In input port or external clock input			-1	μA
				In resonator connection			-10	μA
	I_{LIL4}	P150 to P156	V_I = AV_{SS}				-1	μA
On-chip pull-up resistance	R_{U1}	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P50 to P57, P70 to P77, P140 to P143, P125 to P127	V_I = V_{SS}	2.4 V ≤ V_{DD} ≤ 3.6 V	10	20	100	kΩ
				1.6 V ≤ V_{DD} ≤ 2.4 V	10	30	100	
	R_{U2}	P40 to P46, P80 to P83	V_I = V_{SS}		10	20	100	kΩ

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

34.3.2 Supply current characteristics

(T_A = -40 to +85°C, 1.6 V ≤ V_{DD} ≤ 3.6 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_{HOCO} = 48 MHz Note 3, f_{IH} = 24 MHz Note 3	Basic operation	V_{DD} = 3.6 V		2.2	2.8	mA
						V_{DD} = 3.0 V		2.2	2.8	
					Normal operation	V_{DD} = 3.6 V		4.4	8.5	
						V_{DD} = 3.0 V		4.4	8.5	
				f_{HOCO} = 24 MHz Note 3, f_{IH} = 24 MHz Note 3	Basic operation	V_{DD} = 3.6 V		2.0	2.6	
						V_{DD} = 3.0 V		2.0	2.6	
					Normal operation	V_{DD} = 3.6 V		4.2	6.8	
						V_{DD} = 3.0 V		4.2	6.8	
				f_{HOCO} = 16 MHz Note 3, f_{IH} = 16 MHz Note 3	Normal operation	V_{DD} = 3.6 V		3.1	4.9	
						V_{DD} = 3.0 V		3.1	4.9	
			LS (low-speed main) mode Note 5	f_{HOCO} = 8 MHz Note 3, f_{IH} = 8 MHz Note 3	Normal operation	V_{DD} = 3.0 V		1.4	2.2	mA
						V_{DD} = 2.0 V		1.4	2.2	
			LV (low-voltage main) mode Note 5	f_{HOCO} = 4 MHz Note 3, f_{IH} = 4 MHz Note 3	Normal operation	V_{DD} = 3.0 V		1.3	1.8	mA
						V_{DD} = 2.0 V		1.3	1.8	
			HS (high-speed main) mode Note 5	f_{MX} = 20 MHz Note 2, V_{DD} = 3.6 V	Normal operation	Square wave input		3.5	5.5	mA
						Resonator connection		3.6	5.7	
				f_{MX} = 20 MHz Note 2, V_{DD} = 3.0 V	Normal operation	Square wave input		3.5	5.5	
						Resonator connection		3.6	5.7	
				f_{MX} = 16 MHz Note 2, V_{DD} = 3.6 V	Normal operation	Square wave input		2.9	4.5	
						Resonator connection		3.1	4.6	
				f_{MX} = 16 MHz Note 2, V_{DD} = 3.0 V	Normal operation	Square wave input		2.9	4.5	
						Resonator connection		3.1	4.6	
				f_{MX} = 10 MHz Note 2, V_{DD} = 3.6 V	Normal operation	Square wave input		2.1	3.2	
						Resonator connection		2.2	3.2	
				f_{MX} = 10 MHz Note 2, V_{DD} = 3.0 V	Normal operation	Square wave input		2.1	3.2	
						Resonator connection		2.2	3.2	
			LS (low-speed main) mode Note 5	f_{MX} = 8 MHz Note 2, V_{DD} = 3.6 V	Normal operation	Square wave input		1.2	2.0	mA
						Resonator connection		1.3	2.0	
				f_{MX} = 8 MHz Note 2, V_{DD} = 3.0 V	Normal operation	Square wave input		1.2	2.1	
						Resonator connection		1.3	2.2	
			HS (High-speed main) mode (PLL operation)	f_{PLL} = 48 MHz, f_{CLK} = 24 MHz Note 2	Normal operation	V_{DD} = 3.6 V		4.7	7.5	mA
						V_{DD} = 3.0 V		4.7	7.5	
				f_{PLL} = 48 MHz, f_{CLK} = 12 MHz Note 2	Normal operation	V_{DD} = 3.6 V		3.1	5.1	
						V_{DD} = 3.0 V		3.1	5.1	
				f_{PLL} = 48 MHz, f_{CLK} = 6 MHz Note 2	Normal operation	V_{DD} = 3.6 V		2.3	3.9	
						V_{DD} = 3.0 V		2.3	3.9	
			Subsystem clock operation	f_{SUB} = 32.768 kHz Note 4 T_A = -40°C	Normal operation	Square wave input		4.6	6.9	μA
						Resonator connection		4.7	6.9	
				f_{SUB} = 32.768 kHz Note 4 T_A = +25°C	Normal operation	Square wave input		4.9	7.0	
						Resonator connection		5.0	7.2	
				f_{SUB} = 32.768 kHz Note 4 T_A = +50°C	Normal operation	Square wave input		5.2	7.6	
						Resonator connection		5.2	7.7	
				f_{SUB} = 32.768 kHz Note 4 T_A = +70°C	Normal operation	Square wave input		5.5	9.3	
						Resonator connection		5.6	9.4	
				f_{SUB} = 32.768 kHz Note 4 T_A = +85°C	Normal operation	Square wave input		6.2	13.3	
						Resonator connection		6.2	13.4	

(**Notes** and **Remarks** are listed on the next page.)

Note 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD}, or V_{SS}. The following points apply in the HS (high-speed main), LS (low-speed main), and LV (low-voltage 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 LCD controller/driver, A/D converter, D/A converter, comparator, LVD circuit, USB 2.0 function module, 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 real-time clock 2.

Note 2. When high-speed on-chip oscillator and subsystem clock are stopped.

Note 3. When high-speed system clock and subsystem clock are stopped.

Note 4. When high-speed on-chip oscillator and high-speed system clock are stopped. When AMPHS1 = 1 (Ultra-low power consumption oscillation).

Note 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} ≤ 3.6 V@1 MHz to 24 MHz
	2.4 V ≤ V_{DD} ≤ 3.6 V@1 MHz to 16 MHz
LS (low-speed main) mode:	1.8 V ≤ V_{DD} ≤ 3.6 V@1 MHz to 8 MHz
LV (low-voltage main) mode	1.6 V ≤ V_{DD} ≤ 3.6 V@1 MHz to 4 MHz

Remark 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

Remark 2. f_{HOCO}: High-speed on-chip oscillator clock frequency (48 MHz max.)

Remark 3. f_{IH}: Main system clock source frequency when the high-speed on-chip oscillator clock divided 1, 2, 4, or 8, or the PLL clock divided by 2, 4, or 8 is selected (24 MHz max.)

Remark 4. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

Remark 5. Except subsystem clock operation, temperature condition of the TYP. value is T_A = 25°C

(T_A = -40 to +85°C, 1.6 V ≤ V_{DD} ≤ 3.6 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_{HOCO} = 48 MHz Note 4, f_{IH} = 24 MHz Note 4	V_{DD} = 3.6 V		0.77	2.70	mA
					V_{DD} = 3.0 V		0.77	2.70	
				f_{HOCO} = 24 MHz Note 4, f_{IH} = 24 MHz Note 4	V_{DD} = 3.6 V		0.55	1.91	
					V_{DD} = 3.0 V		0.55	1.90	
				f_{HOCO} = 16 MHz Note 4, f_{IH} = 16 MHz Note 4	V_{DD} = 3.6 V		0.48	1.41	
					V_{DD} = 3.0 V		0.47	1.41	
			LS (low-speed main) mode Note 6	f_{HOCO} = 8 MHz Note 4, f_{IH} = 8 MHz Note 4	V_{DD} = 3.0 V		300	770	μA
					V_{DD} = 2.0 V		300	770	
			LV (low-voltage main) mode Note 6	f_{HOCO} = 4 MHz Note 4, f_{IH} = 4 MHz Note 4	V_{DD} = 3.0 V		440	770	μA
					V_{DD} = 2.0 V		440	770	
			HS (high-speed main) mode Note 6	f_{MX} = 20 MHz Note 3, V_{DD} = 3.6 V	Square wave input		0.35	1.63	mA
					Resonator connection		0.51	1.68	
				f_{MX} = 20 MHz Note 3, V_{DD} = 3.0 V	Square wave input		0.34	1.63	
					Resonator connection		0.51	1.68	
				f_{MX} = 16 MHz Note 3, V_{DD} = 3.6 V	Square wave input		0.30	1.22	
					Resonator connection		0.45	1.39	
				f_{MX} = 16 MHz Note 3, V_{DD} = 3.0 V	Square wave input		0.29	1.20	
					Resonator connection		0.45	1.38	
				f_{MX} = 10 MHz Note 3, V_{DD} = 3.6 V	Square wave input		0.23	0.82	
					Resonator connection		0.30	0.90	
				f_{MX} = 10 MHz Note 3, V_{DD} = 3.0 V	Square wave input		0.22	0.81	
					Resonator connection		0.30	0.89	
			LS (low-speed main) mode Note 6	f_{MX} = 8 MHz Note 3, V_{DD} = 3.0 V	Square wave input		120	510	μA
					Resonator connection		170	560	
				f_{MX} = 8 MHz Note 3, V_{DD} = 2.0 V	Square wave input		130	520	
					Resonator connection		170	570	
			HS (High-speed main) mode (PLL operation)	f_{MX} = 48 MHz, f_{CLK} = 24 MHz Note 3	V_{DD} = 3.6 V		0.99	2.89	mA
					V_{DD} = 3.0 V		0.99	2.88	
				f_{MX} = 48 MHz, f_{CLK} = 12 MHz Note 3	V_{DD} = 3.6 V		0.89	2.48	
					V_{DD} = 3.0 V		0.89	2.47	
				f_{MX} = 48 MHz, f_{CLK} = 6 MHz Note 3	V_{DD} = 3.6 V		0.84	2.27	
					V_{DD} = 3.0 V		0.84	2.27	
			Subsystem clock operation	f_{SUB} = 32.768 kHz Note 5 T_A = -40°C	Square wave input		0.32	0.61	μA
					Resonator connection		0.51	0.80	
				f_{SUB} = 32.768 kHz Note 5 T_A = +25°C	Square wave input		0.41	0.74	
					Resonator connection		0.62	0.91	
				f_{SUB} = 32.768 kHz Note 5 T_A = +50°C	Square wave input		0.52	2.30	
					Resonator connection		0.75	2.49	
				f_{SUB} = 32.768 kHz Note 5 T_A = +70°C	Square wave input		0.82	4.03	
					Resonator connection		1.08	4.22	
				f_{SUB} = 32.768 kHz Note 5 T_A = +85°C	Square wave input		1.38	8.04	
					Resonator connection		1.62	8.23	
	I_{DD3}	STOP mode Note 7	T_A = -40°C				0.18	0.52	μA
			T_A = +25°C				0.25	0.52	
			T_A = +50°C				0.34	2.21	
			T_A = +70°C				0.64	3.94	
			T_A = +85°C				1.18	7.95	

(**Notes** and **Remarks** are listed on the next page.)

Note 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The following points apply in the HS (high-speed main), LS (low-speed main), and LV (low-voltage 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 LCD controller/driver, A/D converter, D/A converter, comparator, LVD circuit, USB 2.0 function module, 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 real-time clock 2.
In the STOP mode, the currents in both the “TYP.” and “MAX.” columns do not include the operating currents of the peripheral modules.

Note 2. During HALT instruction execution by flash memory.

Note 3. When high-speed on-chip oscillator and subsystem clock are stopped.

Note 4. When high-speed system clock and subsystem clock are stopped.

Note 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).

Note 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} ≤ 3.6 V@1 MHz to 24 MHz
	2.4 V ≤ V_{DD} ≤ 3.6 V@1 MHz to 16 MHz
LS (low-speed main) mode:	1.8 V ≤ V_{DD} ≤ 3.6 V@1 MHz to 8 MHz
LV (low-voltage main) mode	1.6 V ≤ V_{DD} ≤ 3.6 V@1 MHz to 4 MHz

Note 7. Regarding the value for current to operate the subsystem clock in STOP mode, refer to that in HALT mode.

Remark 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

Remark 2. f_{HOCO}: High-speed on-chip oscillator clock frequency (48 MHz max.)

Remark 3. f_{IH}: Main system clock source frequency when the high-speed on-chip oscillator clock divided 1, 2, 4, or 8, or the PLL clock divided by 2, 4, or 8 is selected (24 MHz max.)

Remark 4. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

Remark 5. Except subsystem clock operation and STOP mode, temperature condition of the TYP. value is T_A = 25°C

(T_A = -40 to +85°C, 1.6 V ≤ V_{DD} ≤ 3.6 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
RTC2 operating current	I_{RTC} Notes 1, 3						0.02		μA
12-bit interval timer operating current	I_{TMKA} 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 6, 7	AV_{DD} = 3.0 V, when conversion at maximum speed					422	720	μA
$AV_{REF(+)}$ current	I_{AVREF} Note 8	AV_{DD} = 3.0 V, ADREFP1 = 0, ADREFP0 = 0 Note 7					14.0	25.0	μA
		AV_{REFP} = 3.0 V, ADREFP1 = 0, ADREFP0 = 1 Note 10					14.0	25.0	
		ADREFP1 = 1, ADREFP0 = 0 Note 1					14.0	25.0	
A/D converter reference voltage current	I_{ADREF} Notes 1, 9	V_{DD} = 3.0 V					75.0		μA
Temperature sensor operating current	I_{TMPS} Note 1						78		μA
D/A converter operating current	I_{DAC} Notes 1, 11	Per D/A converter channel					0.53	1.5	mA
Comparator operating current	I_{CMP} Notes 1, 12	V_{DD} = 3.6 V, Regulator output voltage = 2.1 V	Window mode				12.5		μA
			Comparator high-speed mode				4.5		μA
			Comparator low-speed mode				1.2		μA
		V_{DD} = 3.6 V, Regulator output voltage = 1.8 V	Window mode				7.05		μA
			Comparator high-speed mode				2.2		μA
			Comparator low-speed mode				0.9		μA
LVD operating current	I_{LVI} Notes 1, 13						0.06		μA
Self-programming operating current	I_{FSP} Notes 1, 14						2.50	12.20	mA
BGO operating current	I_{BGO} Notes 1, 15						1.68	12.20	mA
SNOOZE operating current	I_{SNOZ} Note 1	ADC operation	The mode is performed Note 16				0.34	1.10	mA
			The A/D conversion operations are performed, Low voltage mode, AV_{REFP} = V_{DD} = 3.0 V				0.53	2.04	
		Simplified SPI (CSI)/UART operation					0.70	1.54	mA
LCD operating current	I_{LCD1} Notes 17, 18	External resistance division method	f_{LCD} = f_{SUB} LCD clock = 128 Hz	1/3 bias 4-time slice	V_{DD} = 3.6 V, L_{V4} = 3.6 V		0.14		μA
	I_{LCD2} Note 17	Internal voltage boosting method	f_{LCD} = f_{SUB} LCD clock = 128 Hz	1/3 bias 4-time slice	V_{DD} = 3.0 V, L_{V4} = 3.0 V (VLCD = 04H)		0.61		μA
	I_{LCD3} Note 17	Capacitor split method	f_{LCD} = f_{SUB} LCD clock = 128 Hz	1/3 bias 4-time slice	V_{DD} = 3.0 V, L_{V4} = 3.0 V		0.12		μA
USB current Note 19	I_{USB} Note 20	Operating current during USB communication					4.88		mA
	I_{USB} Note 21	Operating current in the USB suspended state					0.04		mA

(**Notes** and **Remarks** are listed on the next page.)

Note 1. Current flowing to V_{DD}.

Note 2. When high speed on-chip oscillator and high-speed system clock are stopped.

Note 3. Current flowing only to the real-time clock 2 (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 2 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 2.

Note 4. Current flowing only to the 12-bit interval timer (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_{TMKA}, 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. I_{DD2} subsystem clock operation includes the operational current of the 12-bit interval timer.

Note 5. Current flowing only to the watchdog timer (including the operating current of the low-speed on-chip oscillator). The current value of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when the watchdog timer operates in STOP mode.

Note 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}, I_{AVREF}, I_{ADREF} when the A/D converter operates in an operation mode or the HALT mode.

Note 7. Current flowing to the AV_{DD}.

Note 8. Current flowing from the reference voltage source of A/D converter.

Note 9. Operation current flowing to the internal reference voltage.

Note 10. Current flowing to the AV_{REFP}.

Note 11. Current flowing only to the D/A converter. The current value of the RL78 microcontrollers is the sum of I_{DD1} or I_{DD2} and I_{DA} when the D/A converter operates in an operation mode or the HALT mode.

Note 12. Current flowing only to the comparator circuit. The current value of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{CMP} when the comparator circuit operates in the Operating, HALT or STOP mode.

Note 13. Current flowing only to the LVD circuit. The current value of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVI} when the LVD circuit operates in the Operating, HALT or STOP mode.

Note 14. Current flowing only during self-programming.

Note 15. Current flowing only during data flash rewrite.

Note 16. For shift time to the SNOOZE mode, see **23.3.3 SNOOZE mode**.

Note 17. Current flowing only to the LCD controller/driver (V_{DD} pin). The current value of the RL78 microcontrollers is the sum of the LCD operating current (I_{LCD1}, I_{LCD2} or I_{LCD3}) to the supply current (I_{DD1}, or I_{DD2}) when the LCD controller/driver operates in an operation mode or HALT mode. Not including the current that flows through the LCD panel.

Note 18. Not including the current that flows through the external divider resistor divider resistor.

Note 19. Current flowing to the UV_{BUS}.

Note 20. Including the operating current when f_{PLL} = 48 MHz.

Note 21. Including the current supplied from the pull-up resistor of the UDP pin to the pull-down resistor of the host device, in addition to the current consumed by this MCU during the suspended state.

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

Remark 2. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

Remark 3. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 4. Temperature condition of the TYP. value is T_A = 25°C

34.4 AC Characteristics

34.4.1 Basic operation

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V) (1/2)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	HS (high-speed main) mode	2.7 V ≤ V_{DD} ≤ 3.6 V	0.0417		1	μs
				2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	μs
			LS (low-speed main) mode	1.8 V ≤ V_{DD} ≤ 3.6 V	0.125		1	μs
			LV (low-voltage main) mode	1.6 V ≤ V_{DD} ≤ 3.6 V	0.25		1	μs
		Subsystem clock (f_{SUB}) operation		1.8 V ≤ V_{DD} ≤ 3.6 V	28.5	30.5	31.3	μs
		In the self-programming mode	HS (high-speed main) mode	2.7 V ≤ V_{DD} ≤ 3.6 V	0.0417		1	μs
				2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	μs
			LS (low-speed main) mode	1.8 V ≤ V_{DD} ≤ 3.6 V	0.125		1	μs
			LV (low-voltage main) mode	1.8 V ≤ V_{DD} ≤ 3.6 V	0.25		1	μs
External main system clock frequency	f_{EX}	2.7 V ≤ V_{DD} ≤ 3.6 V			1.0		20.0	MHz
		2.4 V ≤ V_{DD} < 2.7 V			1.0		16.0	MHz
		1.8 V ≤ V_{DD} < 2.4 V			1.0		8.0	MHz
		1.6 V ≤ V_{DD} < 1.8 V			1.0		4.0	MHz
	f_{EXT}				32		35	kHz
External main system clock input high-level width, low-level width	t_{EXH}, t_{EXL}	2.7 V ≤ V_{DD} ≤ 3.6 V			24			ns
		2.4 V ≤ V_{DD} < 2.7 V			30			ns
		1.8 V ≤ V_{DD} < 2.4 V			60			ns
		1.6 V ≤ V_{DD} < 1.8 V			120			ns
	t_{EXHS}, t_{EXLS}				13.7			μs
TI00 to TI07 input high-level width, low-level width	t_{TIH}, t_{TIL}				1/f_{MCK} + 10			ns

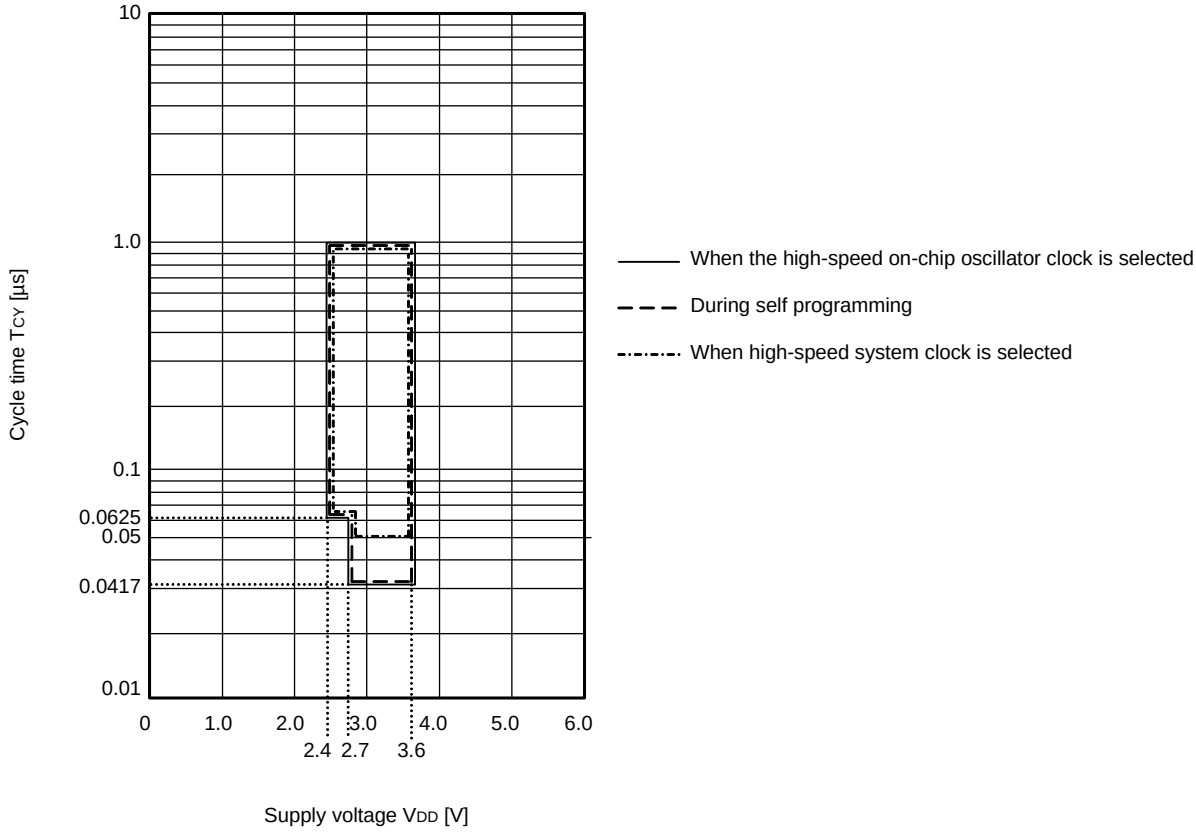
Remark f_{MCK}: Timer array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of timer mode register mn (TMRmn). m: Unit number (m = 0), n: Channel number (n = 0 to 7))

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V) **(2/2)**

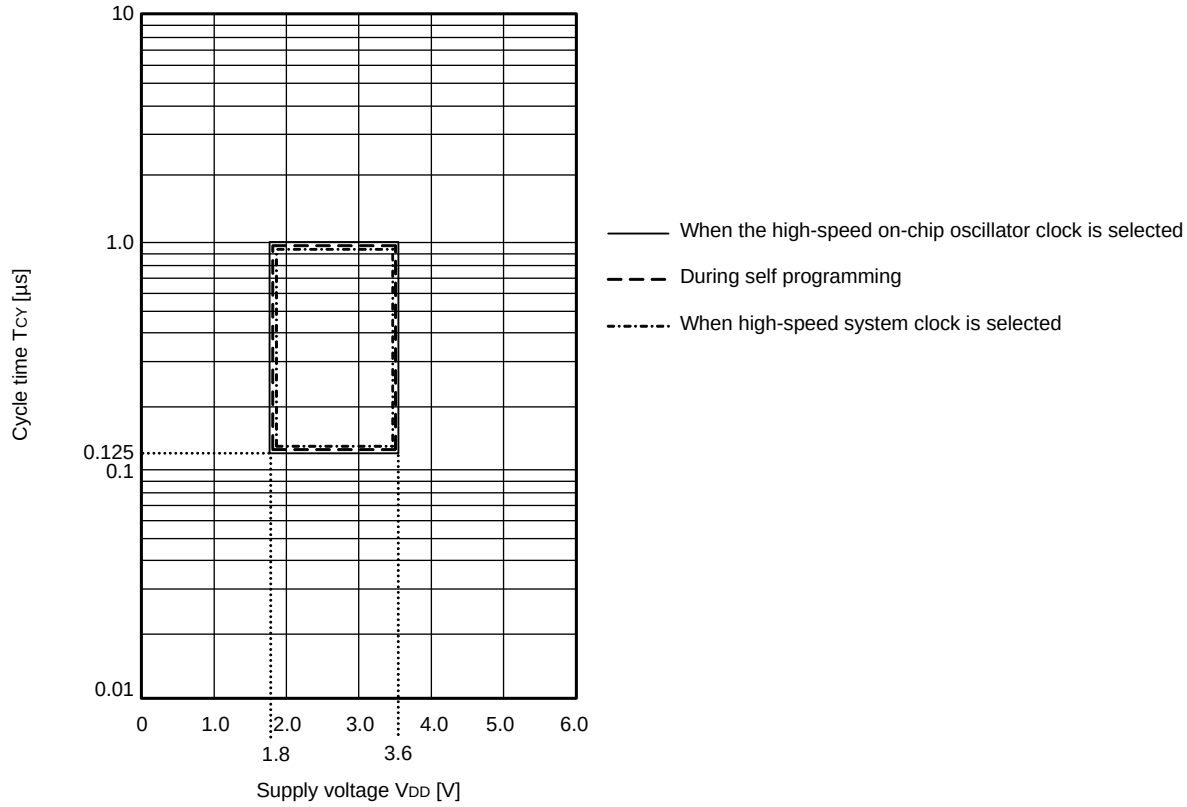
Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
TO00 to TO07, TKBO00, TKBO01, TKBO10, TKBO11, TKBO20, TKBO21 output frequency	f_{TO}	HS (high-speed main) mode	2.7 V ≤ V_{DD} ≤ 3.6 V			8	MHz
			2.4 V ≤ V_{DD} < 2.7 V			8	MHz
		LS (low-speed main) mode	1.8 V ≤ V_{DD} ≤ 3.6 V			4	MHz
		LV (low-voltage main) mode	1.6 V ≤ V_{DD} ≤ 3.6 V			2	MHz
PCLBUZ0, PCLBUZ1 output frequency	f_{PCL}	HS (high-speed main) mode	2.7 V ≤ V_{DD} ≤ 3.6 V			8	MHz
			2.4 V ≤ V_{DD} < 2.7 V			8	MHz
		LS (low-speed main) mode	1.8 V ≤ V_{DD} ≤ 3.6 V			4	MHz
		LV (low-voltage main) mode	1.8 V ≤ V_{DD} ≤ 3.6 V			2	MHz
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}	INTP0 to INTP7	1.6 V ≤ V_{DD} ≤ 3.6 V	1			µs
Key interrupt input low-level width	t_{KR}	1.8 V ≤ V_{DD} ≤ 3.6 V		250			ns
		1.6 V ≤ V_{DD} < 1.8 V		1			µs
TMKB2 forced output stop input high-level width	t_{IHR}	INTP0 to INTP7	f_{CLK} > 16 MHz	125			ns
			f_{CLK} ≤ 16 MHz	2			f_{CLK}
$\overline{\text{RESET}}$ low-level width	t_{RSL}			10			µs

Minimum Instruction Execution Time during Main System Clock Operation

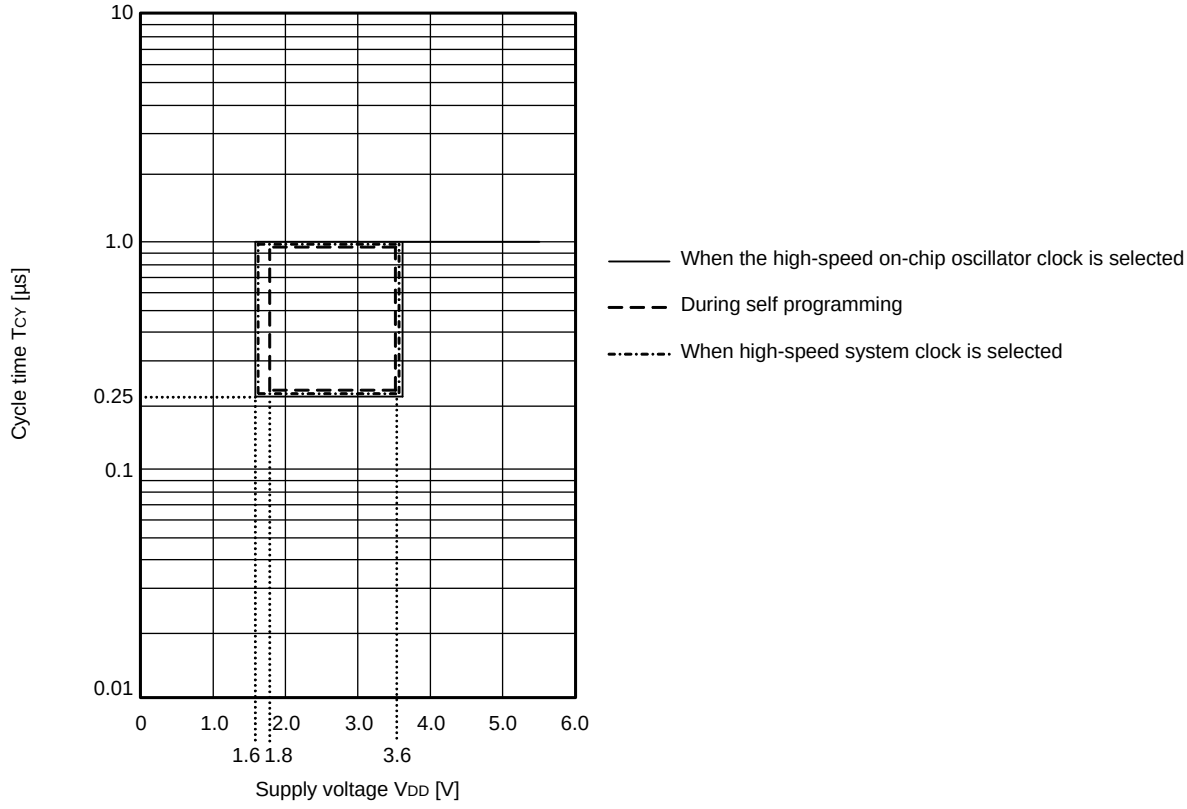
T_{CY} vs V_{DD} (HS (high-speed main) mode)



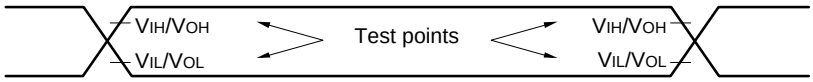
T_{CY} vs V_{DD} (LS (low-speed main) mode)



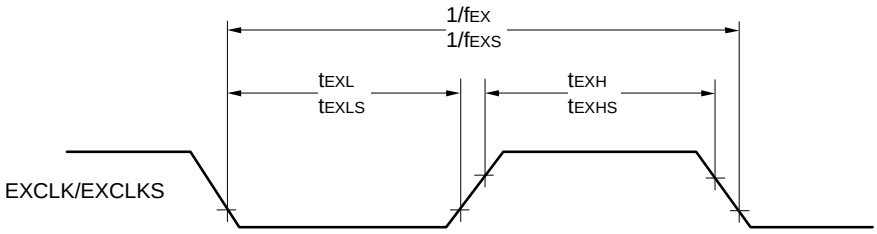
T_{CY} vs V_{DD} (LV (low-voltage main) mode)



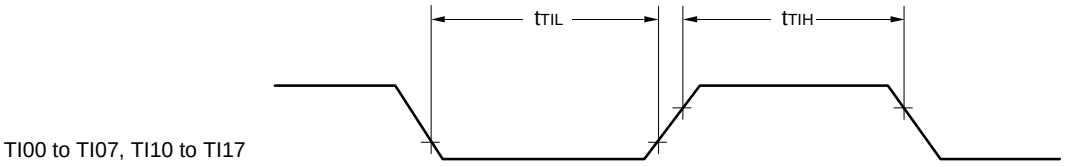
AC Timing Test Points

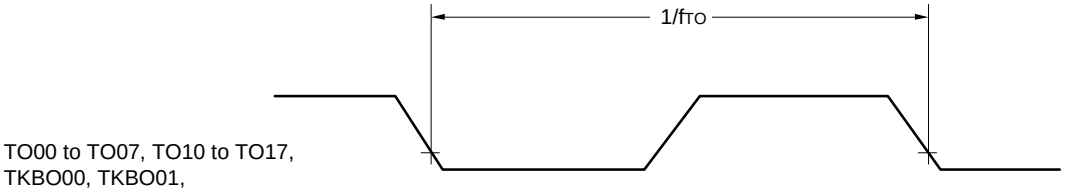


External System Clock Timing

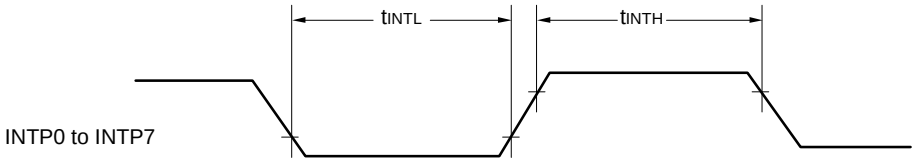


TI/TO Timing

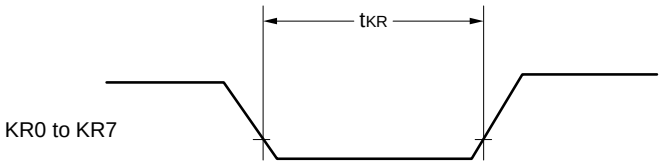




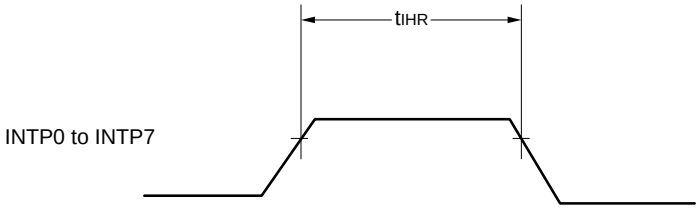
Interrupt Request Input Timing



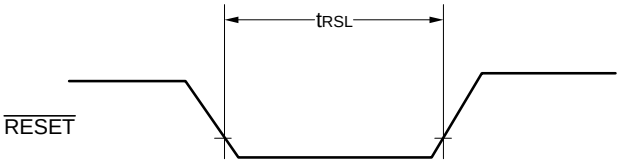
Key Interrupt Input Timing



Timer KB2 Input Timing

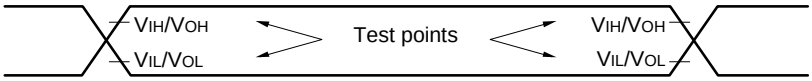


$\overline{\text{RESET}}$ Input Timing



34.5 Peripheral Functions Characteristics

AC Timing Test Points



34.5.1 Serial array unit

(1) During communication at same potential (UART mode)

(T_A = -40 to +85°C, 1.6 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Transfer rate Note 1		2.7 V ≤ V_{DD} ≤ 3.6 V		f_{MCK}/6 Note 2		f_{MCK}/6		f_{MCK}/6	bps
		Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 3		4.0		1.3		0.6	Mbps
		2.4 V ≤ V_{DD} ≤ 3.6 V		f_{MCK}/6 Note 2		f_{MCK}/6		f_{MCK}/6	bps
		Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 3		2.6		1.3		0.6	Mbps
		1.8 V ≤ V_{DD} ≤ 3.6 V	—			f_{MCK}/6 Note 2		f_{MCK}/6	bps
		Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 3	—			1.3		0.6	Mbps
		1.6 V ≤ V_{DD} ≤ 3.6 V	—		—			f_{MCK}/6	bps
		Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 3	—		—			0.6	Mbps

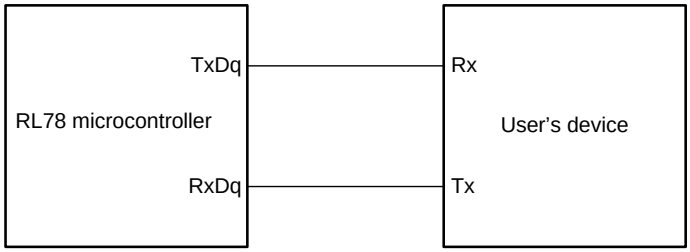
Note 1. Transfer rate in the SNOOZE mode is 4800 bps only.

Note 2. The following conditions are required for low voltage interface.
2.4 V ≤ V_{DD} < 2.7 V: MAX. 2.6 Mbps
1.8 V ≤ V_{DD} < 2.4 V: MAX. 1.3 Mbps
1.6 V ≤ V_{DD} < 1.8 V: MAX. 0.6 Mbps

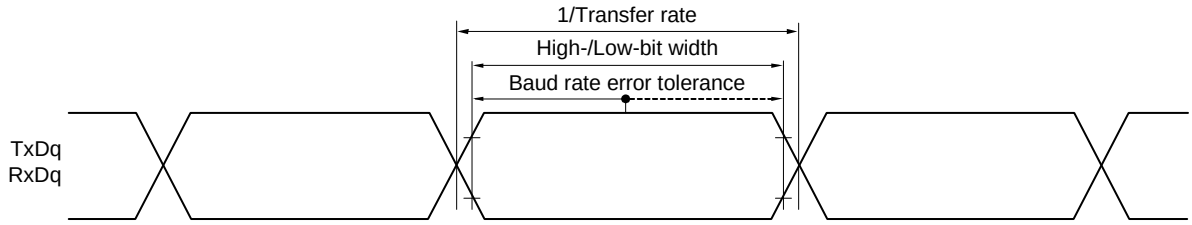
Note 3. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
HS (high-speed main) mode: 24 MHz (2.7 V ≤ V_{DD} ≤ 3.6 V)
16 MHz (2.4 V ≤ V_{DD} ≤ 3.6 V)
LS (low-speed main) mode: 8 MHz (1.8 V ≤ V_{DD} ≤ 3.6 V)
LV (low-voltage main) mode: 4 MHz (1.6 V ≤ V_{DD} ≤ 3.6 V)

Caution Select the normal input buffer for the RxDq pin and the normal output mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg).

UART mode connection diagram (during communication at same potential)



UART mode bit width (during communication at same potential) (reference)



Remark 1. q: UART number (q = 0 to 3), g: PIM and POM number (g = 0 to 3)

Remark 2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00 to 03, 10 to 13))

(2) During communication at same potential (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output, corresponding CSI00 only)

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq f_{CLK}/2$ 2.7 V ≤ V_{DD} ≤ 3.6 V	167		250		500		ns
SCKp high-/low-level width	t_{KL1}	2.7 V ≤ V_{DD} ≤ 3.6 V	$t_{KCY1}/2$ - 10		$t_{KCY1}/2$ - 50		$t_{KCY1}/2$ - 50		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 3.6 V	33		110		110		ns
SIp hold time (from SCKp↑) Note 2	t_{KSI1}	2.7 V ≤ V_{DD} ≤ 3.6 V	10		10		10		ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO1}	C = 20 pF Note 4		10		10		10	ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes “to SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes “from SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes “from SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 4. C is the load capacitance of the SCKp and SOp output lines.

Caution Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remark 1. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM number (g = 2)

Remark 2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00))

(3) During communication at same potential (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +85°C, 1.6 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq f_{CLK}/4$	2.7 V ≤ V_{DD} ≤ 3.6 V	167		500		1000		ns
			2.4 V ≤ V_{DD} ≤ 3.6 V	250		500		1000		ns
			1.8 V ≤ V_{DD} ≤ 3.6 V	—		500		1000		ns
			1.6 V ≤ V_{DD} ≤ 3.6 V	—		—		1000		ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	2.7 V ≤ V_{DD} ≤ 3.6 V		t_{KCY1}/2 - 18		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		ns
		2.4 V ≤ V_{DD} ≤ 3.6 V		t_{KCY1}/2 - 38		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V		—		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		ns
		1.6 V ≤ V_{DD} ≤ 3.6 V		—		—		t_{KCY1}/2 - 100		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 3.6 V		44		110		110		ns
		2.4 V ≤ V_{DD} ≤ 3.6 V		75		110		110		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V		—		110		110		ns
		1.6 V ≤ V_{DD} ≤ 3.6 V		—		—		220		ns
SIp hold time (from SCKp↑) Note 2	t_{KSI1}	2.4 V ≤ V_{DD} ≤ 3.6 V		19		19		19		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V		—		19		19		ns
		1.6 V ≤ V_{DD} ≤ 3.6 V		—		—		19		ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO1}	C = 30 pF Note 4	2.7 V ≤ V_{DD} ≤ 3.6 V		25		50		50	ns
			2.4 V ≤ V_{DD} ≤ 3.6 V		25		50		50	ns
			1.8 V ≤ V_{DD} ≤ 3.6 V		—		50		50	ns
			1.6 V ≤ V_{DD} ≤ 3.6 V		—		—		50	ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 4. C is the load capacitance of the SCKp and SOp output lines.

Caution Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remark 1. p: CSI number (p = 00, 10, 20, 30), m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), g: PIM number (g = 0 to 3)

Remark 2. f_{MCK}: Serial array unit operation clock frequency (Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00 to 03, 10 to 13))

(4) During communication at same potential (Simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(T_A = -40 to +85°C, 1.6 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp cycle time Note 5	t_{KCY2}	2.7 V ≤ V_{DD} < 3.6 V	f_{MCK} > 16 MHz	8/f_{MCK}		—		—		ns
			f_{MCK} ≤ 16 MHz	6/f_{MCK}		6/f_{MCK}		6/f_{MCK}		ns
		2.4 V ≤ V_{DD} < 3.6 V		6/f_{MCK} and 500		6/f_{MCK} and 500		6/f_{MCK} and 500		ns
		1.8 V ≤ V_{DD} < 3.6 V		—		6/f_{MCK} and 750		6/f_{MCK} and 750		ns
		1.6 V ≤ V_{DD} < 3.6 V		—		—		6/f_{MCK} and 1500		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	2.7 V ≤ V_{DD} ≤ 3.6 V		t_{KCY2}/2 - 8		t_{KCY2}/2 - 8		t_{KCY2}/2 - 8		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V		—		t_{KCY2}/2 - 18		t_{KCY2}/2 - 18		ns
		1.6 V ≤ V_{DD} ≤ 3.6 V		—		—		t_{KCY1}/2 - 66		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK2}	2.7 V ≤ V_{DD} ≤ 3.6 V		1/f_{MCK} + 20		1/f_{MCK} + 30		1/f_{MCK} + 30		ns
		2.4 V ≤ V_{DD} ≤ 3.6 V		1/f_{MCK} + 30		1/f_{MCK} + 30		1/f_{MCK} + 30		ns
		1.8 V ≤ V_{DD} < 3.6 V		—		1/f_{MCK} + 30		1/f_{MCK} + 30		ns
		1.6 V ≤ V_{DD} < 3.6 V		—		—		1/f_{MCK} + 40		ns
SIp hold time (from SCKp↑) Note 2	t_{KSI2}	2.4 V ≤ V_{DD} < 3.6 V		1/f_{MCK} + 31		1/f_{MCK} + 31		1/f_{MCK} + 31		ns
		1.8 V ≤ V_{DD} < 3.6 V		—		1/f_{MCK} + 31		1/f_{MCK} + 31		ns
		1.6 V ≤ V_{DD} < 3.6 V		—		—		1/f_{MCK} + 250		ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO2}	C = 30 pF Note 4	2.7 V ≤ V_{DD} ≤ 3.6 V		2/f_{MCK} + 44		2/f_{MCK} + 110		2/f_{MCK} + 110	ns
			2.4 V ≤ V_{DD} < 3.6 V		2/f_{MCK} + 75		2/f_{MCK} + 110		2/f_{MCK} + 110	ns
			1.8 V ≤ V_{DD} < 3.6 V		—		2/f_{MCK} + 110		2/f_{MCK} + 110	ns
			1.6 V ≤ V_{DD} < 3.6 V		—		—		2/f_{MCK} + 220	ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 4. C is the load capacitance of the SOp output lines.

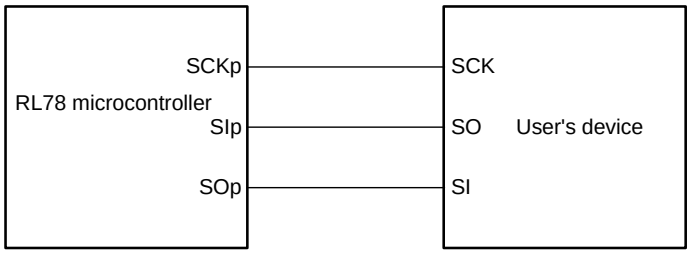
Note 5. The maximum transfer rate when using the SNOOZE mode is 1 Mbps.

Caution Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remark 1. p: CSI number (p = 00, 10, 20, 30), m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), g: PIM number (g = 0 to 3)

Remark 2. f_{MCK}: Serial array unit operation clock frequency (Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00 to 03, 10 to 13))

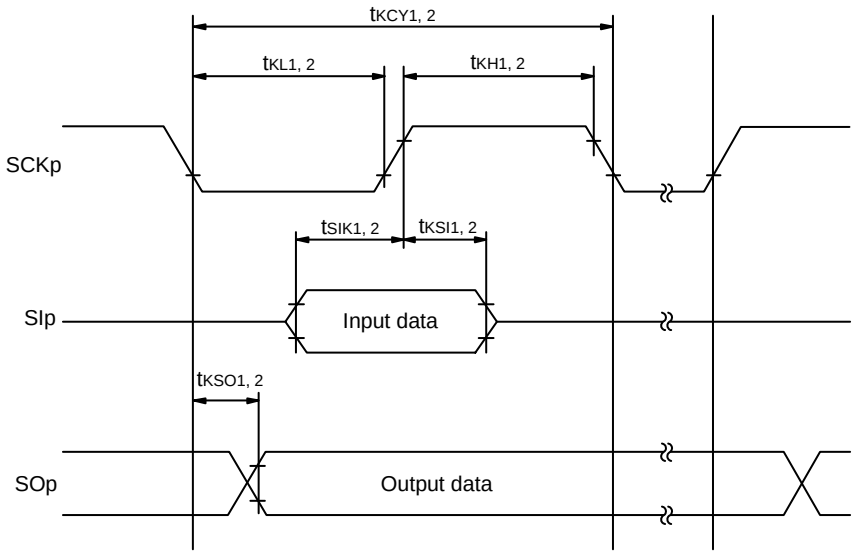
Simplified SPI (CSI) mode connection diagram (during communication at same potential)



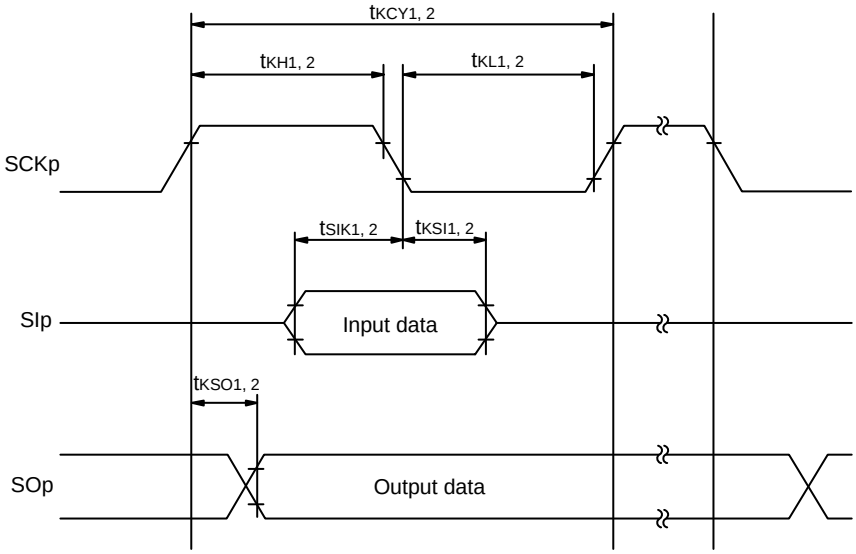
Remark 1. p: CSI number (p = 00, 10, 20, 30)

Remark 2. m: Unit number, n: Channel number (mn = 00 to 03, 10 to 13)

Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark 1. p: CSI number (p = 00, 10, 20, 30)
Remark 2. m: Unit number, n: Channel number (mn = 00 to 03, 10 to 13)

(5) During communication at same potential (simplified I^2C mode)

(T_A = -40 to +85°C, 1.6 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

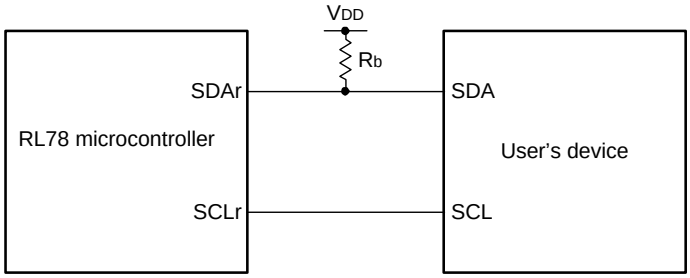
Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLr clock frequency	f_{SCL}	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 50 pF, R_b = 2.7 kΩ		1000 Note 1		400 Note 1		400 Note 1	kHz
		1.8 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ		400 Note 1		400 Note 1		400 Note 1	kHz
		1.8 V ≤ V_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ		300 Note 1		300 Note 1		300 Note 1	kHz
		1.6 V ≤ V_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ		—		—		250	kHz
Hold time when SCLr = “L”	t_{LOW}	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 50 pF, R_b = 2.7 kΩ	475		1150		1150		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ	1150		1150		1150		ns
		1.8 V ≤ V_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	1550		1550		1550		ns
		1.6 V ≤ V_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	—		—		1850		ns
Hold time when SCLr = “H”	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 50 pF, R_b = 2.7 kΩ	475		1150		1150		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ	1150		1150		1150		ns
		1.8 V ≤ V_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	1550		1550		1550		ns
		1.6 V ≤ V_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	—		—		1850		ns
Data setup time (reception)	$t_{SU: DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 50 pF, R_b = 2.7 kΩ	1/f_{MCK} + 85 Note 2		1/f_{MCK} + 145 Note 2		1/f_{MCK} + 145 Note 2		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ	1/f_{MCK} + 145 Note 2		1/f_{MCK} + 145 Note 2		1/f_{MCK} + 145 Note 2		ns
		1.8 V ≤ V_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	1/f_{MCK} + 230 Note 2		1/f_{MCK} + 230 Note 2		1/f_{MCK} + 230 Note 2		ns
		1.6 V ≤ V_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	—		—		1/f_{MCK} + 290 Note 2		ns
Data hold time (transmission)	$t_{HD: DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 50 pF, R_b = 2.7 kΩ	0	305	0	305	0	305	ns
		1.8 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ	0	355	0	355	0	355	ns
		1.8 V ≤ V_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	0	405	0	405	0	405	ns
		1.6 V ≤ V_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	—		—		0	405	ns

Note 1. The value must be equal to or less than f_{MCK}/4.

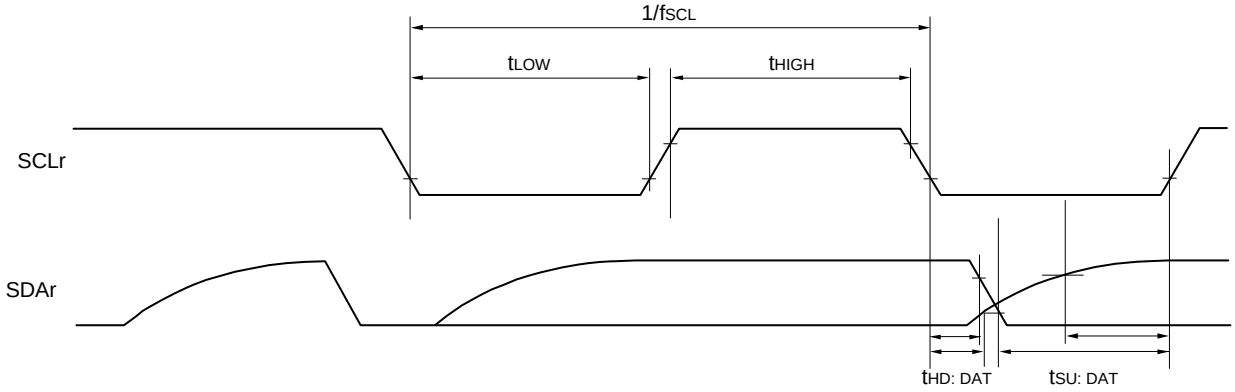
Note 2. Set the f_{MCK} value to keep the hold time of SCLr = “L” and SCLr = “H”.

Caution Select the normal input buffer and the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin and the normal output mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register h (POMh).

Simplified I²C mode connection diagram (during communication at same potential)



Simplified I²C mode serial transfer timing (during communication at same potential)



Remark 1. R_b[Ω]: Communication line (SDAr) pull-up resistance, C_b[F]: Communication line (SDAr, SCLr) load capacitance

Remark 2. r: IIC number (r = 00, 10, 20, 30), g: PIM number (g = 0 to 3),
h: POM number (h = 0 to 3)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number (m = 0, 1),
n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13)

(6) Communication at different potential (1.8 V, 2.5 V) (UART mode)

(T_A = -40 to +85°C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) (1/2)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Transfer rate Notes 1, 2		reception	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V		f_{MCK}/6 Note 1		f_{MCK}/6 Note 1		f_{MCK}/6 Note 1	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 4		4.0		1.3		0.6	Mbps
			1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		f_{MCK}/6 Notes 1, 2, 3		f_{MCK}/6 Notes 1, 2, 3		f_{MCK}/6 Notes 1, 2, 3	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 4		4.0		1.3		0.6	Mbps

Note 1. Transfer rate in the SNOOZE mode is 4,800 bps only.

Note 2. Use it with $V_{DD} \geq V_b$.

Note 3. The following conditions are required for low voltage interface.

2.4 V ≤ V_{DD} < 2.7 V: MAX. 2.6 Mbps

1.8 V ≤ V_{DD} < 2.4 V: MAX. 1.3 Mbps

1.6 V ≤ V_{DD} < 1.8 V: MAX. 0.6 Mbps

Note 4. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:

HS (high-speed main) mode: 24 MHz (2.7 V ≤ V_{DD} ≤ 3.6 V)

16 MHz (2.4 V ≤ V_{DD} ≤ 3.6 V)

LS (low-speed main) mode: 8 MHz (1.8 V ≤ V_{DD} ≤ 3.6 V)

LV (low-voltage main) mode: 4 MHz (1.6 V ≤ V_{DD} ≤ 3.6 V)

Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Remark 1. V_b[V]: Communication line voltage

Remark 2. q: UART number (q = 0 to 3), g: PIM and POM number (g = 0 to 3)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00 to 03, 10 to 13))

(6) Communication at different potential (1.8 V, 2.5V) (UART mode)

(T_A = -40 to +85°C, 1.8 ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Transfer rate Note 2		transmission	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V		Note 1		Note 1		Note 1	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 2		1.2 Note 2		1.2 Note 2	Mbps
			1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		Notes 3, 4		Notes 3, 4		Notes 3, 4	bps
			Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 5.5 kΩ, V_b = 1.6 V		0.43 Note 5		0.43 Note 5		0.43 Note 5	Mbps

Note 1. The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 2.7 V ≤ V_{DD} < 3.6 V and 2.3 V ≤ V_b ≤ 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

Note 2. This value as an example is calculated when the conditions described in the “Conditions” column are met.
Refer to Note 1 above to calculate the maximum transfer rate under conditions of the customer.

Note 3. Use it with $V_{DD} \geq V_b$.

Note 4. The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 1.8 V ≤ V_{DD} < 3.3 V and 1.6 V ≤ V_b ≤ 2.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\} \times 3} \text{ [bps]}$$

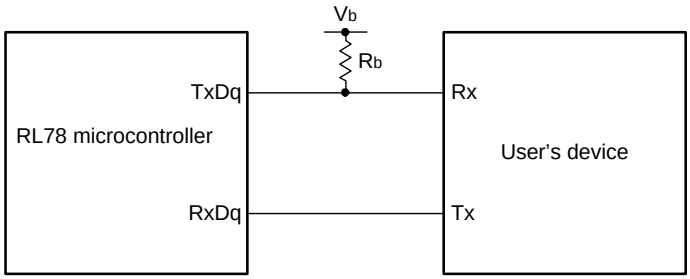
$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

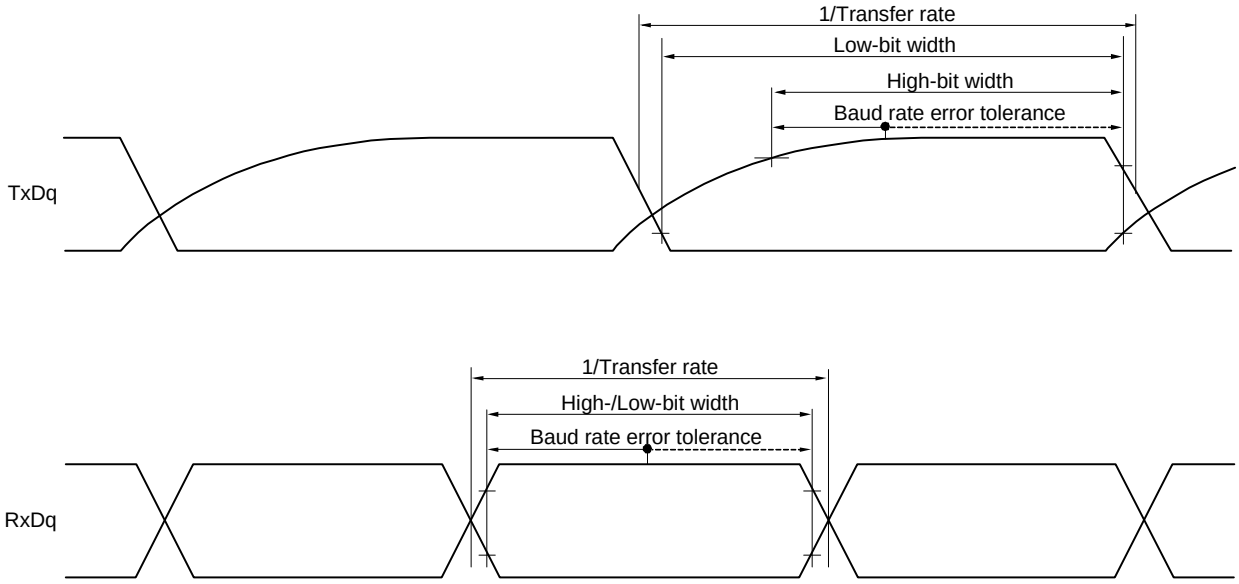
Note 5. This value as an example is calculated when the conditions described in the “Conditions” column are met.
Refer to Note 4 above to calculate the maximum transfer rate under conditions of the customer.

Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

UART mode connection diagram (during communication at different potential)



UART mode bit width (during communication at different potential) (reference)



Remark 1. R_b[Ω]: Communication line (TxDq) pull-up resistance, C_b[F]: Communication line (TxDq) load capacitance, V_b[V]: Communication line voltage

Remark 2. q: UART number (q = 0 to 3), g: PIM and POM number (g = 0 to 3)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00 to 03, 10 to 13))

(7) Communication at different potential (2.5 V) (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output, corresponding CSI00 only)

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq f_{CLK}/2$	2.7V ≤ V_{DD} < 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ	300		1150		1150		ns
SCKp high-level width	t_{KH1}	2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		t_{KCY1}/2 - 120		t_{KCY1}/2 - 120		t_{KCY1}/2 - 120		ns
SCKp low-level width	t_{KL1}	2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 1.4 kΩ		t_{KCY1}/2 - 10		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		121		479		479		ns
SIp hold time (from SCKp↑) Note 1	t_{KSI1}	2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 1.4 kΩ		10		10		10		ns
Delay time from SCKp↓ to SOp output Note 1	t_{KSO1}	2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 1.4 kΩ			130		130		130	ns
SIp setup time (to SCKp↓) Note 2	t_{SIK1}	2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		33		110		110		ns
SIp hold time (from SCKp↓) Note 2	t_{KSI1}	2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		10		10		10		ns
Delay time from SCKp↑ to SOp output Note 2	t_{KSO1}	2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ			10		10		10	ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

Note 2. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Remark 1. R_b[Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

Remark 2. p: CSI number (p = 00), m: Unit number (m = 0),
n: Channel number (n = 0), g: PIM and POM number (g = 2)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number,
n: Channel number (mn = 00))

(8) Communication at different potential (1.8 V, 2.5 V) (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +85°C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ f_{CLK}/4	2.7V ≤ V_{DD} < 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	500 Note		1150		1150		ns
			1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 1.8 V, C_b = 30 pF, R_b = 5.5 kΩ	1150 Note		1150		1150		ns
SCKp high-level width	t_{KH1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 - 170		t_{KCY1}/2 - 170		t_{KCY1}/2 - 170		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 - 458		t_{KCY1}/2 - 458		t_{KCY1}/2 - 458		ns
SCKp low-level width	t_{KL1}	2.7 V ≤ V_{DD} ≤ 3.6 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		t_{KCY1}/2 - 50		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		ns

Note Use it with V_{DD} ≥ V_b.

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** are listed on the page after the next page.)

(8) Communication at different potential (1.8 V, 2.5 V) (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +85°C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	177		479		479		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	479		479		479		ns
SIp hold time (from SCKp↑) Note 1	t_{KSI1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	19		19		19		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	19		19		19		ns
Delay time from SCKp↓ to SOp output Note 1	t_{KSO1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		195		195		195	ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ		483		483		483	ns
SIp setup time (to SCKp↓) Note 2	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	44		110		110		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	110		110		110		ns
SIp hold time (from SCKp↓) Note 2	t_{KSI1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	19		19		19		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	19		19		19		ns
Delay time from SCKp↑ to SOp output Note 2	t_{KSO1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		25		25		25	ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ		25		25		25	ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

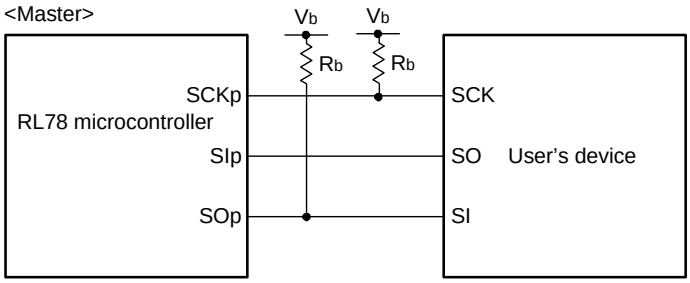
Note 2. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. Use it with V_{DD} ≥ V_b.

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** are listed on the next page.)

Simplified SPI (CSI) mode connection diagram (during communication at different potential)

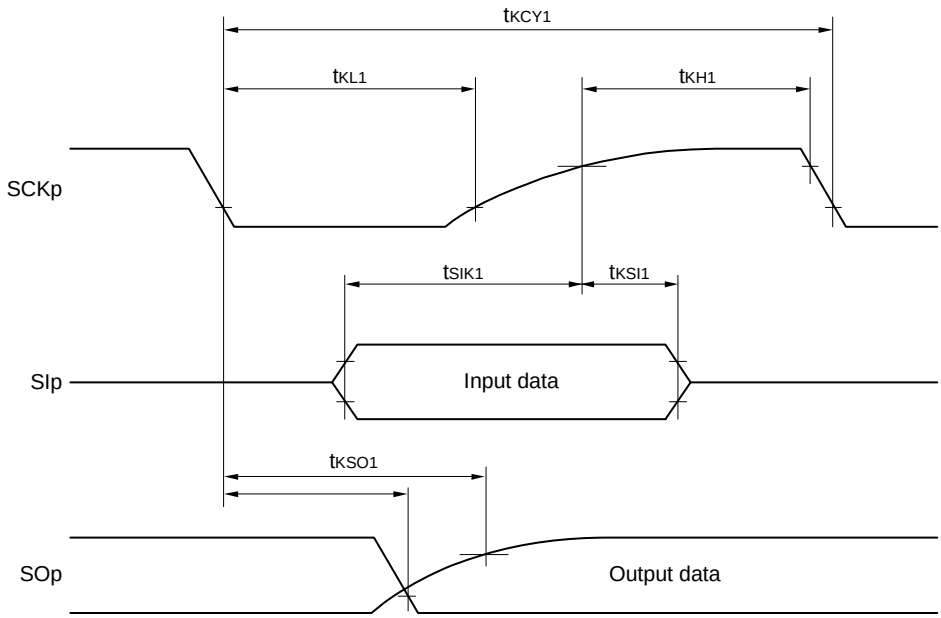


Remark 1. R_b[Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

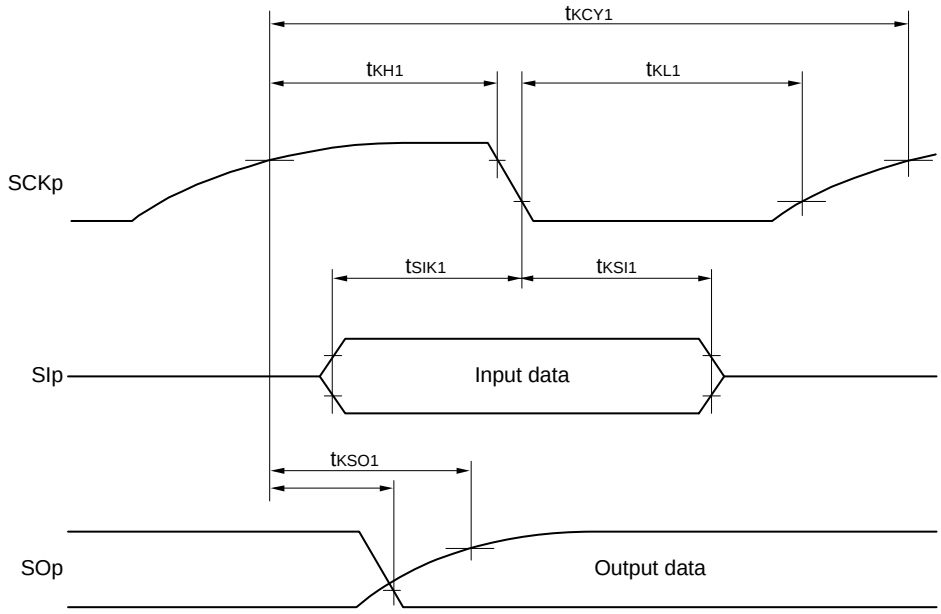
Remark 2. p: CSI number (p = 00, 10, 20, 30), m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), g: PIM and POM number (g = 0 to 3)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00))

Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark p: CSI number (p = 00, 10, 20, 30), m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3),
g: PIM and POM number (g = 0 to 3)

(9) Communication at different potential (1.8 V, 2.5 V) (Simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(T_A = -40 to +85°C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp cycle time Note 1	t_{KCY2}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V	20 MHz < f_{MCK} ≤ 24 MHz	16/f_{MCK}		—		—		ns
			16 MHz < f_{MCK} ≤ 20 MHz	14/f_{MCK}		—		—		ns
			8 MHz < f_{MCK} ≤ 16 MHz	12/f_{MCK}		—		—		ns
			4 MHz < f_{MCK} ≤ 8 MHz	8/f_{MCK}		16/f_{MCK}		—		ns
			f_{MCK} ≤ 4 MHz	6/f_{MCK}		10/f_{MCK}		10/f_{MCK}		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2	20 MHz < f_{MCK} ≤ 24 MHz	36/f_{MCK}		—		—		ns
			16 MHz < f_{MCK} ≤ 20 MHz	32/f_{MCK}		—		—		ns
			8 MHz < f_{MCK} ≤ 16 MHz	26/f_{MCK}		—		—		ns
			4 MHz < f_{MCK} ≤ 8 MHz	16/f_{MCK}		16/f_{MCK}		—		ns
			f_{MCK} ≤ 4 MHz	10/f_{MCK}		10/f_{MCK}		10/f_{MCK}		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V		t_{KCY2}/2 - 18		t_{KCY2}/2 - 50		t_{KCY2}/2 - 50		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2		t_{KCY2}/2 - 50		t_{KCY2}/2 - 50		t_{KCY2}/2 - 50		ns
SIp setup time (to SCKp↑) Note 3	t_{SIK2}	2.7 V ≤ V_{DD} ≤ 3.6 V		1/f_{MCK} + 20		1/f_{MCK} + 30		1/f_{MCK} + 30		ns
		1.8 V ≤ V_{DD} < 3.3 V		1/f_{MCK} + 30		1/f_{MCK} + 30		1/f_{MCK} + 30		ns
SIp hold time (from SCKp↑) Note 4	t_{KSI2}			1/f_{MCK} + 31		1/f_{MCK} + 31		1/f_{MCK} + 31		ns
Delay time from SCKp↓ to SOp output Note 5	t_{KSO2}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V C_b = 30 pF, R_b = 2.7 kΩ			2/f_{MCK} + 214		2/f_{MCK} + 573		2/f_{MCK} + 573	ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2 C_b = 30 pF, R_b = 5.5 kΩ			2/f_{MCK} + 573		2/f_{MCK} + 573		2/f_{MCK} + 573	ns

Note 1. Transfer rate in the SNOOZE mode: MAX. 1 Mbps

Note 2. Use it with V_{DD} ≥ V_b.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

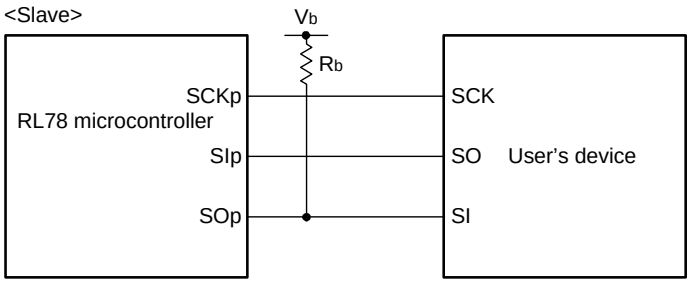
Note 4. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 5. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Caution Select the TTL input buffer for the SIp pin and SCKp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

(**Remarks** are listed on the next page.)

Simplified SPI (CSI) mode connection diagram (during communication at different potential)

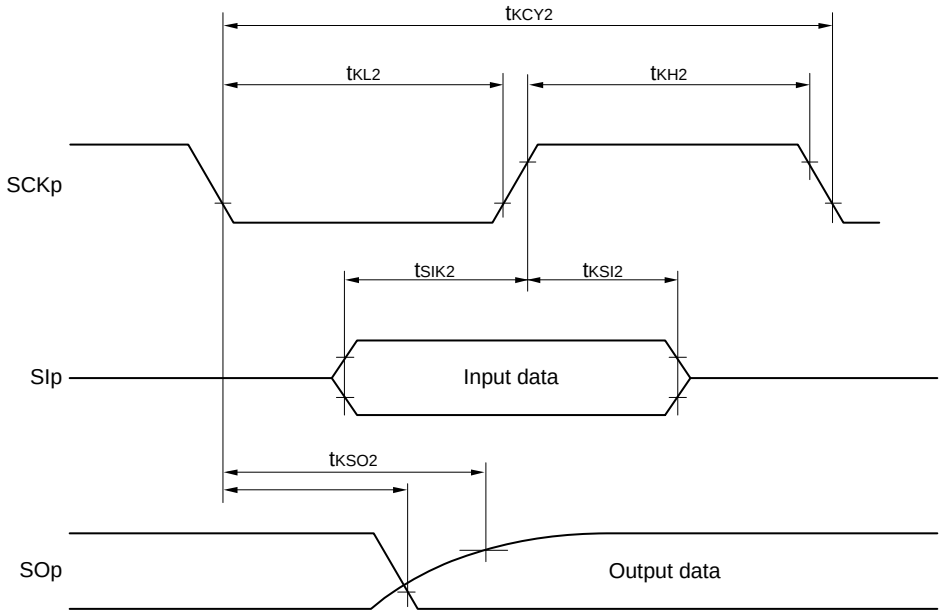


Remark 1. R_b[Ω]: Communication line (SOp) pull-up resistance, C_b[F]: Communication line (SOp) load capacitance, V_b[V]: Communication line voltage

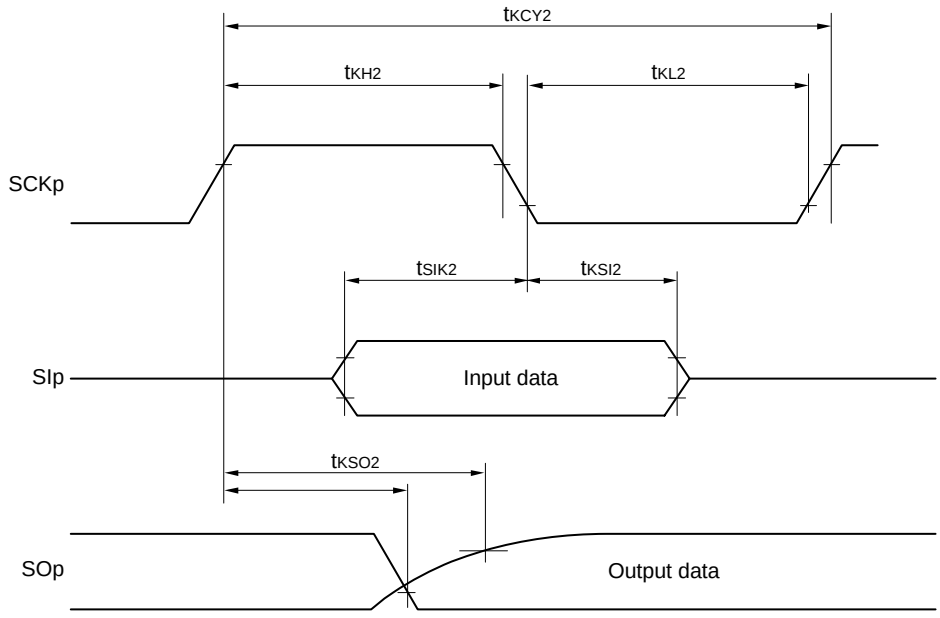
Remark 2. p: CSI number (p = 00, 10, 20, 30), m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), g: PIM and POM number (g = 0 to 3)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 02, 10, 12))

Simplified SPI (CSI) mode serial transfer timing (slave mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (slave mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark p: CSI number (p = 00, 10, 20, 30), m: Unit number (m = 0, 1),
n: Channel number (n = 0 to 3), g: PIM and POM number (g = 0 to 3)

(10) Communication at different potential (1.8 V, 2.5 V) (simplified I^2C mode)

(T_A = -40 to +85°C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLr clock frequency	f_{SCL}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ		1000 Note 1		300 Note 1		300 Note 1	kHz
		2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ		400 Note 1		300 Note 1		300 Note 1	kHz
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ		400 Note 1		300 Note 1		300 Note 1	kHz
Hold time when SCLr = “L”	t_{LOW}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	475		1550		1550		ns
		2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b <2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	1150		1550		1550		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ	1550		1550		1550		ns
Hold time when SCLr = “H”	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	200		610		610		ns
		2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	600		610		610		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ	610		610		610		ns
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	1/f_{MCK} + 135 Note 3		1/f_{MCK} + 190 Note 3		1/f_{MCK} + 190 Note 3		ns
		2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	1/f_{MCK} + 190 Note 3		1/f_{MCK} + 190 Note 3		1/f_{MCK} + 190 Note 3		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ	1/f_{MCK} + 190 Note 3		1/f_{MCK} + 190 Note 3		1/f_{MCK} + 190 Note 3		ns
Data hold time (transmission)	$t_{HD:DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	0	305	0	305	0	305	ns
		2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	0	355	0	355	0	355	ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ	0	405	0	405	0	405	ns

Note 1. The value must be equal to or less than f_{MCK}/4.

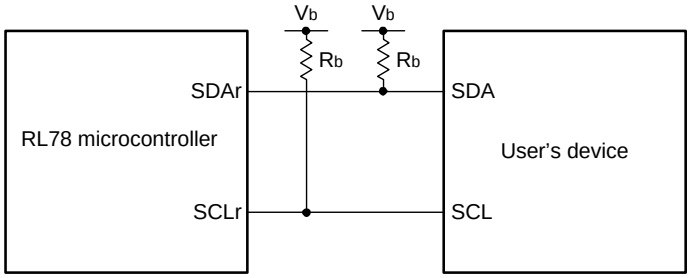
Note 2. Use it with V_{DD} ≥ V_b.

Note 3. Set the f_{MCK} value to keep the hold time of SCLr = “L” and SCLr = “H”.

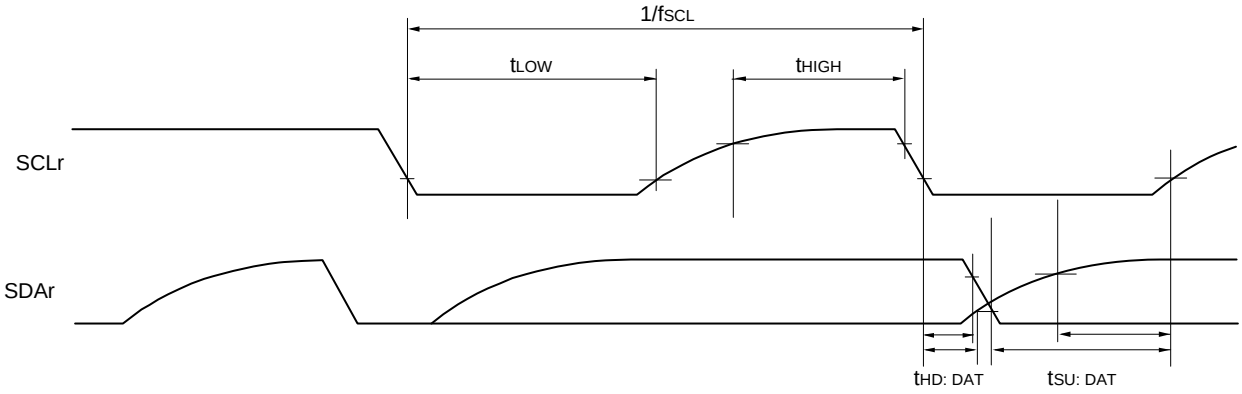
Caution Select the TTL input buffer and the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin and the N-ch open drain output (V_{DD} tolerance) mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

(**Remarks** are listed on the next page.)

Simplified I²C mode connection diagram (during communication at different potential)



Simplified I²C mode serial transfer timing (during communication at different potential)



Remark 1. R_b[Ω]: Communication line (SDAr, SCLr) pull-up resistance, C_b[F]: Communication line (SDAr, SCLr) load capacitance, V_b[V]: Communication line voltage

Remark 2. r: IIC number (r = 00, 10, 20, 30), g: PIM, POM number (g = 0 to 3)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12)

34.5.2 Serial interface IICA

(1) I^2C standard mode

(T_A = -40 to +85°C, 1.6 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Standard mode: f_{CLK} ≥ 1 MHz	2.7 V ≤ V_{DD} ≤ 3.6 V	0	100	0	100	0	100	kHz
			1.8 V ≤ V_{DD} ≤ 3.6 V	—	—	0	100	0	100	kHz
			1.6 V ≤ V_{DD} ≤ 3.6 V	—	—	—	—	0	100	kHz
Setup time of restart condition	$t_{SU: STA}$	2.7 V ≤ V_{DD} ≤ 3.6 V		4.7		4.7		4.7		µs
		1.8 V ≤ V_{DD} ≤ 3.6 V		—		4.7		4.7		µs
		1.6 V ≤ V_{DD} ≤ 3.6 V		—		—		4.7		µs
Hold time Note 1	$t_{HD: STA}$	2.7 V ≤ V_{DD} ≤ 3.6 V		4.0		4.0		4.0		µs
		1.8 V ≤ V_{DD} ≤ 3.6 V		—		4.0		4.0		µs
		1.6 V ≤ V_{DD} ≤ 3.6 V		—		—		4.0		µs
Hold time when SCLA0 = “L”	t_{LOW}	2.7 V ≤ V_{DD} ≤ 3.6 V		4.7		4.7		4.7		µs
		1.8 V ≤ V_{DD} ≤ 3.6 V		—		4.7		4.7		µs
		1.6 V ≤ V_{DD} ≤ 3.6 V		—		—		4.7		µs
Hold time when SCLA0 = “H”	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 3.6 V		4.0		4.0		4.0		µs
		1.8 V ≤ V_{DD} ≤ 3.6 V		—		4.0		4.0		µs
		1.6 V ≤ V_{DD} ≤ 3.6 V		—		—		4.0		µs
Data setup time (reception)	$t_{SU: DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V		250		250		250		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V		—		250		250		ns
		1.6 V ≤ V_{DD} ≤ 3.6 V		—		—		250		ns
Data hold time (transmission) Note 2	$t_{HD: DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V		0	3.45	0	3.45	0	3.45	µs
		1.8 V ≤ V_{DD} ≤ 3.6 V		—	—	0	3.45	0	3.45	µs
		1.6 V ≤ V_{DD} ≤ 3.6 V		—	—	—	—	0	3.45	µs
Setup time of stop condition	$t_{SU: STO}$	2.7 V ≤ V_{DD} ≤ 3.6 V		4.0		4.0		4.0		µs
		1.8 V ≤ V_{DD} ≤ 3.6 V		—		4.0		4.0		µs
		1.6 V ≤ V_{DD} ≤ 3.6 V		—		—		4.0		µs
Bus-free time	t_{BUF}	2.7 V ≤ V_{DD} ≤ 3.6 V		4.7		4.7		4.7		µs
		1.8 V ≤ V_{DD} ≤ 3.6 V		—		4.7		4.7		µs
		1.6 V ≤ V_{DD} ≤ 3.6 V		—		—		4.7		µs

Note 1. The first clock pulse is generated after this period when the start/restart condition is detected.

Note 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 C_b (communication line capacitance) and the value of R_b (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 +85°C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Fast mode: f_{CLK} ≥ 3.5 MHz	2.7 V ≤ V_{DD} ≤ 3.6 V	0	400	0	400	0	400	kHz
			1.8 V ≤ V_{DD} ≤ 3.6 V	0	400	0	400	0	400	kHz
Setup time of restart condition	$t_{SU: STA}$	2.7 V ≤ V_{DD} ≤ 3.6 V		0.6		0.6		0.6		μs
		1.8 V ≤ V_{DD} ≤ 3.6 V		0.6		0.6		0.6		μs
Hold time Note 1	$t_{HD: STA}$	2.7 V ≤ V_{DD} ≤ 3.6 V		0.6		0.6		0.6		μs
		1.8 V ≤ V_{DD} ≤ 3.6 V		0.6		0.6		0.6		μs
Hold time when SCLA0 = “L”	t_{LOW}	2.7 V ≤ V_{DD} ≤ 3.6 V		1.3		1.3		1.3		μs
		1.8 V ≤ V_{DD} ≤ 3.6 V		1.3		1.3		1.3		μs
Hold time when SCLA0 = “H”	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 3.6 V		0.6		0.6		0.6		μs
		1.8 V ≤ V_{DD} ≤ 3.6 V		0.6		0.6		0.6		μs
Data setup time (reception)	$t_{SU: DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V		100		100		100		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V		100		100		100		ns
Data hold time (transmission) Note 2	$t_{HD: DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V		0	0.9	0	0.9	0	0.9	μs
		1.8 V ≤ V_{DD} ≤ 3.6 V		0	0.9	0	0.9	0	0.9	μs
Setup time of stop condition	$t_{SU: STO}$	2.7 V ≤ V_{DD} ≤ 3.6 V		0.6		0.6		0.6		μs
		1.8 V ≤ V_{DD} ≤ 3.6 V		0.6		0.6		0.6		μs
Bus-free time	t_{BUF}	2.7 V ≤ V_{DD} ≤ 3.6 V		1.3		1.3		1.3		μs
		1.8 V ≤ V_{DD} ≤ 3.6 V		1.3		1.3		1.3		μs

Note 1. The first clock pulse is generated after this period when the start/restart condition is detected.

Note 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 C_b (communication line capacitance) and the value of R_b (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

(3) I^2C fast mode plus

(T_A = -40 to +85°C, 2.7 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

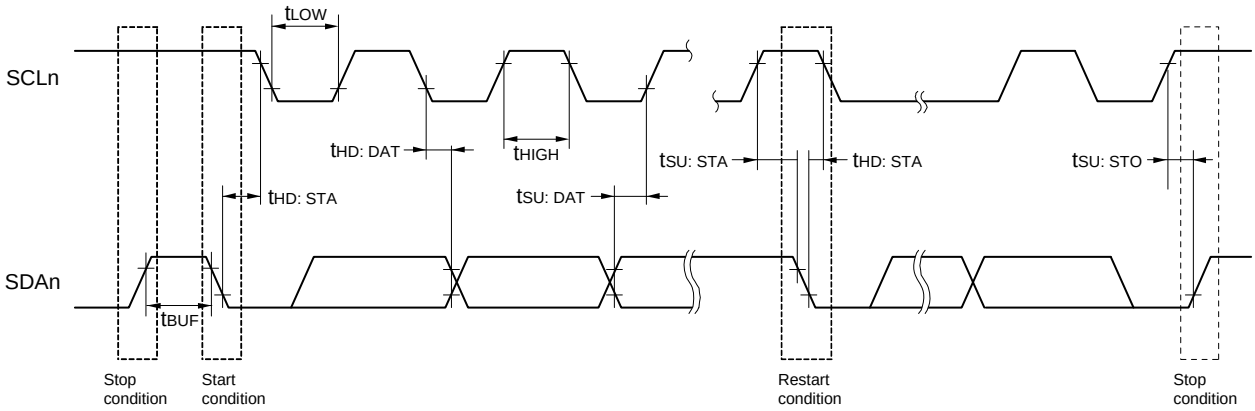
Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Fast mode plus: f_{CLK} ≥ 10 MHz	2.7 V ≤ V_{DD} ≤ 3.6 V	0	1000	—		—		kHz
Setup time of restart condition	$t_{SU: STA}$	2.7 V ≤ V_{DD} ≤ 3.6 V		0.26		—		—		μs
Hold time Note 1	$t_{HD: STA}$	2.7 V ≤ V_{DD} ≤ 3.6 V		0.26		—		—		μs
Hold time when SCLA0 = "L"	t_{LOW}	2.7 V ≤ V_{DD} ≤ 3.6 V		0.5		—		—		μs
Hold time when SCLA0 = "H"	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 3.6 V		0.26		—		—		μs
Data setup time (reception)	$t_{SU: DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V		50		—		—		ns
Data hold time (transmission) Note 2	$t_{HD: DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V		0	0.45	—		—		μs
Setup time of stop condition	$t_{SU: STO}$	2.7 V ≤ V_{DD} ≤ 3.6 V		0.26		—		—		μs
Bus-free time	t_{BUF}	2.7 V ≤ V_{DD} ≤ 3.6 V		0.5		—		—		μs

Note 1. The first clock pulse is generated after this period when the start/restart condition is detected.

Note 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 C_b (communication line capacitance) and the value of R_b (communication line pull-up resistor) at that time in each mode are as follows.
Fast mode plus: C_b = 120 pF, R_b = 1.1 kΩ

IICA serial transfer timing



34.5.3 USB

(1) Electrical specifications

(T_A = -40 to +85°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V, HS (High-speed main) mode only)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
U_{REGC}	U_{REGC} output voltage characteristic	U_{REGC}	UV_{BUS} = 4.0 to 5.5 V, PXXCON = VDDUSBE = 1	3.0	3.3	3.6	V
UV_{BUS}	UV_{BUS} input voltage characteristic	UV_{BUS}	Function	4.35 (4.02 Note)	5.00	5.25	V

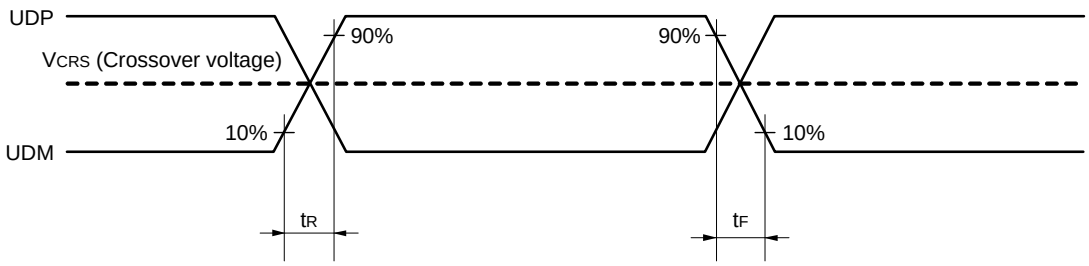
Note Value of instantaneous voltage

(T_A = -40 to +85°C, 4.35 V ≤ UV_{BUS} ≤ 5.25 V, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V, HS (High-speed main) mode only)

Parameter			Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input characteristic (FS/LS receiver)	Input voltage		V_{IH}		2.0			V
			V_{IL}				0.8	V
	Difference input sensitivity		V_{DI}	| UDP voltage - UDM voltage |	0.2			V
	Difference common mode range		V_{CM}		0.8		2.5	V
Output characteristic (FS driver)	Output voltage		V_{OH}	I_{OH} = -200 µA	2.8		3.6	V
			V_{OL}	I_{OL} = 2 mA	0		0.3	V
	Transition time	Rising	t_{FR}	Rising: From 10% to 90% of amplitude, Falling: From 90% to 10% of amplitude, CL = 50 pF	4		20	ns
		Falling	t_{FF}		4		20	ns
	Matching (TFR/TFF)		V_{FRFM}		90		111.1	%
	Crossover voltage		V_{FCRS}		1.3		2.0	V
	Output Impedance		Z_{DRV}		28		44	Ω
Output characteristic (LS driver)	Output voltage		V_{OH}		2.8		3.6	V
			V_{OL}		0		0.3	V
	Transition time	Rising	t_{LR}	Rising: From 10% to 90% of amplitude, Falling: From 90% to 10% of amplitude, CL = 250 pF to 750 pF The UDP and UDM pins are individually pulled down via 15 kΩ	75		300	ns
		Falling	t_{LF}		75		300	ns
	Matching (TFR/TFF) Note		V_{LTFM}		80		125	%
	Crossover voltage Note		V_{LCRS}		1.3		2.0	V
Pull-up, Pull-down	Pull-down resistor		R_{PD}		14.25		24.80	kΩ
	Pull-up resistor	Idle	R_{PUI}		0.9		1.575	kΩ
		Reception	R_{PUA}		1.425		3.09	kΩ
UV_{BUS}	UV_{BUS} pull-down resistor		R_{VBUS}	UV_{BUS} voltage = 5.5 V		1000		kΩ
	UV_{BUS} input voltage		V_{IH}		3.20			V
			V_{IL}				0.8	V

Note Excludes the first signal transition from the idle state.

Timing of UDP and UDM



(2) BC standard

(T_A = -40 to +85°C, 4.35 V ≤ UV_{BUS} ≤ 5.25 V, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V, HS (High-speed main) mode only)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
USB standard BC1.2	UDP sink current	I_{DP_SINK}		25	100	175	μA
	UDM sink current	I_{DM_SINK}		25	100	175	μA
	DCD source current	I_{DP_SRC}		7	10	13	μA
	Data detection voltage	V_{DAT_REF}		0.25	0.325	0.4	V
	UDP source voltage	V_{DP_SRC}	Output current 250 μA	0.5	0.6	0.7	V
	UDM source voltage	V_{DM_SRC}	Output current 250 μA	0.5	0.6	0.7	V

(3) BC option standard

(T_A = -40 to +85°C, 4.35 V ≤ UV_{BUS} ≤ 5.25 V, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V, HS (High-speed main) mode only)

Parameter			Symbol	Conditions	MIN.	TYP.	MAX.	Unit
UDP/UDM input reference voltage (UV_{BUS} divider ratio) (Function)	VDSELi [3: 0] (i = 0, 1)	0000	V_{DDET0}		27	32	37	%UVBUS
		0001	V_{DDET1}		29	34	39	%UVBUS
		0010	V_{DDET2}		32	37	42	%UVBUS
		0011	V_{DDET3}		35	40	45	%UVBUS
		0100	V_{DDET4}		38	43	48	%UVBUS
		0101	V_{DDET5}		41	46	51	%UVBUS
		0110	V_{DDET6}		44	49	54	%UVBUS
		0111	V_{DDET7}		47	52	57	%UVBUS
		1000	V_{DDET8}		51	56	61	%UVBUS
		1001	V_{DDET9}		55	60	65	%UVBUS
		1010	V_{DDET10}		59	64	69	%UVBUS
		1011	V_{DDET11}		63	68	73	%UVBUS
		1100	V_{DDET12}		67	72	73	%UVBUS
		1101	V_{DDET13}		71	76	81	%UVBUS
		1110	V_{DDET14}		75	80	85	%UVBUS
		1111	V_{DDET15}		79	84	89	%UVBUS

34.6 Analog Characteristics

34.6.1 A/D converter characteristics

Classification of A/D converter characteristics

Reference Voltage / Input Channel	Reference voltage (+) = AV_{REFP} Reference voltage (-) = AV_{REFM}	Reference voltage (+) = AV_{DD} Reference voltage (-) = AV_{SS}	Reference voltage (+) = Internal reference voltage Reference voltage (-) = AV_{SS}
High-accuracy channel; ANI0 to ANI6 (input buffer power supply: AV_{DD})	Refer to **34.6.1 (1)**. Refer to **34.6.1 (2)**.	Refer to **34.6.1 (3)**.	Refer to **34.6.1 (6)**.
Standard channel; ANI16 to ANI21 (input buffer power supply: V_{DD})	Refer to **34.6.1 (4)**.	Refer to **34.6.1 (5)**.	
Internal reference voltage, Temperature sensor output voltage	Refer to **34.6.1 (4)**.	Refer to **34.6.1 (5)**.	—

(1) When reference voltage (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (-) = AV_{REFM}/ANI1 (ADREFM = 1), target for conversion: ANI2 to ANI6

(T_A = -40 to +85°C, 2.4 V ≤ AV_{REFP} ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V, AV_{SS} = 0 V, reference voltage (+) = AV_{REFP}, reference voltage (-) = AV_{REFM} = 0 V, HALT mode)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES				12	bit
Overall error Notes 1, 2, 3	AINL	12-bit resolution		±1.7	±3.3	LSB
Conversion time	tCONV	ADTYP = 0, 12-bit resolution	3.375			μs
Zero-scale error Notes 1, 2, 3	EZS	12-bit resolution		±1.3	±3.2	LSB
Full-scale error Notes 1, 2, 3	EFS	12-bit resolution		±0.7	±2.9	LSB
Integral linearity error Notes 1, 2, 3	ILE	12-bit resolution		±1.0	±1.4	LSB
Differential linearity error Notes 1, 2, 3	DLE	12-bit resolution		±0.9	±1.2	LSB
Analog input voltage	VAIN		0		AV_{REFP}	V

Note 1. TYP. Value is the average value at AV_{DD} = AV_{REFP} = 3 V and T_A = 25°C. MAX. value is the average value ±3σ at normalized distribution.

Note 2. These values are the results of characteristic evaluation and are not checked for shipment.

Note 3. Excludes quantization error (±1/2 LSB).

Caution 1. Route the wiring so that noise will not be superimposed on each power line and ground line, and insert a capacitor to suppress noise.
In addition, separate the reference voltage line of AV_{REFP} from the other power lines to keep it free from the influences of noise.

Caution 2. During A/D conversion, keep a pulse, such as a digital signal, that abruptly changes its level from being input to or output from the pins adjacent to the converter pins and P150 to P156.

(2) When reference voltage (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (-) = AV_{REFM}/ANI1 (ADREFM = 1), conversion target: ANI2 to ANI6

(T_A = -40 to +85°C, 1.6 V ≤ AV_{REFP} ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V, AV_{SS} = 0 V, Reference voltage (+) = AV_{REFP}, Reference voltage (-) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES		2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V	8		12	bit
			1.8 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V	8		10 Note 1	
			1.6 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V	8 Note 2			
Overall error Note 3	AINL	12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±6.0	LSB
		10-bit resolution	1.8 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±5.0	
		8-bit resolution	1.6 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±2.5	
Conversion time	t_{CONV}	ADTYP = 0, 12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V	3.375			µs
		ADTYP = 0, 10-bit resolution Note 1	1.8 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V	6.75			
		ADTYP = 0, 8-bit resolution Note 2	1.6 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V	13.5			
		ADTYP = 1, 8-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V	2.5625			
			1.8 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V	5.125			
			1.6 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V	10.25			
Zero-scale error Note 3	E_{ZS}	12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±4.5	LSB
		10-bit resolution	1.8 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±4.5	
		8-bit resolution	1.6 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±2.0	
Full-scale error Note 3	E_{FS}	12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±4.5	LSB
		10-bit resolution	1.8 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±4.5	
		8-bit resolution	1.6 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±2.0	
Integral linearity error Note 3	ILE	12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±2.0	LSB
		10-bit resolution	1.8 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±1.5	
		8-bit resolution	1.6 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±1.0	
Differential linearity error Note 3	DLE	12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±1.5	LSB
		10-bit resolution	1.8 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±1.5	
		8-bit resolution	1.6 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±1.0	
Analog input voltage	V_{AIN}			0		AV_{REFP}	V

Note 1. Cannot be used for lower 2 bit of ADCR register

Note 2. Cannot be used for lower 4 bit of ADCR register

Note 3. Excludes quantization error (±1/2 LSB).

Caution Always use AV_{DD} pin with the same potential as the V_{DD} pin.

(3) When reference voltage (+) = AV_{DD} (ADREFP1 = 0, ADREFP0 = 0), reference voltage (-) = AV_{SS} (ADREFM = 0), conversion target: ANI0 to ANI6

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V, AV_{SS} = 0 V, Reference voltage (+) = AV_{DD}, Reference voltage (-) = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES		2.4 V ≤ AV_{DD} ≤ 3.6 V	8		12	bit
			1.8 V ≤ AV_{DD} ≤ 3.6 V	8		10 Note 1	
			1.6 V ≤ AV_{DD} ≤ 3.6 V	8 Note 2			
Overall error Note 3	AINL	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±7.5	LSB
		10-bit resolution	1.8 V ≤ AV_{DD} ≤ 3.6 V			±5.5	
		8-bit resolution	1.6 V ≤ AV_{DD} ≤ 3.6 V			±3.0	
Conversion time	t_{CONV}	ADTYP = 0, 12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V	3.375			µs
		ADTYP = 0, 10-bit resolution Note 1	1.8 V ≤ AV_{DD} ≤ 3.6 V	6.75			
		ADTYP = 0, 8-bit resolution Note 2	1.6 V ≤ AV_{DD} ≤ 3.6 V	13.5			
		ADTYP = 1, 8-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V	2.5625			
			1.8 V ≤ AV_{DD} ≤ 3.6 V	5.125			
			1.6 V ≤ AV_{DD} ≤ 3.6 V	10.25			
Zero-scale error Note 3	E_{ZS}	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±6.0	LSB
		10-bit resolution	1.8 V ≤ AV_{DD} ≤ 3.6 V			±5.0	
		8-bit resolution	1.6 V ≤ AV_{DD} ≤ 3.6 V			±2.5	
Full-scale error Note 3	E_{FS}	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±6.0	LSB
		10-bit resolution	1.8 V ≤ AV_{DD} ≤ 3.6 V			±5.0	
		8-bit resolution	1.6 V ≤ AV_{DD} ≤ 3.6 V			±2.5	
Integral linearity error Note 3	ILE	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±3.0	LSB
		10-bit resolution	1.8 V ≤ AV_{DD} ≤ 3.6 V			±2.0	
		8-bit resolution	1.6 V ≤ AV_{DD} ≤ 3.6 V			±1.5	
Differential linearity error Note 3	DLE	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±2.0	LSB
		10-bit resolution	1.8 V ≤ AV_{DD} ≤ 3.6 V			±2.0	
		8-bit resolution	1.6 V ≤ AV_{DD} ≤ 3.6 V			±1.5	
Analog input voltage	V_{AIN}	ANI0 to ANI6		0		AV_{DD}	V

Note 1. Cannot be used for lower 2 bit of ADCR register

Note 2. Cannot be used for lower 4 bit of ADCR register

Note 3. Excludes quantization error (±1/2 LSB).

Caution Always use AV_{DD} pin with the same potential as the V_{DD} pin.

(4) When reference voltage (+) = AVREFP/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (-) = AVREFM/ANI1 (ADREFM = 1), conversion target: ANI16 to ANI21, internal reference voltage, temperature sensor output voltage

(T_A = -40 to +85°C, 1.6 V ≤ V_{DD} ≤ 3.6 V, 1.6 V ≤ AVREFP ≤ AVDD = V_{DD} ≤ 3.6 V, V_{SS} = 0 V, AVSS = 0 V, Reference voltage (+) = AVREFP, Reference voltage (-) = AVREFM = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES		2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	8		12	bit
			1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	8		10 Note 1	
			1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	8 Note 2			
Overall error Note 3	AINL	12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±7.0	LSB
		10-bit resolution	1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±5.5	
		8-bit resolution	1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±3.0	
Conversion time	tCONV	ADTYP = 0, 12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	4.125			µs
		ADTYP = 0, 10-bit resolution Note 1	1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	9.5			
		ADTYP = 0, 8-bit resolution Note 2	1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	57.5			
		ADTYP = 1, 8-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	3.3125			
			1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	7.875			
			1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	54.25			
Zero-scale error Note 3	EZS	12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±5.0	LSB
		10-bit resolution	1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±5.0	
		8-bit resolution	1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±2.5	
Full-scale error Note 3	EFS	12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±5.0	LSB
		10-bit resolution	1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±5.0	
		8-bit resolution	1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±2.5	
Integral linearity error Note 3	ILE	12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±3.0	LSB
		10-bit resolution	1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±2.0	
		8-bit resolution	1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±1.5	
Differential linearity error Note 3	DLE	12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±2.0	LSB
		10-bit resolution	1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±2.0	
		8-bit resolution	1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±1.5	
Analog input voltage	VAIN			0		AVREFP	V
		Internal reference voltage (2.4 V ≤ V_{DD} ≤ 3.6 V, HS (high-speed main) mode)		V_{BGR} Note 4			
		Temperature sensor output voltage (2.4 V ≤ V_{DD} ≤ 3.6 V, HS (high-speed main) mode)		V_{TMP25} Note 4			

Note 1. Cannot be used for lower 2 bits of ADCR register

Note 2. Cannot be used for lower 4 bits of ADCR register

Note 3. Excludes quantization error (±1/2 LSB).

Note 4. Refer to **34.6.2 Temperature sensor, internal reference voltage output characteristics**.

Caution Always use AVDD pin with the same potential as the V_{DD} pin.

(5) When reference voltage (+) = AV_{DD} (ADREFP1 = 0, ADREFP0 = 0), reference voltage (-) = AV_{SS} (ADREFM = 0), conversion target: ANI16 to ANI21, internal reference voltage, temperature sensor output voltage

(T_A = -40 to +85°C, 1.6 V ≤ V_{DD} ≤ 3.6 V, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V, AV_{SS} = 0 V, Reference voltage (+) = AV_{DD}, Reference voltage (-) = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES		2.4 V ≤ AV_{DD} ≤ 3.6 V	8		12	bit
			1.8 V ≤ AV_{DD} ≤ 3.6 V	8		10 Note 1	
			1.6 V ≤ AV_{DD} ≤ 3.6 V	8 Note 2			
Overall error Note 3	AINL	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±8.5	LSB
		10-bit resolution	1.8 V ≤ AV_{DD} ≤ 3.6 V			±6.0	
		8-bit resolution	1.6 V ≤ AV_{DD} ≤ 3.6 V			±3.5	
Conversion time	tCONV	ADTYP = 0, 12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V	4.125			µs
		ADTYP = 0, 10-bit resolution Note 1	1.8 V ≤ AV_{DD} ≤ 3.6 V	9.5			
		ADTYP = 0, 8-bit resolution Note 2	1.6 V ≤ AV_{DD} ≤ 3.6 V	57.5			
		ADTYP = 1, 8-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V	3.3125			
			1.8 V ≤ AV_{DD} ≤ 3.6 V	7.875			
			1.6 V ≤ AV_{DD} ≤ 3.6 V	54.25			
Zero-scale error Note 3	EZS	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±8.0	LSB
		10-bit resolution	1.8 V ≤ AV_{DD} ≤ 3.6 V			±5.5	
		8-bit resolution	1.6 V ≤ AV_{DD} ≤ 3.6 V			±3.0	
Full-scale error Note 3	EFS	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±8.0	LSB
		10-bit resolution	1.8 V ≤ AV_{DD} ≤ 3.6 V			±5.5	
		8-bit resolution	1.6 V ≤ AV_{DD} ≤ 3.6 V			±3.0	
Integral linearity error Note 3	ILE	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±3.5	LSB
		10-bit resolution	1.8 V ≤ AV_{DD} ≤ 3.6 V			±2.5	
		8-bit resolution	1.6 V ≤ AV_{DD} ≤ 3.6 V			±1.5	
Differential linearity error Note 3	DLE	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±2.5	LSB
		10-bit resolution	1.8 V ≤ AV_{DD} ≤ 3.6 V			±2.5	
		8-bit resolution	1.6 V ≤ AV_{DD} ≤ 3.6 V			±2.0	
Analog input voltage	VAIN			0		AV_{DD}	V
		Internal reference voltage (2.4 V ≤ V_{DD} ≤ 3.6 V, HS (high-speed main) mode)		V_{BGR} Note 4			
		Temperature sensor output voltage (2.4 V ≤ V_{DD} ≤ 3.6 V, HS (high-speed main) mode)		V_{TMP25} Note 4			

Note 1. Cannot be used for lower 2 bits of ADCR register

Note 2. Cannot be used for lower 4 bits of ADCR register

Note 3. Excludes quantization error (±1/2 LSB).

Note 4. Refer to **34.6.2 Temperature sensor, internal reference voltage output characteristics**.

Caution Always use AV_{DD} pin with the same potential as the V_{DD} pin.

(6) When reference voltage (+) = Internal reference voltage (1.45 V) (ADREFP1 = 1, ADREFP0 = 0), reference voltage (-) = AVSS (ADREFM = 0), conversion target: ANI0 to ANI6, ANI16 to ANI21

(T_A = -40 to +85°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, 1.6 V ≤ V_{DD}, 1.6 V ≤ AV_{DD} = V_{DD}, V_{SS} = 0 V, AV_{SS} = 0 V, Reference voltage (+) = internal reference voltage, Reference voltage (-) = AV_{SS} = 0 V, HS (high-speed main) mode)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES			8		bit
Conversion time	tCONV	8-bit resolution	16			µs
Zero-scale error Note	EZS	8-bit resolution			±4.0	LSB
Integral linearity error Note	ILE	8-bit resolution			±2.0	LSB
Differential linearity error Note	DLE	8-bit resolution			±2.5	LSB
Reference voltage (+)	$AV_{REF(+)}$	= Internal reference voltage (V_{BGR})	1.38	1.45	1.5	V
Analog input voltage	VAIN		0		V_{BGR}	V

Note Excludes quantization error (±1/2 LSB).

Caution Always use AV_{DD} pin with the same potential as the V_{DD} pin.

34.6.2 Temperature sensor, internal reference voltage output characteristics

(T_A = -40 to +85°C, 2.4 V ≤ V_{DD} ≤ 3.6 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	FVTMPS	Temperature sensor output voltage that depends on the temperature		-3.6		mV/°C
Operation stabilization wait time	tAMP		10			µs

34.6.3 D/A converter characteristics

(T_A = -40 to +85°C, 1.6 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					8	bit
Overall error	AINL	Rload = 4 MΩ	1.8 V ≤ V_{DD} ≤ 3.6 V			±2.5	LSB
		Rload = 8 MΩ	1.8 V ≤ V_{DD} ≤ 3.6 V			±2.5	LSB
Settling time	tSET	Cload = 20 pF	2.7 V ≤ V_{DD} ≤ 3.6 V			3	µs
			1.6 V ≤ V_{DD} < 2.7 V			6	µs

34.6.4 Comparator

(T_A = -40 to +85°C, 1.6 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage range	Ivref			0		V_{DD} - 1.4	V
	Ivcmp			-0.3		V_{DD} + 0.3	V
Output delay	td	V_{DD} = 3.0 V Input slew rate > 50 mV/μs	High-speed comparator mode, standard mode			1.2	μs
			High-speed comparator mode, window mode			2.0	μs
			Low-speed comparator mode, standard mode		3	5.0	μs
High-electric-potential judgment voltage	VTW+	High-speed comparator mode, window mode			0.76 V_{DD}		V
Low-electric-potential judgment voltage	VTW-	High-speed comparator mode, window mode			0.24 V_{DD}		V
Operation stabilization wait time	t_{CMP}			100			μs
Internal reference voltage Note	V_{BGR}			1.38	1.45	1.50	V

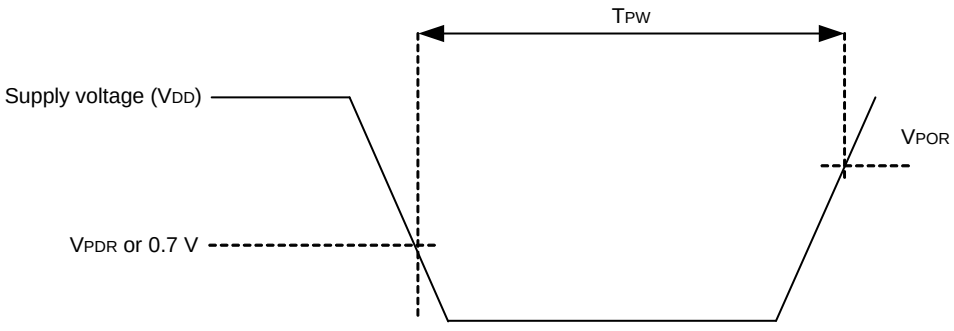
Note Not usable in LS (low-speed main) mode, LV (low-voltage main) mode, sub-clock operation, or STOP mode.

34.6.5 POR circuit characteristics

(T_A = -40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}	Power supply rise time	1.47	1.51	1.55	V
	V_{PDR}	Power supply fall time Note	1.46	1.50	1.54	V
Minimum pulse width	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).



34.6.6 LVD circuit characteristics

(T_A = -40 to +85°C, $V_{PDR} \leq V_{DD} \leq 3.6$ V $\leq V_{SS}$ = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVD2}	Power supply rise time	3.07	3.13	3.19	V
			Power supply fall time	3.00	3.06	3.12	V
		V_{LVD3}	Power supply rise time	2.96	3.02	3.08	V
			Power supply fall time	2.90	2.96	3.02	V
		V_{LVD4}	Power supply rise time	2.86	2.92	2.97	V
			Power supply fall time	2.80	2.86	2.91	V
		V_{LVD5}	Power supply rise time	2.76	2.81	2.87	V
			Power supply fall time	2.70	2.75	2.81	V
		V_{LVD6}	Power supply rise time	2.66	2.71	2.76	V
			Power supply fall time	2.60	2.65	2.70	V
		V_{LVD7}	Power supply rise time	2.56	2.61	2.66	V
			Power supply fall time	2.50	2.55	2.60	V
		V_{LVD8}	Power supply rise time	2.45	2.50	2.55	V
			Power supply fall time	2.40	2.45	2.50	V
		V_{LVD9}	Power supply rise time	2.05	2.09	2.13	V
			Power supply fall time	2.00	2.04	2.08	V
		V_{LVD10}	Power supply rise time	1.94	1.98	2.02	V
			Power supply fall time	1.90	1.94	1.98	V
		V_{LVD11}	Power supply rise time	1.84	1.88	1.91	V
			Power supply fall time	1.80	1.84	1.87	V
		V_{LVD12}	Power supply rise time	1.74	1.77	1.81	V
			Power supply fall time	1.70	1.73	1.77	V
		V_{LVD13}	Power supply rise time	1.64	1.67	1.70	V
			Power supply fall time	1.60	1.63	1.66	V
Minimum pulse width		t_{LW}		300			μs
Detection delay time						300	μs

Caution Set the detection voltage (V_{LVD}) to be within the operating voltage range. The operating voltage range depends on the setting of the user option byte (000C2H/010C2H). The following shows the operating voltage range.

HS (high-speed main) mode: V_{DD} = 2.7 to 3.6 V at 1 MHz to 24 MHz
V_{DD} = 2.4 to 3.6 V at 1 MHz to 16 MHz
LS (low-speed main) mode: V_{DD} = 1.8 to 3.6 V at 1 MHz to 8 MHz
LV (low-voltage main) mode: V_{DD} = 1.6 to 3.6 V at 1 MHz to 4 MHz

LVD Detection Voltage of Interrupt & Reset Mode

(T_A = -40 to +85°C, $V_{PDR} \leq V_{DD} \leq 3.6$ V, V_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Interrupt and reset mode	V_{LVDA0}	V_{POC0}, V_{POC1}, V_{POC2} = 0, 0, 0, falling reset voltage: 1.6 V			1.60	1.63	1.66	V
	V_{LVDA1}	LVIS0, LVIS1 = 1, 0		Rising release reset voltage	1.74	1.77	1.81	V
				Falling interrupt voltage	1.70	1.73	1.77	V
	V_{LVDA2}	LVIS0, LVIS1 = 0, 1		Rising release reset voltage	1.84	1.88	1.91	V
				Falling interrupt voltage	1.80	1.84	1.87	V
	V_{LVDA3}	LVIS0, LVIS1 = 0, 0		Rising release reset voltage	2.86	2.92	2.97	V
				Falling interrupt voltage	2.80	2.86	2.91	V
	V_{LVDB0}	V_{POC0}, V_{POC1}, V_{POC2} = 0, 0, 1, falling reset voltage: 1.8 V			1.80	1.84	1.87	V
	V_{LVDB1}	LVIS0, LVIS1 = 1, 0		Rising release reset voltage	1.94	1.98	2.02	V
				Falling interrupt voltage	1.90	1.94	1.98	V
	V_{LVDB2}	LVIS0, LVIS1 = 0, 1		Rising release reset voltage	2.05	2.09	2.13	V
				Falling interrupt voltage	2.00	2.04	2.08	V
	V_{LVDB3}	LVIS0, LVIS1 = 0, 0		Rising release reset voltage	3.07	3.13	3.19	V
				Falling interrupt voltage	3.00	3.06	3.12	V
	V_{LVDC0}	V_{POC0}, V_{POC1}, V_{POC2} = 0, 1, 0, falling reset voltage: 2.4 V			2.40	2.45	2.50	V
	V_{LVDC1}	LVIS0, LVIS1 = 1, 0		Rising release reset voltage	2.56	2.61	2.66	V
				Falling interrupt voltage	2.50	2.55	2.60	V
	V_{LVDC2}	LVIS0, LVIS1 = 0, 1		Rising release reset voltage	2.66	2.71	2.76	V
				Falling interrupt voltage	2.60	2.65	2.70	V
	V_{LVDD0}	V_{POC0}, V_{POC1}, V_{POC2} = 0, 1, 1, falling reset voltage: 2.7 V			2.70	2.75	2.81	V
	V_{LVDD1}	LVIS0, LVIS1 = 1, 0		Rising release reset voltage	2.86	2.92	2.97	V
				Falling interrupt voltage	2.80	2.86	2.91	V
	V_{LVDD2}	LVIS0, LVIS1 = 0, 1		Rising release reset voltage	2.96	3.02	3.08	V
				Falling interrupt voltage	2.90	2.96	3.02	V

34.7 Power supply voltage rising slope characteristics

(T_A = -40 to +85°C, V_{SS} = 0 V)

Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Power supply voltage rising slope	SV_{DD}			54	V/ms

Caution Make sure to keep the internal reset state by the LVD circuit or an external reset until V_{DD} reaches the operating voltage range shown in 34.4 AC Characteristics.

34.8 LCD Characteristics

34.8.1 Resistance division method

(1) Static display mode

(T_A = -40 to +85°C, V_{L4} (MIN.) ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	V_{L4}		2.0		V_{DD}	V

(2) 1/2 bias method, 1/4 bias method

(T_A = -40 to +85°C, V_{L4} (MIN.) ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	V_{L4}		2.7		V_{DD}	V

(3) 1/3 bias method

(T_A = -40 to +85°C, V_{L4} (MIN.) ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	V_{L4}		2.5		V_{DD}	V

34.8.2 Internal voltage boosting method

(1) 1/3 bias method

(T_A = -40 to +85°C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
LCD output voltage variation range	V_{L1}	C1 to C4 Note 1 = 0.47 μF Note 2	VLCD = 04H	0.90	1.00	1.08	V
			VLCD = 05H	0.95	1.05	1.13	V
			VLCD = 06H	1.00	1.10	1.18	V
			VLCD = 07H	1.05	1.15	1.23	V
			VLCD = 08H	1.10	1.20	1.28	V
			VLCD = 09H	1.15	1.25	1.33	V
			VLCD = 0AH	1.20	1.30	1.38	V
			VLCD = 0BH	1.25	1.35	1.43	V
			VLCD = 0CH	1.30	1.40	1.48	V
			VLCD = 0DH	1.35	1.45	1.53	V
			VLCD = 0EH	1.40	1.50	1.58	V
			VLCD = 0FH	1.45	1.55	1.63	V
			VLCD = 10H	1.50	1.60	1.68	V
			VLCD = 11H	1.55	1.65	1.73	V
			VLCD = 12H	1.60	1.70	1.78	V
			VLCD = 13H	1.65	1.75	1.83	V
Doubler output voltage	V_{L2}	C1 to C4 Note 1 = 0.47 μF		2 V_{L1} - 0.1	2 V_{L1}	2 V_{L1}	V
Tripler output voltage	V_{L3}	C1 to C4 Note 1 = 0.47 μF		3 V_{L1} - 0.15	3 V_{L1}	3 V_{L1}	V
Reference voltage setup time Note 2	t_{VWAIT1}			5			ms
Voltage boost wait time Note 3	t_{VWAIT2}	C1 to C4 Note 1 = 0.47μF		500			ms

Note 1. This is a capacitor that is connected between voltage pins used to drive the LCD.
C1: A capacitor connected between CAPH and CAPL
C2: A capacitor connected between V_{L1} and GND
C3: A capacitor connected between V_{L2} and GND
C4: A capacitor connected between V_{L4} and GND
C1 = C2 = C3 = C4 = 0.47 μF±30%

Note 2. This is the time required to wait from when the reference voltage is specified by using the VLCD register (or when the internal voltage boosting method is selected (by setting the MDSET1 and MDSET0 bits of the LCDM0 register to 01B) if the default value reference voltage is used) until voltage boosting starts (VLCON = 1).

Note 3. This is the wait time from when voltage boosting is started (VLCON = 1) until display is enabled (LCDON = 1).

(2) 1/4 bias method

(T_A = -40 to +85°C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
LCD output voltage variation range	V_{L1}	C1 to C4 Note 1 = 0.47 μF Note 2	VLCD = 04H	0.90	1.00	1.08	V
			VLCD = 05H	0.95	1.05	1.13	V
			VLCD = 06H	1.00	1.10	1.18	V
			VLCD = 07H	1.05	1.15	1.23	V
			VLCD = 08H	1.10	1.20	1.28	V
			VLCD = 09H	1.15	1.25	1.33	V
			VLCD = 0AH	1.20	1.30	1.38	V
Doubler output voltage	V_{L2}	C1 to C4 Note 1 = 0.47 μF		2 V_{L1} - 0.08	2 V_{L1}	2 V_{L1}	V
Tripler output voltage	V_{L3}	C1 to C4 Note 1 = 0.47 μF		3 V_{L1} - 0.12	3 V_{L1}	3 V_{L1}	V
Quadruply output voltage	V_{L4}	C1 to C5 Note 1 = 0.47 μF		4 V_{L1} - 0.16	4 V_{L1}	4 V_{L1}	V
Reference voltage setup time Note 2	t_{VWAIT1}			5			ms
Voltage boost wait time Note 3	t_{VWAIT2}	C1 to C5 Note 1 = 0.47μF		500			ms

Note 1. This is a capacitor that is connected between voltage pins used to drive the LCD.
C1: A capacitor connected between CAPH and CAPL
C2: A capacitor connected between V_{L1} and GND
C3: A capacitor connected between V_{L2} and GND
C4: A capacitor connected between V_{L3} and GND
C5: A capacitor connected between V_{L4} and GND
C1 = C2 = C3 = C4 = 0.47 μF±30%

Note 2. This is the time required to wait from when the reference voltage is specified by using the VLCD register (or when the internal voltage boosting method is selected (by setting the MDSET1 and MDSET0 bits of the LCDM0 register to 01B) if the default value reference voltage is used) until voltage boosting starts (VLCON = 1).

Note 3. This is the wait time from when voltage boosting is started (VLCON = 1) until display is enabled (LCDON = 1).

34.8.3 Capacitor split method

(1) 1/3 bias method

(T_A = -40 to +85°C, 2.2 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{L4} voltage	V_{L4}	C1 to C4 = 0.47 µF Note 2		V_{DD}		V
V_{L2} voltage	V_{L2}	C1 to C4 = 0.47 µF Note 2	2/3 V_{L4} - 0.1	2/3 V_{L4}	2/3 V_{L4} + 0.1	V
V_{L1} voltage	V_{L1}	C1 to C4 = 0.47 µF Note 2	1/3 V_{L4} - 0.1	1/3 V_{L4}	1/3 V_{L4} + 0.1	V
Capacitor split wait time Note 1	tVWAIT		100			ms

Note 1. This is the wait time from when voltage bucking is started (VLCON = 1) until display is enabled (LCDON = 1).

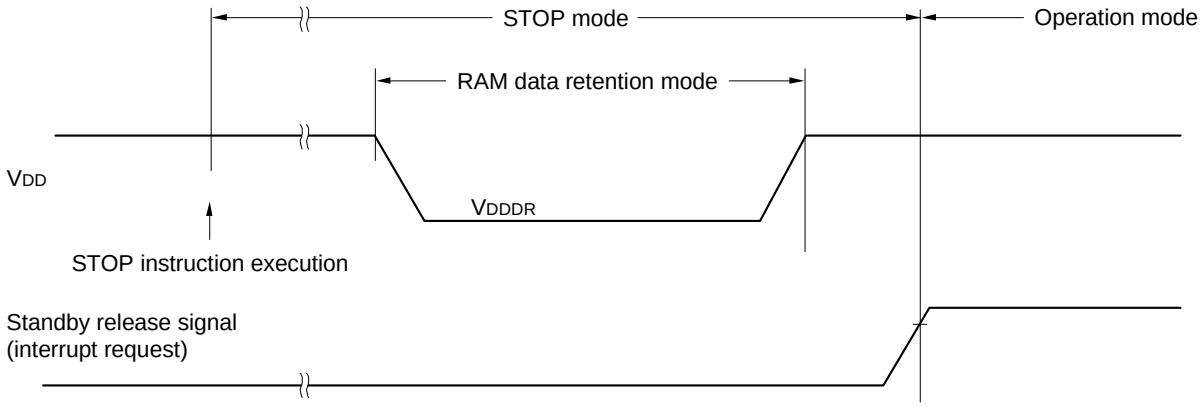
Note 2. This is a capacitor that is connected between voltage pins used to drive the LCD.
C1: A capacitor connected between CAPH and CAPL
C2: A capacitor connected between V_{L1} and GND
C3: A capacitor connected between V_{L2} and GND
C4: A capacitor connected between V_{L4} and GND
C1 = C2 = C3 = C4 = 0.47 µF±30%

34.9 RAM Data Retention Characteristics

(T_A = -40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.46 Note		3.6	V

Note This depends on the POR detection voltage. For a falling voltage, data in RAM are retained until the voltage reaches the level that triggers a POR reset but not once it reaches the level at which a POR reset is generated.



34.10 Flash Memory Programming Characteristics

(T_A = -40 to +85°C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
CPU/peripheral hardware clock frequency	f_{CLK}	2.4 V ≤ V_{DD} ≤ 3.6 V	1		24	MHz
Number of code flash rewrites Notes 1, 2, 3	C_{erwr}	Retained for 20 years T_A = 85°C	1,000			Times
Number of data flash rewrites Notes 1, 2, 3		Retained for 1 year T_A = 25°C		1,000,000		
		Retained for 5 years T_A = 85°C	100,000			
		Retained for 20 years T_A = 85°C	10,000			

Note 1. 1 erase + 1 write after the erase is regarded as 1 rewrite. The retaining years are until next rewrite after the rewrite.

Note 2. When using flash memory programmer and Renesas Electronics self programming library

Note 3. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

34.11 Dedicated Flash Memory Programmer Communication (UART)

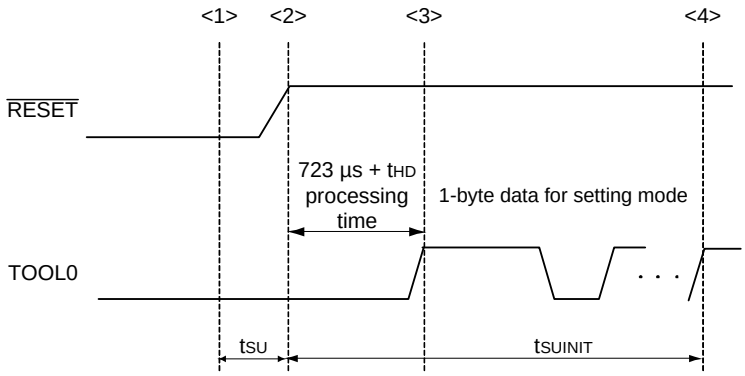
(T_A = -40 to +85°C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		During serial programming	115,200		1,000,000	bps

34.12 Timing of Entry to Flash Memory Programming Modes

(T_A = -40 to +85°C, 1.8 V ≤ V_{DD} ≤ 3.6 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
Time to hold the TOOL0 pin at the low level after an external reset is released (excluding the processing time of the firmware to control the flash memory)	t_{HD}	POR and LVD reset must end before the external reset ends.	1			ms



<1> The low level is input to the TOOL0 pin.
<2> The external reset ends (POR and LVD reset must end before the external reset ends.).
<3> The TOOL0 pin is set to the high level.
<4> Setting of the flash memory programming mode by UART reception and complete the baud rate setting.

Remark t_{SUINIT}: The segment shows that it is necessary to finish specifying the initial communication settings within 100 ms from when the resets end.
t_{SU}: How long from when the TOOL0 pin is placed at the low level until a 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 35 ELECTRICAL SPECIFICATIONS (G: INDUSTRIAL APPLICATIONS T_A = -40 to +105°C)

This chapter describes the following electrical specifications.
Target products G: Industrial applications T_A = -40 to +105°C
R5F110xxGxx, R5F111xxGxx

Caution 1. The RL78 microcontroller has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

Caution 2. The pins mounted depend on the product. Refer to 2.1 Port Function to 2.2.1 With functions for each product.

Caution 3. Please contact Renesas Electronics sales office for derating of operation under T_A = +85°C to +105°C. Derating is the systematic reduction of load for the sake of improved reliability.

Remark When the RL78 microcontroller is used in the range of T_A = -40 to +85°C, see **CHAPTER 34 ELECTRICAL SPECIFICATIONS (T_A = -40 to +85°C)**.

The following functions differ between the products “G: Industrial applications (T_A = -40 to +105°C)” and the products “A: Consumer applications and G: Industrial applications (when used in the range of T_A = -40 to +85°C)”.

Parameter	A: Consumer applications	G: Industrial applications
Operating ambient temperature	T_A = -40 to +85°C	T_A = -40 to +105°C
Operating mode Operating voltage range	HS (high-speed main) mode: 2.7 V ≤ V_{DD} ≤ 3.6 V@1 MHz to 24 MHz 2.4 V ≤ V_{DD} ≤ 3.6 V@1 MHz to 16 MHz LS (low-speed main) mode: 1.8 V ≤ V_{DD} ≤ 3.6 V@1 MHz to 8 MHz LV (low-voltage main) mode: 1.6 V ≤ V_{DD} ≤ 3.6 V@1 MHz to 4 MHz	HS (high-speed main) mode only: 2.7 V ≤ V_{DD} ≤ 3.6 V@1 MHz to 24 MHz 2.4 V ≤ V_{DD} ≤ 3.6 V@1 MHz to 16 MHz
High-speed on-chip oscillator clock accuracy	1.8 V ≤ V_{DD} ≤ 3.6 V: ±1.0% @ T_A = -20 to +85°C ±1.5% @ T_A = -40 to -20°C 1.6 V ≤ V_{DD} ≤ 1.8 V: ±5.0% @ T_A = -20 to +85°C ±5.5% @ T_A = -40 to -20°C	2.4 V ≤ V_{DD} ≤ 3.6 V: ±2.0% @ T_A = +85 to +105°C ±1.0% @ T_A = -20 to +85°C ±1.5% @ T_A = -40 to -20°C
Serial array unit	UART Simplified SPI (CSI): f_{CLK}/4 Simplified I^2C communication	UART Simplified SPI (CSI): f_{CLK}/4 Simplified I^2C communication
IICA	Normal mode Fast mode Fast mode plus	Normal mode Fast mode
Voltage detector	• Rise detection: 1.67 V to 3.13 V (12 levels) • Fall detection: 1.63 V to 3.06 V (12 levels)	• Rise detection: 2.61 V to 3.13 V (6 levels) • Fall detection: 2.55 V to 3.06 V (6 levels)

Remark The electrical characteristics of the products G: Industrial applications (T_A = -40 to +105°C) are different from those of the products “A: Consumer applications”. For details, refer to **35.1** to **35.12**.

35.1 Absolute Maximum Ratings

Absolute Maximum Ratings (T_A = 25°C) (1/3)

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}		-0.5 to + 6.5	V
	UV_{BUS}		-0.5 to + 6.5	V
	AV_{DD}	$AV_{DD} \leq V_{DD}$	-0.5 to + 4.6	V
REGC pin input voltage	V_{IREGC}	REGC	-0.3 to + 2.8 and -0.3 to V_{DD} + 0.3 Note 1	V
U_{REGC} pin input voltage	V_{IUREGC}	U_{REGC}	-0.3 to UV_{BUS} + 0.3 Note 2	V
Input voltage	V_{I1}	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P70 to P77, P80 to P83, P125 to P127, P137, P140 to P143, EXCLK, EXCLKS, $\overline{\text{RESET}}$	-0.3 to V_{DD} + 0.3 Note 3	V
	V_{I2}	P60, P61 (N-ch open-drain)	-0.3 to + 6.5	V
	V_{I3}	UDP, UDM	-0.3 to + 6.5	V
	V_{I4}	P150 to P156	-0.3 to AV_{DD} + 0.3 Note 4	V
Output voltage	V_{O1}	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P60, P61, P70 to P77, P80 to P83, P125 to P127, P130, P140 to P143	-0.3 to V_{DD} + 0.3 Note 3	V
	V_{O2}	P150 to P156	-0.3 to AV_{DD} + 0.3 Note 3	V
	V_{O3}	UDP, UDM	-0.3 to + 3.8	V
Analog input voltage	V_{AI1}	ANI16 to ANI21	-0.3 to V_{DD} + 0.3 and $AV_{REF(+)}$ + 0.3 Notes 3, 5	V
	V_{AI2}	ANI0 to ANI6	-0.3 to AV_{DD} + 0.3 and $AV_{REF(+)}$ + 0.3 Notes 3, 5	V

Note 1. Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

Note 2. Connect the U_{REGC} pin to Vss via a capacitor (0.33 μF). This value regulates the absolute maximum rating of the U_{REGC} pin. Do not use this pin with voltage applied to it.

Note 3. Must be 6.5 V or lower.

Note 4. Must be 4.6 V or lower.

Note 5. Do not exceed $AV_{REF(+)}$ + 0.3 V in case of A/D conversion target pin.

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

Remark 2. $AV_{REF(+)}$: + side reference voltage of the A/D converter.

Remark 3. V_{SS}: Reference voltage

Absolute Maximum Ratings (T_A = 25°C) **(2/3)**

Parameter	Symbols	Conditions		Ratings	Unit
LCD voltage	V_{LI1}	V_{L1} input voltage Note 1		-0.3 to +2.8	V
	V_{LI2}	V_{L2} input voltage Note 1		-0.3 to +6.5	V
	V_{LI3}	V_{L3} input voltage Note 1		-0.3 to +6.5	V
	V_{LI4}	V_{L4} input voltage Note 1		-0.3 to +6.5	V
	V_{LI5}	CAPL, CAPH input voltage Note 1		-0.3 to +6.5	V
	V_{LO1}	V_{L1} output voltage		-0.3 to +2.8	V
	V_{LO2}	V_{L2} output voltage		-0.3 to +6.5	V
	V_{LO3}	V_{L3} output voltage		-0.3 to +6.5	V
	V_{LO4}	V_{L4} output voltage		-0.3 to +6.5	V
	V_{LO5}	CAPL, CAPH output voltage		-0.3 to +6.5	V
	V_{LO6}	COM0 to COM7 SEG0 to SEG55 output voltage	External resistance division method	-0.3 to V_{DD} + 0.3 Note 2	V
			Capacitor split method	-0.3 to V_{DD} + 0.3 Note 2	V
			Internal voltage boosting method	-0.3 to V_{LI4} + 0.3 Note 2	V

Note 1. This value only indicates the absolute maximum ratings when applying voltage to the V_{L1}, V_{L2}, V_{L3}, and V_{L4} pins; it does not mean that applying voltage to these pins is recommended. When using the internal voltage boosting method or capacitance split method, connect these pins to V_{SS} via a capacitor (0.47 ± 30%) and connect a capacitor (0.47 ± 30%) between the CAPL and CAPH pins.

Note 2. Must be 6.5 V or lower.

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.

Absolute Maximum Ratings (T_A = 25°C) **(3/3)**

Parameter	Symbols	Conditions		Ratings	Unit
Output current, high	I_{OH1}	Per pin	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P70 to P77, P80 to P83, P125 to P127, P130, P140 to P143	-40	mA
		Total of all pins -170 mA	P40 to P46	-70	mA
			P00 to P07, P10 to P17, P20 to P27, P30 to P37, P50 to P57, P70 to P77, P80 to P83, P125 to P127, P130, P140 to P143	-100	mA
	I_{OH2}	Per pin	P150 to P156	-0.1	mA
		Total of all pins		-0.7	mA
	I_{OH3}	Per pin	UDP, UDM	-3	mA
Output current, low	I_{OL1}	Per pin	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P60, P61, P70 to P77, P80 to P83, P125 to P127, P130, P140 to P143	40	mA
		Total of all pins 170 mA	P40 to P46	70	mA
			P00 to P07, P10 to P17, P20 to P27, P30 to P37, P50 to P57, P70 to P77, P80 to P83, P125 to P127, P130, P140 to P143	100	mA
	I_{OL2}	Per pin	P150 to P156	0.4	mA
		Total of all pins		2.8	mA
	I_{OL3}	Per pin	UDP, UDM	3	mA
Operating ambient temperature	T_A	In normal operation mode		-40 to +105	°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.

35.2 Oscillator Characteristics

35.2.1 X1 and XT1 oscillator characteristics

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Resonator	Conditions	MIN.	TYP.	MAX.	Unit
X1 clock oscillation frequency (f_X) Note	Ceramic resonator/crystal resonator	2.7 V ≤ V_{DD} ≤ 3.6 V	1.0		20.0	MHz
		2.4 V ≤ V_{DD} < 2.7 V	1.0		16.0	
XT1 clock oscillation frequency (f_{XT}) Note	Crystal resonator		32	32.768	35	kHz

Note Indicates only permissible oscillator frequency ranges. Refer to **AC Characteristics** for instruction execution time. Request evaluation by the manufacturer of the oscillator circuit mounted on a board to check the oscillator characteristics.

Caution **Since the CPU is started by the high-speed on-chip oscillator clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and the oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.**

Remark When using the X1 and XT1 oscillator, refer to **5.4 System Clock Oscillator**.

35.2.2 On-chip oscillator characteristics

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Oscillators	Parameters	Conditions	MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency Notes 1, 2	f_{HOCO}		1		24	MHz
High-speed on-chip oscillator clock frequency accuracy		-20 to +85°C	-1.0		+1.0	%
		-40 to -20°C	-1.5		+1.5	%
		+85 to +105°C	-2.0		+2.0	%
Low-speed on-chip oscillator clock frequency	f_{IL}			15		kHz
Low-speed on-chip oscillator clock frequency accuracy			-15		+15	%

Note 1. High-speed on-chip oscillator frequency is selected with bits 0 to 4 of the option byte (000C2H) and bits 0 to 2 of the HOCODIV register.

Note 2. This only indicates the oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

35.2.3 PLL oscillator characteristics

(T_A = -40 to +105°C, 2.4 V ≤ VDD ≤ 3.6 V, V_{SS} = 0 V)

Oscillators	Parameters	Conditions	MIN.	TYP.	MAX.	Unit
PLL input frequency Note	f_{PLLIN}	High-speed system clock	6.00		16.00	MHz
PLL output frequency Note	f_{PLL}			48.00		MHz

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

35.3 DC Characteristics

35.3.1 Pin characteristics

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high Note 1	IOH1	Per pin for P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P70 to P77, P80 to P83, P125 to P127, P130, P140 to P143				-3.0 Note 2	mA
		Total of P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P70 to P77, P80 to P83, P125 to P127, P130, P140 to P143 (When duty ≤ 70% Note 3)	2.7 V ≤ V_{DD} ≤ 3.6 V			-15.0	mA
			2.4 V ≤ V_{DD} < 2.7 V			-7.0	mA
	I_{OH2}	Per pin for P150 to P156				-0.1 Note 2	mA
		Total of all pins	2.4 V ≤ V_{DD} ≤ 3.6 V			-0.7	mA

Note 1. Value of current at which the device operation is guaranteed even if the current flows from the V_{DD} pin to an output pin.

Note 2. However, do not exceed the total current value.

Note 3. Specification under conditions where the duty factor ≤ 70%.
The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).

- Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OH} = -10.0 mA
Total output current of pins = (-10.0 × 0.7)/(80 × 0.01) ≈ -8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Caution P00 to P02, P10 to P12, P24 to P26, P33 to P35, and P42 to P44 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, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low Note 1	I_{OL1}	Per pin for P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P70 to P77, P80 to P83, P125 to P127, P130, P140 to P143				8.5 Note 2	mA
		Per pin for P60 and P61				15.0 Note 2	mA
		Total of P40 to P46, P130 (When duty ≤ 70% Note 3)	2.7 V ≤ V_{DD} ≤ 3.6 V			15.0	mA
			2.4 V ≤ V_{DD} < 2.7 V			9.0	mA
		Total of P00 to P07, P10 to P17, P20 to P27, P30 to P37, P50 to P57, P60, P61, P70 to P77, P80 to P83, P125 to P127, P140 to P143 (When duty ≤ 70% Note 3)	2.7 V ≤ V_{DD} ≤ 3.6 V			35.0	mA
			2.4 V ≤ V_{DD} < 2.7 V			20.0	mA
		Total of all pins (When duty ≤ 70% Note 3)				50.0	mA
	I_{OL2}	Per pin for P150 to P156				0.4 Note 2	mA
		Total of all pins	2.4 V ≤ V_{DD} ≤ 3.6 V			2.8	mA

Note 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to the V_{SS} pin.

Note 2. However, do not exceed the total current value.

Note 3. Specification under conditions where the duty factor ≤ 70%.
The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression
(when changing the duty factor from 70% to n%).

- Total output current of pins = (I_{OL} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OL} = 10.0 mA
Total output current of pins = (10.0 × 0.7)/(80 × 0.01) ≈ 8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	VIH1	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P70 to P77, P80 to P83, P125 to P127, P140 to P143	Normal input buffer	0.8 V_{DD}		V_{DD}	V
	V_{IH2}	P00, P01, P10, P11, P24, P25, P33, P34, P43, P44	TTL input buffer 3.3 V ≤ V_{DD} ≤ 3.6 V	2.0		V_{DD}	V
			TTL input buffer 2.4 V ≤ V_{DD} < 3.3 V	1.50		V_{DD}	V
	V_{IH3}	P150 to P156		0.7 AV_{DD}		AV_{DD}	V
	V_{IH4}	P60, P61		0.7 V_{DD}		6.0	V
	V_{IH5}	P121 to P124, P137, EXCLK, EXCLKS, $\overline{\text{RESET}}$		0.8 V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P70 to P77, P80 to P83, P125 to P127, P140 to P143	Normal input buffer	0		0.2 V_{DD}	V
	V_{IL2}	P00, P01, P10, P11, P24, P25, P33, P34, P43, P44	TTL input buffer 3.3 V ≤ V_{DD} ≤ 3.6 V	0		0.5	V
			TTL input buffer 2.4 V ≤ V_{DD} < 3.3 V	0		0.32	V
	V_{IL3}	P150 to P156		0		0.3 AV_{DD}	V
	V_{IL4}	P60, P61		0		0.3 V_{DD}	V
	V_{IL5}	P121 to P124, P137, EXCLK, EXCLKS, $\overline{\text{RESET}}$		0		0.2 V_{DD}	V

Caution The maximum value of V_{IH} of pins P00 to P02, P10 to P12, P24 to P26, P33 to P35, and P42 to P44 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, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH1}	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P70 to P77, P80 to P83, P125 to P127, P130, P140 to P143	2.7 V ≤ V_{DD} ≤ 3.6 V, I_{OH1} = -2.0 mA	V_{DD} - 0.6			V
			2.4 V ≤ V_{DD} ≤ 3.6 V, I_{OH1} = -1.5 mA	V_{DD} - 0.5			V
	V_{OH2}	P150 to P156	2.4 V ≤ V_{DD} ≤ 3.6 V, I_{OH2} = -100 μA	AV_{DD} - 0.5			V
Output voltage, low	V_{OL1}	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P70 to P77, P80 to P83, P125 to P127, P130, P140 to P143	2.7 V ≤ V_{DD} ≤ 3.6 V, I_{OL1} = 3.0 mA			0.6	V
			2.4 V ≤ V_{DD} ≤ 3.6 V, I_{OL1} = 1.5 mA			0.4	V
			2.4 V ≤ V_{DD} ≤ 3.6 V, I_{OL1} = 0.6 mA			0.4	V
	V_{OL2}	P150 to P156	2.4 V ≤ V_{DD} ≤ 3.6 V, I_{OL2} = 400 μA			0.4	V
	V_{OL3}	P60, P61	2.7 V ≤ V_{DD} ≤ 3.6 V, I_{OL3} = 3.0 mA			0.4	V
			2.4 V ≤ V_{DD} ≤ 3.6 V, I_{OL3} = 2.0 mA			0.4	V

Caution P00 to P02, P10 to P12, P24 to P26, P33 to P35, and P42 to P44 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, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P60, P61, P70 to P77, P80 to P83, P125 to P127, P137, P140 to P143, $\overline{RESET}$	V_I = V_{DD}				1	μA
	I_{LIH2}	P20, P21, P140 to P143	V_I = V_{DD}				1	μA
	I_{LIH3}	P121 to P124 (X1, X2, EXCLK, XT1, XT2, EXCLKS)	V_I = V_{DD}	In input port or external clock input			1	μA
				In resonator connection			10	μA
	I_{LIH4}	P150 to P156	V_I = AV_{DD}				1	μA
Input leakage current, low	I_{LIL1}	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P46, P50 to P57, P60, P61, P70 to P77, P80 to P83, P125 to P127, P137, P140 to P143, $\overline{RESET}$	V_I = V_{SS}				-1	μA
	I_{LIL2}	P20, P21, P140 to P143	V_I = V_{SS}				-1	μA
	I_{LIL3}	P121 to P124 (X1, X2, EXCLK, XT1, XT2, EXCLKS)	V_I = V_{SS}	In input port or external clock input			-1	μA
				In resonator connection			-10	μA
	I_{LIL4}	P150 to P156	V_I = AV_{SS}				-1	μA
On-chip pull-up resistance	R_{U1}	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P50 to P57, P70 to P77, P140 to P143, P125 to P127	V_I = V_{SS}	2.4 V ≤ V_{DD} ≤ 3.6 V	10	20	100	kΩ
	R_{U2}	P40 to P46, P80 to P83	V_I = V_{SS}		10	20	100	kΩ

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

35.3.2 Supply current characteristics

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 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_{HOCO} = 48 MHz Note 3, f_{IH} = 24 MHz Note 3	Basic operation	V_{DD} = 3.6 V		2.2	2.9	mA
						V_{DD} = 3.0 V		2.2	2.9	
					Normal operation	V_{DD} = 3.6 V		4.4	9.2	
						V_{DD} = 3.0 V		4.4	9.2	
				f_{HOCO} = 24 MHz Note 3, f_{IH} = 24 MHz Note 3	Basic operation	V_{DD} = 3.6 V		2.0	2.6	
						V_{DD} = 3.0 V		2.0	2.6	
					Normal operation	V_{DD} = 3.6 V		4.2	7.0	
						V_{DD} = 3.0 V		4.2	7.0	
				f_{HOCO} = 16 MHz Note 3, f_{IH} = 16 MHz Note 3	Normal operation	V_{DD} = 3.6 V		3.1	5.0	
						V_{DD} = 3.0 V		3.1	5.0	
			HS (high-speed main) mode Note 5	f_{MX} = 20 MHz Note 2, V_{DD} = 3.6 V	Normal operation	Square wave input		3.5	5.9	mA
						Resonator connection		3.6	6.0	
				f_{MX} = 20 MHz Note 2, V_{DD} = 3.0 V	Normal operation	Square wave input		3.5	5.9	
						Resonator connection		3.6	6.0	
				f_{MX} = 16 MHz Note 2, V_{DD} = 3.6 V	Normal operation	Square wave input		2.9	4.5	
						Resonator connection		3.1	4.6	
				f_{MX} = 16 MHz Note 2, V_{DD} = 3.0 V	Normal operation	Square wave input		2.9	4.5	
						Resonator connection		3.1	4.6	
				f_{MX} = 10 MHz Note 2, V_{DD} = 3.6 V	Normal operation	Square wave input		2.1	3.5	
						Resonator connection		2.2	3.5	
				f_{MX} = 10 MHz Note 2, V_{DD} = 3.0 V	Normal operation	Square wave input		2.1	3.5	
						Resonator connection		2.2	3.5	
			HS (High-speed main) mode (PLL operation)	f_{PLL} = 48 MHz, f_{CLK} = 24 MHz Note 2	Normal operation	V_{DD} = 3.6 V		4.7	7.6	mA
						V_{DD} = 3.0 V		4.7	7.6	
				f_{PLL} = 48 MHz, f_{CLK} = 12 MHz Note 2	Normal operation	V_{DD} = 3.6 V		3.1	5.2	
						V_{DD} = 3.0 V		3.1	5.1	
				f_{PLL} = 48 MHz, f_{CLK} = 6 MHz Note 2	Normal operation	V_{DD} = 3.6 V		2.3	3.9	
						V_{DD} = 3.0 V		2.3	3.9	
			Subsystem clock operation	f_{SUB} = 32.768 kHz Note 4 T_A = -40°C	Normal operation	Square wave input		4.6	6.9	µA
						Resonator connection		4.7	6.9	
				f_{SUB} = 32.768 kHz Note 4 T_A = +25°C	Normal operation	Square wave input		4.9	7.0	
						Resonator connection		5.0	7.2	
				f_{SUB} = 32.768 kHz Note 4 T_A = +50°C	Normal operation	Square wave input		5.2	7.6	
						Resonator connection		5.2	7.7	
				f_{SUB} = 32.768 kHz Note 4 T_A = +70°C	Normal operation	Square wave input		5.5	9.3	
						Resonator connection		5.6	9.4	
				f_{SUB} = 32.768 kHz Note 4 T_A = +85°C	Normal operation	Square wave input		6.2	13.3	
						Resonator connection		6.2	13.4	
				f_{SUB} = 32.768 kHz Note 4 T_A = +105°C	Normal operation	Square wave input		8.3	46.0	
						Resonator connection		8.4	46.0	

(**Notes** and **Remarks** are listed on the next page.)

Note 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD}, or V_{SS}. The following points apply in the HS (high-speed main) mode.
- 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 LCD controller/driver, A/D converter, D/A converter, comparator, LVD circuit, USB 2.0 function module, 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 real-time clock 2.

Note 2. When high-speed on-chip oscillator and subsystem clock are stopped.

Note 3. When high-speed system clock and subsystem clock are stopped.

Note 4. When high-speed on-chip oscillator and high-speed system clock are stopped. When AMPHS1 = 1 (Ultra-low power consumption oscillation).

Note 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} ≤ 3.6 V@1 MHz to 24 MHz
2.4 V ≤ V_{DD} ≤ 3.6 V@1 MHz to 16 MHz

Remark 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

Remark 2. f_{HOCO}: High-speed on-chip oscillator clock frequency (48 MHz max.)

Remark 3. f_{IH}: Main system clock source frequency when the high-speed on-chip oscillator clock divided 1, 2, 4, or 8, or the PLL clock divided by 2, 4, or 8 is selected (24 MHz max.)

Remark 4. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

Remark 5. Except subsystem clock operation, temperature condition of the TYP. value is T_A = 25°C

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 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_{HOCO} = 48 MHz Note 4, f_{IH} = 24 MHz Note 4	V_{DD} = 3.6 V		0.77	3.4	mA
					V_{DD} = 3.0 V		0.77	3.4	
				f_{HOCO} = 24 MHz Note 4, f_{IH} = 24 MHz Note 4	V_{DD} = 3.6 V		0.55	2.7	
					V_{DD} = 3.0 V		0.55	2.7	
				f_{HOCO} = 16 MHz Note 4, f_{IH} = 16 MHz Note 4	V_{DD} = 3.6 V		0.48	1.9	
					V_{DD} = 3.0 V		0.47	1.9	
			HS (high-speed main) mode Note 6	f_{MX} = 20 MHz Note 3, V_{DD} = 3.6 V	Square wave input		0.35	2.10	mA
					Resonator connection		0.51	2.20	
				f_{MX} = 20 MHz Note 3, V_{DD} = 3.0 V	Square wave input		0.34	2.10	
					Resonator connection		0.51	2.20	
				f_{MX} = 16 MHz Note 3, V_{DD} = 3.6 V	Square wave input		0.30	1.25	
					Resonator connection		0.45	1.41	
				f_{MX} = 16 MHz Note 3, V_{DD} = 3.0 V	Square wave input		0.29	1.23	
					Resonator connection		0.45	1.41	
				f_{MX} = 10 MHz Note 3, V_{DD} = 3.6 V	Square wave input		0.23	1.10	
					Resonator connection		0.30	1.20	
				f_{MX} = 10 MHz Note 3, V_{DD} = 3.0 V	Square wave input		0.22	1.10	
					Resonator connection		0.30	1.20	
			HS (High-speed main) mode (PLL operation)	f_{MX} = 48 MHz, f_{CLK} = 24 MHz Note 3	V_{DD} = 3.6 V		0.99	2.93	mA
					V_{DD} = 3.0 V		0.99	2.92	
				f_{MX} = 48 MHz, f_{CLK} = 12 MHz Note 3	V_{DD} = 3.6 V		0.89	2.51	
					V_{DD} = 3.0 V		0.89	2.50	
				f_{MX} = 48 MHz, f_{CLK} = 6 MHz Note 3	V_{DD} = 3.6 V		0.84	2.30	
					V_{DD} = 3.0 V		0.84	2.29	
			Subsystem clock operation	f_{SUB} = 32.768 kHz Note 5 T_A = -40°C	Square wave input		0.32	0.61	µA
					Resonator connection		0.51	0.80	
				f_{SUB} = 32.768 kHz Note 5 T_A = +25°C	Square wave input		0.41	0.74	
					Resonator connection		0.62	0.91	
				f_{SUB} = 32.768 kHz Note 5 T_A = +50°C	Square wave input		0.52	2.30	
					Resonator connection		0.75	2.49	
				f_{SUB} = 32.768 kHz Note 5 T_A = +70°C	Square wave input		0.82	4.03	
					Resonator connection		1.08	4.22	
				f_{SUB} = 32.768 kHz Note 5 T_A = +85°C	Square wave input		1.38	8.04	
					Resonator connection		1.62	8.23	
				f_{SUB} = 32.768 kHz Note 5 T_A = +105°C	Square wave input		3.29	41.00	
					Resonator connection		3.63	41.00	
	I_{DD3}	STOP mode Note 7	T_A = -40°C				0.18	0.52	µA
			T_A = +25°C				0.25	0.52	
			T_A = +50°C				0.34	2.21	
			T_A = +70°C				0.64	3.94	
			T_A = +85°C				1.18	7.95	
			T_A = +105°C				2.92	40.00	

(**Notes** and **Remarks** are listed on the next page.)

Note 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The following points apply in the HS (high-speed main) mode.
- 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 LCD controller/driver, A/D converter, D/A converter, comparator, LVD circuit, USB 2.0 function module, 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 real-time clock 2.
In the STOP mode, the currents in both the “TYP.” and “MAX.” columns do not include the operating currents of the peripheral modules.

Note 2. During HALT instruction execution by flash memory.

Note 3. When high-speed on-chip oscillator and subsystem clock are stopped.

Note 4. When high-speed system clock and subsystem clock are stopped.

Note 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).

Note 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} ≤ 3.6 V@1 MHz to 24 MHz
2.4 V ≤ V_{DD} ≤ 3.6 V@1 MHz to 16 MHz

Note 7. Regarding the value for current to operate the subsystem clock in STOP mode, refer to that in HALT mode.

Remark 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

Remark 2. f_{HOCO}: High-speed on-chip oscillator clock frequency (48 MHz max.)

Remark 3. f_{IH}: Main system clock source frequency when the high-speed on-chip oscillator clock divided 1, 2, 4, or 8, or the PLL clock divided by 2, 4, or 8 is selected (24 MHz max.)

Remark 4. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

Remark 5. Except subsystem clock operation and STOP mode, temperature condition of the TYP. value is T_A = 25°C

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 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
RTC2 operating current	I_{RTC} Notes 1, 3						0.02		μA
12-bit interval timer operating current	I_{TMKA} 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 6, 7	AV_{DD} = 3.0 V, when conversion at maximum speed					422	720	μA
$AV_{REF(+)}$ current	I_{AVREF} Note 8	AV_{DD} = 3.0 V, ADREFP1 = 0, ADREFP0 = 0 Note 7					14.0	25.0	μA
		AV_{REFP} = 3.0 V, ADREFP1 = 0, ADREFP0 = 1 Note 10					14.0	25.0	
		ADREFP1 = 1, ADREFP0 = 0 Note 1					14.0	25.0	
A/D converter reference voltage current	I_{ADREF} Notes 1, 9	V_{DD} = 3.0 V					75.0		μA
Temperature sensor operating current	I_{TMPS} Note 1						78		μA
D/A converter operating current	I_{DAC} Notes 1, 11	Per D/A converter channel					0.53	1.5	mA
Comparator operating current	I_{CMP} Notes 1, 12	V_{DD} = 3.6 V, Regulator output voltage = 2.1 V	Window mode				12.5		μA
			Comparator high-speed mode				4.5		μA
			Comparator low-speed mode				1.2		μA
LVD operating current	I_{LVD} Notes 1, 13						0.06		μA
Self-programming operating current	I_{FSP} Notes 1, 14						2.50	12.20	mA
BGO operating current	I_{BGO} Notes 1, 15						1.68	12.20	mA
SNOOZE operating current	I_{SNOZ} Note 1	ADC operation	The mode is performed Note 16				0.34	1.10	mA
			The A/D conversion operations are performed, Low voltage mode, AV_{REFP} = V_{DD} = 3.0 V				0.53	2.04	
		Simplified SPI (CSI)/UART operation					0.70	1.54	mA
LCD operating current	I_{LCD1} Notes 17, 18	External resistance division method	f_{LCD} = f_{SUB} LCD clock = 128 Hz	1/3 bias 4-time slice	V_{DD} = 3.6 V, L_{V4} = 3.6 V		0.14		μA
	I_{LCD2} Note 17	Internal voltage boosting method	f_{LCD} = f_{SUB} LCD clock = 128 Hz	1/3 bias 4-time slice	V_{DD} = 3.0 V, L_{V4} = 3.0 V (VLCD = 04H)		0.61		μA
	I_{LCD3} Note 17	Capacitor split method	f_{LCD} = f_{SUB} LCD clock = 128 Hz	1/3 bias 4-time slice	V_{DD} = 3.0 V, L_{V4} = 3.0 V		0.12		μA
USB current Note 19	I_{USB} Note 20	Operating current during USB communication					4.88		mA
	I_{USB} Note 21	Operating current in the USB suspended state					0.04		mA

(**Notes** and **Remarks** are listed on the next page.)

Note 1. Current flowing to V_{DD}.

Note 2. When high speed on-chip oscillator and high-speed system clock are stopped.

Note 3. Current flowing only to the real-time clock 2 (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 2 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 2.

Note 4. Current flowing only to the 12-bit interval timer (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_{TMKA}, 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. I_{DD2} subsystem clock operation includes the operational current of the 12-bit interval timer.

Note 5. Current flowing only to the watchdog timer (including the operating current of the low-speed on-chip oscillator). The current value of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when the watchdog timer operates in STOP mode.

Note 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}, I_{AVREF}, I_{ADREF} when the A/D converter operates in an operation mode or the HALT mode.

Note 7. Current flowing to the AV_{DD}.

Note 8. Current flowing from the reference voltage source of A/D converter.

Note 9. Operation current flowing to the internal reference voltage.

Note 10. Current flowing to the AV_{REFP}.

Note 11. Current flowing only to the D/A converter. The current value of the RL78 microcontrollers is the sum of I_{DD1} or I_{DD2} and I_{DA} when the D/A converter operates in an operation mode or the HALT mode.

Note 12. Current flowing only to the comparator circuit. The current value of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{CMP} when the comparator circuit operates in the Operating, HALT or STOP mode.

Note 13. Current flowing only to the LVD circuit. The current value of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVI} when the LVD circuit operates in the Operating, HALT or STOP mode.

Note 14. Current flowing only during self-programming.

Note 15. Current flowing only during data flash rewrite.

Note 16. For shift time to the SNOOZE mode, see **23.3.3 SNOOZE mode**.

Note 17. Current flowing only to the LCD controller/driver (V_{DD} pin). The current value of the RL78 microcontrollers is the sum of the LCD operating current (I_{LCD1}, I_{LCD2} or I_{LCD3}) to the supply current (I_{DD1}, or I_{DD2}) when the LCD controller/driver operates in an operation mode or HALT mode. Not including the current that flows through the LCD panel.

Note 18. Not including the current that flows through the external divider resistor divider resistor.

Note 19. Current flowing to the UV_{BUS}.

Note 20. Including the operating current when f_{PLL} = 48 MHz.

Note 21. Including the current supplied from the pull-up resistor of the UDP pin to the pull-down resistor of the host device, in addition to the current consumed by this MCU during the suspended state.

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

Remark 2. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

Remark 3. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 4. Temperature condition of the TYP. value is T_A = 25°C

35.4 AC Characteristics

35.4.1 Basic operation

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V) (1/2)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	HS (high-speed main) mode	2.7 V ≤ V_{DD} ≤ 3.6 V	0.0417		1	μs
				2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	μs
		Subsystem clock (f_{SUB}) operation		2.4 V ≤ V_{DD} ≤ 3.6 V	28.5	30.5	31.3	μs
		In the self-programming mode	HS (high-speed main) mode	2.7 V ≤ V_{DD} ≤ 3.6 V	0.0417		1	μs
				2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	μs
External main system clock frequency	f_{EX}	2.7 V ≤ V_{DD} ≤ 3.6 V			1.0		20.0	MHz
		2.4 V ≤ V_{DD} < 2.7 V			1.0		16.0	MHz
	f_{EXT}				32		35	kHz
External main system clock input high-level width, low-level width	t_{EXH}, t_{EXL}	2.7 V ≤ V_{DD} ≤ 3.6 V			24			ns
		2.4 V ≤ V_{DD} < 2.7 V			30			ns
	t_{EXHS}, t_{EXLS}				13.7			μs
TI00 to TI07 input high-level width, low-level width	t_{TIH}, t_{TIL}				1/f_{MCK} + 10			ns

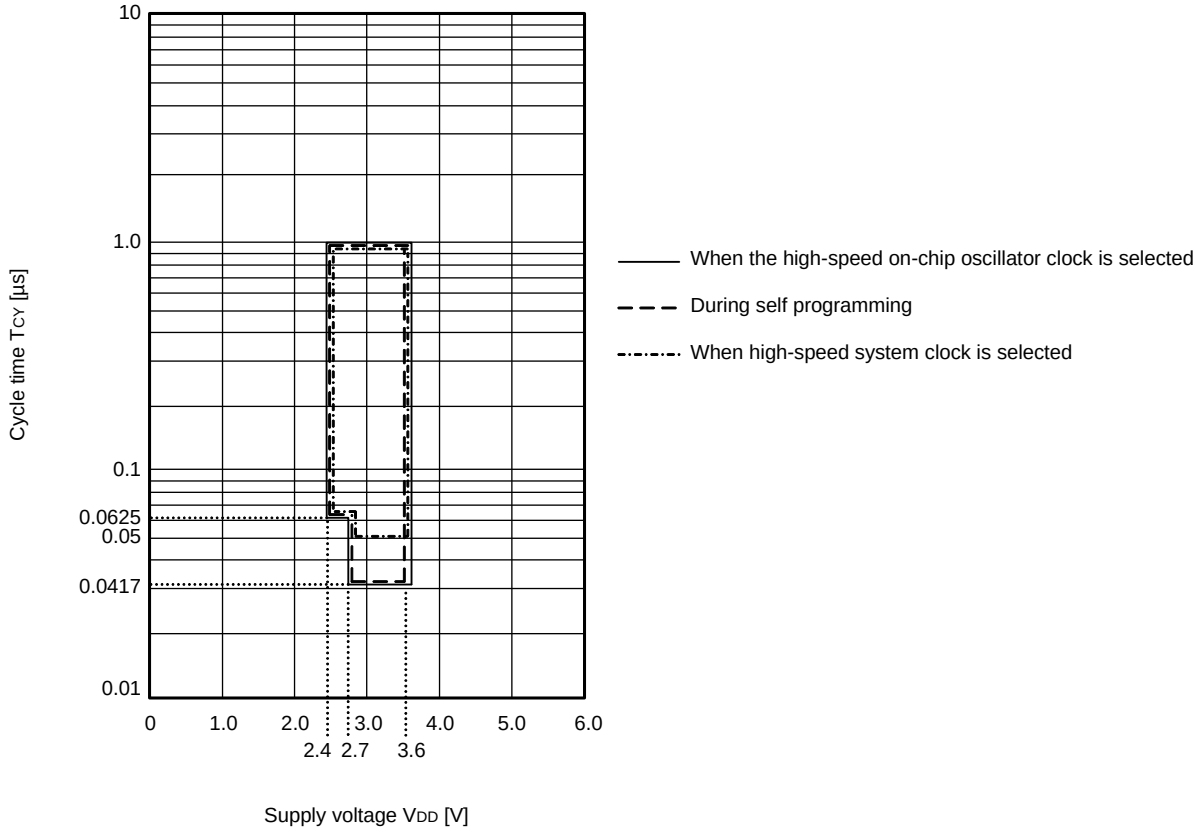
Remark f_{MCK}: Timer array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of timer mode register mn (TMRmn). m: Unit number (m = 0), n: Channel number (n = 0 to 7))

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V) **(2/2)**

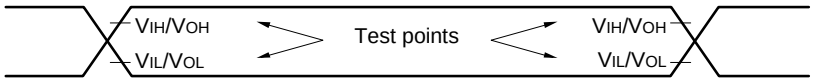
Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
TO00 to TO07, TKBO00, TKBO01, TKBO10, TKBO11, TKBO20, TKBO21 output frequency	f_{TO}	HS (high-speed main) mode	2.7 V ≤ V_{DD} ≤ 3.6 V			8	MHz
			2.4 V ≤ V_{DD} < 2.7 V			8	MHz
PCLBUZ0, PCLBUZ1 output frequency	f_{PCL}	HS (high-speed main) mode	2.7 V ≤ V_{DD} ≤ 3.6 V			8	MHz
			2.4 V ≤ V_{DD} < 2.7 V			8	MHz
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}	INTP0 to INTP7	2.4 V ≤ V_{DD} ≤ 3.6 V	1			µs
Key interrupt input low-level width	t_{KR}	2.4 V ≤ V_{DD} ≤ 3.6 V		250			ns
TMKB2 forced output stop input high-level width	t_{IHR}	INTP0 to INTP7	f_{CLK} > 16 MHz	125			ns
			f_{CLK} ≤ 16 MHz	2			f_{CLK}
$\overline{\text{RESET}}$ low-level width	t_{RSL}			10			µs

Minimum Instruction Execution Time during Main System Clock Operation

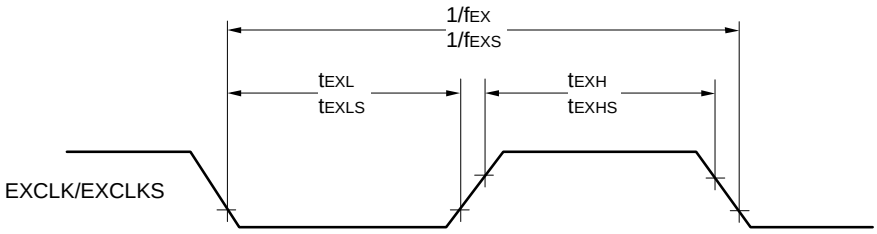
T_{CY} vs V_{DD} (HS (high-speed main) mode)



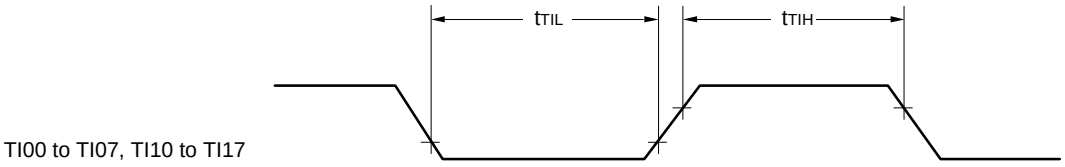
AC Timing Test Points

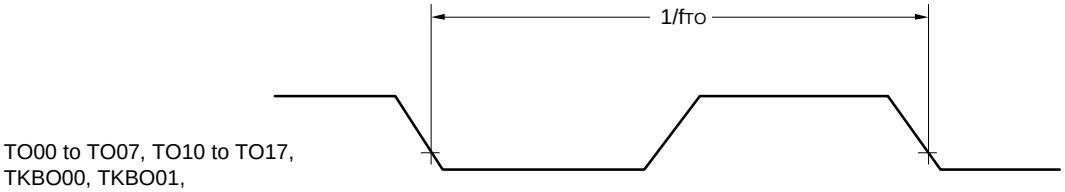


External System Clock Timing

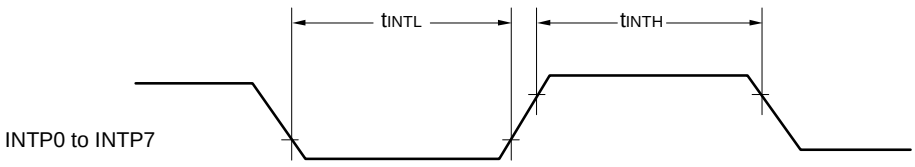


TI/TO Timing

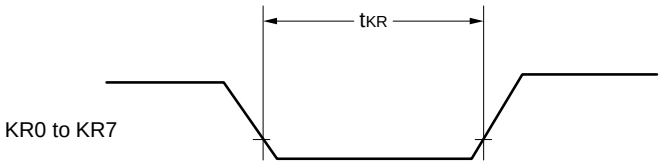




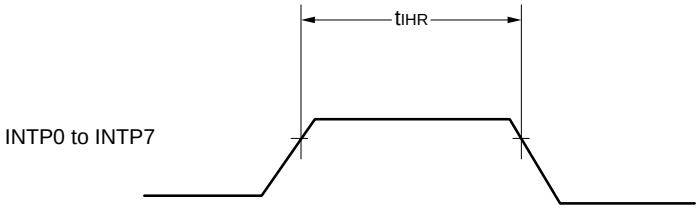
Interrupt Request Input Timing



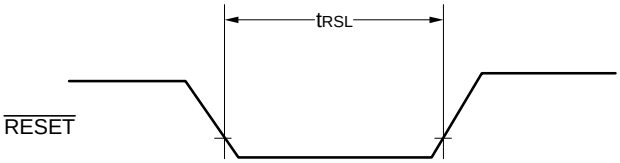
Key Interrupt Input Timing



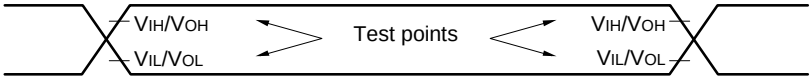
Timer KB2 Input Timing



$\overline{\text{RESET}}$ Input Timing



35.5 Peripheral Functions Characteristics



35.5.1 Serial array unit

(1) During communication at same potential (UART mode)

(TA = -40 to +105°C, 2.4 V ≤ VDD ≤ 3.6 V, VSS = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
Transfer rate Note 1		2.4 V ≤ VDD ≤ 3.6 V		fMCK/12 Note 2	bps
		Theoretical value of the maximum transfer rate fMCK = fCLK Note 3		2.0	Mbps

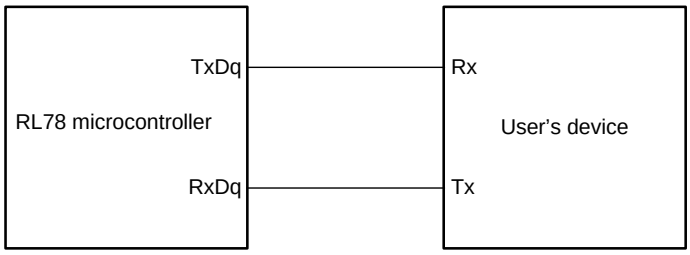
Note 1. Transfer rate in the SNOOZE mode is 4800 bps only.

Note 2. The following conditions are required for low voltage interface.
2.4 V ≤ VDD < 2.7 V: MAX. 1.3 Mbps

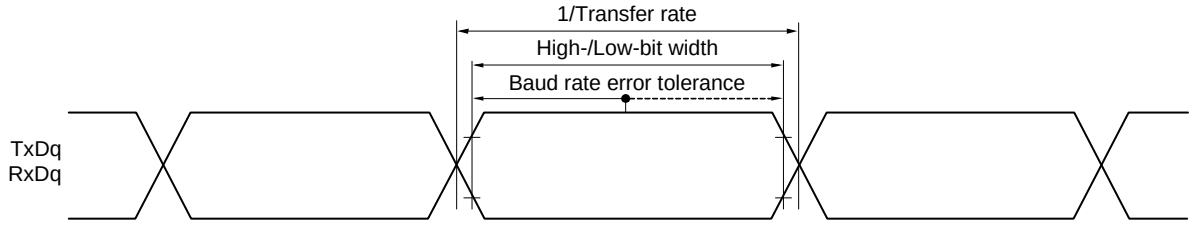
Note 3. The maximum operating frequencies of the CPU/peripheral hardware clock (fCLK) are:
HS (high-speed main) mode: 24 MHz (2.7 V ≤ VDD ≤ 3.6 V)
16 MHz (2.4 V ≤ VDD ≤ 3.6 V)

Caution Select the normal input buffer for the RxDq pin and the normal output mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg).

UART mode connection diagram (during communication at same potential)



UART mode bit width (during communication at same potential) (reference)



Remark 1. q: UART number (q = 0 to 3), g: PIM and POM number (g = 0 to 3)

Remark 2. fMCK: 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, 10 to 13))

(2) During communication at same potential (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq f_{CLK}/4$	2.7 V ≤ V_{DD} ≤ 3.6 V	250		ns
			2.4 V ≤ V_{DD} ≤ 3.6 V	500		ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	2.7 V ≤ V_{DD} ≤ 3.6 V		t_{KCY1}/2 - 36		ns
		2.4 V ≤ V_{DD} ≤ 3.6 V		t_{KCY1}/2 - 76		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 3.6 V		66		ns
		2.4 V ≤ V_{DD} ≤ 3.6 V		133		ns
SIp hold time (from SCKp↑) Note 2	t_{KSI1}			38		ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO1}	C = 30 pF Note 4			50	ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes “to SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes “from SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes “from SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 4. C is the load capacitance of the SCKp and SOp output lines.

Caution Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remark 1. p: CSI number (p = 00, 10, 20, 30), m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), g: PIM number (g = 0 to 3)

Remark 2. f_{MCK}: Serial array unit operation clock frequency (Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00 to 03, 10 to 13))

(3) During communication at same potential (Simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
SCKp cycle time Note 5	t_{KCY2}	2.7 V ≤ V_{DD} < 3.6 V	f_{MCK} > 16 MHz	16/f_{MCK}		ns
			f_{MCK} ≤ 16 MHz	12/f_{MCK}		ns
		2.4 V ≤ V_{DD} < 3.6 V		12/f_{MCK} and 1000		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	2.7 V ≤ V_{DD} ≤ 3.6 V		t_{KCY2}/2 - 16		ns
		2.4 V ≤ V_{DD} ≤ 3.6 V		t_{KCY2}/2 - 36		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK2}	2.7 V ≤ V_{DD} ≤ 3.6 V		1/f_{MCK} + 40		ns
		2.4 V ≤ V_{DD} ≤ 3.6 V		1/f_{MCK} + 60		ns
SIp hold time (from SCKp↑) Note 2	t_{KSI2}			1/f_{MCK} + 62		ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO2}	C = 30 pF Note 4	2.7 V ≤ V_{DD} ≤ 3.6 V		2/f_{MCK} + 66	ns
			2.4 V ≤ V_{DD} < 3.6 V		2/f_{MCK} + 113	ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes “to SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes “from SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes “from SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 4. C is the load capacitance of the SOp output lines.

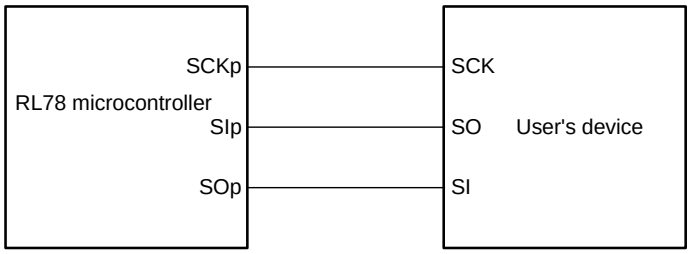
Note 5. The maximum transfer rate when using the SNOOZE mode is 1 Mbps.

Caution Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remark 1. p: CSI number (p = 00, 10, 20, 30), m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), g: PIM number (g = 0 to 3)

Remark 2. f_{MCK}: Serial array unit operation clock frequency (Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00 to 03, 10 to 13))

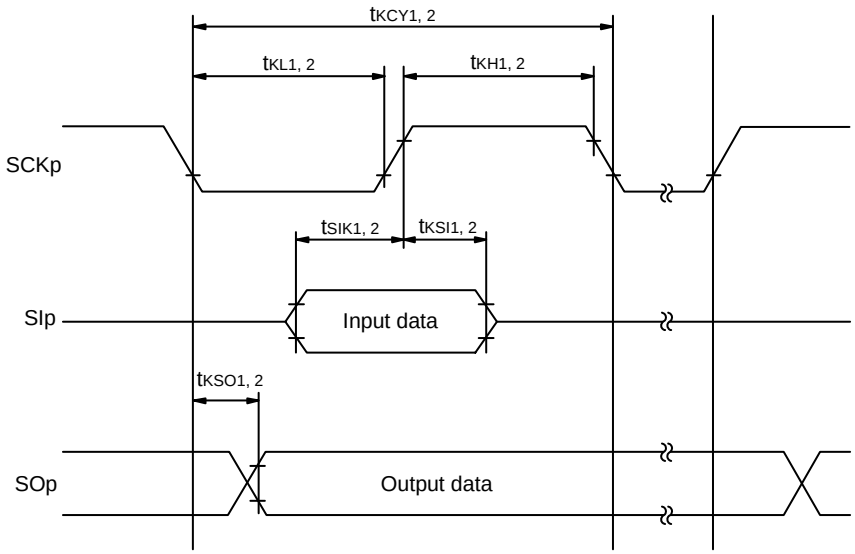
Simplified SPI (CSI) mode connection diagram (during communication at same potential)



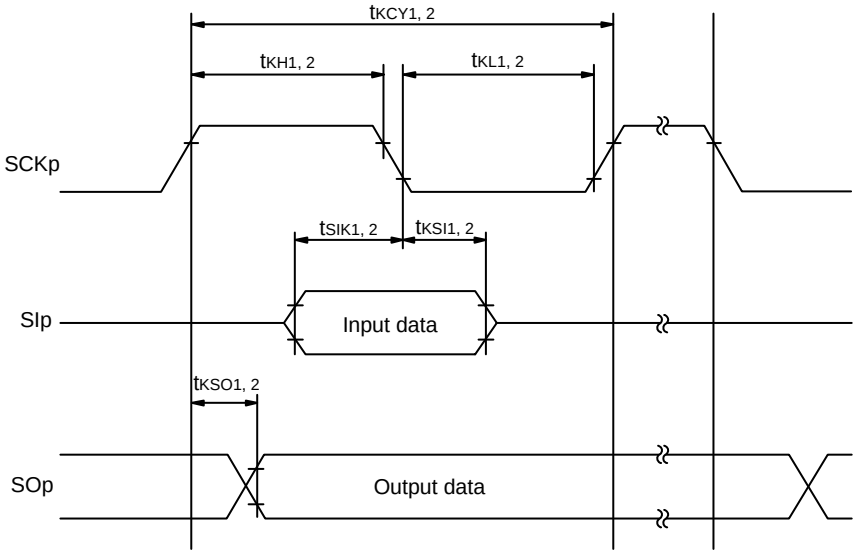
Remark 1. p: CSI number (p = 00, 10, 20, 30)

Remark 2. m: Unit number, n: Channel number (mn = 00 to 03, 10 to 13)

Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark 1. p: CSI number (p = 00, 10, 20, 30)
Remark 2. m: Unit number, n: Channel number (mn = 00 to 03, 10 to 13)

(4) During communication at same potential (simplified I²C mode)

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

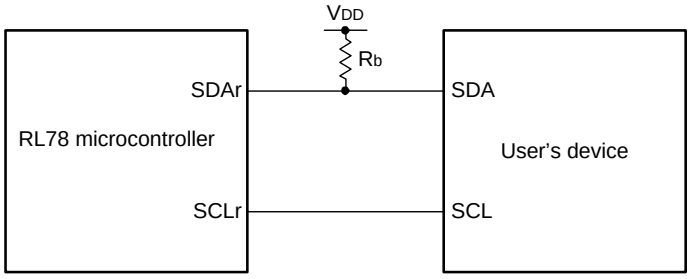
Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
SCLr clock frequency	f_{SCL}	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 50 pF, R_b = 2.7 kΩ		400 Note 1	kHz
		2.4 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ		100 Note 1	kHz
Hold time when SCLr = "L"	t_{LOW}	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 50 pF, R_b = 2.7 kΩ	1200		ns
		2.4 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ	4600		ns
Hold time when SCLr = "H"	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 50 pF, R_b = 2.7 kΩ	1200		ns
		2.4 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ	4600		ns
Data setup time (reception)	$t_{SU: DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 50 pF, R_b = 2.7 kΩ	1/f_{MCK} + 200 Note 2		ns
		2.4 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ	1/f_{MCK} + 580 Note 2		ns
Data hold time (transmission)	$t_{HD: DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 50 pF, R_b = 2.7 kΩ	0	770	ns
		2.4 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ	0	1420	ns

Note 1. The value must be equal to or less than f_{MCK}/4.

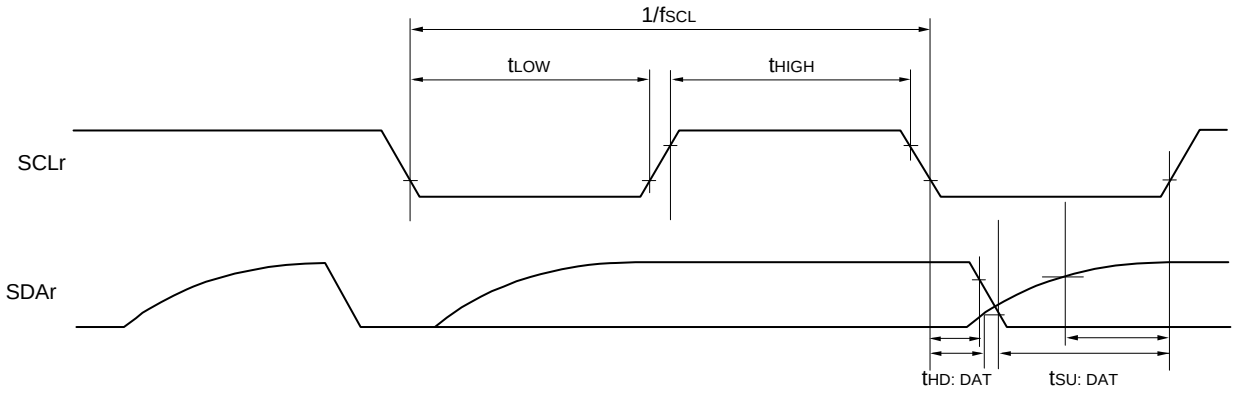
Note 2. Set the f_{MCK} value to keep the hold time of SCLr = "L" and SCLr = "H".

Caution Select the normal input buffer and the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin and the normal output mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register h (POMh).

Simplified I²C mode connection diagram (during communication at same potential)



Simplified I²C mode serial transfer timing (during communication at same potential)



Remark 1. R_b[Ω]: Communication line (SDAr) pull-up resistance, C_b[F]: Communication line (SDAr, SCLr) load capacitance

Remark 2. r: IIC number (r = 00, 10, 20, 30), g: PIM number (g = 0 to 3),
h: POM number (h = 0 to 3)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number (m = 0, 1),
n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13)

(5) Communication at different potential (1.8 V, 2.5 V) (UART mode)

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) (1/2)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
Transfer rate Notes 1, 2		Reception	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V		f_{MCK}/12 Note 1	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 4		2.0	Mbps
			1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		f_{MCK}/12 Notes 1, 2, 3	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 4		1.3	Mbps

Note 1. Transfer rate in the SNOOZE mode is 4,800 bps only.

Note 2. Use it with $V_{DD} \geq V_b$.

Note 3. The following conditions are required for low voltage interface.
2.4 V ≤ V_{DD} < 2.7 V: MAX. 2.6 Mbps

Note 4. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
HS (high-speed main) mode: 24 MHz (2.7 V ≤ V_{DD} ≤ 3.6 V)
16 MHz (2.4 V ≤ V_{DD} ≤ 3.6 V)

Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Remark 1. V_b[V]: Communication line voltage

Remark 2. q: UART number (q = 0 to 3), g: PIM and POM number (g = 0 to 3)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00 to 03, 10 to 13))

(5) Communication at different potential (1.8 V, 2.5V) (UART mode)

(T_A = -40 to +105°C, 2.4 ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
Transfer rate Note 2		Transmission	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V		Note 1	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 2	Mbps
			1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		Notes 3, 4	bps
			Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 5.5 kΩ, V_b = 1.6 V		0.43 Note 5	Mbps

Note 1. The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 2.7 V ≤ V_{DD} < 3.6 V and 2.3 V ≤ V_b ≤ 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

Note 2. This value as an example is calculated when the conditions described in the “Conditions” column are met.
Refer to Note 1 above to calculate the maximum transfer rate under conditions of the customer.

Note 3. Use it with V_{DD} ≥ V_b.

Note 4. 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 2.4 V ≤ V_{DD} < 3.3 V and 1.6 V ≤ V_b ≤ 2.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\} \times 3} \text{ [bps]}$$

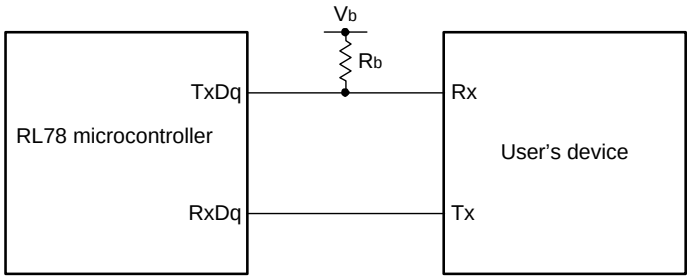
$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

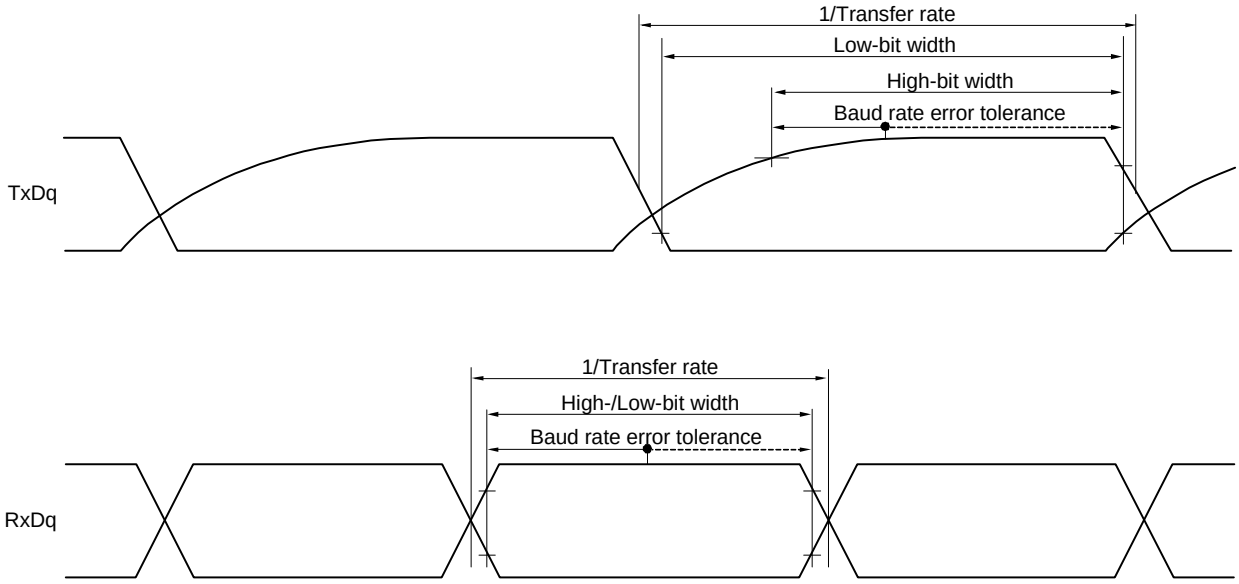
Note 5. This value as an example is calculated when the conditions described in the “Conditions” column are met.
Refer to Note 4 above to calculate the maximum transfer rate under conditions of the customer.

Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

UART mode connection diagram (during communication at different potential)



UART mode bit width (during communication at different potential) (reference)



Remark 1. R_b[Ω]: Communication line (TxDq) pull-up resistance, C_b[F]: Communication line (TxDq) load capacitance, V_b[V]: Communication line voltage

Remark 2. q: UART number (q = 0 to 3), g: PIM and POM number (g = 0 to 3)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00 to 03, 10 to 13))

(6) Communication at different potential (1.8 V, 2.5 V) (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) (1/2)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq f_{CLK}/4$	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	1000 Note		ns
			2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 1.8 V, C_b = 30 pF, R_b = 5.5 kΩ	2300 Note		ns
SCKp high-level width	t_{KH1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 - 340		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 - 916		ns
SCKp low-level width	t_{KL1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 - 36		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 - 100		ns

Note Use it with $V_{DD} \geq V_b$.

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** are listed on the page after the next page.)

(6) Communication at different potential (1.8 V, 2.5 V) (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	354		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	958		ns
SIp hold time (from SCKp↑) Note 1	t_{KSI1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	38		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	38		ns
Delay time from SCKp↓ to SOp output Note 1	t_{KSO1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		390	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ		966	ns
SIp setup time (to SCKp↓) Note 2	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	88		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	220		ns
SIp hold time (from SCKp↓) Note 2	t_{KSI1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	38		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	38		ns
Delay time from SCKp↑ to SOp output Note 2	t_{KSO1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		50	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ		50	ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

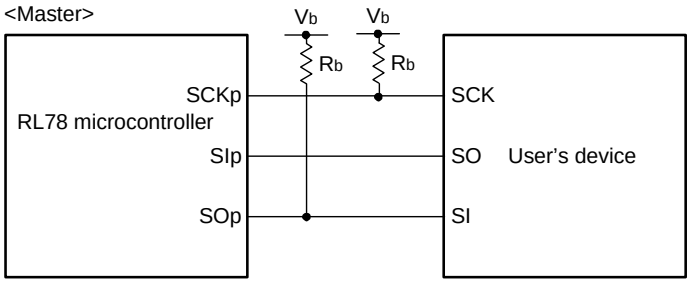
Note 2. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. Use it with $V_{DD} \geq V_b$.

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** are listed on the next page.)

Simplified SPI (CSI) mode connection diagram (during communication at different potential)

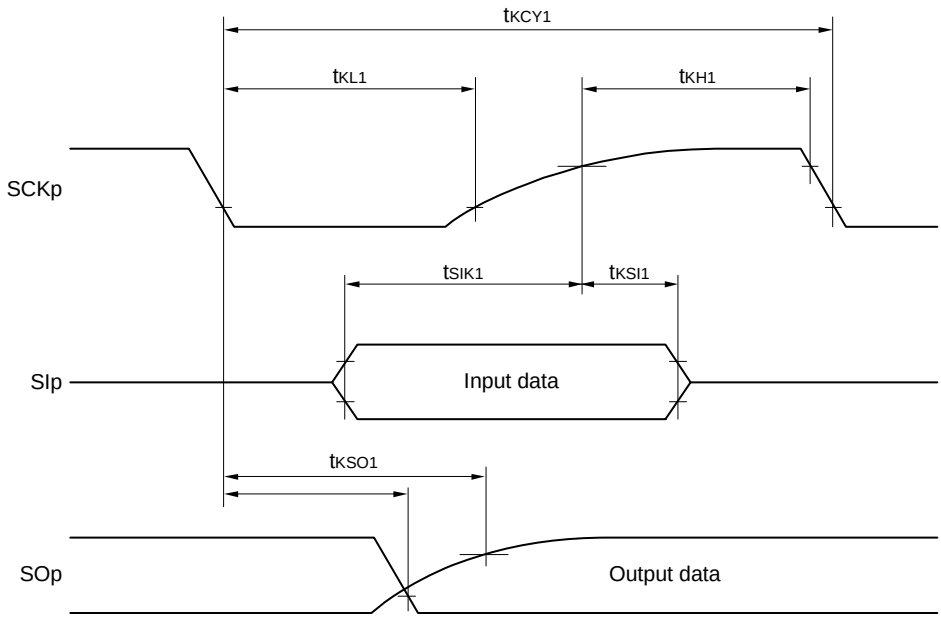


Remark 1. R_b[Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

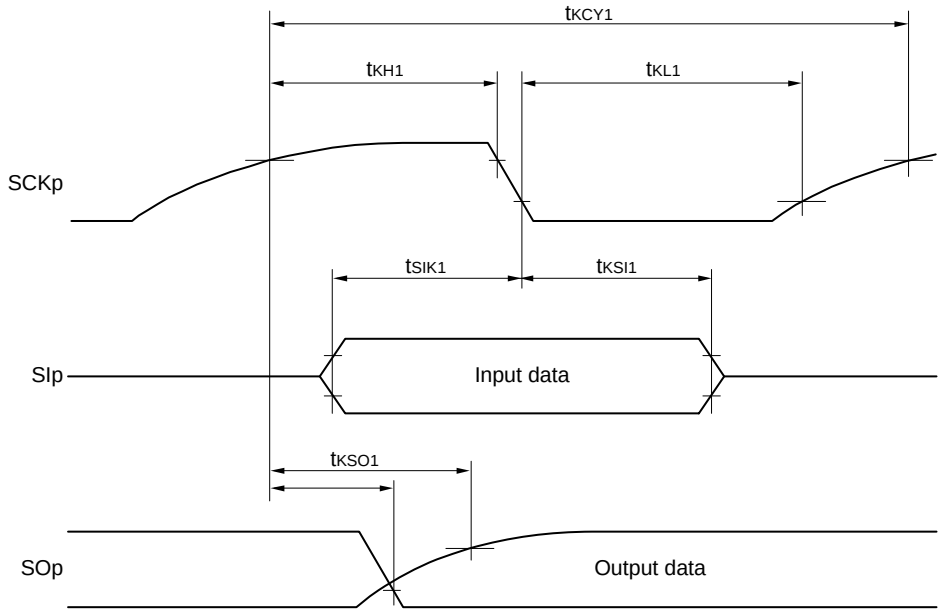
Remark 2. p: CSI number (p = 00, 10, 20, 30), m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), g: PIM and POM number (g = 0 to 3)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00))

Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark p: CSI number (p = 00, 10, 20, 30), m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3),
g: PIM and POM number (g = 0 to 3)

(7) Communication at different potential (1.8 V, 2.5 V) (Simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
SCKp cycle time Note 1	t_{KCY2}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V	20 MHz < f_{MCK} ≤ 24 MHz	32/f_{MCK}		ns
			16 MHz < f_{MCK} ≤ 20 MHz	28/f_{MCK}		ns
			8 MHz < f_{MCK} ≤ 16 MHz	24/f_{MCK}		ns
			4 MHz < f_{MCK} ≤ 8 MHz	16/f_{MCK}		ns
			f_{MCK} ≤ 4 MHz	12/f_{MCK}		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2	20 MHz < f_{MCK} ≤ 24 MHz	72/f_{MCK}		ns
			16 MHz < f_{MCK} ≤ 20 MHz	64/f_{MCK}		ns
			8 MHz < f_{MCK} ≤ 16 MHz	52/f_{MCK}		ns
			4 MHz < f_{MCK} ≤ 8 MHz	32/f_{MCK}		ns
			f_{MCK} ≤ 4 MHz	20/f_{MCK}		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V		t_{KCY2}/2 - 36		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2		t_{KCY2}/2 - 100		ns
SIp setup time (to SCKp↑) Note 3	t_{SIK2}	2.7 V ≤ V_{DD} ≤ 3.6 V		1/f_{MCK} + 40		ns
		2.4 V ≤ V_{DD} < 3.3 V		1/f_{MCK} + 60		ns
SIp hold time (from SCKp↑) Note 4	t_{KSI2}			1/f_{MCK} + 62		ns
Delay time from SCKp↓ to SOp output Note 5	t_{KSO2}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V C_b = 30 pF, R_b = 2.7 kΩ			2/f_{MCK} + 428	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2 C_b = 30 pF, R_b = 5.5 kΩ			2/f_{MCK} + 1146	ns

Note 1. Transfer rate in the SNOOZE mode: MAX. 1 Mbps

Note 2. Use it with $V_{DD} \geq V_b$.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes “to SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

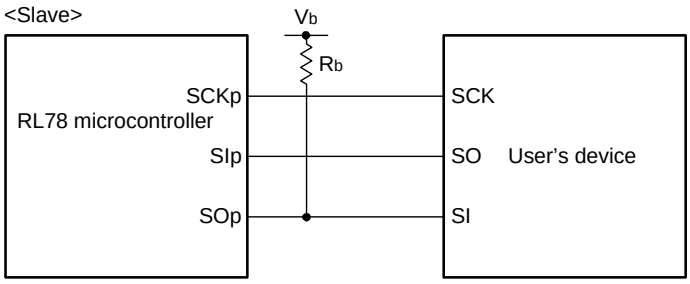
Note 4. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes “from SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 5. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes “from SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Caution Select the TTL input buffer for the SIp pin and SCKp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

(**Remarks** are listed on the next page.)

Simplified SPI (CSI) mode connection diagram (during communication at different potential)

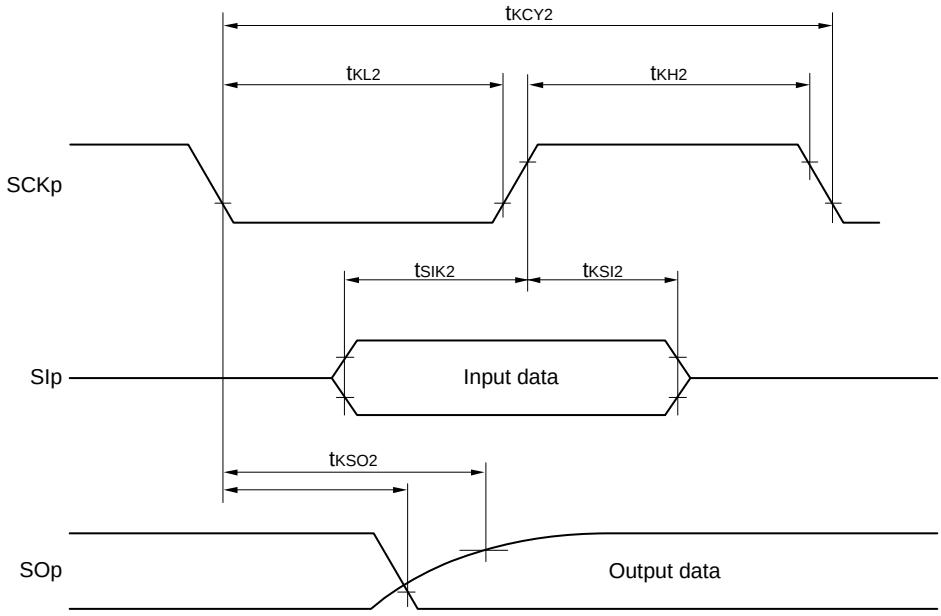


Remark 1. R_b[Ω]: Communication line (SOp) pull-up resistance, C_b[F]: Communication line (SOp) load capacitance, V_b[V]: Communication line voltage

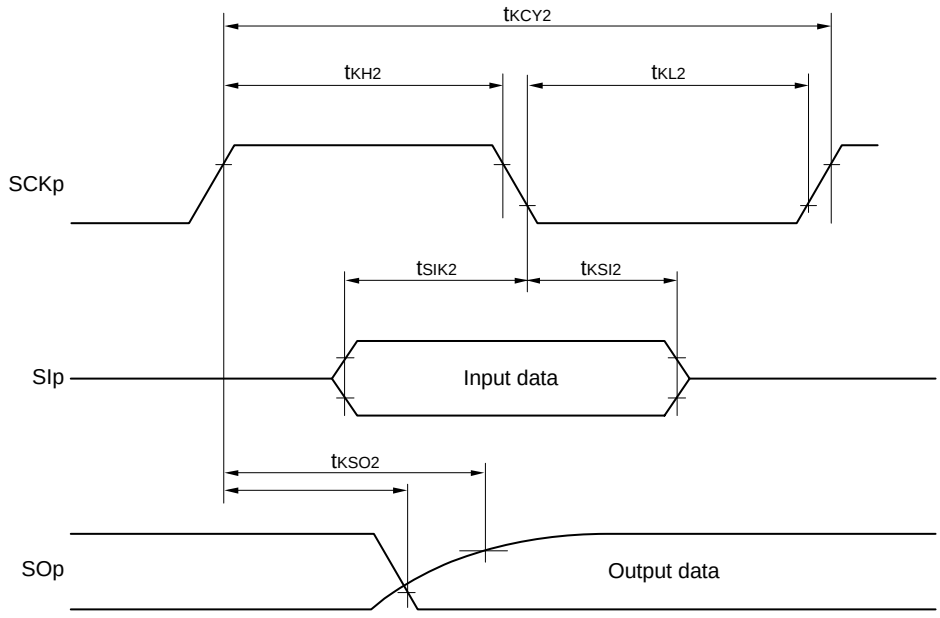
Remark 2. p: CSI number (p = 00, 10, 20, 30), m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), g: PIM and POM number (g = 0 to 3)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 02, 10, 12))

Simplified SPI (CSI) mode serial transfer timing (slave mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (slave mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark p: CSI number (p = 00, 10, 20, 30), m: Unit number (m = 0, 1),
n: Channel number (n = 0 to 3), g: PIM and POM number (g = 0 to 3)

(8) Communication at different potential (1.8 V, 2.5 V) (simplified I^2C mode)

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
SCLr clock frequency	f_{SCL}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ		400 Note 1	kHz
		2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ		100 Note 1	kHz
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ		100 Note 1	kHz
Hold time when SCLr = “L”	t_{LOW}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	1200		ns
		2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b <2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	4600		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ	4650		ns
Hold time when SCLr = “H”	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	500		ns
		2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	2400		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ	1830		ns
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	1/f_{MCK} + 340 Note 3		ns
		2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	1/f_{MCK} + 760 Note 3		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ	1/f_{MCK} + 570 Note 3		ns
Data hold time (transmission)	$t_{HD:DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	0	770	ns
		2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b < 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	0	1420	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ	0	1215	ns

Note 1. The value must be equal to or less than f_{MCK}/4.

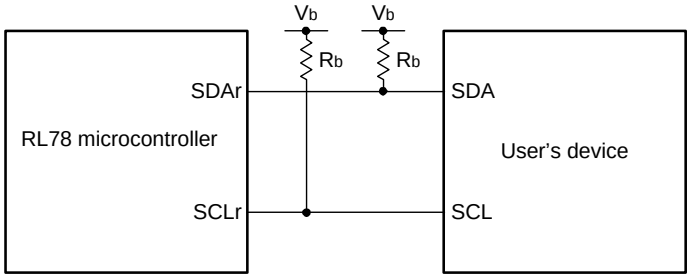
Note 2. Use it with V_{DD} ≥ V_b.

Note 3. Set the f_{MCK} value to keep the hold time of SCLr = “L” and SCLr = “H”.

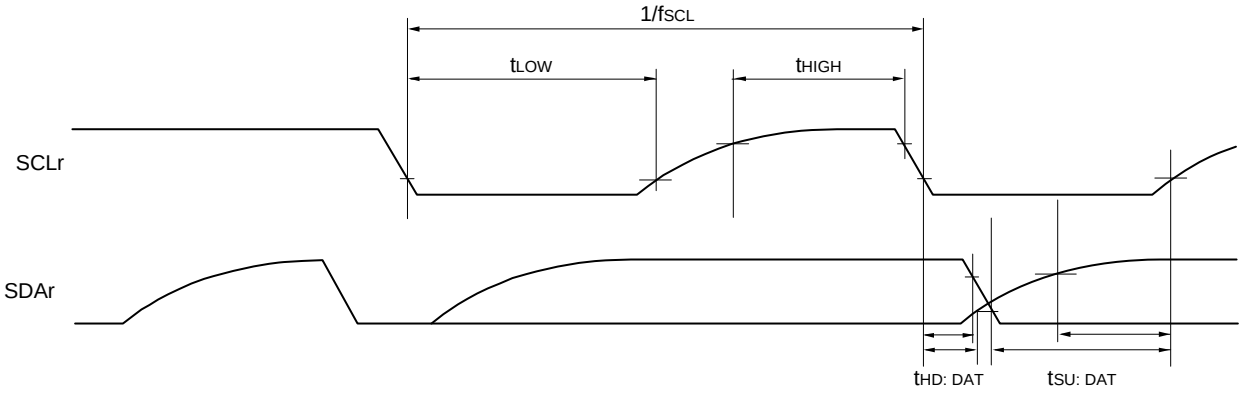
Caution Select the TTL input buffer and the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin and the N-ch open drain output (V_{DD} tolerance) mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

(**Remarks** are listed on the next page.)

Simplified I²C mode connection diagram (during communication at different potential)



Simplified I²C mode serial transfer timing (during communication at different potential)



Remark 1. R_b[Ω]: Communication line (SDAr, SCLr) pull-up resistance, C_b[F]: Communication line (SDAr, SCLr) load capacitance, V_b[V]: Communication line voltage

Remark 2. r: IIC number (r = 00, 10, 20, 30), g: PIM, POM number (g = 0 to 3)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 02, 10, 12)

35.5.2 Serial interface IICA

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode				Unit
			Standard mode		Fast mode		
			MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Fast mode: f_{CLK} ≥ 3.5 MHz	—	—	0	400	kHz
		Standard mode: f_{CLK} ≥ 1 MHz	0	100	—	—	kHz
Setup time of restart condition	$t_{SU: STA}$		4.7		0.6		μs
Hold time Note 1	$t_{HD: STA}$		4.0		0.6		μs
Hold time when SCLA0 = “L”	t_{LOW}		4.7		1.3		μs
Hold time when SCLA0 = “H”	t_{HIGH}		4.0		0.6		μs
Data setup time (reception)	$t_{SU: DAT}$		250		100		ns
Data hold time (transmission) Note 2	$t_{HD: DAT}$		0	3.45	0	0.9	μs
Setup time of stop condition	$t_{SU: STO}$		4.0		0.6		μs
Bus-free time	t_{BUF}		4.7		1.3		μs

Note 1. The first clock pulse is generated after this period when the start/restart condition is detected.

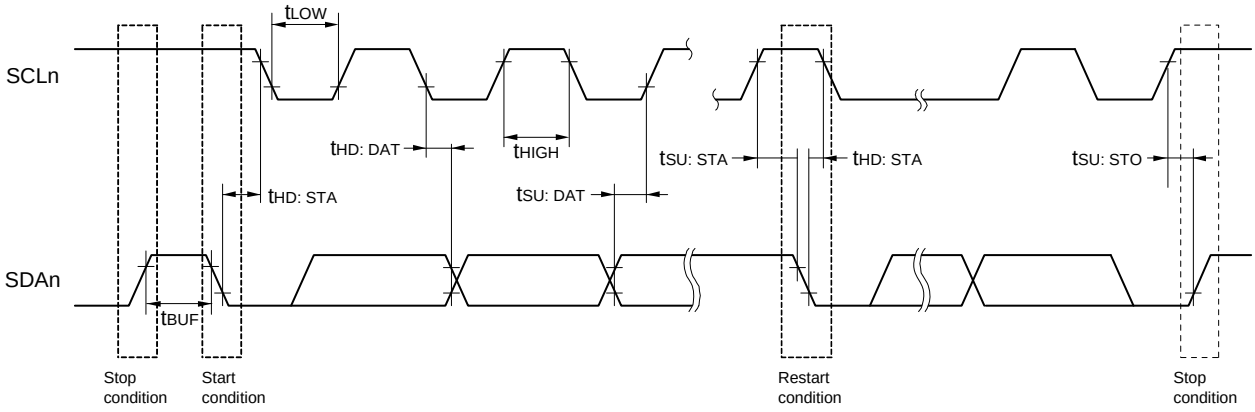
Note 2. The maximum value (MAX.) of $t_{HD:DAT}$ is during normal transfer and a clock stretch state is inserted in the ACK (acknowledge) timing.

Remark The maximum value of C_b (communication line capacitance) and the value of R_b (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

Fast mode: C_b = 320 pF, R_b = 1.1 kΩ

IICA serial transfer timing



35.5.3 USB

(1) Electrical specifications

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
UREGC	UREGC output voltage characteristic	UREGC	UVBUS = 4.0 to 5.5 V, PXXCON = VDDUSBE = 1	3.0	3.3	3.6	V
UVBUS	UVBUS input voltage characteristic	UVBUS	Function	4.35 (4.02 Note)	5.00	5.25	V

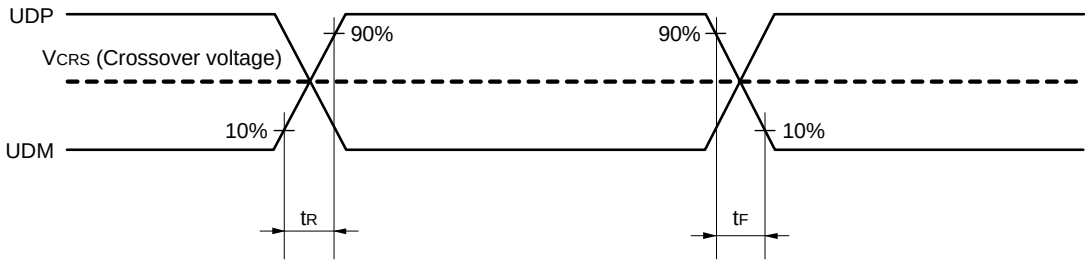
Note Value of instantaneous voltage

(T_A = -40 to +105°C, 4.35 V ≤ UVBUS ≤ 5.25 V, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter			Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input characteristic (FS/LS receiver)	Input voltage		VIH		2.0			V
			VIL				0.8	V
	Difference input sensitivity		VDI	| UDP voltage - UDM voltage |	0.2			V
	Difference common mode range		VCM		0.8		2.5	V
Output characteristic (FS driver)	Output voltage		VOH	IOH = -200 µA	2.8		3.6	V
			VOL	IOL = 2 mA	0		0.3	V
	Transition time	Rising	tFR	Rising: From 10% to 90% of amplitude, Falling: From 90% to 10% of amplitude, CL = 50 pF	4		20	ns
		Falling	tFF		4		20	ns
	Matching (TFR/TFF)		VFRFM		90		111.1	%
	Crossover voltage		VFCRS		1.3		2.0	V
	Output Impedance		ZDRV		28		44	Ω
Output characteristic (LS driver)	Output voltage		VOH		2.8		3.6	V
			VOL		0		0.3	V
	Transition time	Rising	tLR	Rising: From 10% to 90% of amplitude, Falling: From 90% to 10% of amplitude, CL = 250 pF to 750 pF The UDP and UDM pins are individually pulled down via 15 kΩ	75		300	ns
		Falling	tLF		75		300	ns
	Matching (TFR/TFF) Note		VLTFM		80		125	%
	Crossover voltage Note		VLCRS		1.3		2.0	V
Pull-up, Pull-down	Pull-down resistor		RPD		14.25		24.80	kΩ
	Pull-up resistor	Idle	RPUI		0.9		1.575	kΩ
		Reception	RPUA		1.425		3.09	kΩ
UVBUS	UVBUS pull-down resistor		RVBUS	UVBUS voltage = 5.5 V		1000		kΩ
	UVBUS input voltage		VIH		3.20			V
			VIL				0.8	V

Note Excludes the first signal transition from the idle state.

Timing of UDP and UDM



(2) BC standard

(T_A = -40 to +105°C, 4.35 V ≤ UV_{BUS} ≤ 5.25 V, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
USB standard BC1.2	UDP sink current	I_{DP_SINK}		25	100	175	μA
	UDM sink current	I_{DM_SINK}		25	100	175	μA
	DCD source current	I_{DP_SRC}		7	10	13	μA
	Data detection voltage	V_{DAT_REF}		0.25	0.325	0.4	V
	UDP source voltage	V_{DP_SRC}	Output current 250 μA	0.5	0.6	0.7	V
	UDM source voltage	V_{DM_SRC}	Output current 250 μA	0.5	0.6	0.7	V

(3) BC option standard

(T_A = -40 to +105°C, 4.35 V ≤ UV_{BUS} ≤ 5.25 V, 2.4 V ≤ V_{DD} ≤ 3.6, V_{SS} = 0 V)

Parameter			Symbol	Conditions	MIN.	TYP.	MAX.	Unit
UDP/UDM input reference voltage (UV_{BUS} divider ratio) (Function)	VDSELi [3: 0] (i = 0, 1)	0000	V_{DDET0}		27	32	37	%UVBUS
		0001	V_{DDET1}		29	34	39	%UVBUS
		0010	V_{DDET2}		32	37	42	%UVBUS
		0011	V_{DDET3}		35	40	45	%UVBUS
		0100	V_{DDET4}		38	43	48	%UVBUS
		0101	V_{DDET5}		41	46	51	%UVBUS
		0110	V_{DDET6}		44	49	54	%UVBUS
		0111	V_{DDET7}		47	52	57	%UVBUS
		1000	V_{DDET8}		51	56	61	%UVBUS
		1001	V_{DDET9}		55	60	65	%UVBUS
		1010	V_{DDET10}		59	64	69	%UVBUS
		1011	V_{DDET11}		63	68	73	%UVBUS
		1100	V_{DDET12}		67	72	73	%UVBUS
		1101	V_{DDET13}		71	76	81	%UVBUS
		1110	V_{DDET14}		75	80	85	%UVBUS
		1111	V_{DDET15}		79	84	89	%UVBUS

35.6 Analog Characteristics

35.6.1 A/D converter characteristics

Classification of A/D converter characteristics

Reference Voltage / Input Channel	Reference voltage (+) = AV_{REFP} Reference voltage (-) = AV_{REFM}	Reference voltage (+) = AV_{DD} Reference voltage (-) = AV_{SS}	Reference voltage (+) = Internal reference voltage Reference voltage (-) = AV_{SS}
High-accuracy channel; ANI0 to ANI6 (input buffer power supply: AV_{DD})	Refer to **35.6.1 (1)**.	Refer to **35.6.1 (2)**.	Refer to **35.6.1 (5)**.
Standard channel; ANI16 to ANI21 (input buffer power supply: V_{DD})	Refer to **35.6.1 (3)**.	Refer to **35.6.1 (4)**.	
Internal reference voltage, Temperature sensor output voltage	Refer to **35.6.1 (3)**.	Refer to **35.6.1 (4)**.	—

(1) When reference voltage (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (-) = AV_{REFM}/ANI1 (ADREFM = 1), conversion target: ANI2 to ANI6

(T_A = -40 to +105°C, 2.4 V ≤ AV_{REFP} ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V, AV_{SS} = 0 V, Reference voltage (+) = AV_{REFP}, Reference voltage (-) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}		2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V	8		12	bit
Overall error Note	AINL	12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±6.0	LSB
Conversion time	t_{CONV}	ADTYP = 0, 12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V	3.375			µs
Zero-scale error Note	E_{ZS}	12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±4.5	LSB
Full-scale error Note	E_{FS}	12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±4.5	LSB
Integral linearity error Note	ILE	12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±2.0	LSB
Differential linearity error Note	DLE	12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±1.5	LSB
Analog input voltage	V_{AIN}			0		AV_{REFP}	V

Note Excludes quantization error (±1/2 LSB).

Caution Always use AV_{DD} pin with the same potential as the V_{DD} pin.

(2) When reference voltage (+) = AV_{DD} (ADREFP1 = 0, ADREFP0 = 0), reference voltage (-) = AV_{SS} (ADREFM = 0), conversion target: ANI0 to ANI6

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V, AV_{SS} = 0 V, Reference voltage (+) = AV_{DD}, Reference voltage (-) = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES		2.4 V ≤ AV_{DD} ≤ 3.6 V	8		12	bit
Overall error Note	AINL	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±7.5	LSB
Conversion time	t_{CONV}	ADTYP = 0, 12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V	3.375			µs
Zero-scale error Note	E_{ZS}	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±6.0	LSB
Full-scale error Note	E_{FS}	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±6.0	LSB
Integral linearity error Note	ILE	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±3.0	LSB
Differential linearity error Note	DLE	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±2.0	LSB
Analog input voltage	V_{AIN}			0		AV_{DD}	V

Note Excludes quantization error (±1/2 LSB).

Caution Always use AV_{DD} pin with the same potential as the V_{DD} pin.

(3) When reference voltage (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (-) = AV_{REFM}/ANI1 (ADREFM = 1), conversion target ANI16 to ANI21, internal reference voltage, temperature sensor output voltage

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, 2.4 V ≤ AV_{REFP} ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V, AV_{SS} = 0 V, Reference voltage (+) = AV_{REFP}, Reference voltage (-) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES		2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V	8		12	bit
Overall error Note 1	AINL	12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±7.0	LSB
Conversion time	t_{CONV}	ADTYP = 0, 12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V	4.125			µs
Zero-scale error Note 1	EZS	12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±5.0	LSB
Full-scale error Note 1	EFS	12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±5.0	LSB
Integral linearity error Note 1	ILE	12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±3.0	LSB
Differential linearity error Note 1	DLE	12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±2.0	LSB
Analog input voltage	V_{AIN}			0		AV_{REFP}	V
		Internal reference voltage (2.4 V ≤ V_{DD} ≤ 3.6 V, HS (high-speed main) mode)		V_{BGR} Note 2			
		Temperature sensor output voltage (2.4 V ≤ V_{DD} ≤ 3.6 V, HS (high-speed main) mode)		V_{TMP25} Note 2			

Note 1. Excludes quantization error (±1/2 LSB).

Note 2. Refer to **35.6.2 Temperature sensor, internal reference voltage output characteristics**.

Caution Always use AV_{DD} pin with the same potential as the V_{DD} pin.

(4) When reference voltage (+) = AV_{DD} (ADREFP1 = 0, ADREFP0 = 0), reference voltage (-) = AV_{SS} (ADREFM = 0), conversion target: ANI16 to ANI21, internal reference voltage, temperature sensor output voltage

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V, AV_{SS} = 0 V, Reference voltage (+) = AV_{DD}, Reference voltage (-) = AV_{SS} = 0)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES		2.4 V ≤ AV_{DD} ≤ 3.6 V	8		12	bit
Overall error Note 1	AINL	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±8.5	LSB
Conversion time	tCONV	ADTYP = 0, 12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V	4.125			µs
Zero-scale error Note 1	EZS	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±8.0	LSB
Full-scale error Note 1	EFS	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±8.0	LSB
Integral linearity error Note 1	ILE	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±3.5	LSB
Differential linearity error Note 1	DLE	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±2.5	LSB
Analog input voltage	VAIN			0		AV_{DD}	V
		Internal reference voltage (2.4 V ≤ V_{DD} ≤ 3.6 V, HS (high-speed main) mode)		V_{BGR} Note 2			
		Temperature sensor output voltage (2.4 V ≤ V_{DD} ≤ 3.6 V, HS (high-speed main) mode)		V_{TMP25} Note 2			

Note 1. Excludes quantization error (±1/2 LSB).

Note 2. Refer to **35.6.2 Temperature sensor, internal reference voltage output characteristics**.

Caution Always use AV_{DD} pin with the same potential as the V_{DD} pin.

(5) When reference voltage (+) = Internal reference voltage (1.45 V) (ADREFP1 = 1, ADREFP0 = 0), reference voltage (-) = AVSS (ADREFM = 0), conversion target: ANI0 to ANI6, ANI16 to ANI21

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, 2.4 V ≤ V_{DD}, 2.4 V ≤ AV_{DD} = V_{DD}, V_{SS} = 0 V, AV_{SS} = 0 V, Reference voltage (+) = internal reference voltage, Reference voltage (-) = AV_{SS} = 0 V, HS (high-speed main) mode)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES			8		bit
Conversion time	tCONV	8-bit resolution	16.0			µs
Zero-scale error Note	EZS	8-bit resolution			±4.0	LSB
Integral linearity error Note	ILE	8-bit resolution			±2.0	LSB
Differential linearity error Note	DLE	8-bit resolution			±2.5	LSB
Reference voltage (+)	$AV_{REF(+)}$	= Internal reference voltage (V_{BGR})	1.38	1.45	1.5	V
Analog input voltage	V_{AIN}		0		V_{BGR}	V

Note Excludes quantization error (±1/2 LSB).

Caution Always use AV_{DD} pin with the same potential as the V_{DD} pin.

35.6.2 Temperature sensor, internal reference voltage output characteristics

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 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 output voltage that depends on the temperature		-3.6		mV/°C
Operation stabilization wait time	tAMP		10			µs

35.6.3 D/A converter characteristics

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					8	bit
Overall error	AINL	Rload = 4 MΩ	2.4 V ≤ V_{DD} ≤ 3.6 V			±2.5	LSB
		Rload = 8 MΩ	2.4 V ≤ V_{DD} ≤ 3.6 V			±2.5	LSB
Settling time	tSET	Cload = 20 pF	2.7 V ≤ V_{DD} ≤ 3.6 V			3	µs
			2.4 V ≤ V_{DD} < 2.7 V			6	µs

35.6.4 Comparator

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage range	Ivref			0		V_{DD} - 1.4	V
	Ivcmp			-0.3		V_{DD} + 0.3	V
Output delay	td	V_{DD} = 3.0 V Input slew rate > 50 mV/μs	High-speed comparator mode, standard mode			1.2	μs
			High-speed comparator mode, window mode			2.0	μs
			Low-speed comparator mode, standard mode		3	5.0	μs
High-electric-potential judgment voltage	VTW+	High-speed comparator mode, window mode			0.76 V_{DD}		V
Low-electric-potential judgment voltage	VTW-	High-speed comparator mode, window mode			0.24 V_{DD}		V
Operation stabilization wait time	t_{CMP}			100			μs
Internal reference voltage Note	V_{BGR}	2.4 V ≤ V_{DD} ≤ 3.6 V, HS (high-speed main) mode		1.38	1.45	1.50	V

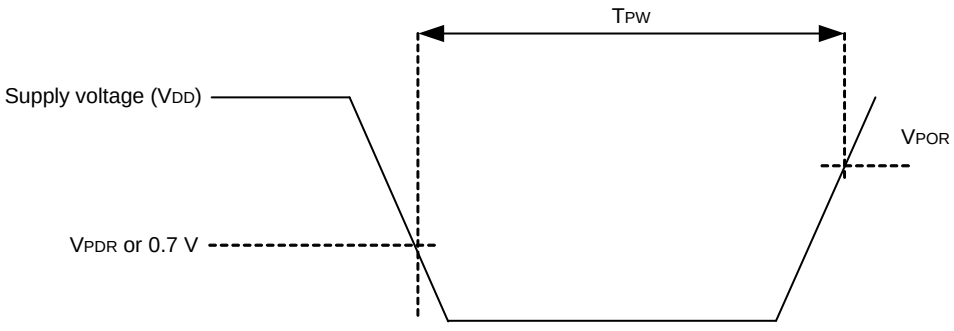
Note Not usable in sub-clock operation or STOP mode.

35.6.5 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 Note	1.44	1.50	1.56	V
Minimum pulse width	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).



35.6.6 LVD circuit characteristics

(T_A = -40 to +105°C, $V_{PDR} \leq V_{DD} \leq 3.6$ V $\leq V_{SS}$ = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVD2}	Power supply rise time	3.01	3.13	3.25	V
			Power supply fall time	2.94	3.06	3.18	V
		VLVD3	Power supply rise time	2.90	3.02	3.14	V
			Power supply fall time	2.85	2.96	3.07	V
		V_{LVD4}	Power supply rise time	2.81	2.92	3.03	V
			Power supply fall time	2.75	2.86	2.97	V
		V_{LVD5}	Power supply rise time	2.71	2.81	2.92	V
			Power supply fall time	2.64	2.75	2.86	V
		V_{LVD6}	Power supply rise time	2.61	2.71	2.81	V
			Power supply fall time	2.55	2.65	2.75	V
		V_{LVD7}	Power supply rise time	2.51	2.61	2.71	V
			Power supply fall time	2.45	2.55	2.65	V
Minimum pulse width		t_{LW}		300			μs
Detection delay time						300	μs

Caution Set the detection voltage (V_{LVD}) to be within the operating voltage range. The operating voltage range depends on the setting of the user option byte (000C2H/010C2H). The following shows the operating voltage range.
HS (high-speed main) mode: V_{DD} = 2.7 to 3.6 V at 1 MHz to 24 MHz
V_{DD} = 2.4 to 3.6 V at 1 MHz to 16 MHz

LVD Detection Voltage of Interrupt & Reset Mode

(T_A = -40 to +105°C, $V_{PDR} \leq V_{DD} \leq$ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Interrupt and reset mode	V_{LVDD0}	V_{POC0}, V_{POC1}, V_{POC2} = 0, 1, 1, falling reset voltage: 2.7 V			2.64	2.75	2.86	V
	V_{LVDD1}		LVIS0, LVIS1 = 1, 0	Rising release reset voltage	2.81	2.92	3.03	V
				Falling interrupt voltage	2.75	2.86	2.97	V
	V_{LVDD2}		LVIS0, LVIS1 = 0, 1	Rising release reset voltage	2.90	3.02	3.14	V
				Falling interrupt voltage	2.85	2.96	3.07	V

35.7 Power supply voltage rising slope characteristics

(T_A = -40 to +105°C, V_{SS} = 0 V)

Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Power supply voltage rising slope	SV_{DD}			54	V/ms

Caution Make sure to keep the internal reset state by the LVD circuit or an external reset until V_{DD} reaches the operating voltage range shown in 35.4 AC Characteristics.

35.8 LCD Characteristics

35.8.1 Resistance division method

(1) Static display mode

(T_A = -40 to +105°C, V_{L4} (MIN.) ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	V_{L4}		2.0		V_{DD}	V

(2) 1/2 bias method, 1/4 bias method

(T_A = -40 to +105°C, V_{L4} (MIN.) ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	V_{L4}		2.7		V_{DD}	V

(3) 1/3 bias method

(T_A = -40 to +105°C, V_{L4} (MIN.) ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	V_{L4}		2.5		V_{DD}	V

35.8.2 Internal voltage boosting method

(1) 1/3 bias method

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
LCD output voltage variation range	V_{L1}	C1 to C4 Note 1 = 0.47 μF Note 2	VLCD = 04H	0.90	1.00	1.08	V
			VLCD = 05H	0.95	1.05	1.13	V
			VLCD = 06H	1.00	1.10	1.18	V
			VLCD = 07H	1.05	1.15	1.23	V
			VLCD = 08H	1.10	1.20	1.28	V
			VLCD = 09H	1.15	1.25	1.33	V
			VLCD = 0AH	1.20	1.30	1.38	V
			VLCD = 0BH	1.25	1.35	1.43	V
			VLCD = 0CH	1.30	1.40	1.48	V
			VLCD = 0DH	1.35	1.45	1.53	V
			VLCD = 0EH	1.40	1.50	1.58	V
			VLCD = 0FH	1.45	1.55	1.63	V
			VLCD = 10H	1.50	1.60	1.68	V
			VLCD = 11H	1.55	1.65	1.73	V
			VLCD = 12H	1.60	1.70	1.78	V
			VLCD = 13H	1.65	1.75	1.83	V
Doubler output voltage	V_{L2}	C1 to C4 Note 1 = 0.47 μF		2 V_{L1} - 0.1	2 V_{L1}	2 V_{L1}	V
Tripler output voltage	V_{L3}	C1 to C4 Note 1 = 0.47 μF		3 V_{L1} - 0.15	3 V_{L1}	3 V_{L1}	V
Reference voltage setup time Note 2	t_{VWAIT1}			5			ms
Voltage boost wait time Note 3	t_{VWAIT2}	C1 to C4 Note 1 = 0.47μF		500			ms

Note 1. This is a capacitor that is connected between voltage pins used to drive the LCD.
C1: A capacitor connected between CAPH and CAPL
C2: A capacitor connected between V_{L1} and GND
C3: A capacitor connected between V_{L2} and GND
C4: A capacitor connected between V_{L4} and GND
C1 = C2 = C3 = C4 = 0.47 μF±30%

Note 2. This is the time required to wait from when the reference voltage is specified by using the VLCD register (or when the internal voltage boosting method is selected (by setting the MDSET1 and MDSET0 bits of the LCDM0 register to 01B) if the default value reference voltage is used) until voltage boosting starts (VLCON = 1).

Note 3. This is the wait time from when voltage boosting is started (VLCON = 1) until display is enabled (LCDON = 1).

(2) 1/4 bias method

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
LCD output voltage variation range	V_{L1}	C1 to C4 Note 1 = 0.47 μF Note 2	VLCD = 04H	0.90	1.00	1.08	V
			VLCD = 05H	0.95	1.05	1.13	V
			VLCD = 06H	1.00	1.10	1.18	V
			VLCD = 07H	1.05	1.15	1.23	V
			VLCD = 08H	1.10	1.20	1.28	V
			VLCD = 09H	1.15	1.25	1.33	V
			VLCD = 0AH	1.20	1.30	1.38	V
Doubler output voltage	V_{L2}	C1 to C4 Note 1 = 0.47 μF		2 V_{L1} - 0.08	2 V_{L1}	2 V_{L1}	V
Tripler output voltage	V_{L3}	C1 to C4 Note 1 = 0.47 μF		3 V_{L1} - 0.12	3 V_{L1}	3 V_{L1}	V
Quadruply output voltage	V_{L4}	C1 to C5 Note 1 = 0.47 μF		4 V_{L1} - 0.16	4 V_{L1}	4 V_{L1}	V
Reference voltage setup time Note 2	t_{VWAIT1}			5			ms
Voltage boost wait time Note 3	t_{VWAIT2}	C1 to C5 Note 1 = 0.47μF		500			ms

Note 1. This is a capacitor that is connected between voltage pins used to drive the LCD.
C1: A capacitor connected between CAPH and CAPL
C2: A capacitor connected between V_{L1} and GND
C3: A capacitor connected between V_{L2} and GND
C4: A capacitor connected between V_{L3} and GND
C5: A capacitor connected between V_{L4} and GND
C1 = C2 = C3 = C4 = 0.47 μF±30%

Note 2. This is the time required to wait from when the reference voltage is specified by using the VLCD register (or when the internal voltage boosting method is selected (by setting the MDSET1 and MDSET0 bits of the LCDM0 register to 01B) if the default value reference voltage is used) until voltage boosting starts (VLCON = 1).

Note 3. This is the wait time from when voltage boosting is started (VLCON = 1) until display is enabled (LCDON = 1).

35.8.3 Capacitor split method

(1) 1/3 bias method

(T_A = -40 to +105°C, 2.4 V ≤ VDD ≤ 3.6 V, Vss = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{L4} voltage	V_{L4}	C1 to C4 = 0.47 μF Note 2		V_{DD}		V
V_{L2} voltage	V_{L2}	C1 to C4 = 0.47 μF Note 2	2/3 V_{L4} - 0.07	2/3 V_{L4}	2/3 V_{L4} + 0.07	V
V_{L1} voltage	V_{L1}	C1 to C4 = 0.47 μF Note 2	1/3 V_{L4} - 0.08	1/3 V_{L4}	1/3 V_{L4} + 0.08	V
Capacitor split wait time Note 1	t_{VWAIT}		100			ms

Note 1. This is the wait time from when voltage bucking is started (VLCON = 1) until display is enabled (LCDON = 1).

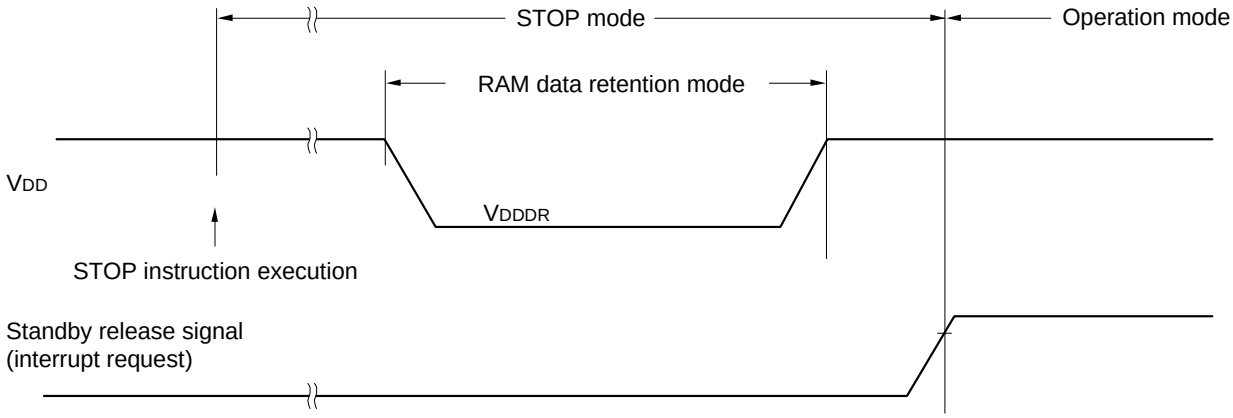
Note 2. This is a capacitor that is connected between voltage pins used to drive the LCD.
C1: A capacitor connected between CAPH and CAPL
C2: A capacitor connected between V_{L1} and GND
C3: A capacitor connected between V_{L2} and GND
C4: A capacitor connected between V_{L4} and GND
C1 = C2 = C3 = C4 = 0.47 μF±30%

35.9 RAM Data Retention Characteristics

(T_A = -40 to +105°C, Vss = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.44 Note		3.6	V

Note This depends on the POR detection voltage. For a falling voltage, data in RAM are retained until the voltage reaches the level that triggers a POR reset but not once it reaches the level at which a POR reset is generated.



35.10 Flash Memory Programming Characteristics

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
CPU/peripheral hardware clock frequency	f_{CLK}	2.4 V ≤ V_{DD} ≤ 3.6 V	1		24	MHz
Number of code flash rewrites Notes 1, 2, 3	C_{erwr}	Retained for 20 years T_A = 85°C Note 4	1,000			Times
Number of data flash rewrites Notes 1, 2, 3		Retained for 1 year T_A = 25°C		1,000,000		
		Retained for 5 years T_A = 85°C Note 4	100,000			
		Retained for 20 years T_A = 85°C Note 4	10,000			

Note 1. 1 erase + 1 write after the erase is regarded as 1 rewrite. The retaining years are until next rewrite after the rewrite.

Note 2. When using flash memory programmer and Renesas Electronics self programming library

Note 3. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

Note 4. This temperature is the average value at which data are retained.

35.11 Dedicated Flash Memory Programmer Communication (UART)

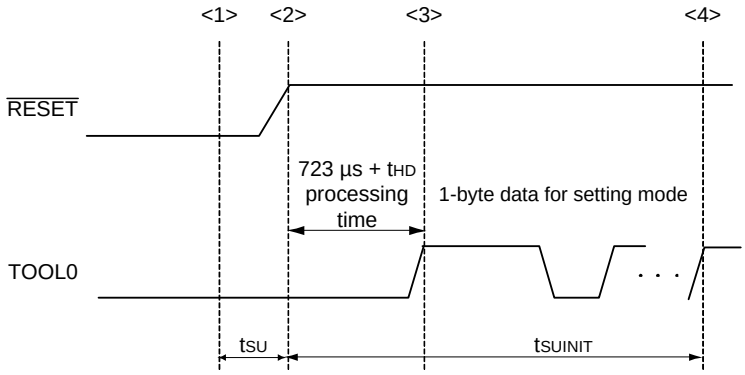
(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		During serial programming	115,200		1,000,000	bps

35.12 Timing of Entry to Flash Memory Programming Modes

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 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
Time to hold the TOOL0 pin at the low level after an external reset is released (excluding the processing time of the firmware to control the flash memory)	t_{HD}	POR and LVD reset must end before the external reset ends.	1			ms



<1> The low level is input to the TOOL0 pin.
<2> The external reset ends (POR and LVD reset must end before the external reset ends.).
<3> The TOOL0 pin is set to the high level.
<4> Setting of the flash memory programming mode by UART reception and complete the baud rate setting.

Remark t_{SUINIT}: The segment shows that it is necessary to finish specifying the initial communication settings within 100 ms from when the resets end.
t_{SU}: How long from when the TOOL0 pin is placed at the low level until a 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)

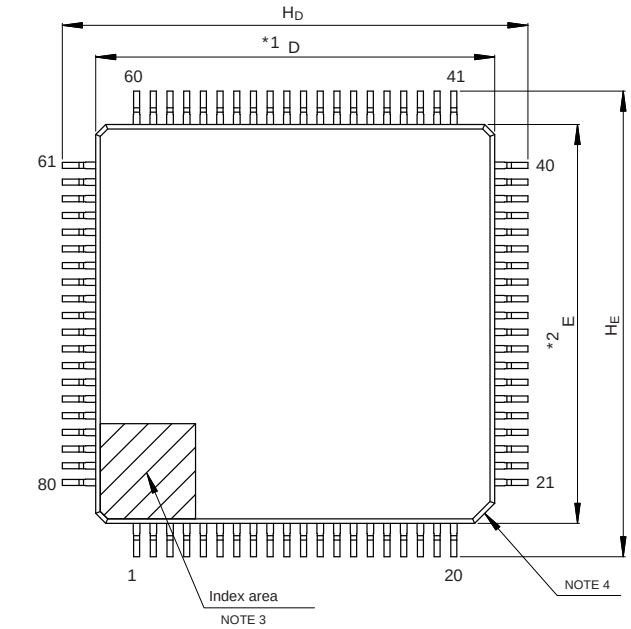
CHAPTER 36 PACKAGE DRAWINGS

36.1 80-pin products

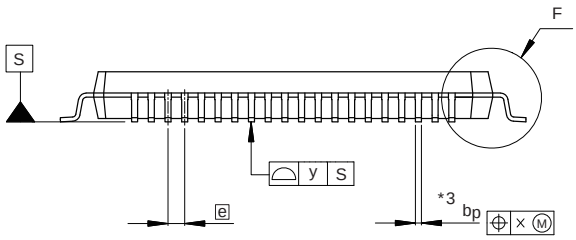
R5F110MEAFB, R5F110MFAFB, R5F110MGAFB, R5F110MHAFB, R5F110MJAFB
R5F111MEAFB, R5F111MFAFB, R5F111MGAFB, R5F111MHAFB, R5F111MJAFB
R5F110MEGFB, R5F110MFGFB, R5F110MGGFB, R5F110MHGFB, R5F110MJGFB
R5F111MEGFB, R5F111MFGFB, R5F111MGGFB, R5F111MHGFB, R5F111MJGFB

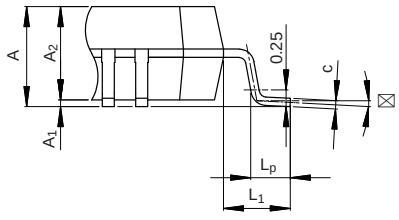
JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP80-12x12-0.50	PLQP0080KB-B	—	0.5

Unit: mm



NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

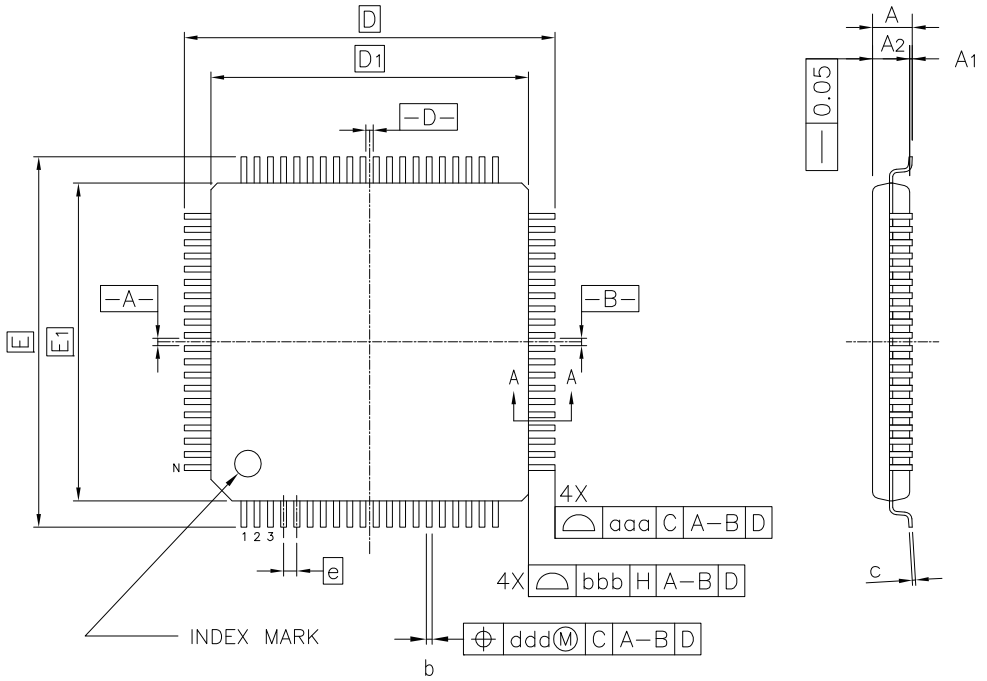


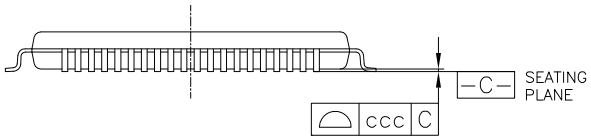


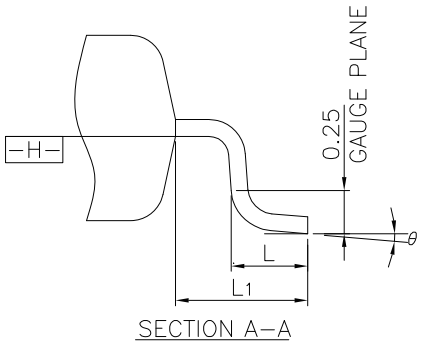
Detail F

Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	11.9	12.0	12.1
E	11.9	12.0	12.1
A_2	—	1.4	—
H_D	13.8	14.0	14.2
H_E	13.8	14.0	14.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.15	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LFQFP80-12x12-0.50	PLQP0080KJ-A	0.49





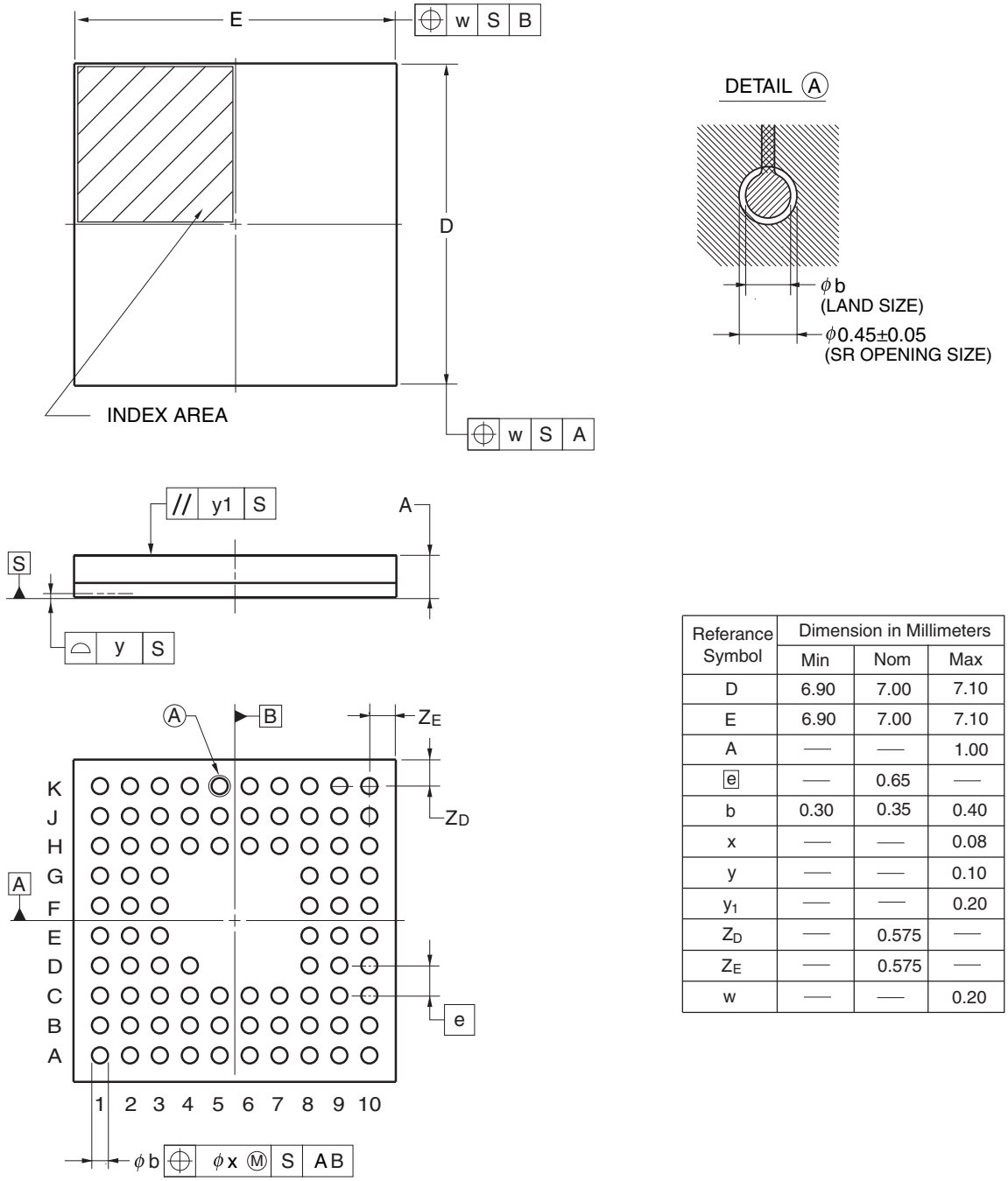


Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	–	–	1.60
A_1	0.05	–	0.15
A_2	1.35	1.40	1.45
D	–	14.00	–
D_1	–	12.00	–
E	–	14.00	–
E_1	–	12.00	–
N	–	80	–
e	–	0.50	–
b	0.17	0.22	0.27
c	0.09	–	0.20
☒	0°	3.5°	7°
L	0.45	0.60	0.75
L_1	–	1.00	–
aaa	–	–	0.20
bbb	–	–	0.20
ccc	–	–	0.08
ddd	–	–	0.08

36.2 85-pin products

R5F110NEALA, R5F110NFALA, R5F110NGALA, R5F110NHALA, R5F110NJALA
R5F111NEALA, R5F111NFALA, R5F111NGALA, R5F111NHALA, R5F111NJALA
R5F110NEGLA, R5F110NFGLA, R5F110NGGLA, R5F110NHGLA, R5F110NJGLA
R5F111NEGLA, R5F111NFGLA, R5F111NGGLA, R5F111NHGLA, R5F111NJGLA

JEITA Package code	RENESAS code	Previous code	MASS(TYP.)[g]
P-VFLGA85-7x7-0.65	PVLG0085JA-A	P85FC-65-BN4	0.1

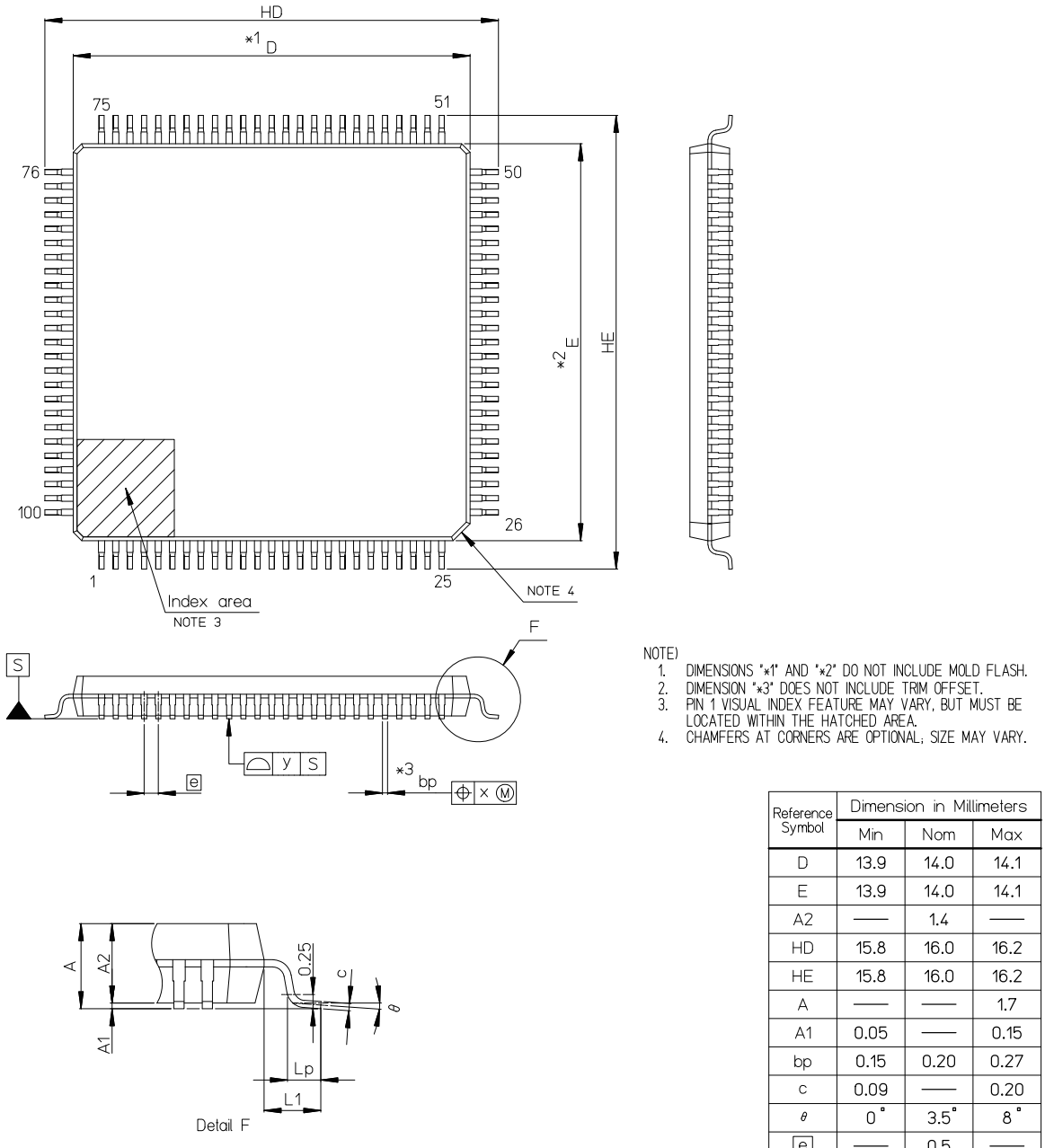


Referance Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	6.90	7.00	7.10
E	6.90	7.00	7.10
A	—	—	1.00
e	—	0.65	—
b	0.30	0.35	0.40
x	—	—	0.08
y	—	—	0.10
y_1	—	—	0.20
Z_D	—	0.575	—
Z_E	—	0.575	—
w	—	—	0.20

36.3 100-pin products

R5F110PEAFB, R5F110PFAFB, R5F110PGAFB, R5F110PHAFB, R5F110PJAFB
R5F111PEAFB, R5F111PFAFB, R5F111PGAFB, R5F111PHAFB, R5F111PJAFB
R5F110PEGFB, R5F110PFGFB, R5F110PGGFB, R5F110PHGFB, R5F110PJGFB
R5F111PEGFB, R5F111PFGFB, R5F111PGGFB, R5F111PHGFB, R5F111PJGFB

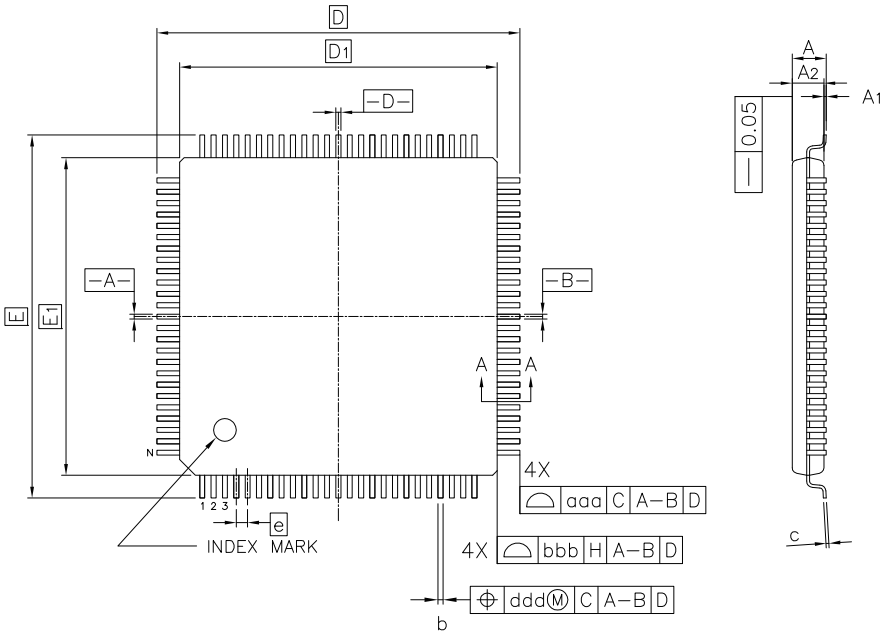
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP100-14x14-0.50	PLQP0100KB-B	——	0.6g

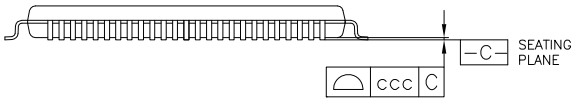


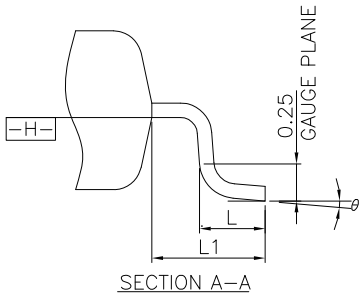
NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL; SIZE MAY VARY.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	13.9	14.0	14.1
E	13.9	14.0	14.1
A2	——	1.4	——
HD	15.8	16.0	16.2
HE	15.8	16.0	16.2
A	——	——	1.7
A1	0.05	——	0.15
bp	0.15	0.20	0.27
c	0.09	——	0.20
θ	0°	3.5°	8°
e	——	0.5	——
x	——	——	0.08
y	——	——	0.08
Lp	0.45	0.6	0.75
L1	——	1.0	——

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LFQFP100-14x14-0.50	PLQP0100KP-A	0.67







Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	–	–	1.60
A_1	0.05	–	0.15
A_2	1.35	1.40	1.45
D	–	16.00	–
D_1	–	14.00	–
E	–	16.00	–
E_1	–	14.00	–
N	–	100	–
e	–	0.50	–
b	0.17	0.22	0.27
c	0.09	–	0.20
☒	0°	3.5°	7°
L	0.45	0.60	0.75
L_1	–	1.00	–
aaa	–	–	0.20
bbb	–	–	0.20
ccc	–	–	0.08
ddd	–	–	0.08

APPENDIX A REVISION HISTORY

A.1 Major Revisions in This Edition

Page	Description	Classification
CHAPTER 1 OUTLINE		
p.5	Modification of description in Figure 1 - 1 Part Number, Memory Size, and Package of RL78/L1C	(d)
p.6	Modification of description in Table 1 - 1 List of Ordering Part Numbers	(d)
CHAPTER 2 PIN FUNCTIONS		
p.24	Modification of description of P60 in the table (2/3) of 2.1.1 80/85-pin products (with USB)	(a)
p.27	Modification of description of P60 in the table (2/2) of 2.1.2 80/85-pin products (without USB)	(a)
p.30	Modification of description of P60 in the table (2/3) of 2.1.3 100-pin products (with USB)	(a)
p.33	Modification of description of P60 in the table (2/3) of 2.1.4 100-pin products (without USB)	(a)
p.42	Modification of description in Table 2 - 3 Connection of Unused Pins	(c)
p.56	Modification of title and description in Figure 2 - 16 Pin Block Diagram of Pin Type 12-1-3	(a)
CHAPTER 11 WATCHDOG TIMER		
p.496	Addition of Note in Table 11 - 3 Setting of Overflow Time of Watchdog Timer	(c)
CHAPTER 15 SERIAL ARRAY UNIT		
p.663	Modification of Figure 15 - 74 Timing Chart of SNOOZE Mode Operation (once startup) (Type 1: DAPmn = 0, CKPmn = 0)	(c)
p.665	Modification of Figure 15 - 76 Timing Chart of SNOOZE Mode Operation (continuous startup) (Type 1: DAPmn = 0, CKPmn = 0)	(c)
CHAPTER 27 SAFETY FUNCTIONS		
p.1091	Modification of description in 27.1 Overview of Safety Functions	(c)
p.1096	Modification of description in 27.3.2 CRC operation function (general-purpose CRC)	(c)
p.1099	Modification of description in 27.3.4 RAM guard function	(c)
p.1100	Modification of description in 27.3.5 SFR guard function	(c)
CHAPTER 30 FLASH MEMORY		
p.1144	Addition of Caution 4 in 30.8.3 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.

(1/15)

Edition	Description	Chapter
Rev. 2.30	Modification of description in Figure 8 - 8 Format of Real-time clock control register 1 (RTCC1) (3/3)	CHAPTER 8 REAL-TIME CLOCK 2
	Addition of description in 8.4.3 Reading real-time clock 2	
	Addition of Figure 8 - 23 Procedure for Reading Real-time Clock 2 (When the Alarm Interrupt is in Use)	
	Addition of description in 8.4.4 Writing to real-time clock 2 counter	
	Modification of remark in Figure 8 - 24 Procedure for Writing Real-time Clock 2	
	Addition of Figure 8 - 25 Procedure for Writing Real-time Clock 2 (When the Alarm Interrupt is in Use)	
	Modification of description in 19.5.1 Setting DTC registers and vector table	CHAPTER 19 DATA TRANSFER CONTROLLER (DTC)
	Modification of notes in 34.3.2 Supply current characteristics (T_A = -40 to +85°C, 1.6 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) (1/2)	CHAPTER 34 ELECTRICAL SPECIFICATIONS (TA = -40 to +85°C)
	Modification of notes and remarks in 34.3.2 Supply current characteristics (T_A = -40 to +85°C, 1.6 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) (2/2)	
	Modification of notes in 35.3.2 Supply current characteristics (T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) (1/2)	CHAPTER 35 ELECTRICAL SPECIFICATIONS (G: INDUSTRIAL APPLICATIONS T_A = -40 to +105°C)
	Modification of notes and remark in 35.3.2 Supply current characteristics (T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) (2/2)	
	Modification of package drawing of PLQP0080KB-B in 36.1 80-pin products	CHAPTER 36 PACKAGE DRAWINGS
	Modification of package drawing of PLQP0100KB-B in 36.3 100-pin products	
Rev. 2.21	"3-Wire Serial I/O" and "3-wire serial" were modified to " Simplified SPI"	Throughout
	The module name for CSI was changed to Simplified SPI	
	"wait" for IIC was modified to "clock stretch"	
	Addition of Note 1 in 1 Features	CHAPTER 1 OUTLINE
	Modification of Table 1-1 List of Ordering Part Numbers	
	Addition of Note in 4.4.4 Handling different potential (1.8 V, 2.5 V) by using I/O buffers	CHAPTER 4 PORT FUNCTIONS
	Addition of Note in CHAPTER 15 SERIAL ARRAY UNIT	CHAPTER 15 SERIAL ARRAY UNIT
	Addition of package drawing in 31.1 80-pin Package	CHAPTER 36 PACKAGE DRAWINGS
	Addition of package drawing in 31.3 100-pin Package	

(2/15)

Edition	Description	Chapter
Rev. 2.20	PG-FP5 has been modified to PG-FP6, FL-PR5 has been modified to FL-PR6, and description of E2, E2 Lite, and E20 has been added.	Throughout
	Modification of Table 3 - 11 Extended SFR (2nd SFR) List (1/15)	CHAPTER 3 CPU ARCHITECTURE
	Modification of Table 4 - 1 Port Configuration	CHAPTER 4 PORT FUNCTIONS
	Modification of description in 4.3 Registers Controlling Port Function	
	Deletion of 4.3.9 Global analog input disable register (GAIDIS)	
	Modification of description in 5.4.4 Low-speed on-chip oscillator	CHAPTER 5 CLOCK GENERATOR
	Addition of note in Table 11 - 4 Setting Window Open Period of Watchdog Timer	CHAPTER 11 WATCHDOG TIMER
	Modification of Figure 15 - 103 Transmission Operation of LIN	CHAPTER 15 SERIAL ARRAY UNIT
	Modification of Figure 15 - 104 Flowchart for LIN Transmission	
	Modification of Figure 15 - 105 Reception Operation of LIN	
	Modification of Figure 15 - 106 Flowchart of LIN Reception	
	Addition of note 3 in Figure 29 - 1 Format of User Option Byte (000C0H/010C0H)	CHAPTER 29 OPTION BYTE
	Modification of Figure 30 - 7 Setting of Flash Memory Programming Mode	CHAPTER 30 FLASH MEMORY
	Modification of table and note 3 in 34.3.1 Pin characteristics	CHAPTER 34 ELECTRICAL SPECIFICATIONS (T_A = -40 to +85°C)
	Modification of figure in 34.12 Timing of Entry to Flash Memory Programming Modes	
	Modification of table in 35.1 Absolute Maximum Ratings	CHAPTER 35 ELECTRICAL SPECIFICATIONS (G: INDUSTRIAL APPLICATIONS T_A = -40 to +105°C)
	Modification of table and note 3 in 35.3.1 Pin characteristics	
	Modification of figure in 35.12 Timing of Entry to Flash Memory Programming Modes	

(3/15)

Edition	Description	Chapter
Rev. 2.10	Addition of product name (RL78/L1C) and description (Top View) in 1.3.1 80-pin products (with USB)	CHAPTER 1 OUTLINE
	Addition of product name (RL78/L1C) and description (Top View) in 1.3.2 80-pin products (without USB)	
	Addition of product name (RL78/L1C) and description (Top View) in 1.3.5 100-pin products (with USB)	
	Addition of product name (RL78/L1C) and description (Top View) in 1.3.6 100-pin products (without USB)	
	Modification of 1.6 Outline of Functions	
	Deletion of P130 in Table 2 - 1 Pin I/O Buffer Power Supplies	CHAPTER 2 PIN FUNCTIONS
	Modification of the header of the table in 2.1.1 80/85-pin products (with USB)	
	Modification of the header of the table in 2.1.2 80/85-pin products (without USB)	
	Modification of the header of the table in 2.1.3 100-pin products (with USB)	
	Modification of the header of the table in 2.1.4 100-pin products (without USB)	
	Modification of the order of Figures 2 - 7 to 2 – 15 and Figures 2 – 17 to 2 - 19 Pin Block Diagram	
	Addition of caution in Figure 2 - 7 Pin Block Diagram of Pin Type 7-3-4	
	Addition of caution in Figure 2 - 12 Pin Block Diagram of Pin Type 7-5-10	
	Addition of cautions 1 and 2 in Figure 2 - 14 Pin Block Diagram of Pin Type 8-3-4	
	Addition of cautions 1 and 2 in Figure 2 - 15 Pin Block Diagram of Pin Type 8-5-10	
	Deletion of 3.1 Overview	CHAPTER 3 CPU ARCHITECTURE
	Modification of vector table addresses in Tables 3 - 4 and 3 – 5 Vector Table	
	Modification of the address (00000H and 00001H) in 3.2.1 Control registers, (1) Program counter (PC)	
	Modification of the symbol for FFF14H and FFF15H in Table 3 - 7 SFR List (1/4)	
	Modification of the symbol for FFF48H and FFF49H in Table 3 - 8 SFR List (2/4)	
	Modification of Table 3 - 16 Extended SFR (2nd SFR) List (6/15), (7/15)	
	Addition of description in 5.1 (1) Main system clock	CHAPTER 5 CLOCK GENERATOR
	Addition of f_{PLL} and f_{USB} in remark of Figure 5 - 2 Block Diagram of Clock Generator (Products without USB)	
	Modification of caution 5 in 5.3.2 System clock control register (CKC)	
	Modification of description of RTCWEN bit and deletion of note in Figure 5 - 8 Format of Peripheral enable register 0 (PER0) (1/2)	
	Deletion of note in Figure 5 - 9 Format of Peripheral enable register 0 (PER0) (2/2)	
	Addition of cautions 2 and 3 in Figure 5 - 15 Format of PLL control register (DSCCTL)	
	Modification of caution 2 in 5.4.5 PLL (Phase Locked Loop)	
	Addition of notes 1 and 2, and note changed to note 3 in 5.6.1 Example of setting high-speed on-chip oscillator	
	Modification of description of <1> in 5.6.3 Example of setting XT1 oscillation clock	
	Addition of <1>, <7>, and <8>, addition of description to <3> and modification of <4> and <6>, addition of notes 1 and 3, and note changed to note 2 in 5.6.4 Example of setting PLL circuit	
	Modification of Figure 5 - 24 CPU Clock Status Transition Diagram (Products with USB)	
	Modification of the table in (2) of Table 5 - 4 CPU Clock Transition and SFR Register Setting Examples (1/7)	
	Modification of the table in (4) of Table 5 - 5 CPU Clock Transition and SFR Register Setting Examples (2/7)	
	Modification of note in (6) and (8) of Table 5 - 6 CPU Clock Transition and SFR Register Setting Examples (3/7)	
	Modification of description in (10) of Table 5 - 8, Table 5 - 9 CPU Clock Transition and SFR Register Setting Examples (5/7), (6/7)	
	Deletion of Table 5 - 9 CPU Clock Transition and SFR Register Setting Examples (5/7)	
	Modification of description in (11) of Table 5 - 9 CPU Clock Transition and SFR Register Setting Examples (6/7)	
	Modification of description in (12) of Table 5 - 10 CPU Clock Transition and SFR Register Setting Examples (7/7)	
	Modification of description in (15) of Table 5 - 10 CPU Clock Transition and SFR Register Setting Examples (7/7)	
	Modification of Condition Before Change in Table 5 - 16 Changing CPU Clock (1/3)	
	Modification of Condition Before Change in Table 5 - 17 Changing CPU Clock (2/3)	

(4/15)

Edition	Description	Chapter
Rev. 2.10	Modification of description in 5.6.8 Conditions before clock oscillation is stopped	CHAPTER 5 CLOCK GENERATOR
	Modification of description in 5.7 Resonator and Oscillator Constants	
	Modification of note 3, and note 3 changed to note 2 in (1) X1 oscillation of 5.7 Resonator and Oscillator Constants	
	Deletion of note in Figure 6 - 11 Format of Peripheral enable register 0 (PER0)	CHAPTER 6 TIMER ARRAY UNIT
	Modification of description for CCSmn in Figure 6 - 14 Format of Timer mode register mn (TMRmn) (1/4)	
	Modification of description for Setting of starting counting and interrupt in Figure 6 - 17 Format of Timer mode register mn (TMRmn) (4/4)	
	Modification of formula in (1) Interval timer of 6.8.1 Operation as interval timer/square wave output	
	Modification of description for Software Operation in Figure 6 - 64 Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used	
	Modification of caution in 6.9.1 Operation as one-shot pulse output function	
	Modification of description	CHAPTER 7 16-BIT TIMER KB20, KB21, KB22
	Modification of description and deletion of note in Figure 8 - 2 Format of Peripheral enable register 0 (PER0)	CHAPTER 8 REAL-TIME CLOCK 2
	Addition of notes 1 and 2 in Figure 8 - 8 Format of Real-time clock control register 1 (RTCC1) (3/3)	
	Modification of description in 10.5 Cautions of clock output/buzzer output controller	CHAPTER 10 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER
	Deletion of note in Figure 12 - 2 Format of Peripheral enable register 0 (PER0)	CHAPTER 12 A/D CONVERTER
	Modification of description in (2) Input range of ANI0 to ANI6 and ANI16 to ANI21 pins of 12.10 Cautions for A/D Converter	
	Deletion of note in Figure 15 - 5 Format of Peripheral enable register 0 (PER0)	CHAPTER 15 SERIAL ARRAY UNIT
	Modification of caution 4 in Figure 15 - 11 Format of Serial data register mn (SDRmn)	
	Modification of pin name (SOUT0n → TXDq) in Figure 15 - 21 Examples of Reverse Transmit Data	
	Modification of description in 15.5.7 SNOOZE mode function	
	Modification of Figure 15 - 74 Timing Chart of SNOOZE Mode Operation (once startup) (Type 1: DAPmn = 0, CKPmn = 0), and deletion of note 2	
	Modification of Figure 15 - 75 Flowchart of SNOOZE Mode Operation (once startup)	
	Modification of Figure 15 - 76 Timing Chart of SNOOZE Mode Operation (continuous startup) (Type 1: DAPmn = 0, CKPmn = 0)	
	Modification of Figure 15 - 77 Flowchart of SNOOZE Mode Operation (continuous startup)	
	Modification of description and addition of caution 5 in 15.6.3 SNOOZE mode function	
	Modification of Figure 15 - 95 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)	
	Modification of Figure 15 - 96 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)	
	Modification of Figure 15 - 97 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)	
	Modification of Figure 15 - 98 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)	
	Modification of Figure 15 - 99 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)	
	Addition of f_{MCK} in Figure 16 - 1 Block Diagram of Serial Interface IICA	CHAPTER 16 SERIAL INTERFACE IICA
	Deletion of note in Figure 16 - 5 Format of Peripheral enable register 0 (PER0)	
	Modification of description in Figure 16 - 14 Format of IICA control register 01 (IICCTL01) (1/2)	
	Modification of description for PRSn bit, addition of caution 1, and modification of caution 2 in Figure 16 - 15 Format of IICA control register 01 (IICCTL01) (2/2)	
	Addition of description in 16.3.6 IICA low-level width setting register 0 (IICWL0)	
	Modification of description (f_{CLK} → f_{MCK}), addition of caution 1, and modification of caution 2 and remark 2 in 16.4.2 Setting transfer clock by using IICWLn and IICWHn registers	
	Modification of description in Figure 16 - 29 Flow When Setting WUPn = 1	

(5/15)

Edition	Description	Chapter
Rev. 2.10	Modification of description in Figure 16 - 30 Flow When Setting WUPn = 0 upon Address Match (Including Extension Code Reception)	CHAPTER 16 SERIAL INTERFACE IICA
	Modification of description in Figure 16 - 31 When Operating as Master Device after Releasing STOP Mode other than by INTIICAn	
	Modification of formula and remark 1 in 16.5.14 Communication reservation	
	Modification of note 1 and remark 1 in Figure 16 - 34 Communication Reservation Protocol	
	Modification of description in (2) When communication reservation function is disabled (bit 0 (IICRSVn) of IICA flag register n (IICFn) = 1) of 16.5.14 Communication reservation	
	Modification of description in <4> of 16.5.15 (3) If other I^2C communications are already in progress	
	Addition of Setting the PER0 register in Figure 16 - 35 Master Operation in Single-Master System	
	Addition of Setting the PER0 register in Figure 16 - 36 Master Operation in Multi-Master System (1/3)	
	Addition of description and modification of remark 1 in Figure 16 - 37 Master Operation in Multi-Master System (2/3)	
	Addition of Setting the PER0 register in Figure 16 - 39 Slave Operation Flowchart (1)	
	Modification of Table 17 - 22 DTC Settings, and addition of caution	CHAPTER 17 USB 2.0 FUNCTION MODULE (USB)
	Addition of description	CHAPTER 19 DATA TRANSFER CONTROLLER (DTC)
	Modification of note in Table 19 - 1 DTC Specifications	
	Addition of Figure 19 - 4 Start Address of Control Data and Vector Table	
	Modification of value for DTCCT12 in Figure 19 - 16 Example 1 of using normal mode: Consecutively capturing A/D conversion results	
	Modification of description in 19.5.3 DTC Pending Instruction	
	Modification of vector table addresses in Tables 21 - 1 to 21 - 4 Interrupt Source List	CHAPTER 21 INTERRUPT FUNCTIONS
	Addition of caution in 21.4.4 Interrupt servicing during division instruction	
	Addition of instructions to 21.4.5 Interrupt request hold	
	Modification of description in Table 23 - 1 Operating Statuses in HALT Mode (1/2)	CHAPTER 23 STANDBY FUNCTION
	Modification of description in Table 23 - 1 Operating Statuses in HALT Mode (2/2)	
	Modification of addresses (00000H and 00001H)	CHAPTER 24 RESET FUNCTION
	Deletion of caution in 24.1 Timing of Reset Operation	
	Modification of description in Table 24 - 1 Operation Statuses During Reset Period	
	Modification of addresses (00000H, 00001H) in Table 24 - 2 State of Hardware After Receiving a Reset Signal	
	Modification of description for after reset in Figure 24 - 4 Format of Reset control flag register (RESF)	
	Addition of caution in Figure 24 - 5 Example of Procedure for Checking Reset Source	
	Modification of note 3 in (2) of Figure 25 - 3 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (2/3)	CHAPTER 25 POWER-ON-RESET CIRCUIT
	Modification of description for interrupt mode in the table in 26.1 Functions of Voltage Detector	CHAPTER 26 VOLTAGE DETECTOR
	Modification of description in the table in Figure 26 - 5 Format of User Option Byte (000C1H/010C1H) (2/2)	
	Modification of description in 26.4.2 When used as interrupt mode	
	Modification of description in (3) of 29.1.1 User option byte (000C0H to 000C2H/010C0H to 010C2H)	CHAPTER 29 OPTION BYTE
	Modification of the header of the table in Figure 29 - 4 Format of User Option Byte (000C2H/010C2H)	
	Modification of signal name in Table 30 - 1 Wiring Between RL78/L1C and Dedicated Flash Memory Programmer	CHAPTER 30 FLASH MEMORY
	Addition of FLMD1 in Figure 30 - 2 Communication with Dedicated Flash Memory Programmer	
	Change of signal names in Table 30 - 2 Pin Connection	
	Modification of remark 1 in 30.6 Self-Programming	
	Addition of description in 30.8.3 Procedure for accessing data flash memory	
	Addition of note in Table 31 - 1 On-Chip Debug Security ID	CHAPTER 31 ON-CHIP DEBUG FUNCTION

(6/15)

Edition	Description	Chapter
Rev. 2.10	Addition of note 3 and caution, and deletion of remark 1 in Table 33 - 16 Operation List (12/18)	CHAPTER 33 INSTRUCTION SET
	Correction of conditions of V_{O1}, V_{O2} in Absolute Maximum Ratings (1/3)	CHAPTER 34 ELECTRICAL SPECIFICATIONS (A: T_A = -40 to +85°C)
	Correction of conditions of I_{OH1}, I_{OL1} and ratings I_{OH2}, I_{OL2} (total of all pins) in Absolute Maximum Ratings (3/3)	
	Correction of conditions of I_{OH1}, I_{OH2}, I_{OL1}, I_{OL2} and ratings I_{OH2}, I_{OL2} (total of all pins) in 34.3.1 Pin characteristics	
	Correction of conditions of V_{OH1}, V_{OH2}, V_{OL1}, V_{OL2} in 34.3.1 Pin characteristics	
	Modification of the graphs for Minimum Instruction Execution Time during Main System Clock Operation	
	Modification of conditions in (1) of 34.6.1 A/D converter characteristics	
	Modification of the title and note in 34.9 RAM Data Retention Characteristics	
	Modification of conditions in 34.10 Flash Memory Programming Characteristics	
	Modification of description	CHAPTER 35 ELECTRICAL SPECIFICATIONS (G: INDUSTRIAL APPLICATIONS T_A = -40 to +105°C)
	Correction of conditions of V_{O1}, V_{O2} in Absolute Maximum Ratings (1/3)	
	Correction of conditions of I_{OH1}, I_{OL1} and ratings I_{OH2}, I_{OL2} (total of all pins) in Absolute Maximum Ratings (3/3)	
	Correction of conditions of I_{OH1}, I_{OH2}, I_{OL1}, I_{OL2} and ratings I_{OH2}, I_{OL2} (total of all pins) in 35.3.1 Pin characteristics	
	Correction of conditions of V_{OH1}, V_{OH2}, V_{OL1}, V_{OL2} in 35.3.1 Pin characteristics	
	Modification of the graph for Minimum Instruction Execution Time during Main System Clock Operation	
	Modification of the title and note in 35.9 RAM Data Retention Characteristics	
	Modification of conditions and addition of note 4 in 35.10 Flash Memory Programming Characteristics	
Rev. 2.00	Addition of 85-pin product information	All
	Modification from 80-pin to 80/85-pin	
	Modification from x = M, P to x = M, N, P	
	Modification from high-accuracy real-time clock to real-time clock 2	
	Modification from RTC to RTC2	
	Modification of LCD controller/driver in 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/L1C	
	Modification of Table 2 - 1 Pin I/O Buffer Power Supplies	CHAPTER 2 PIN FUNCTIONS
	Modification of (6) in 4.5.2 Register settings for alternate function whose output function is not used	CHAPTER 4 PORT FUNCTIONS
	Addition of 5.7 Resonator and Oscillator Constants	CHAPTER 5 CLOCK GENERATOR
	Modification of Figure 7 - 25 Format of Forced output stop function control register 0p (TKBPACTL0p) (3/3)	CHAPTER 7 16-BIT TIMER KB20, KB21, KB22
	Modification of Figure 7 - 28 Format of Forced output stop function control register 1p (TKBPACTL1p) (3/3)	
	Modification of Figure 7 - 31 Format of Forced output stop function control register 2p (TKBPACTL2p) (3/3)	
	Modification of Figure 8 - 4 Format of Real-time clock control register 0 (RTCC0) (1/2)	CHAPTER 8 REAL-TIME CLOCK 2
	Modification of Figure 8 - 5 Format of Real-time clock control register 0 (RTCC0) (2/2)	
	Modification of Figure 8 - 6 Format of Real-time clock control register 1 (RTCC1) (1/3)	
	Modification of Figure 8 - 25 1 Hz Output Setting Procedure	
	Modification of 17.4.1.1 Starting Operation	CHAPTER 17 USB 2.0 FUNCTION MODULE (USB)
	Modification of Figure 17 - 45 USB Connector Connection Example in Self-powered (3.3 V)	
	Modification of Figure 17 - 46 USB Connector Connection Example in Bus-powered (3.3 V)	
	Modification of Figure 23 - 8 When the Interrupt Request Signal is not Generated in the SNOOZE Mode	CHAPTER 23 STANDBY FUNCTION

(7/15)

Edition	Description	Chapter
Rev. 2.00	Modification of (1) Electrical specifications in 34.5.3 USB	CHAPTER 34 ELECTRICAL SPECIFICATIONS (A: T_A = -40 to +85°C)
	Modification of note 1 in (1) 1/3 bias method in 34.8.2 Internal voltage boosting method	
	Modification of (1) Electrical specifications in 35.5.3 USB	CHAPTER 35 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)
	Modification of note 1 in (1) 1/3 bias method in 35.8.2 Internal voltage boosting method	
Rev. 1.00	Deletion of overline of SCK and ACK xx	All
	Deletion of COMEXP and COMEXP pins	
	Deletion of memory-type liquid crystal control register (MLCD)	
	Addition of high-speed on-chip oscillator trimming register (HIOTRM)	
	Addition of interrupt enable register 1 (INTENB1) (only in products with USB)	
	Addition of interrupt status register 1 (INTSTS1) (only in products with USB)	
	Addition of USB clock select register (UCKSEL) (only in products with USB)	
	Modification of 1.1 Features	CHAPTER 1 OUTLINE
	Modification of 1.2 Ordering Information	
	Modification of package type in 1.3 Pin Configuration (Top View)	
	Modification of vectored interrupt sources in 1.6 Outline of Functions	
	Modification of operating ambient temperature in 1.6 Outline of Functions	
	Addition of caution in 2.1 Port Function	CHAPTER 2 PIN FUNCTIONS
	Modification of tables and remarks in 2.1.1 to 2.1.4	
	Modification of description and table in 2.3 Connection of Unused Pins	
	Modification of 2.4 Pin Block Diagrams	
	Modification of 3.1 Overview	CHAPTER 3 CPU ARCHITECTURE
	Modification of notes in Figure 3 - 1 to Figure 3 - 5	
	Modification of Table 3 - 4 (deletion of INTLCD0)	
	Modification of description and caution in 3.2.3 Internal data memory space	
	Modification of figure in 3.2.6 Data memory addressing	
	Modification of error in (3) Stack pointer (SP)	
	Modification of description in 3.3.2 General-purpose registers	
	Modification of Figure 3 - 13 Extension of Data Area Which Can Be Accessed	
	Modification of description in 3.3.4 Special function registers (SFRs)	
	Modification of description in 3.3.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)	
	Modification of Table 3 - 11 to Table 3 - 25	
	Modification of Table 4 - 1 Port Configuration	CHAPTER 4 PORT FUNCTIONS
	Modification of descriptions in 4.2.1 Port 0 to 4.2.13 Port 15	
	Addition of caution in 4.3 Registers Controlling Port Function	
	Modification of description in 4.3.5 Port output mode registers (POMxx)	
	Modification of Figure 4 - 8 Format of Peripheral I/O redirection register (PIOR)	
	Modification of description in 4.3.9 Global analog input disable register (GAIDIS)	
	Addition of caution in Table 4 - 6 Segment Output Pins in Each Product and Correspondence with PFSEG Register (PFSEG Bits)	
	Modification of description in 4.4.4 Handling different potential (1.8 V, 2.5 V) by using I/O buffers	

(8/15)

Edition	Description	Chapter
Rev. 1.00	Modification of description in 4.5 Register Settings When Using Alternate Function	CHAPTER 4 PORT FUNCTIONS
	Modification of description in 5.1 Functions of Clock Generator	CHAPTER 5 CLOCK GENERATOR
	Modification of Figure 5 - 1 and Figure 5 - 2	
	Addition of caution in 5.3 Registers Controlling Clock Generator	
	Modification of description, remark, and caution in 5.3.2 System clock control register (CKC)	
	Modification of description in 5.3.5 Oscillation stabilization time select register (OSTS)	
	Modification of description and note in Figure 5 - 10 Format of Peripheral enable register 1 (PER1)	
	Modification of note in Figure 5 - 11 Format of Peripheral enable register 2 (PER2)	
	Modification of description in 5.3.7 Subsystem clock supply mode control register (OSMC)	
	Modification of description in 5.3.8 High-speed on-chip oscillator frequency select register (HOCODIV)	
	Addition of 5.3.9 High-speed on-chip oscillator trimming register (HIOTRM)	
	Modification of description, note, and caution in 5.3.11 Main clock control register (MCKC)	
	Addition of 5.3.12 USB clock selection register (UCKSEL)	
	Extension of clock in 5.4.3 High-speed on-chip oscillator	
	Modification of description in 5.4.4 Low-speed on-chip oscillator	
	Modification of description in Figure 5 - 23 Clock Generator Operation When Power Supply Voltage Is Turned On	
	Extension of selected clock in 5.6.1 Example of setting high-speed on-chip oscillator	
	Modification of description in 5.6.2 Example of setting X1 oscillation clock	
	Modification of description in 5.6.5 CPU clock status transition diagram	
	Modification of table in 5.6.6 Condition before changing CPU clock and processing after changing CPU clock	
	Modification of description of timer array unit	CHAPTER 6 TIMER ARRAY UNIT
	Modification of figure in (1) One-shot pulse output	
	Modification of Figure 6 - 1, Figure 6 - 4, and Figure 6 - 5	
	Addition of caution in 6.3 Registers Controlling Timer Array Unit	
	Modification of caution in Figure 6 - 11 Format of Peripheral enable register 0 (PER0)	
	Modification of description in 6.3.2 Timer clock select register m (TPSm)	
	Modification of description in 6.3.7 Timer channel stop register m (TTm)	
	Modification of description in Figure 6 - 22 Format of Timer input select register 0 (TIS0)	
	Modification of description in Figure 6 - 24 Format of Timer output enable register m (TOEm)	
	Modification of description in 6.3.15 Noise filter enable register 1 (NFEN1)	
	Modification of description in 6.3.16 Registers controlling port functions of pins to be used for timer I/O	
	Modification of remarks in 6.5.3 Operation of counter (2) and (3)	
	Modification of description in 6.6.2 TOmn Pin Output Setting	
	Addition of caution in 6.7 Timer Input (TImn) Control	
	Modification of errors in Figure 6 - 50 and Figure 6 - 51	
	Modification of description in Figure 6 - 53	
	Modification of errors in Figure 6 - 55 and Figure 6 - 56	
	Addition of caution in Figure 6 - 60	
	Modification of description in Figure 6 - 61	

(9/15)

Edition	Description	Chapter
Rev. 1.00	Modification of description in Figure 6 - 64	CHAPTER 6 TIMER ARRAY UNIT
	Addition of caution in Figure 6 - 65	
	Modification of errors and addition of descriptions in Figure 6 - 67 and Figure 6 - 68	
	Addition of caution in Figure 6 - 69	
	Modification of error in Figure 6 - 71	
	Modification of descriptions in Figure 6 - 73 and Figure 6 - 74	
	Modification of errors and addition of descriptions in Figure 6 - 77 and Figure 6 - 78	
	Modification of errors and addition of descriptions in Figure 6 - 83 and Figure 6 - 84	
	Modification of title and table in CHAPTER 7 16-BIT TIMER KB20, KB21, KB22	CHAPTER 7 16-BIT TIMER KB20, KB21, KB22
	Modification of description in (5) Timer restart function	
	Modification of remark in (6), addition of remark in (7)	
	Modification of configuration (modification of name) in Table 7 - 1 Configuration of 16-bit Timer KB0, KB1, and KB22	
	Modification of description in Figure 7 - 1 Block Diagram of 16-bit Timer KB20, KB21, KB22	
	Addition of (1) 16-bit timer counter register n (TKBCNTn)	
	Modification of descriptions in (2) and (3)	
	Modification of caution in Figure 7 - 5 Format of Peripheral enable register 1 (PER1)	
	Extension of operating clock in Figure 7 - 7 Format of 16-bit timer KB2 clock division ratio select register n (TKBPSCSn)	
	Modification of description in 7.3.4 16-bit timer KB2 operation control register n0 (TKBCTLn0)	
	Addition of note and modification of caution in Figure 7 - 9 Format of 16-bit timer KB2 operation control register n0 (TKBCTLn0) (2/2)	
	Modification of description in Figure 7 - 10 Format of 16-bit timer KB2 operation control register n1 (TKBCTLn1)	
	Modification of error in Figure 7 - 11 Format of 16-bit timer KB2 output control register n0 (TKBIOCn0)	
	Modification of description in Figure 7 - 12 Format of 16-bit timer KB2 output control register n1 (TKBIOCn1)	
	Modification of error in Figure 7 - 13 Format of 16-bit timer KB2 flag register n (TKBFLGn)	
	Addition of description in Figure 7 - 14 Format of 16-bit timer KB2 trigger register n (TKBTRGn)	
	Modification of description and caution in Figure 7 - 15 Format of 16-bit timer KB2 flag clear trigger register n (TKBCLRn)	
	Modification of description in 7.3.17 Forced output stop function control register 0p (TKBPACTL0p)	
	Modification of description in 7.3.18 Forced output stop function control register 1p (TKBPACTL1p)	
	Modification of description in 7.3.19 Forced output stop function control register 2p (TKBPACTL2p)	
	Modification of description (bit symbol) in Figure 7 - 33 Format of Forced output stop function flag register n (TKBPAFLGn)	
	Modification of title and description in 7.4 Operation of 16-bit Timer KB20, KB21, KB22, Modification of description in 7.4.1 (1)	
	Modification of description in 7.4.2 Default Level and Active Level	
	Modification of description in 7.4.3 Stop/Restart Operation	
	Modification of error in (1) Timing of batch overwrite	
	Modification of description in Figure 7 - 50 Configuration of Standalone Mode (Period Controlled by TKBCRn0)	
	Modification of description in (5) Sample of Register Setting Details at Standalone Mode (Period Controlled by TKBCRn0)	

(10/15)

Edition	Description	Chapter
Rev. 1.00	Modification of description in (2) Batch Overwrite Function (In Standalone Operation during Period Controlled by External Trigger Input, Buffer Updating during Counting Operation (TKBTSEn bit set to 1))	CHAPTER 7 16-BIT TIMER KB20, KB21, KB22
	Modification of description in (4) Sample of Register Setting Details at Standalone Mode (Period Controlled by External Trigger Input)	
	Modification of description in 7.4.7 Simultaneous Start / Stop Mode	
	Modification of information in (3) Simultaneous Start/Stop Mode to (6) Simultaneous Start / Stop Mode	
	Modification of information in Figure 7 - 57	
	Modification of description in 7.4.8 Synchronous Start / Clear Mode	
	Modification of (2) Synchronous Start / Clear Mode: List of register setting by Master and (3) Synchronous Start / Clear Mode: List of register setting by Slave	
	Modification of information in Figure 7 - 59	
	Modification of information in (1) Output Conditions of TKBOn1 at Interleave PFC	
	Modification of description in (2) List of Register Setting at Interleave PFC Output Mode	
	Modification of table in 7.5 Option Functions of 16-bit Timers KB20, KB2, and KB22	
	Modification of error in Figure 7 - 77 Figure of Waveform at Dithering Operation	
	Modification of caution 1 in (1) Available Operation Mode	
	Modification of description in 7.5.3 PWM Output Smooth Start Function	
	Modification of description in (2) Operation Mode Available for Maximum Frequency Limit Function	
	Deletion of 7.5.5 PWM Output Function for IH Control	
	Modification of Figure 7 - 82 System Structure of Forced Output Stop Function	
	Modification of description in 7.6.1 Forced Output Stop Function 1 and 2	
	Modification of description, caution, and remark, and deletion of note in 7.7.1 I/O Setting for Forced Output Stop Function 1	
	Modification of description, caution, and remark in 7.8.1 I/O Setting for Forced Output Stop Function 2	
	Modification of description in 7.9 Usage Notes	
	Addition of setting in Figure 8 - 20 Procedure for Starting Operation of High-Accuracy Real-time Clock	CHAPTER 8 HIGH-ACCURACY REAL-TIME CLOCK
	Modification of Figure 8 - 21 Procedure for Shifting to HALT/STOP Mode After Setting RTCE bit to 1	
	Modification of descriptions and figures and addition of notes in 8.4.3 and 8.4.4	
	Addition of settings in Figure 8 - 24 and Figure 8 - 25	
	Modification of 8.4.7 Clock error correction register setting procedure	
	Modification of description in 8.4.8 Example of watch error correction of high-accuracy real-time clock	
	Modification of error in Figure 9 - 1 Block Diagram	CHAPTER 9 12-BIT INTERVAL TIMER
	Modification of caution in Figure 9 - 2 Format of Peripheral enable register 1 (PER1)	
	Modification of error in 9.4.1 12-bit interval timer operation timing	
	Addition of 9.4.2 Start of count operation and re-enter to HALT/STOP mode after returned from HALT/STOP mode	
	Modification of note in Figure 10 - 2 Format of Clock output select registers n (CKSn)	CHAPTER 10 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

(11/15)

Edition	Description	Chapter
Rev. 1.00	Addition of 10.3.2 Registers controlling port functions of pins to be used for clock or buzzer output	CHAPTER 10 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER
	Modification of description in 10.4.1 Operation as output pin	
	Modification of description in 10.5 Cautions of clock output/buzzer output controller	
	Modification of description in 11.1 Functions of Watchdog Timer	CHAPTER 11 WATCHDOG TIMER
	Addition of description in 11.2 Configuration of Watchdog Timer	
	Modification of caution in 11.4.1 Controlling operation of watchdog timer	
	Modification of error in table of A/D converter	CHAPTER 12 A/D CONVERTER
	Modification of description in 12.1 Function of A/D Converter	
	Modification of caution in 12.3.1 Peripheral enable register 0 (PER0)	
	Modification of description in 12.3.2 A/D converter mode register 0 (ADM0)	
	Modification of caution in 12.3.3 A/D converter mode register 1 (ADM1)	
	Addition of description in 12.3.4 A/D converter mode register 2 (ADM2)	
	Modification of description in 12.3.5 12-bit A/D conversion result register (ADCR)	
	Modification of note and caution in 12.3.7 Analog input channel specification register (ADS)	
	Modification of description in 12.3.10 A/D test register (ADTES)	
	Modification of description in 12.3.11 Registers controlling port function of analog input pins	
	Addition of note and modification of description in 12.4 A/D Converter Conversion Operations	
	Modification of error in 12.7.4 Setup when temperature sensor output voltage/internal reference voltage is selected (example for software trigger mode and one-shot conversion mode)	
	Modification of figure and remark in 12.8 SNOOZE Mode Function	
	Addition of Figure 12 - 38 Flowchart for Setting up SNOOZE Mode	
	Modification of description in (2) Input range of ANI0 to ANI6 and ANI16 to ANI21 pins	
	Modification of description in (4) Noise countermeasures	
	Modification of description in (5) Analog input (ANIn) pins	
	Modification of description in (6) Input impedance of analog input (ANIn) pins	
	Modification of description in (8) Conversion results just after A/D conversion start	
	Modification of description in (10) Internal equivalent circuit	
	Addition of (12)	
	Modification of description in 13.1 Functions of D/A Converter	CHAPTER 13 D/A CONVERTER
	Addition of 13.3.1 Port mode control register 4 (PMC4)	
	Modification of description in 13.3.5 Port mode register 4 (PM4)	
	Modification of caution in 13.4 Operations of D/A Converter	
	Modification of description in 13.4.2 Operation in Real-Time Output Mode	
	Addition of 13.4.3 Timing for Outputting D/A Conversion Value	
	Modification of description in 13.5 Cautions for D/A Converter	
	Modification of Functions of Comparator	CHAPTER 14 COMPARATOR
	Modification of Figure 14 - 1 Comparator Block Diagram	
	Modification of notes in 14.3.2 Comparator mode setting register (COMPMDR) to 14.3.4 Comparator output control register (COMPOCR)	
	Modification of Table 14 - 4 Procedure for Setting Comparator Associated Registers	
	Modification of descriptions in 14.4.1 to 14.4.3	
	Modification of caution in 14.4.5 Stopping or Supplying Comparator Clock	

(12/15)

Edition	Description	Chapter
Rev. 1.00	Modification of description in 15.1.1 3-wire serial I/O (CSI00, CSI10, CSI20, CSI30)	CHAPTER 15 SERIAL ARRAY UNIT
	Modification of remark in 15.1.3 Simplified I^2C (IIC00, IIC10, IIC20, IIC30)	
	Modification of Figure 15 - 1 Block Diagram of Serial Array Unit 0	
	Modification of Figure 15 - 2 Block Diagram of Serial Array Unit 1	
	Modification of description in 15.2.2 Lower 8/9 bits of the serial data register mn (SDRmn)	
	Modification of description in 15.3 Registers Controlling Serial Array Unit	
	Modification of caution in 15.3.1 Peripheral enable register 0 (PER0)	
	Modification of description in 15.3.4 Serial communication operation setting register mn (SCRmn)	
	Modification of description in 15.3.5 Serial data register mn (SDRmn)	
	Modification of caution in 15.3.7 Serial status register mn (SSRmn)	
	Modification of description in 15.3.13 Serial output level register m (SOLm)	
	Modification of description in 15.3.14 Serial standby control register m (SSCm)	
	Modification of description in 15.3.16 Noise filter enable register 0 (NFEN0)	
	Modification of description in 15.3.17 Registers controlling port functions of serial input/output pins	
	Modification of caution in 15.4.1 Stopping the operation by units	
	Modification of description in 15.5 Operation of 3-Wire Serial I/O (CSI00, CSI10, CSI20, CSI30) Communication	
	Modification of description in 15.5.1 Master transmission	
	Modification of description in 15.5.2 Master reception	
	Modification of description in 15.5.3 Master transmission/reception	
	Modification of description in 15.5.4 Slave transmission	
	Modification of description in 15.5.5 Slave reception	
	Modification of description in 15.5.6 Slave transmission/reception	
	Modification of description in 15.5.7 SNOOZE mode function	
	Modification of description in 15.6 Operation of UART (UART0 to UART3) Communication	
	Modification of description in 15.6.1 UART transmission	
	Modification of description in 15.6.2 UART reception	
	Modification of description in 15.6.3 SNOOZE mode function	
	Modification of description in 15.7.1 LIN transmission	
	Modification of description in 15.7.2 LIN reception	
	Modification of description in 15.8 Operation of Simplified I^2C (IIC00, IIC10, IIC20, IIC30) Communication	
	Modification of description in 15.8.1 Address field transmission	
	Modification of description in 15.8.2 Data transmission	
	Modification of description in 15.8.3 Data reception	
	Modification of description in 15.8.6 Procedure for processing errors that occurred during simplified I^2C (IIC00, IIC10, IIC20, IIC30) communication	
	Modification of caution in 16.3.1 Peripheral enable register 0 (PER0)	CHAPTER 16 SERIAL INTERFACE IICA
	Modification of notes and remarks in Figure 16 - 8 and Figure 16 - 9	
	Modification of description in 16.3.5 IICA control register 01 (IICCTL01)	
	Modification of description in 16.3.6 IICA low-level width setting register 0 (IICWL0)	
	Modification of description in 16.3.7 IICA high-level width setting register 0 (IICWH0)	
	Modification of description in 16.5.13 Wakeup function	
	Modification of Figure 16 - 31 When Operating as Master Device after Releasing STOP Mode other than by INTIICAn	

(13/15)

Edition	Description	Chapter
Rev. 1.00	Modification of error in Figure 16 - 35 Master Operation in Single-Master System	CHAPTER 16 SERIAL INTERFACE IICA
	Modification of error in Figure 16 - 38 Master Operation in Multi-Master System (3/3)	
	Modification of error in Figure 16 - 39 Slave Operation Flowchart (1)	
	Addition of description in Table 17 - 1 USB Module Specifications	CHAPTER 17 USB 2.0 FUNCTION MODULE (USB)
	Modification of error in Table 17 - 2 USB Module I/O Pins	
	Modification of Table 17.3 Registers Used in USB 2.0 Function Module	
	Modification of description in Figure 17 - 10 Format of DnFIFO port select register (DnFIFOSEL) (n = 0, 1)	
	Addition of 17.3.10 Interrupt enable register 1 (INTENB1)	
	Addition of 17.3.16 Interrupt status register 1 (INTSTS1)	
	Addition of 17.3.37 USB clock selection register (UCKSEL)	
	Modification of description and figure in 17.4.1.2 Controlling USB Data Bus Resistors	
	Modification of table and figure in17.4.2 Interrupt Sources	
	Modification of figure and note in Figure 17 - 48 Timing of BRDY Interrupt Generation	
	Modification of Figure 17 - 49 Timing of NRDY Interrupt Generation	
	Addition of 17.4.3.10 DTC Transfer End Interrupts for D0FIFO/D1FIFO	
	Modification of Figure 17 - 53 Procedure for Setting DTC Transfer	
	Addition of 17.4.12 Battery Charging Detection Processing	
	Modification of Table 18 - 1 Number of LCD Display Function Pins of Each Product	CHAPTER 18 LCD CONTROLLER/DRIVER
	Modification of description and table in 18.1 Functions of LCD Controller/Driver	
	Modification of Figure 18.1 Block Diagram of LCD Controller/Driver	
	Modification of description in 18.3.1 LCD mode register 0 (LCDM0)	
	Modification of description in 18.3.2 LCD mode register 1 (LCDM1)	
	Deletion of Caution 2 in 18.3.3 Subsystem clock supply mode control register (OSMC)	
	Modification of description in 18.3.4 LCD clock control register 0 (LCDC0)	
	Modification of description in 18.3.5 LCD boost level control register (VLCD)	
	Modification of caution in 18.3.6 LCD input switch control register (ISCLCD)	
	Modification of caution in 18.5 Selection of LCD Display Register	
	Modification of 18.6 Setting the LCD Controller/Driver	
	Modification of 18.7 Operation stop procedure (deletion of (b) During memory-type liquid crystal waveform)	
	Modification of description in 18.8 Supplying LCD Drive Voltages VL_1, VL_2, VL_3, and VL_4	
	Modification of Figure 18 - 24 Examples of LCD Drive Power Connections (Capacitor Split Method)	
	Deletion of 18.11 Memory-Type Liquid Crystal Waveform Example	
	Modification of Figure 19 - 1 DTC Block Diagram	CHAPTER 19 DATA TRANSFER CONTROLLER (DTC)
	Modification of error in 19.3.3 Vector Table	
	Modification of error in Table 19 - 6 Register Functions in Normal Mode	
	Modification of error in (1) Example 1 of using normal mode: Consecutively capturing A/D conversion results	
	Modification of error in (2) Example 2 of using normal mode: UART consecutive transmission	
	Modification of error in (1) Example 1 of using repeat mode: Outputting a stepping motor control pulse using ports	
	Modification of error in (2) Example 2 of using repeat mode: Outputting a sine wave using the 8-bit D/A converter	
	Modification of description in 19.4.4 Chain Transfers	

(14/15)

Edition	Description	Chapter
Rev. 1.00	Modification of error in (1) Example of using chain transfers: Consecutively capturing A/D conversion results and UART0 transmission	CHAPTER 19 DATA TRANSFER CONTROLLER (DTC)
	Addition of description in Table 20 - 1 Registers Controlling ELC	CHAPTER 20 EVENT LINK CONTROLLER (ELC)
	Modification of description in 20.3.1 Event output destination select register n (ELSELRn) (n = 00 to 30)	
	Addition of Table 20 - 7 Response of Peripheral Functions That Receive Events	
	Modification of table in interrupt functions	CHAPTER 21 INTERRUPT FUNCTIONS
	Modification of description in 21.2 Interrupt Sources and Configuration (deletion of LCD frame interrupt)	
	Modification of caution in Table 21 - 7 Flags Corresponding to Interrupt Request Sources (3/4)	
	Modification of descriptions in 21.3.1 to 21.3.5	
	Modification of Figure 21 - 11 to Figure 21 - 13	
	Modification of Table 21 - 11 Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing	
	Modification of bit symbol in Figure 24 - 6 Format of Power-on-reset status register (PORSR)	
	Modification of Table 22 - 1 Assignment of Key Interrupt Detection Pins	CHAPTER 22 KEY INTERRUPT FUNCTION
	Modification of caution in Figure 22 - 3 Format of Key return mode register (KRM0)	
	Modification of description in 22.3.4 Port mode register 7 (PM7)	
	Modification of Table 23 - 1 Operating Statuses in HALT Mode (1/2)	CHAPTER 23 STANDBY FUNCTION
	Modification of Table 23 - 2 Operating Statuses in HALT Mode (2/2)	
	Modification of description in 23.3.2 STOP mode	
	Modification of description in 23.3.3 SNOOZE mode	
	Modification of description in reset function	CHAPTER 24 RESET FUNCTION
	Modification of 24.1 Timing of Reset Operation	
	Modification of 24.2 States of Operation During Reset Periods	
	Modification of 24.3 Register for Confirming Reset Source	
	Modification of description in 25.1 Functions of Power-on-reset Circuit	CHAPTER 25 POWER-ON-RESET CIRCUIT
	Modification of 25.3 Operation of Power-on-reset Circuit	
	Modification of description in 26.1 Functions of Voltage Detector	CHAPTER 26 VOLTAGE DETECTOR
	Modification of description in Figure 26 - 1 Block Diagram of Voltage Detector	
	Modification of 26.3 Registers Controlling Voltage Detector	
	Modification of 26.4 Operation of Voltage Detector	
	Modification of 26.5 Cautions for Voltage Detector	
	Addition of description in Figure 27 - 3 Flowchart of Flash Memory CRC Operation Function (High-speed CRC)	CHAPTER 27 SAFETY FUNCTIONS
	Modification of caution and remark in 27.3.3.1 RAM parity error control register (RPECTL)	
	Modification of note in 27.3.5.1 Invalid memory access detection control register (IAWCTL)	
	Modification of error in 27.3.6 Invalid memory access detection function	
	Modification of description in 27.3.7 Frequency detection function	
	Modification of description in 27.3.9 A/D test function	
	Modification of description in 29.1 Functions of Option Bytes	CHAPTER 29 OPTION BYTE

(15/15)

Edition	Description	Chapter
Rev. 1.00	Modification of error and description inn 29.2 Format of User Option Byte	CHAPTER 29 OPTION BYTE
	Addition of description in FLASH MEMORY	CHAPTER 30 FLASH MEMORY
	Addition of 30.1 Serial Programming Using Flash Memory Programmer	
	Modification of 30.2 Serial Programming Using External Device (that Incorporates UART)	
	Addition of description in 30.3 Connection of Pins on Board	
	Modification of 30.4 Serial Programming Method	
	Addition of 30.5 Processing Time for Each Command When PG-FP5 Is in Use (Reference Value)	
	Modification of description in 30.6 Self-Programming	
	Modification of description in 30.7 Security Settings	
	Addition of 30.8.1 Data flash overview	
	Addition of 30.8.3 Procedure for accessing data flash memory	
	Modification of error in 31.1 Connecting E1 On-chip Debugging Emulator	CHAPTER 31 ON-CHIP DEBUG FUNCTION
	Addition of description in Table 33 - 2 Symbols in “Operation” Column	CHAPTER 33 INSTRUCTION SET
	Modification of Note 2 in 33.2 Operation List	
	Modification of error of flags when CMP0 instruction is executed in 33.2 Operation List	
	Modification of description in tables in 34.1 Absolute Maximum Ratings	CHAPTER 34 ELECTRICAL SPECIFICATIONS (A: T_A = -40 to +85°C)
	Modification of description in 34.2 Oscillator Characteristics	
	Modification of low-level output current in 34.3.1 Pin characteristics	
	Modification of error of high-level input voltage conditions in 34.3.1 Pin characteristics	
	Modification of error of low-level input voltage conditions in 34.3.1 Pin characteristics	
	Modification of low-level output voltage in 34.3.1 Pin characteristics	
	Modification of error of internal pull-up resistor conditions in 34.3.1 Pin characteristics	
	Modification of 34.3.2 Supply current characteristics	
	Modification of 34.4 AC Characteristics	
	Addition of minimum instruction execution time during main system clock operation	
	Addition of LS mode and LV mode characteristics in 34.5.1 Serial array unit	
	Addition of LS mode and LV mode characteristics in 34.5.2 Serial interface IICA	
	Modification of conditions in 34.5.3 USB	
	Addition of (3) BC option standard in 34.5.3 USB	
	Addition of characteristics about conversion of internal reference voltage and temperature sensor in 34.6.1 A/D converter characteristics	
	Addition of characteristic in 34.6.4 Comparator	
	Deletion of detection delay in 34.6.5 POR circuit characteristics	
	Modification of 34.7 Power supply voltage rising slope characteristics	
	Modification of 34.8 LCD Characteristics	
	Modification of 34.9 Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics	
	Modification of 34.10 Flash Memory Programming Characteristics	
	Addition of 34.12 Timing Specs for Switching Modes	
	New	CHAPTER 35 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)

RL78/L1C User’s Manual: Hardware

Publication Date: Rev.0.01 Jan 10, 2013
Rev.2.40 Mar 22, 2024

Published by: Renesas Electronics Corporation

RL78/L1C

Renesas Electronics Corporation

R01UH0409EJ0240